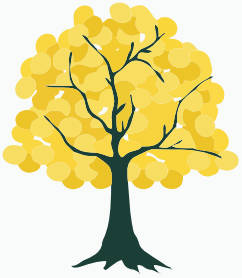
02 a 05
de outubro 2017
Uberlândia - MG

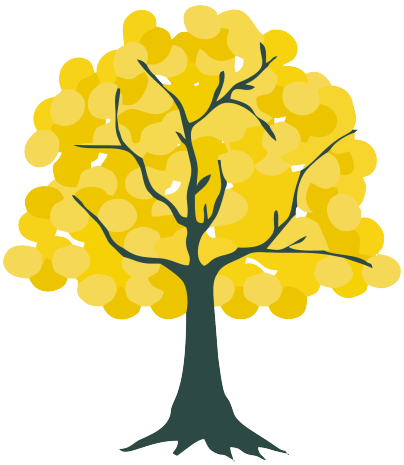
SBBD
Proceedings
Satellite Events of the 32nd Brazilian Symposium on Databases

Carmem S. Hara, Bernadette F. Lóscio,
Daniel de Oliveira, Carina F. Dorneles,
Vânia M. P. Vidal, Fernanda Baião,
Mirella M. Moro, Kary Ocaña,
Humberto L. Razente, Maria Camila N. Barioni (Org.)

Sociedade Brasileira de Computação



SBBD2017



32nd BRAZILIAN SYMPOSIUM ON DATABASES
October 2nd to 5th, 2017
Uberlândia – MG – Brazil

PROCEEDINGS OF THE SATELLITE EVENTS OF THE 32nd BRAZILIAN SYMPOSIUM ON DATABASES

Publisher
Sociedade Brasileira de Computação – SBC

Organization
Carmem Satie Hara, Bernadette Farias Lóscio, Daniel de Oliveira,
Carina Friedrich Dorneles, Vânia Maria P. Vidal, Fernanda Baião,
Mirella M. Moro, Kary Ocaña, Humberto Luiz Razente,
Maria Camila Nardini Barioni

Realization
Sociedade Brasileira de Computação – SBC
Comissão Especial de Banco de Dados (CEBD) da SBC
Universidade Federal de Uberlândia – UFU

ISBN: 978-85-7669-399-4

Dados Internacionais de Catalogação na Publicação (CIP)

B827s Brazilian Symposium on databases (SBBD) (32. : 2017 : Uberlândia, MG, Brazil)
Proceedings of the satellite events [recurso eletrônico] [do] 32 Brazilian Symposium on databases (SBBD), October, 2nd a 5th, 2017, Uberlândia, Minas Gerais; organizadores: Carmem Satie Hara, et al. - Uberlândia: SBC, 2017.

ISBN: 978-85-7669-399-4
Inclui bibliografia.
342 p.: il.
Modo de acesso: http://www.sbbd.org.br/2017

1. Computação - Congressos. 2. Bases de Dados - Congressos. I. Hara, Carmem Satie. II. Lóscio, Bernadete Farias. III. Brazilian Symposium on databases (SBBD) (32.: 2017 : Uberlândia, MG, Brazil) IV. Universidade Federal de Uberlândia. V. Sociedade Brasileira de Computação. VI. Título.

CDU: 681.3(061.3)

Message from the Local Organization Committee Chairs

Welcome to the 32nd Brazilian Symposium on Databases and to Uberlândia, Minas Gerais! The Brazilian Symposium on Databases is the official database event of the Brazilian Computer Society (SBC) and the largest venue in Latin America for presentation and discussion of research results in the database domain. The 32^{nd} edition of the symposium (SBBD 2017) was held in Uberlândia, in the state of Minas Gerais, from October 2^{nd} to October 5^{th}, 2017. The local organization was performed by the Federal University of Uberlândia (UFU) through the Computing Faculty (FACOM). This year, for the first time, SBBD had the Brazilian Conference on Intelligent Systems (BRACIS) and the Symposium on Knowledge Discovery, Mining and Learning (KDMiLe) as co-located events providing a rich environment for the discussion of researches of their interrelated areas.

The SBBD 2017 program offers a variety of activities, suited for an audience ranging from undergraduate to Ph.D. students, database professionals, practitioners and researchers. The program includes: 3 invited talks and 3 tutorials, presented by distinguished speakers from Brazil, USA and France; 9 technical sessions; 3 short courses about hot topics in the area, presented by specialists in their research fields; demos and applications session; posters sessions; thesis and dissertations workshop; the biannual thesis and dissertations contest; 2 co-located workshops; the 1^{st} KDD-BR (Brazilian Knowledge Discovery in Databases) competition; and a panel.

The excellence of SBBD 2017 program is the result of the competence and effort of a large community, which we gratefully acknowledge. The various sections of these proceedings list in detail those that contributed to the SBBD 2017 edition. We thank the symposium chairs and our colleagues of the local organization committee who donated their precious time to made SBBD 2017 a reality. We also thank the Computing Faculty (FACOM) of the Federal University of Uberlândia (UFU). We are also grateful to the SBC board for their support and to the steering committee members for their help, advice and support. Further, we thank the program committee members and external reviewers for the high quality reviews, and the authors who submitted their papers to SBBD 2017. Finally, we are grateful to our sponsors. Without their support we would not be able to organize this annual event that brings together our community.

We hope you all enjoy SBBD 2017 in Uberlândia, Minas Gerais!

Maria Camila Nardini Barioni, UFU
Humberto Luiz Razente, UFU
SBBD 2017 Local Organization Committee Chairs

Table of Contents

32th Brazilian Symposium on Databases

demos

October 2nd to 5th, 2017
Uberlândia – MG – Brazil

DEMOS AND APPLICATIONS SESSION

PROCEEDINGS

Promotion

Brazilian Computer Society – SBC
SBC Special Interest Group on Databases

Organization

Universidade Federal de Uberlândia – UFU

Demos and Applications Session Chair

Daniel de Oliveira, UFF

Editorial

The Brazilian Symposium on Databases (SBBD) is the largest venue in Latin America for presenting research results in the database domain. In its 32nd edition, SBBD will be held in Uberlândia, in the beautiful state of Minas Gerais, from October 2nd to 5th, 2017.

The Demonstrations and Applications Session (or just Demos Session) is organized since 2004 within SBBD. The Demos Session has become an important venue for sharing prototype data management systems with the SBBD community. The session aims at revealing new approaches and systems that contribute to data management research among researchers, developers and professionals, from both academia and industry.

In this edition issue, we had 7 interesting demo papers selected from a total of 17 submissions (an acceptance rate of 41%). Each paper was evaluated by 3 reviewers selected from a committee of 22 researchers from both academia and industry.

The Demos Session is the result of the collective effort of the SBBD community, which we gratefully acknowledge. First, we are very thankful to all authors of submitted papers for their interest in Demos Session. Second, we would like to thank the reviewers for their high quality evaluations.

Finally, we would like to thank the SBBD 2017 organizers for all local arrangements that provide the necessary infrastructure for Demos Session.

We hope you all enjoy SBBD Demos Session in Uberlândia!

Daniel de Oliveira, UFF
SBBD Demos and Applications Session 2017 Chair

32nd Brazilian Symposium on Databases

October 2nd to 5th, 2017
Uberlândia – MG – Brazil

Promotion

Brazilian Computer Society – SBC
SBC Special Interest Group on Databases

Organization

Universidade Federal de Uberlândia – UFU

SBBD Steering Commitee

Agma Juci Machado Traina, USP
Bernadette Lóscio, UFPE
Caetano Traina Jr., USP
Carmem Hara, UFPR
Javam Machado, UFC
Mirella M. Moro, UFMG
Vanessa Braganholo, UFF

SBBD 2017 Commitee

Steering Committee Chair
Javam Machado, UFC

Local Organization Chairs
Maria Camila N. Barioni, UFU and Humberto L. Razente, UFU

Program Committee Chair
Carmem S. Hara, UFPR

Short papers Chairs
Bernadette Lóscio, UFPE and Damires Souza, IFPB

Demos and Applications Session Chair
Daniel de Oliveira, UFF

Short Courses Chair
Vaninha Vieira, UFBA

Workshop on Thesis and Dissertations in Databases Chair
Carina Dorneles, UFSC

Tutorials Chair
Ana Carolina Salgado, UFPE

Thesis and Dissertation Contest Chair
Vânia Vidal, UFC

Workshops Chair
Fernanda Baião (UNIRIO)

Local Organization Committee

Maria Camila N. Barioni, UFU
Humberto L. Razente, UFU
José Gustavo de Souza Paiva, UFU
Marcelo Zanchetta do Nascimento, UFU
Elaine Ribeiro de Faria Paiva, UFU
João Henrique de Souza Pereira, UFU

Demos and Applications Session Program Committee

Angelo Brayner (UFC)
Carlos Eduardo Pires (UFCG)
Daniel de Oliveira (UFF) – Chair
Eduardo Bezerra (CEFET/RJ)
Eduardo de Almeida (UFPR)
Eduardo Ogasawara (CEFET/RJ)
Fabio Porto (LNCC)
Fernanda Baião (UNIRIO)
Humberto Razente (UFU)
Jonas Dias (DELL-EMC)
José Antonio Macêdo (UFC)
José Maria Monteiro (UFC)
Kary Ocaña (LNCC)
Leonardo Azevedo (IBM Research Brazil and UNIRIO)
Luiz André Paes Leme (UFF)
Marcela Ribeiro (UFSCar)
Marta Mattoso (COPPE/UFRJ)
Mirella M. Moro (UFMG)
Renata Galante (UFRGS)
Rodrigo Monteiro (UFF)

Vania Vidal (UFC)
Victor de Almeida (UFF and Petrobrás)

Table of Contents (Demos Session)

AgriExt: Uma Ferramenta para Estimativa da Evapotranspiração de Referência

Hinessa Dantas Caminha[1], Antonio Raimundo Rocha Mendonça[1], Ticiana L. Coelho da Silva[1], Atslands Rego da Rocha[1], Carlos Diego Andrade de Almeida[1], José A. F. de Macêdo [1]

[1]Universidade Federal do Ceará (UFC)
Ceará - Brasil

{hinessacaminha,raimundo}@alu.ufc.br,

{diego.andrade,ticianalc,atslands}@ufc.br,

{jose.macedo}@dc.ufc.br

Abstract. *Sustainable use of water is a global challenge. Evapotranspiration is the combination process of transferring moisture from the earth to the atmosphere by evaporation and transpiration from plants. By estimating this rate of loss, farmers can efficiently manage the crop water requirement and how much water is available. This paper proposes AgriExt tool in order to assist farmers and agronomists with meteorological and climatic data. As well as, AgriExt provides analysis of such data by means of prediction models that estimate reference evapotranspiration and help to propose a sustainable solution for water use, especially in regions strongly affected by drought.*

Resumo. *A utilização da água de forma sustentável é um desafio mundial. Evapotranspiração é o processo de combinação de transferir a umidade da terra para a atmosfera por evaporação e transpiração das plantas. Ao estimar essa taxa de perda, os agricultores podem gerenciar eficientemente o requisito de água da cultura e a quantidade de água disponível. Este artigo propõe a ferramenta AgriExt[1], a fim de auxiliar agricultores e agrônomos na coleta de dados de estações meteorológicas e prover análise desses dados construindo modelos de predição, que estimam a evapotranspiração de referência e ajudem a propor uma solução sustentável para utilização da água, principalmente em regiões fortemente afetadas pela seca.*

1. Introdução

A agricultura irrigada exerce um papel importante para garantir a segurança alimentar, proporcionando abordagens inovadoras que levam a uma maior produtividade. No Brasil, a irrigação é responsável por 75% da demanda de água consumida anualmente [ANA 2016]. À medida que outros setores como abastecimento urbano, indústria, manufatura e o próprio meio ambiente se expandem, a concorrência por essa demanda aumenta. Assim, cabe ao setor de agricultura revisar e ajustar seus métodos de acordo com a quantidade de água disponível para utilização [Garces-Restrepo et al. 2007].

[1]http://tiny.cc/agri_ext

O manejo da irrigação tem por finalidade estabelecer técnicas que possibilitem aumentar a conservação de água e energia sem reduzir a produção econômica da cultura. Dentre as diversas técnicas existentes, pode-se citar o monitoramento via clima, que consiste na utilização de dados climáticos para estimativa do consumo de água de uma cultura. A estimativa é determinada pela evapotranspiração, conceito este que designa a ocorrência simultânea dos processos de evaporação e transpiração de uma superfície vegetada [Frizzone et al. 2013]. Entretanto, os métodos tradicionais para realizar esta estimativa podem ser dispendiosos e complexos. Entre os diversos métodos existentes na literatura, o recomendado pela FAO é a equação de *Penman-Monteith* [Allen et al. 1998]. No entanto, seu uso é complexo e requer que todas as variáveis climáticas estejam presentes. Desta forma, vários procedimentos foram desenvolvidos para estimar o valor dos parâmetros climáticos ausentes dos dados coletados. A mineração de dados é uma técnica que consiste na descoberta de padrões e modelos sobre os dados, de modo que gerem algum conhecimento, e pode ser uma alternativa viável à esses modelos de uso complexo.

No trabalho de [Rahimikhoob 2014], foram utilizadas técnicas de mineração de dados para criar modelos que estimassem a evapotranspiração de referência. Os experimentos foram executados sobre os dados de quatro estações meteorológicas presentes na província de Sistan and Baluchestan, Irã. Já no estudo de [Xavier et al. 2016], os autores obtiveram as bases de dados meteorológicos do INMET (Instituto Nacional de Meteorologia), as quais correspondiam às seis estações meteorológicas presentes em cidades do Rio de Janeiro. O objetivo dos autores era criar um modelo de predição da evapotranspiração para cada base, o qual conseguisse gerar bons resultados mesmo que houvesse a indisponibilidade de alguns atributos. Todas as bases possuíam pelo menos um atributo com valores ausentes. Os modelos geraram equações para calcular a evapotranspiração, e os resultados gerados pelo modelo obtiveram alto grau de correlação com as séries históricas disponibilizadas pelo INMET. É importante ressaltar que em nenhuma das equações criadas pelo modelo preditivo utilizaram todos os atributos das bases de dados.

O trabalho [Caminha et al. 2017] apresenta modelos de predição de alta acurácia para a estimativa da evapotranspiração de referência (ET_0), gerados a partir de dados meteorológicos da cidade de Quixadá-CE. Partindo deste cenário, este trabalho propõe a ferramenta AgriExt[2] a fim de disponibilizar os modelos propostos por [Caminha et al. 2017], entre outros. Além disso, a AgriExt é uma ferramenta capaz de criar modelos da ET_0, a partir de um conjunto de dados climáticos fornecido como entrada pelo usuário. A ferramenta AgriExt é uma solução sustentável para utilização da água, principalmente em regiões fortemente afetadas pela seca.

Este artigo está dividido da seguinte maneira: Na Seção 2 são apresentados os principais conceitos envolvidos neste trabalho. Na Seção 3 o funcionamento da ferramenta AgriExt é discutido. Por fim, na Seção 4 desenha as conclusões do artigo e trabalhos futuros.

2. Fundamentação Teórica

Nesta Seção, serão abordados brevemente os principais conceitos envolvidos na idealização da ferramenta.

[2] `http://tiny.cc/agri_ext`

2.1. Evapotranspiração de Referência

O termo evapotranspiração (ET) designa a ocorrência simultânea da evaporação e transpiração em uma superfície vegetada. Sua taxa é normalmente descrita em milímetros (mm) por uma determinada unidade de tempo e expressa a quantidade de água perdida da cultura. Analogamente, a evapotranspiração de referência (ET_0) é a quantidade de água perdida da cultura, entretanto, para uma superfície de grama padrão. Os únicos fatores que influenciam na ET_0 são parâmetros climáticos e, consequentemente, seus valores podem ser computados a partir de dados meteorológicos [Allen et al. 1998].

A ferramenta proposta neste trabalho realiza a estimativa da ET_0 fazendo uso de modelos de predição. Os algoritmos responsáveis pela geração dos modelos de predição na ferramenta AgriExt são descritos a seguir.

2.2. Regressão Linear

A análise de regressão é uma técnica estatística que possui a finalidade de investigar e modelar a relação entre variáveis por meio de uma equação matemática. Segundo [Montgomery et al. 2015], a regressão linear múltipla pode ser definida como:

$$y = \beta_0 + \beta_1 x_1 + \beta_2 x_2 + ... + \beta_k x_k + \epsilon \quad (1)$$

Onde y corresponde à variável a ser prevista e β_j, $\forall j, j \in \{0, 1, ..., k\}$ os coeficientes de regressão, que representam a variação do valor-resposta de y por cada unidade de x_j. Considere que ϵ corresponde ao erro estatístico para mapear os dados de acordo com o modelo.

A regressão linear é empregada pela ferramenta AgriExt para criar as equações de estimativa da ET_0.

2.3. M5'

A árvore M5', proposta por [Wang and Witten 1996], é uma melhoria da árvore M5 proposta por [Quinlan 1992]. A M5' trabalha com o mesmo conceito das árvores de decisão tradicionais, entretanto, é utilizada para dados de valores contínuos, pois oferece a possibilidade de utilização de modelos de regressão linear multivariados em suas folhas.

A árvore M5' é usada na ferramenta AgriExt para criar modelos de predição da ET_0, assim como a regressão linear. Tais algoritmos de classificação são utilizados na ferramenta AgriExt por terem reportado resultados satisfatórios em outros trabalhos como [Caminha et al. 2017, Xavier et al. 2016]. No entanto, é possível gerar modelos para predição de ET_0 utilizando outros algoritmos.

3. AgriExt

A ferramenta AgriExt possui três principais funcionalidades: visualização dos dados climáticos coletados a partir de uma estação metereológica, criação de modelos de predição da ET_0 e a estimativa da ET_0 para novos dados climáticos fornecidos de entrada. Dessa última funcionalidade, é possível calcular a necessidade de água diária para irrigação de uma cultura (ET_c), uma vez que esse valor pode ser obtido da equação abaixo. Considere K_c o coeficiente individual de cada cultura:

$$ET_c = K_c \times ET_0 \quad (2)$$

Na visualização dos dados, o usuário pode carregar seu conjunto de dados através do *upload* de um arquivo *.csv* ou similar, ou pode optar por utilizar as bases de dados já disponibilizadas pela ferramenta. Após o carregamento dos dados, são apresentados para o usuário os atributos presentes no conjunto de dados escolhido, e seus respectivos valores máximos, mínimos e média. Além disso, a ferramenta AgriExt permite visualizar os valores de cada atributo da base de dados em gráficos, a fim de melhorar o entendimento dos usuários. As Figuras 1 e 2 mostram as telas referentes a esta funcionalidade.

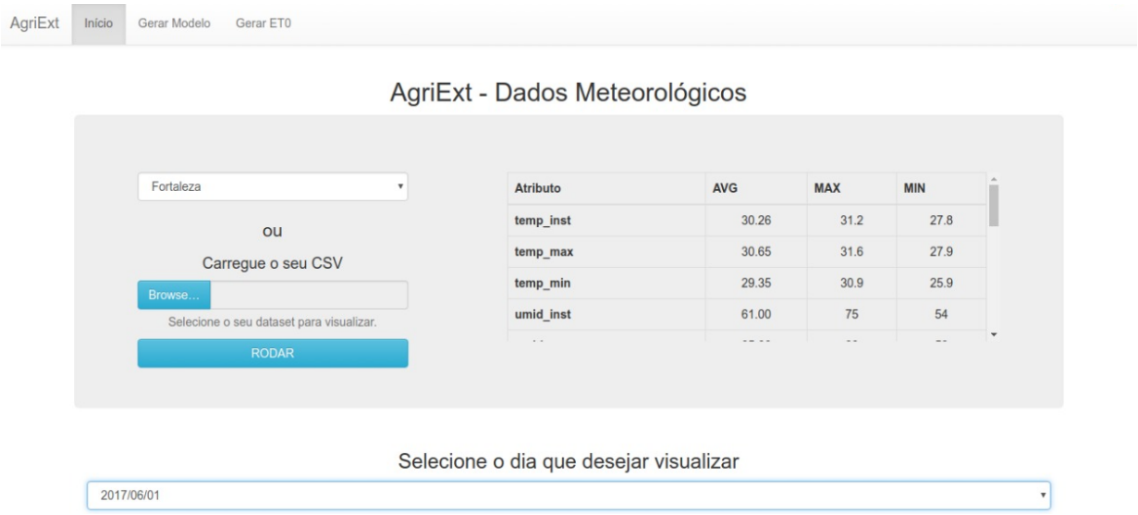


Figura 1. Importação dos dados metereológicos pela AgriExt

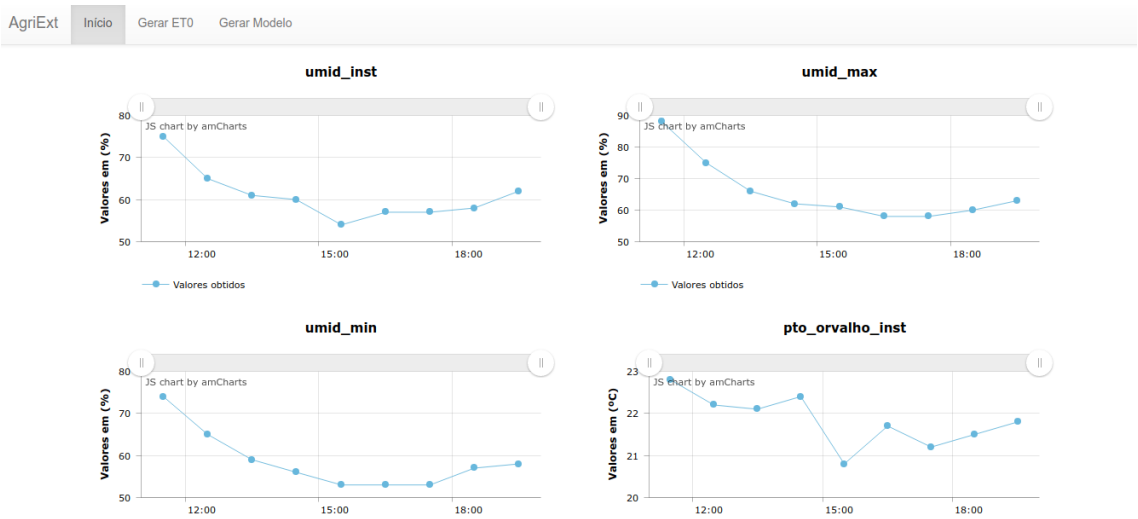


Figura 2. Visualização dos dados

Na funcionalidade de geração dos modelos de predição, são disponibilizados dois algoritmos de classificação: Regressão Linear e *M5'*. Estes algoritmos são executados sobre os dados fornecidos, e ao final, o modelo de predição da ET_0 e suas métricas de

avaliação são apresentadas, conforme a Figura 3. O modelo gerado pela ferramenta AgriExt também pode ser exportado, viabilizando posteriormente o cálculo da ET_0 para novos dados.

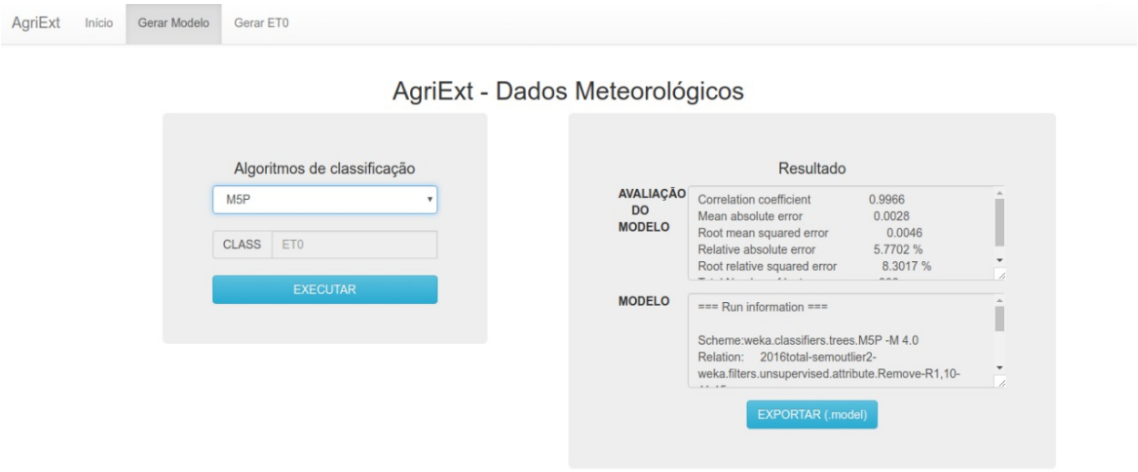


Figura 3. Geração de modelo de predição para ET_0 pela AgriExt

Por fim, na funcionalidade de estimativa da ET_0, são disponibilizados os modelos capazes de calcular o valor de ET_0 a partir dos dados fornecidos de entrada. Por padrão, a ferramenta já apresenta o método de Jensen-Hayse [Fernandes et al. 2010], indicado para regiões de clima semi-árido, e o modelo QuixadaHC, proposto no trabalho de [Caminha et al. 2017]. Caso nenhum dos dois se aplique às necessidades do usuário, também é permitido o *upload* de um modelo externo (com extensão *.model*). Ao final, os valores da ET_0 são estimados para cada hora, e ainda o valor diário. Além disso, o usuário também pode inserir um coeficiente de cultura (K_c), e a quantidade de água diária necessária para irrigação é calculada e apresentada ao usuário. A Figura 4 exibe a tela desta funcionalidade.

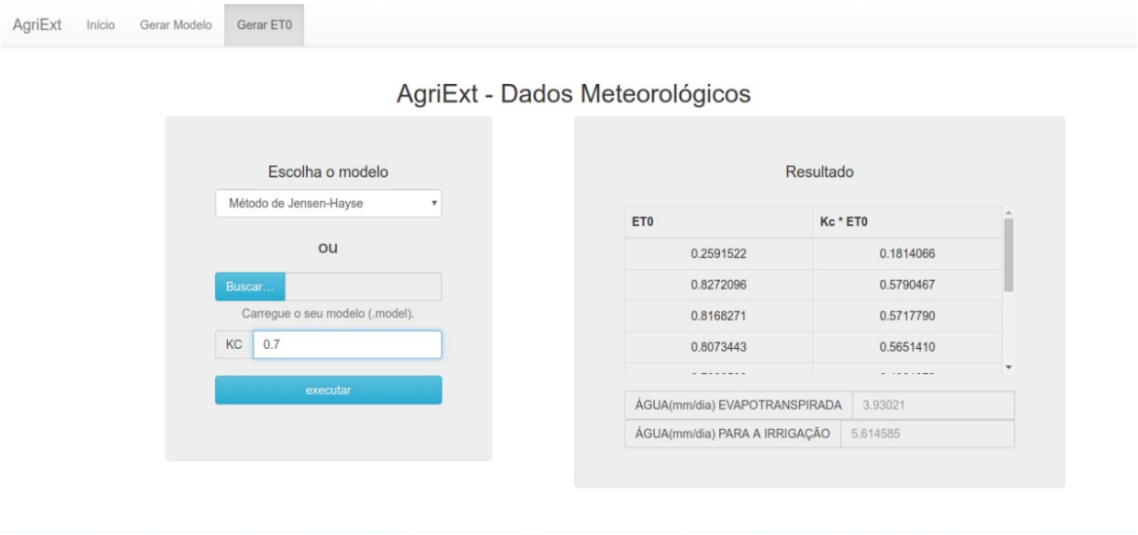


Figura 4. Estimativa da ET_0

4. Conclusão

Estimar a evapotranspiração de referência é uma atividade de grande importância para agricultores e agrônomos que necessitam irrigar suas culturas de modo eficiente sem desperdiçar água. Para auxiliar nesse processo, foi desenvolvida a AgriExt, ferramenta de fácil manuseio, que faz uso de modelos preditivos para realizar as estimativas. Como trabalhos futuros, pode-se mencionar a disponibilização de mais modelos (inclusive, técnicas para análise de séries temporais) e algoritmos na ferramenta. Além disso, a proposta de um modelo unificado capaz de estimar a ET_0 de regiões com condições climáticas semelhantes.

Referências

Allen, R. G., Pereira, L. S., Raes, D., and Smith, M. (1998). Crop evapotranspiration-guidelines for computing crop water requirements-fao irrigation and drainage paper 56. *FAO, Rome*, 300(9):D05109.

ANA (2016). *Conjuntura dos recursos hídricos no Brasil: Informe 2016*. Agência Nacional de Águas.

Caminha, H., Coelho da Silva, T., Rocha, A., and Lima, S. (2017). Estimating reference evapotranspiration using data mining prediction models and feature selection. *Proceedings of the 19th International Conference on Enterprise Information Systems (ICEIS 2017)*, 1:272–279.

Fernandes, D. S. F., Heinemann, A. B., Amorim, A. d. O., et al. (2010). *Evapotranspiração: uma revisão sobre os métodos empíricos*. Embrapa Arroz e Feijão.

Frizzone, J. A., de Souza, F., and Lima, S. C. R. V. (2013). *Manejo da irrigação: Quando, Quanto e Como Irrigar*. INOVAGRI.

Garces-Restrepo, C., Vermillion, D., and Muoz, G. (2007). Irrigation management transfer: Worldwide efforts and results. *FAO Water Reports*.

Montgomery, D. C., Peck, E. A., and Vining, G. G. (2015). *Introduction to linear regression analysis*. John Wiley & Sons.

Quinlan, J. R. (1992). Learning with continuous classes. *5th Australian joint conference on artificial intelligence*, 92:343–348.

Rahimikhoob, A. (2014). Comparison between m5 model tree and neural networks for estimating reference evapotranspiration in an arid environment. *Water resources management*, 28(3):657–669.

Wang, Y. and Witten, I. H. (1996). Induction of model trees for predicting continuous classes. *Working paper series*.

Xavier, F., Tanaka, A. K., and Amorim, F. A. B. (2016). Application of data science techniques in evapotranspiration estimation. *Dissertação (Mestrado em Informática) - Universidade Federal do Estado do Rio de Janeiro, Rio de Janeiro, 2016*.

DataChain: Uma Ferramenta para Assegurar a Propriedade e Imutabilidade de Documentos Digitais

Gabriel O. Mendanha [1], **Lívia A. Cruz** [1], **Regis P. Magalhães**[1]

[1]Universidade Federal do Ceará – Campus de Quixadá
Quixadá – CE – Brasil

gabrielmendanha@alu.ufc.br, {livia.almada,regismagalhaes}@ufc.br

Abstract. *After the creation of the Bitcoin cryptocurrency with the study and research of the technology that made it possible – the blockchain – some initiatives take the opportunity to use it in other use cases not related to financial transactions or digital money. Motivated by a reality that people easily adulterate, copy and corrupt digital information, this paper presents a proof of concept using blockchain, taking advantage of the characteristics of immutability and ownership of the data. This article focuses on digital documents and opens up a range of use cases and possibilities for a variety of consumers who want a way to prove the authenticity and ownership of documents as well as transfer them to others.*

Resumo. *Após a criação da criptomoeda Bitcoin, com o estudo e pesquisa da tecnologia que a tornou possível, a blockchain, viu-se a possibilidade de utiliza-lá para outros casos de uso não relacionados a transações financeiras ou dinheiro digital. Motivado por uma realidade que pessoas facilmente adulteram, copiam e corrompem informações digitais, este trabalho apresenta uma prova de conceito utilizando a blockchain, tirando proveito das características de imutabilidade e propriedade dos dados. Voltado para documentos digitais, o presente trabalho também abre uma gama de casos de uso e possibilidades para uma variedade de consumidores que desejam uma maneira de provar a autenticidade e posse de documentos, assim como transferí-los a outras pessoas.*

1. Introdução

Em 2009 foi publicada por Satoshi Nakamoto[1] a primeira criptomoeda digital bem sucedida, o Bitcoin [Nakamoto 2008]. Uma combinação de fatores permitiu essa moeda ser difundida. Além de ser um ativo digital, ela também é um sistema descentralizado que permite transações financeiras seguras em uma rede *peer-to-peer* com participantes não-confiáveis, sem depender de um instituição financeira. A descentralização é uma dentre muitas características que foram incorporadas, graças a tecnologia *blockchain*, que é fundamentalmente um banco de dados transparente e descentralizado que contém o registro de todas as transações.

Este trabalho propõe uma ferramenta[2] que, utilizando a tecnologia de *blockchain* pública, permite às pessoas uma maneira de provar a posse e autenticidade de um documento digital, assim como transferi-lo a outra pessoa. O sistema também oferece um

[1]Satoshi Nakamoto é um pseudônimo. Até a conclusão deste trabalho não se sabe a identidade da pessoa ou organização responsável pela criação do Bitcoin.

[2]https://vimeo.com/225034651

meio para armazenar o documento com a garantia de que o mesmo não poderá ser modificado ou removido pelo dono, por outra pessoa, ou pelo administrador do sistema. A plataforma proposta tem em seu núcleo o BigchainDB, um banco de dados distribuído e descentralizado que incorporou as características da *blockchain* sem perda de escalabilidade [McConaghy et al. 2016].

A principal contribuição deste trabalho é a proposta de uma arquitetura descentralizada para armazenamento de documentos que integra a *blockchain* com um sistema de arquivos distribuído para garantir a posse, a integridade e a imutabilidade de documentos digitais. Vale ressaltar que a *blockchain* é uma tecnologia emergente e que seu uso fora do contexto de criptomoedas, como por exemplo, na prova de existência de documentos legais traz novos desafios. [Crosby et al. 2016].

2. Estrutura e características da *Blockchain*

A *blockchain* é uma lista encadeada e ordenada de blocos que contém transações como conteúdo principal. Cada bloco referencia o seu anterior e os mesmos são identificados pelo resultado de uma função *hash* criptográfica, que mapeia um dado de tamanho variável para um dado de tamanho fixo. Na *blockchain* é desejável que a função *hash* produza dois resultados iguais se, e somente se, as entradas forem as mesmas. Portanto, deve existir um valor *hash* diferente para cada bloco [Antonopoulos 2014].

A Figura 1 ilustra a estrutura da *blockchain*. Observe que os blocos são representados por retângulos. H representa uma função *hash* que recebe como parâmetro o conteúdo do bloco anterior, indicado pela seta. Logo, cada bloco sabe quem é o seu anterior através do valor *hash* obtido como resultado obtido da chamada da função H.

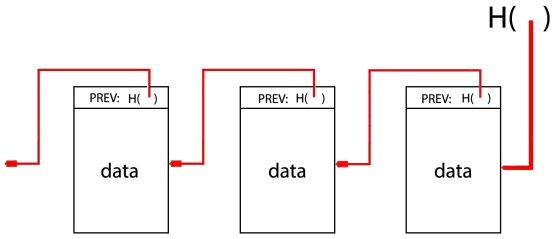


Figura 1. Estrutura da *blockchain*

2.1. Distribuição, integridade e segurança na *Blockchain*

Considerando o contexto da tecnologia da informação, um livro-razão é um mecanismo de armazenamento que permite apenas inserções [Peter Evans-Greenwood 2016]. Em um livro-razão, as informações são imutáveis e podem conter dados genéricos, portanto a *blockchain* pode ser vista como uma tecnologia de livro-razão. Na *blockchain*, não existe um agente central responsável pelo gerenciamento do sistema, ou seja, o controle é descentralizado. Este controle é feito através um conjunto de nós em uma rede *peer-to-peer* [McConaghy et al. 2016]. Assim, a responsabilidade de decidir o que incluir, em que ordem incluir e de garantir que um registro não seja alterado após sua inclusão é distribuída. Um grupo de nós, através de um algoritmo de consenso, divide essa responsabilidade.

Assim como em qualquer outro sistema distribuído, a *blockchain* possui um problema de resolução de conflitos. Se dois fatos incompatíveis chegarem no mesmo instante,

o sistema deve possuir regras que determinem qual dos fatos será considerado válido. A Figura 2 exemplifica o problema de resolução de conflitos. Nesta Figura, Alice envia $10 para Bob e os mesmos $10 para Charlie. O problema está no fato de que Alice possui somente $10 e está tentando gastar duas vezes este valor. Uma maneira de resolver este problema é ordenando os fatos, o primeiro que for registrado é o vencedor.

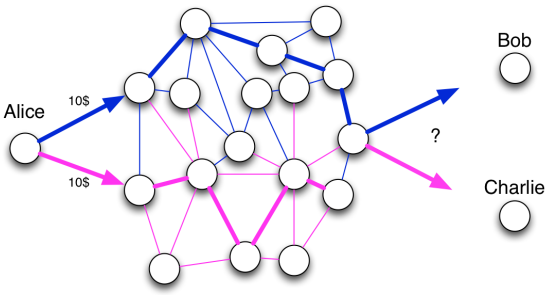


Figura 2. O problema do gasto duplo [Zaninotto 2016]

Porém, ambos os fatos podem aparecer em ordens diferentes em nós distantes um do outro. Para que toda a rede concorde na ordem dos fatos e preserve sua integridade é necessário um sistema de sincronização de dados, um algoritmo de consenso [Zaninotto 2016].

No quesito segurança na *blockchain*, a criptografia de chave pública/privada, que é um dos fundamentos da segurança moderna é usada para possibilitar que as pessoas assinem digitalmente documentos como: arquivos de texto, imagens, etc. A *blockchain* utiliza-se deste sistema de criptografia para assinar digitalmente as transações. O acesso e a utilização dos ativos digitais não é possível sem o conhecimento da chave privada do dono atual. [Peters and Panayi 2016]

3. O Sistema DataChain

O sistema proposto, DataChain[3], armazena os documentos e estabelece um vínculo de posse, que pode ser transferível a outra pessoa através de um par de chaves criptográficas, além de também permitir a validação do título de posse de documentos entre os indivíduos que utilizam a plataforma. O público-alvo são pessoas que querem uma forma de garantir o reconhecimento como autor de uma obra intelectual, como: artistas, compositores, pesquisadores, etc. Assim como pessoas de negócio que podem transferir, por exemplo, a escritura de uma casa para outra pessoa de forma ágil e segura, ou até mesmo por entidades que desejam de alguma forma combater fraude de documentos sensíveis, como: documentos de identidade pessoal, prontuários médicos, diplomas, etc. O sistema proposto, além de utilizar tecnologias inovadoras, também serve como base ou inspiração para o construção de aplicações mais robustas e especializadas.

3.1. Arquitetura

O sistema possui como principais componentes de sua arquitetura o banco de dados BigchainDB [McConaghy et al. 2016] e um sistema de arquivos distribuído denominado *InterPlanetary File System* (IPFS) [Benet 2014].

[3]Código-fonte disponível em: https://github.com/gabrielmendanha/tcc2

O BigchainDB é um banco de dados *open-source*, descentralizado e distribuído e incorpora as melhores características da *blockchain* e de bancos de dados distribuídos (BDD) [McConaghy et al. 2016]. No BigchainDB, a posse de determinado ativo digital é garantida através do par de chaves pública e privada. Para todas as transações é necessário gerar uma assinatura digital que é calculada com base na chave privada e na chave pública da pessoa. Logo, para uma pessoa expressar o seu consentimento em querer transferir um ativo digital para outra pessoa, ela deve informar ao sistema a sua chave privada.

O IPFS é um sistema de arquivos que implementa o modelo *peer-to-peer*. Ele tem como objetivo conectar vários dispositivos ao mesmo sistema de arquivos, provendo um modelo de armazenamento endereçado ao conteúdo, ao mesmo tempo em que os nós não precisam confiar uns nos outros e todos possuem a mesma influência na rede. Os nós se conectam entre si para transferir os arquivos.

O sistema de arquivos gera um valor *hash* único e imutável para cada documento digital, além de possibilitar encontrar o arquivo na rede *peer-to-peer* a partir deste mesmo valor *hash*. Este sistema replica o arquivo nos nós que o requisitam, diminuindo assim as chances de determinado arquivo ficar indisponível temporariamente ou permanentemente em caso de catástrofe (falha no disco rígido, quedas de energia ou interrupção do serviço de internet do servidor que provê os documentos, por exemplo).

A utilização do IPFS com o BigchainDB torna desnecessário o armazenamento dos documentos diretamente no banco de dados, o que implica ganhos de desempenho uma vez que o sistema não está mais limitado ao tipo e tamanho dos dados aceitos pelo BigchainDB. Portando, não é necessário que sejam processadas conversões toda vez que um documento for consultado ou inserido.

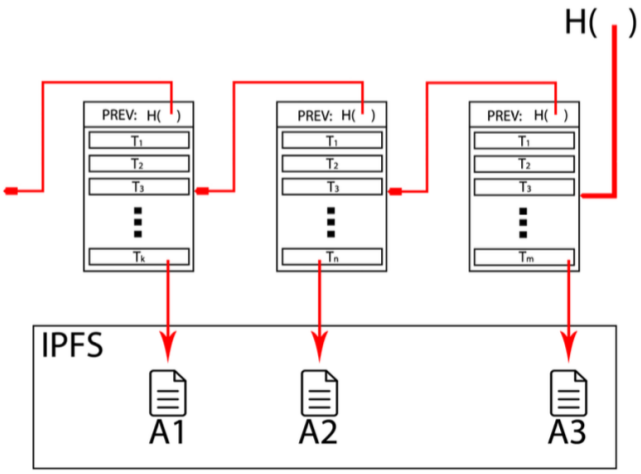


Figura 3. Integração entre a *blockchain* e o IPFS

A Figura 3 ilustra como a *blockchain* provida pelo BigchainDB integra-se aos objetos no IPFS. Como o valor *hash* dos objetos no IPFS é único, imutável e basta para localizar determinado arquivo no sistema, é viável e seguro armazená-lo como uma referência para o documento binário. Embora omitido na imagem, cada transação dentro de um bloco armazena o valor *hash* do nó inicial. Quaisquer modificações em documentos no IPFS geram um novo objeto que não está incluso na *blockchain*. Entretanto, o

objeto original permanece intacto e seu valor *hash* continua vinculado à *blockchain*. O novo objeto, com as modificações não é reconhecido pelo sistema, garantindo assim a imutabilidade do documento original.

4. Visão Geral da Demonstração

A submissão, recuperação e transferência de um documento de um usuário são realizadas através dos passos definidos a seguir.

1. Realização do download do par de chaves criptográficas, se necessário.
2. Escolha do documento que deseja submeter à *blockchain*.
3. Realização do download do comprovante
4. Com os dados fornecidos no comprovante, o usuário pode consultar determinado documento, além de fornecer uma assinatura pública (opcional) para verificar se o documento pertence a assinatura fornecida.
5. Em posse da assinatura privada e da referência do documento, ela pode transferir a posse a quem desejar, representado pela assinatura pública.

(a) Upload de documento

(b) Detalhes da transação submetida

(c) Consulta de documento

(d) Transferência de posse

5. Considerações Finais

Neste trabalho, foi apresentado o DataChain, um sistema baseado na *blockchain* que integrada ao sistema de arquivos IPFS possibilita prover funcionalidades interessantes ao usufruir das características de imutabilidade e posse da *blockchain* de maneira escalável. Voltado para pessoas e entidades que de alguma forma desejam uma maneira de provar o título de posse de determinado documento digital ou combater fraudes de documento sensíveis a adulteração.

Referências

Antonopoulos, A. M. (2014). *Mastering Bitcoin: unlocking digital cryptocurrencies*. "O'Reilly Media, Inc.".

Benet, J. (2014). Ipfs-content addressed, versioned, p2p file system. *arXiv preprint arXiv:1407.3561*.

Crosby, M., Pattanayak, P., Verma, S., and Kalyanaraman, V. (2016). Blockchain technology: Beyond bitcoin. *Applied Innovation*, 2:6–10.

McConaghy, T., Marques, R., Müller, A., De Jonghe, D., McConaghy, T., McMullen, G., Henderson, R., Bellemare, S., and Granzotto, A. (2016). Bigchaindb: a scalable blockchain database. *white paper, BigChainDB*.

Nakamoto, S. (2008). Bitcoin: A peer-to-peer electronic cash system.

Peter Evans-Greenwood, Robert Hillard, I. H. P. W. (2016). Bitcoin, blockchain and distributed ledgers: Caught between promise and reality.

Peters, G. W. and Panayi, E. (2016). Understanding modern banking ledgers through blockchain technologies: Future of transaction processing and smart contracts on the internet of money. In *Banking Beyond Banks and Money*, pages 239–278. Springer.

Zaninotto, F. (2016). The blockchain explained to web developers. `http://marmelab.com/blog/2016/04/28/blockchain-for-web-developers-the-theory.html`. Acesso em: 21 jun. 2016.

DCluster: Um sistema para análise exploratória de grandes volumes de dados georreferenciados

Cláudio Gustavo S. Capanema[1], **Fabrício A. Silva**[1], **Thais R. M. Braga Silva**[1]

[1] Universidade Federal de Viçosa (UFV), Florestal, Brasil

{claudio.capanema, fabricio.asilva, thais.braga}@ufv.br

Resumo. *O crescimento e a diversificação do uso de dispositivos móveis, especialmente smartphones, fez surgir um grande volume de dados georreferenciados oriundos de aplicativos móveis. Como consequência, as empresas estão cada vez mais interessadas em analisar tais dados para conhecer melhor seus usuários e, assim, oferecer melhores serviços. A análise de dados georreferenciados é uma área ainda pouco explorada, e que pode trazer informações mais úteis que os dados puros. Em geral, as ferramentas atuais de análise de dados não tratam atributos georreferenciados e exigem conhecimento prévio dos usuários para a utilização de técnicas avançadas. Este trabalho propõe um sistema que visa auxiliar analistas de dados na exploração e visualização de grandes volumes de tipos variados de dados, incluindo os georreferenciados.*

1. Introdução

A utilização de dispositivos móveis está se disseminando cada vez mais rapidamente. Esse fenômeno de popularização tem trazido uma crescente geração de grandes volumes de dados oriundos da utilização de aplicativos de dispositivos como *smartphones* e *tablets*. Isso se deve a fatores como o baixo custo de aquisição e mobilidade, além do interesse dos provedores de serviços em conhecer os seus usuários. Segundo [Statista 2017], no primeiro quadrimestre de 2012, o número de usuários ativos diariamente no Facebook através de uma plataforma móvel era de aproximadamente 266 milhões. No mesmo período do ano de 2016, esse número subiu para 1,146 bilhão.

Em geral, grande parte dos dados provenientes de dispositivos móveis são georreferenciados, ou seja, incluem a localização do usuário. Um exemplo está relacionado à possibilidade de se realizar *check-ins* de localização ao interagir em redes sociais. Em outros casos, a presença do georreferenciamento é essencial para o funcionamento da ferramenta, como ocorre com os aplicativos Waze e Uber. Por fim, dados de diferentes segmentos de empresas (e.g., bancos, operadoras de telecomunicações, sistemas de comércio eletrônico, dentre outros) também possuem informações georreferenciadas. Esses dados de localização podem trazer informações relevantes para empresas e pesquisadores.

Essa crescente demanda por análise de dados fez surgir ferramentas analíticas no mercado. Entretanto, atualmente ainda existe uma deficiência no tratamento de atributos georreferenciados, que muitas vezes inexiste ou esbarra em limitantes como o preço da licença. Além disso, algumas dessas ferramentas exigem que o usuário tenha uma experiência técnica em análise de dados. Isso dificulta que pessoas com menos conhecimento técnico possam utilizar os recursos de aprendizagem de máquina, por exemplo, para trazê-los a realidade da sua área de pesquisa, o que é um aspecto que dificulta a popularização da análise de dados.

Este trabalho apresenta o DCluster, um sistema para análise exploratória de grandes volumes de dados georreferenciados. Por se tratar de uma ferramenta Web, a sua utilização não é restrita a máquinas com grande poder computacional por parte do usuário. Além disso, também é proposta uma interface simples e intuitiva, caracterizada pela ausência de informações desnecessárias, a fim de que a experiência do usuário transcorra da maneira mais amigável possível. Esses aspectos têm o objetivo de contribuir com a disseminação e a popularização da análise de dados georreferenciados.

2. Ferramentas Relacionadas

Esta seção discute as características das principais ferramentas para análise exploratória de grandes volumes de dados existentes, e as relaciona com o sistema proposto neste trabalho. Dentre os aspectos que caracterizam essas ferramentas, os mais relevantes são: suporte a dados georreferenciados, arquitetura do sistema (centralizada ou distribuída), usabilidade e licença.

Um dos sistemas mais populares é o Weka [of Waikato 2017], uma ferramenta gratuita e com interface simples. Porém, seu desenvolvimento iniciado nos anos 90, e interrompido com sua aquisição pela Pentaho em 2006, não favorece um ambiente ideal para a análise de grandes volumes de dados, uma vez que sua arquitetura é centralizada e não oferece suporte a dados georreferenciados. Além disso, a criação de filtros depende de conhecimento prévio em expressões regulares.

O Geo-Data Visualizer [Xavier et al. 2017] é uma ferramenta gratuita desenvolvida em Javascript/Jquery que utiliza dados georreferenciados para gerar mapas e estatísticas associadas. Suas principais funcionalidades são a visualização de mapas com agrupamentos, e a filtragem de dados a partir de métricas temporais.

O BigML [BigML 2017] e o Azure Machine Learning [Microsoft 2017a] são boas alternativas para quem deseja utilizar recursos de aprendizado de máquina. Possuem interface amigável, um variado conjunto de funcionalidades, e correspondem a ferramentas Web. Ambas as ferramentas possuem planos acessíveis, porém com algumas restrições como tamanho dos dados. No entanto, ambas as ferramentas esbarram na falta de suporte a dados georreferenciados.

Os sistemas Pentaho Big Data [Hitachi 2017], Tableau [Tableau 2017], Microsoft Power BI [Microsoft 2017b], SAS [SAS 2017], Qlik [Qlik 2017] e Sisense [Sisense 2017] se enquadram no conjunto de ferramentas pagas de análise de dados mais completas da atualidade: suportam diversas fontes de dados, possuem versões online, interfaces intuitivas, e trabalham com dados georreferenciados. Essas ferramentas se enquadram na categoria de BI (*Business Intelligence*), e sua ampla lista de funcionalidades faz com que o aprendizado seja relativamente complexo. Além disso, o preço das licenças é muitas vezes inviável para pequenas e médias empresas.

3. DCluster

O DCluster consiste em um sistema Web para análise exploratória de grandes volumes de dados, com foco no georreferenciamento. Sua arquitetura cliente-servidor tem por finalidade facilitar a utilização da ferramenta, uma vez que não é necessária a instalação em máquina local. O DCluster também se destaca pela utilização de recursos

interativos para a visualização de gráficos, e por ser desenvolvido na linguagem Python, que oferece uma lista de APIs para análise, visualização de dados e aprendizagem de máquina. O sistema apresenta um conjunto de funcionalidades para processar, visualizar e exportar dados. A Figura 1 ilustra os principais componentes do DCluster.

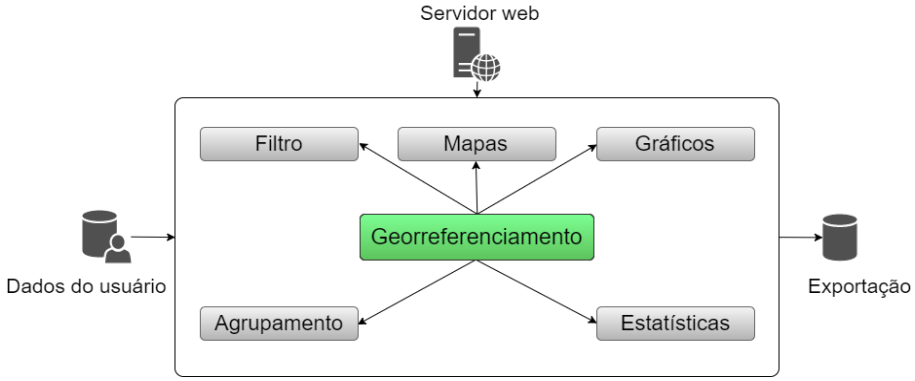
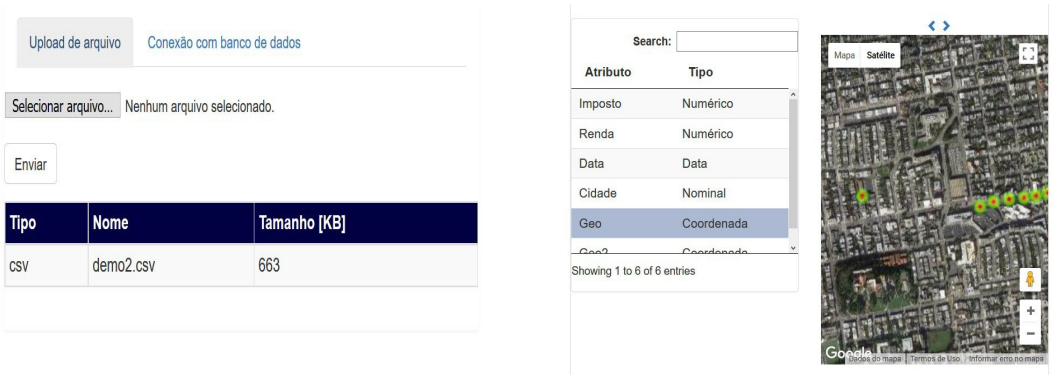


Figura 1. Diagrama de funcionalidades

3.1. Entrada de Dados

O fluxo de execução se inicia com o usuário enviando ao servidor os dados a serem processados (veja a Figura 2(a)), que podem ser provenientes de um arquivo no formato CSV (*Comma Separated Values*), ou pela conexão a um banco de dados MySQL. O sistema então identifica automaticamente os tipos de cada atributo (coluna) dos dados enviados, que podem ser: numérico, nominal ou data/hora. Vale destacar que os atributos georreferenciados são difíceis de serem identificados automaticamente, por se tratarem em geral de valores numéricos reais representando a latitude e longitude. Com isso, os atributos de latitude e longitude são inicialmente assumidos como numéricos. A indicação de quais atributos representam a latitude e longitude de uma localização deve ser feita manualmente pelo usuário, como descrito a seguir.



(a) Entrada de dados (b) Visualização de mapa

Figura 2. Telas do DCluster

3.2. Georeferenciamento

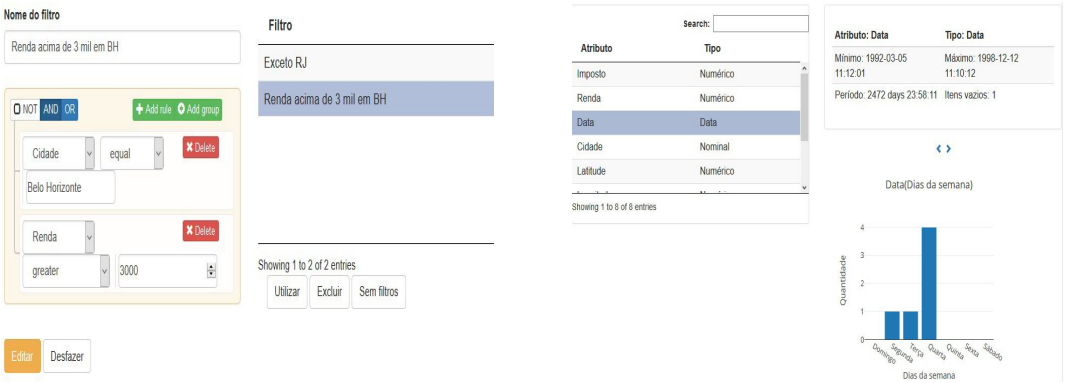
Dados georreferenciados do sistema de coordenadas são compostos pela associação de atributos que correspondam a latitude e longitude. Dependendo dos dados,

essa correspondência pode não estar representada explicitamente. Nesse caso, o usuário deve indicar quais atributos possuem a associação latitude/longitude, formando assim um atributo composto do tipo coordenada.

Uma vez definidos os tipos coordenadas, é possível visualizar os dados em um mapa (Veja Figura 2(b)). Algoritmos de aprendizado de máquina podem ser usados para realizar o agrupamento de dados do tipo coordenada. Nesse aspecto, a visualização das centróides encontradas através de pontos no mapa é essencial para que o usuário tenha um entendimento melhor sobre a sua base de dados.

3.3. Filtro

A ferramenta permite que o usuário filtre os dados, selecionando itens (linhas) com base nos valores dos atributos (veja Figura 3(a)). Essa funcionalidade é caracterizada pela criação de filtros totalmente flexíveis, não sendo necessário conhecimento prévio em expressões regulares.



(a) Filtro (b) Estatísticas

Figura 3. Telas do DCluster

Um filtro é definido pelas associações de regras, que podem ser por meio das operações lógicas *NOT*, *AND* ou *OR*. Cada regra corresponde a um atributo associado a uma operação e um valor, que pode ser numérico ou nominal, dependendo do tipo do atributo selecionado. As regras podem ser aninhadas, possibilitando que sejam elaboradas expressões lógicas complexas. O conjunto de operações para atributos numéricos são: diferente de, igual, maior, menor, maior ou igual, menor ou igual, intervalo de valores, e verificação de nulidade ou não nulidade de itens. Para atributos nominais, as operações são: igual, diferente, e verificação de nulidade ou não nulidade de itens.

3.4. Gráficos

O DCluster oferece um conjunto de gráficos para cada um dos tipos de atributos aceitos: numérico, nominal e data/hora (Veja Figura 3(b)). Para atributos numéricos, estão disponíveis os gráficos de barras e linhas. Os atributos data/hora e nominal possuem gráficos em barras, sendo que o último ainda permite o tipo de gráfico *pizza*.

3.5. Estatísticas

Além dos gráficos, são geradas estatísticas descritivas conforme o tipo de cada atributo da base de dados. Para atributos numéricos são calculados a média, variância,

mínimo e máximo. Para data/hora, são calculados os intervalos de valores, o mínimo e o máximo. No caso de atributos nominais, são geradas informações da quantidade de itens diferentes. Para todos os tipos de atributos, ainda é informada a quantidade de itens com o respectivo valor vazio ou nulo. Um exemplo é ilustrado na Figura 3(b).

3.6. Agrupamento

Para realizar o agrupamento de dados, o DCluster implementa uma versão paralela do algoritmo *K-means*. O conceito de paralelismo utilizado como base da implementação do algoritmo foi o de processos mestre/escravo, como apresentado no trabalho [Hadian and Shahrivari 2014]. O processo mestre inicialmente é responsável pela divisão dos dados em partes que são enviadas para os escravos processarem. Os processos escravos retornam as centróides encontradas para o mestre, que posteriormente utiliza esses dados recebidos como entrada do *K-means* para o processamento final. A Figura 4 apresenta um exemplo de agrupamento utilizando um atributo de coordenada.

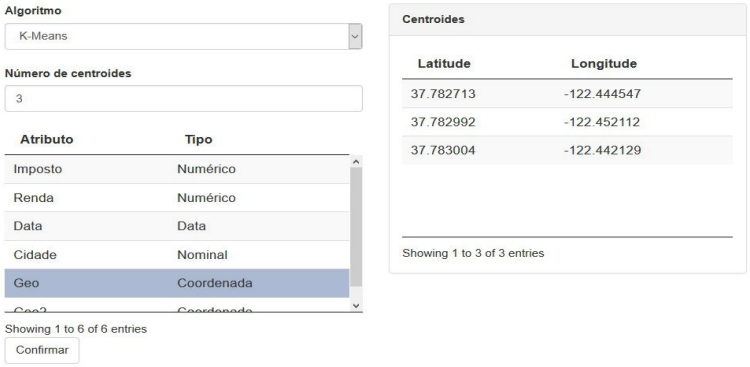


Figura 4. Agrupamento

Para avaliar a implementação paralela do agrupamento, foram realizados testes sobre duas bases de dados de diferentes tamanhos. Os resultados mostraram uma melhoria em desempenho de 26,6% para uma base de 73 MBytes e 35% para uma base de 310 MBytes, quando comparado com a versão tradicional do *k-means*.

3.7. Exportação

Após a exploração dos dados, o usuário poderá exportar os gráficos para gerar relatórios locais. O usuário também tem a opção de exportar a base de dados utilizada para o formato CSV. Essa é uma funcionalidade interessante quando se deseja obter os dados transformados e filtrados.

4. Conclusão e Trabalhos Futuros

Este trabalho apresentou o DCluster, um sistema para a análise de grandes volumes de dados com foco em georreferenciamento. O processo de desenvolvimento do DCluster tem o objetivo de unir quatro importantes aspectos: suporte a dados georreferenciados, licença acessível, usabilidade e disponibilidade via Web. Esses fatores são essenciais para a popularização da análise de dados, afim de torná-la mais acessível a pesquisadores e analistas de dados que não possuem licenças de ferramentas pagas, mas que ao mesmo tempo precisam de um conjunto abrangente de funcionalidades.

Existem vários desafios a serem tratados no DCluster. Primeiramente, serão implementadas a análise exploratória de atributos par-a-par e a customização no tratamento dos dados de entrada em diferentes formatos. Posteriormente, será feita a integração do DCluster com ferramentas de Big Data como *Hadoop* e *Spark*. Por fim, serão disponibilizados outros algoritmos para análise de dados georreferenciados.

Com o objetivo de tornar o sistema financeiramente acessível, um outro desafio se refere à definição do modelo de negócios mais adequado para o DCluster. Nosso maior interesse é torná-lo acessível principalmente a estudantes e pesquisadores.

5. Agradecimento

Este trabalho teve o apoio financeiro da Fundação de Amparo à Pesquisa do Estado de Minas Gerais (FAPEMIG).

Referências

BigML (2017). Bigml: Machine learning made easy. `https://bigml.com/`. Acessado em 01/07/2017.

Hadian, A. and Shahrivari, S. (2014). High performance parallel k-means clustering for disk-resident datasets on multi-core cpus. *The Journal of Supercomputing*, 69(2):845–863.

Hitachi (2017). Pentaho: Data integration, business analytics, and big data. `http://www.pentaho.com/`. Acessado em 01/07/2017.

Microsoft (2017a). Azure machine learning. `https://azure.microsoft.com/pt-br/services/machine-learning/`. Acessado em 01/07/2017.

Microsoft (2017b). Power bi: ferramentas do bi dde visualização de dados interativa. `https://powerbi.microsoft.com/pt-br/`. Acessado em 01/07/2017.

of Waikato, U. (2017). Weka 3: Data mining software in java. `http://www.cs.waikato.ac.nz/ml/weka/index.html`. Acessado em 01/07/2017.

Qlik (2017). Qlik: Business intelligence — ferramentas de visualização de dados. `http://www.qlik.com/pt-br`. Acessado em 01/07/2017.

SAS (2017). Sas: Software de business analytics e business intelligence. `https://www.sas.com/en_us/home.html`. Acessado em 01/07/2017.

Sisense (2017). Sisense: Business itenlligence (bi), software and analytics tools. `https://www.sisense.com/`. Acessado em 01/07/2017.

Statista (2017). Statista: the portal of statistics. `https://www.statista.com/statistics/346195/facebook-global-mobile-dau/`. Acessado em 01/07/2017.

Tableau (2017). Tableau: Análise e business intelligence. `https://www.tableau.com/pt-br`. Acessado em 01/07/2017.

Xavier, W. Z., Xavier, F. H. Z., and Marques-Neto, H. T. (2017). Visualizing and analyzing georeferenced workloads of mobile networks. In *IEEE International Conference on Pervasive Computing and Communications Workshops*, pages 306–310.

eTRC: Uma Ferramenta de e-Learning para Ensino de Cálculo Relacional de Tuplas
Sessão de Demos

Matheus Mayron Lima, Júlio Tavares
José Maria Monteiro, Angelo Brayner, Javam Machado

[1]Universidade Federal do Ceara (UFC)
Fortaleza – CE – Brazil

{matheus,julio,monteiro,brayner,javam}@dc.ufc.br

Abstract. *This paper presents a m-Learning environment, called eTRC, to assist the processes of teaching and learning Tuple Relational Calculus. The proposed tool is based on the concepts of Perception to provide a suitable environment for carrying out practical activities using different devices. The eTRC can be used in both environments, distance education and traditional education. Through eTRC apprentices may perform exercise, group activities and assessments, collaboratively, anytime and anywhere, and still receive an automatic feedback.*

Resumo. *Este artigo apresenta uma ferramenta de m-Learning, denominada eTRC [1], para auxiliar os processos de ensino e aprendizagem de Cálculo Relacional de Tuplas. A abordagem proposta baseia-se nos conceitos de Percepção para fornecer um ambiente adequado à realização de atividades práticas. A eTRC pode ser utilizada tanto em ambientes de educação à distância quanto na educação presencial. Por meio da eTRC os aprendizes podem realizar exercícios, atividades em grupo e avaliações, de forma colaborativa, a qualquer instante, em qualquer lugar e ainda receber um feedback de forma automática.*

1. Introdução

Segundo as Diretrizes Curriculares Nacionais para os cursos de graduação em Computação, Parecer CNE/CES Nº:136/2012, homologado em 28/10/2016, os conteúdos tecnológicos e básicos comuns a todos os cursos desta área incluem a disciplina de Banco de Dados (BD). Um dos objetivos de um curso de Banco de Dados é o ensino de Cálculo Relacional de Tuplas (CRT), uma linguagem de consulta formal. Esta linguagem utiliza uma expressão declarativa, baseada na lógica de primeira ordem, para especificar uma consulta sobre um determinado banco de dados. Uma expressão de CRT permite a descrição da consulta desejada sem especificar os procedimentos para obtenção dos dados desejados, sendo assim classificada como uma linguagem não-procedural. Contudo, tal consulta deve ser capaz de descrever formalmente, com exatidão, os dados que devem ser recuperados.

[1]Um vídeo sobre a ferramenta eTRC pode ser encontrado no link: http://tiny.cc/sbbd-educational-tool.

Tipicamente, uma unidade de um curso de BD pode ser organizada da seguinte forma: introdução e motivação; sintaxe do comando; lista de exemplos comentados; lista de exercícios (conceituais e práticos) e avaliação. Contudo, segundo [Prior and Lister 2004], a habilidade em programação em uma nova linguagem não pode ser adquirida sem um esforço significativo nas atividades práticas de laboratório. Logo, é fundamental oferecer ao aprendiz de um curso de BD um ambiente de laboratório para praticar listas de exercícios e realizar avaliações praticas [Lino et al. 2007b].

Neste sentido, para a realização de um curso de BD, é necessário investir em infraestrutura, *softwares* e formação docente, o que demanda tempo e recursos financeiros. Outro aspecto a ser considerado é que existem poucos softwares disponíveis para apoiar o ensino do Cálculo Relacional de Tupla. Por estes motivos, em geral, o ensino do CRL limita-se a atividades teóricas. Nestes casos, o aprendiz não vivencia a experiência de executar uma consulta especificada em CRT, sobre um determinado banco de dados, e observar os dados retornados, comparando-os com o resultado esperado. Sem atividades práticas de laboratório, os aprendizes demonstram dificuldades no aprendizado do CRT, e relatam que o conteúdo se torna abstrato.

Por outro lado, as tecnologias de computação encontram-se atualmente em franca evolução e têm influenciado fortemente os processos de ensino-aprendizagem, dando origem a diferentes modelos, tais como: *e-learning*, *b-learning* e *m-learning*. O *e-learning* é um modelo de ensino não presencial apoiado em tecnologia. Neste modelo o aprendiz se desenvolve a partir de conteúdos disponibilizados no computador e/ou Internet e em que o professor, se existir, está à distância. O sistema que inclui aulas presenciais no sistema *e-learning* é denominado *b-learning* (*blended learning*). A utilização de dispositivos móveis na educação criou um novo conceito, chamado *Mobile Learning* ou *m-Learning*. Os recursos computacionais têm sido utilizadas com sucesso para apoiar o ensino de outros conteúdos de Bancos de Dados, tais como, Álgebra Relacional [Soler et al. 2007, Litoriy and Ranjan 2010, Gorman et al. 2014] e SQL [Silveira et al. 2009, da Silveira et al. 2010, Lino et al. 2007a, Lobato and Favero 2008, Sadiq et al. 2004, Prior and Lister 2004, Prior 2003, Freire et al. 2004]. Porém, pouco se tem investigado sobre o ensino de Cálculo Relacional de Tuplas. Neste contexto, a utilização de ferramentas de *e-Learning* que forneçam suporte à realização de atividades práticas apresenta um enorme potencial para o processo de ensino e aprendizagem do Cálculo Relacional de Tupla.

Este trabalho apresenta um ambiente de *e-Learning*, denominado eTRC, para apoiar o processo ensino-aprendizagem em cursos de Cálculo Relacional de Tuplas. A solução proposta possibilita a realização de exercício, atividades em grupo e avaliações.

2. Solução Proposta

Este trabalho apresenta uma ferramenta, denominada eTRC, para apoiar o processo ensino-aprendizagem de Cálculo Relacional de Tuplas. A eTRC foi desenvolvida utilizando-se o *Spring Framework* e as APIs *Bootstrap* e *AngularJS*.

A principais características da ferramenta eTRC são:

1. Possibilita a construção de consultas (expressões) em Cálculo Relacional de Tuplas, por meio de uma interface visual;

2. Converte automaticamente uma consulta em CRT para a linguagem SQL;
3. Exibe a consulta SQL equivalente;
4. Permite executar a consulta SQL gerada automaticamente sobre uma base de dados e visualizar o seu resultado;
5. Permite a utilização de diferentes SGBDs, tais como: SQLite, PostgreSQL, MySQL e SQLServer;

Uma das tarefas mais importantes realizadas pela ferramenta eTRC é a reescrita de uma expressão em CRT para uma cláusula SQL. Para isso, utilizamos um formato denominado *SQL-Normal-Form* (SQLNF). Uma expressão (fórmula) f em CRT está em SQLNF se não contém:

1. Dupla negação;
2. Quantificador Universal;
3. Implicação e
4. Negação *Sanduíche*.

Desta forma, inicialmente, a ferramenta eTRC recebe como entrada uma consulta (expressão) f especificada em CTR. Em seguida, utilizamos a ferramenta JavaCC a fim de verificar se f realmente representa uma expressão em CRT ou se existe algum erro de sintaxe. Posteriormente, a fórmula f é convertida para SQLNF, gerando uma nova expressão $f^{'}$. Esse processo de normalização envolve: eliminar todas implicações, eliminar quantificadores universais, eliminar duplas negações e eliminar negação sanduíche. Por fim, a ferramenta eTRC verifica se é possível converter $f^{'}$ em uma cláusula SQL. Em caso afirmativo, é gerada uma cláusula SQL q equivalente a $f^{'}$ (Algoritmo 1). Caso esta conversão não seja possível, um alerta é gerado. Vale destacar que nem toda expressão em SQLNF pode ser traduzida para SQL. O processo utilizado para a normalização, assim como o processo de tradução da expressão SQLNF para SQL são descritos em [Kawash 2000]. Contudo, neste trabalho percebemos que a utilização da SQLNF não apenas facilita a tradução de uma expressão f em CRT para uma cláusula SQL, como também simplifica as validações necessárias para verificar se a expressão f é ou não segura. O Algoritmo 1 destaca as principais atividades associados ao processo de conversão de SQLNF para SQL.

Algorithm 1 Conversão de SQLNF para SQL

Require: uma expressão $f^{'}$ em SQLNF como entrada
1: A cláusula SELECT principal consiste no que está a esquerda de | em $f^{'}$
2: A cláusula FROM principal consiste nas fórmulas R(t) no escopo mais externo, isto é, não estão dentro do escopo de nenhum quantificador
3: A cláusula WHERE consiste das fórmulas F que não são do tipo R(t) e que não estão dentro do escopo de nenhum quantificador.

Como mencionado anteriormente, nem toda expressão em SQLNF pode ser convertida para SQL. Por este motivo, limitamos o escopo das expressões em CRT não permitindo que fórmulas R(t) estejam envolvidas em uma disjunção, pois isso poderia gerar fórmulas inseguras. Além disso, a tradução seria extremamente complexa de ser validada, já que a tradução de um *OR* entre duas fórmulas R(t) seria equivalente a uma operação *UNION*. Além disso, também validamos as relações e projeções utilizadas nas fórmulas sobre um banco específico, a fim de poder executar a consulta sobre este banco de dados. A Figura 1 ilustra o *Dashboard* da ferramenta eTRC.

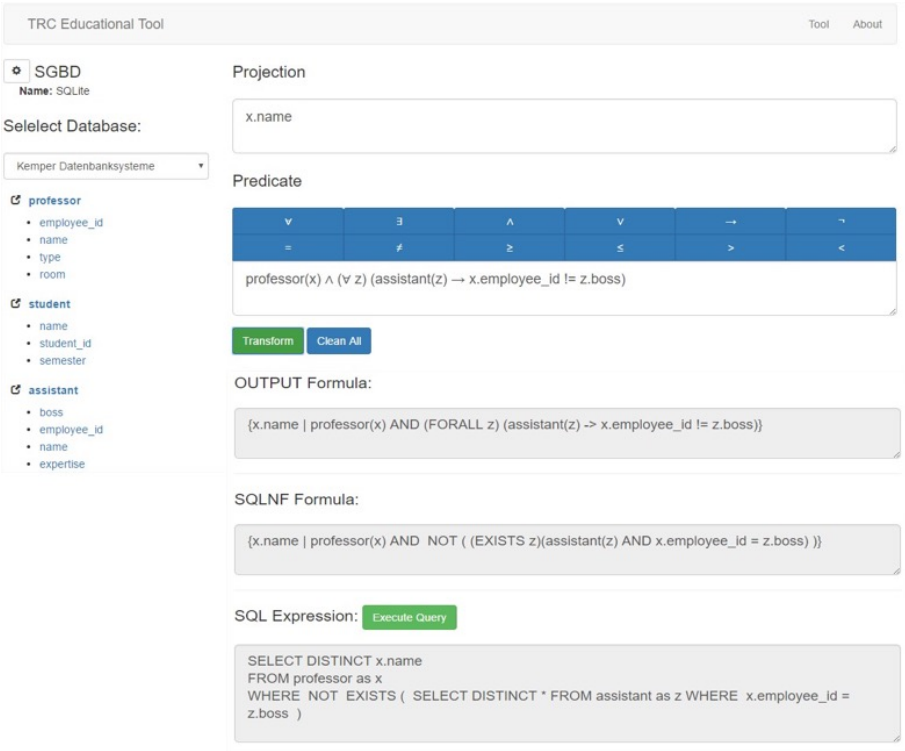


Figura 1. *Dashboard* **da Ferramenta eTRC.**

3. Trabalhos Relacionados

Algumas poucas ferramentas voltadas para facilitar o processo de ensino-aprendizagem de Cálculo Relacional de Tuplas podem ser encontradas na literatura. A ferramenta *Relational Calculus Emulator* [Bhandari 2011] é uma aplicação *desktop* desenvolvida em java que permite a elaboração de expressões em CRT (basicamente em modo texto), sua tradução para SQL e a execução dessa consulta unicamente sobre o MySQL. Além disso, essa ferramenta não valida as fórmulas em CRT, possibilitando a construção de expressões inseguras, as quais são traduzidas para SQL de forma equivocada. Em [Dietrich 1993, Dietrich et al. 1997] o autor apresenta uma ferramenta, denominada WinRDBI (Windows Relational DataBase Interpreter), implementada no Prolog, que fornece um avaliador para linguagens de consulta relacionais (Álgebra Relacional, Cálculo Relacional de Domínio, Cálculo Relacional de Tupla e SQL). Contudo, a WinRDBI não possibilita a tradução automática de expressões em CRT para cláusulas SQL e nem a utilização de SGBDs comerciais (uma vez que as bases de dados utilizadas são armazenadas internamente na aplicação Prolog).

4. Conclusões e Trabalhos Futuros

A ferramenta eTRC foi concebida para auxiliar os processos de ensino e aprendizagem de Cálculo Relacional de Tuplas. A eTRC pode ser utilizada tanto em ambientes de educação à distância quanto na educação presencial. Por meio da eTRC os aprendizes podem realizar exercícios, atividades em grupo e avaliações, de forma colaborativa, a qualquer instante, em qualquer lugar e ainda receber um *feedback* de forma automática. Como trabalho futuro pretende-se investigar se o fato da eTRC utilizar apenas fórmulas seguras proporciona algum impacto no processo de aprendizagem.

Referências

Bhandari, S. (2011). *RELATIONAL CALCULUS EMULATOR*. PhD thesis, Faculty of San Diego State University.

da Silveira, M. C., Monteiro, J. M., and de Souza, J. T. (2010). Um ambiente de m-learning para ensino da linguagem sql. In *SBIE'10: Proceedings of the XI Simpósio Brasileiro de Informática na Educação*.

Dietrich, S. W. (1993). An educational tool for formal relational database query languages. *Computer Science Education*, 4(2):157–184.

Dietrich, S. W., Eckert, E., and Piscator, K. (1997). Winrdbi: A windows-based relational database educational tool. In *Proceedings of the Twenty-eighth SIGCSE Technical Symposium on Computer Science Education*, SIGCSE '97, pages 126–130, New York, NY, USA. ACM.

Freire, J. A., Silva, A. S., Brito, S. R., Favero, E. L., and Harb, M. P. (2004). Iets: Interactive environment for teaching sql. In *Proceedings of the World Congress on Engineering and Technology Education*.

Gorman, J., Gsell, S., and Mayfield, C. (2014). Learning relational algebra by snapping blocks. In *Proceedings of the 45th ACM Technical Symposium on Computer Science Education*, SIGCSE '14, pages 73–78, New York, NY, USA. ACM.

Kawash, J. (2000). Writing complex sql queries that require universal quantifiers.

Lino, A., Favero, E. L., Harb, M. P., Brito, S. R., and Silva, A. S. (2007a). Avaliação automática de consultas sql em ambiente virtual de ensino-aprendizagem. In *CISTI'07: Proceedings of the Conferencia Ibérica de Sistemas y Tecnologias de la Informacion*.

Lino, A., Favero, E. L., and Silva, A. S. (2007b). Aplicando lógica difuso para avaliar qualitativamente o aprendiz no labsql. In *CLEI'07: Proceedings of the XXXIII Conferencia Latinoamericana de informática*.

Litoriy, R. and Ranjan, A. (2010). Implementation of relational algebra interpreter using another query language.

Lobato, A. and Favero, E. L. (2008). Aplicando rubrica para avaliar qualitativamente o estudante no labsql. In *CLEI'08: Proceedings of the XXXIV Conferencia Latinoamericana de Informática*.

Prior, J. C. (2003). Online assessment of sql query formulation skills. In *Proceedings of the Fifth Australasian Conference on Computing Education*.

Prior, J. C. and Lister, R. (2004). The backwash effect on sql skills grading. In *Proceedings of the 9th Annual SIGCSE Conference on innovation and Technology in Computer Science Education*.

Sadiq, S., Orlowska, M., Sadiq, W., and Lin, J. (2004). Sqlator: an online sql learning workbench. In *Proceedings of the 9th Annual SIGCSE Conference on innovation and Technology in Computer Science Education*.

Silveira, C., Eloy, L., and Monteiro, J. M. (2009). A query language for data access in ubiquitous environments. In *CLEI'09: Proceedings of the XXXV Conferência Latino-Americana de Informática*.

Soler, J., Boada, I., Prados, F., Poch, J., and Fabregat, R. (2007). An automatic correction tool for relational algebra queries. In *Computational Science and Its Applications - ICCSA 2007, International Conference, Kuala Lumpur, Malaysia, August 26-29, 2007. Proceedings, Part II*, pages 861–872.

Seal-DB*: Uma Ferramenta de Suporte ao Aprendizado de Banco de Dados

Gustavo Moraes[1], José de Aguiar Moraes Filho[1], Angelo Brayner[2]

[1]Universidade de Fortaleza – Unifor
Washungton Soares, 1321 – Fortaleza – CE – Brasil

[2]Universidade Federal do Ceará – UFC
Campus do Pici, BL 910 – Fortaleza – CE – Brasil

{gustavo, jaguiar}@unifor.br, brayner@dc.ufc.br

Abstract. *This paper presents a tool, called* ***Seal-DB*** *(Slightly Easy to Learn DB).* ***Seal-DB****, to support teaching database. Actually,* ***Seal-DB*** *is a relational database management system (DBMS) with a graphical and interactive interface which projects the real operation execution of DBMS components.* ***Seal-DB*** *enables students to watch on internals of the DBMS "engine". Therefore, this tool facilitates absorption of concepts and techniques of database systems.*

Resumo. *Este artigo apresenta a ferramenta* ***Seal-DB*** *(Slightly Easy to Learn DB) para apoiar o ensino de banco de dados. O* ***Seal-DB*** *é, na realidade, um sistema de gerenciamento de banco de dados (SGBD), com uma interface gráfica e interativa, que projeta a execução real de operações dos diversos componentes de um SGBD. O* ***Seal-DB*** *possibilita ao estudante uma visão do funcionamento interno da "máquina"de um sistema de gerenciamento de banco de dados. Desta forma, a ferramenta proposta facilita a absorção de conceitos e técnicas da área de banco de dados.*

1. Introdução

Os conceitos subjacentes da tecnologia de banco de dados requerem um alto grau de abstração por parte de alunos. Isto se deve à mudança de paradigma exigida pela tecnologia frente às outras áreas da computação. A arquitetura em camadas, a álgebra relacional, os protocolos de controle de concorrência e as técnicas de processamento de consultas demandam uma forma de raciocinar diferente daquela que o estudante está normalmente habituado com linguagens de programação, por exemplo.

O uso de meios áudio-visuais fornecem uma ajuda válida, porém a necessidade de uma ferramenta interativa que simule o ambiente real de execução de componentes de um sistema de gerenciamento de banco de dados é de fundamental importância na transmissão e assimilação do conhecimento de banco de dados.

O uso de ferramentas assistidas por computador fornece uma flexibilidade maior e uma proximidade da realidade de operação de banco de dados. Este tipo de ferramenta pode ser facilmente usada pelos professores e estudantes para auxiliar o entendimento de

*Uma apresentação em vídeo da ferramenta está disponível em https://youtu.be/0_MFatE_j-g

conceitos de banco de dados e, adicionalmente, prover um ambiente visual e interativo no qual o estudante se sinta inserido na tecnologia. Consequentemente, ele/ela pode reter melhor os conceitos e técnicas envolvidas na área de banco de dados.

Com o objetivo de apoiar o aprendizado de técnicas de banco de dados, apresentamos a **Seal-DB**, uma ferramenta assistida por computador para o ensino de banco de dados. A ferramenta permite a visualização do funcionamento interno dos componentes básicos de um sistema de banco de dados, como, por exemplo, o processador de consultas e os gerenciadores de transações, da área de *buffer* e de armazenamento/indexação.

Deve-se ressaltar que **Seal-DB** objetiva o ensino sobre o funcionamento interno da engrenagem de um SGBD, e não o ensino específico de modelagem de dados. Portanto, a ferramenta proposta torna-se mais apropriada ao apoio de disciplinas avançadas de banco de dados, como Técnicas de Implementação de Sistemas Banco de Dados (Banco de Dados II).

Este artigo está assim estruturado. A Seção 2 aborda os trabalhos relacionados e ferramentas de apoio ao ensino de banco de dados. A Seção 3 detalha a ferramenta proposta, **Seal-DB**, sua arquitetura, componentes e usos, juntamente com exemplos da interface gráfica. A Seção 4 conclui o artigo.

2. Trabalhos Relacionados

Pesquisadores vêm fazendo reflexões sobre o ensino de banco de dados [North 2008, Sedbrook 1994, McNeil 1990, Murray and Guimaraes 2009, Kreie and Ernst 2013, Mitra 2009, Douglas and Barker 2004]. Nesta seção discutiremos os principais trabalhos nesta área.

North [North 2008] realiza uma aferição do ensino de banco de dados usando três sistemas de banco de dados existentes no mercado, Oracle, SQL Server e MySQL. Este trabalho apenas indica o uso de um modelo de currículo chamado IS2002, avaliando este modelo pelas notas obtidas no uso de cada um dos três sistemas. Kreie e Ernst [Kreie and Ernst 2013] também não implementam uma ferramenta gráfica, mas restringem-se à aplicação da abordagem de aprendizado baseado em problema, com foco específico no ensino de conceito de modelagem de dados. Para tanto, fazem uso de sistemas de banco de dados de uso livre, disponíveis na Internet.

A proposta de Sedbrook [Sedbrook 1994] já faz uso de uma ferramenta baseada em hipertexto para o ensino de projeto de aplicações de banco de dados e modelagem de dados. Murray e Guimaraes [Murray and Guimaraes 2009] apresentam uma ferramenta de animação para o ensino de modelagem de dados, usando a técnica de Entidade-Relacionamento. McNeil [McNeil 1990], por sua vez, se preocupa com ensino de conceitos de banco de dados em outras áreas de conhecimento, especificamente estatística. Porém, ele não faz uso de nenhuma ferramenta direcionada a tal área.

Outras ferramentas que podem ser utilizadas no ensino de Banco de Dados são mais específicas ainda, pois restringem-se a sub-áreas da tecnologia de banco de dados. Por exemplo, RALT [Mitra 2009] apresenta uma ferramenta para o ensino específico de álgebra relacional, e Douglas e Barker [Douglas and Barker 2004] apresentam uma ferramenta, baseada nas linguagens *Java* e *Prolog*, para o ensino de dependências funcionais.

Nossa abordagem, ao contrário, não está vinculada a nenhum modelo teórico par-

ticular de ensino, mas suporta qualquer um deles. Comportando-se, portanto, como um substrato que pode ser utilizado sobre qualquer ferramenta de ensino. Além do mais, **Seal-DB** apresenta graficamente o ambiente de operação de um sistema de gerenciamento de banco de dados, tentando tornar transparente, com vistas ao aprendizado, o modus-operandi de um sistema de banco de dados.

3. Slightly Easy to Learn DB: Seal-DB

A ferramenta **Seal-DB** foi projetada para apoiar o ensino de técnicas de banco de dados e para fornecer uma ambiente interativo no qual o estudante possa se aproximar da realidade de um sistema de gerenciamento de banco de dados.

Seal-DB é um gerenciador de bancos de dados (SGBD) relacional completamente funcional provendo uma interface gráfica para o estudante observar o comportamento dos seus componentes arquiteturais [Härder and Rahm 2001, Garcia-Molina et al. 2008, Haerder and Reuter 1983] durante sua execução e interagir com eles. Componentes tais como sistema de armazenamento e indexação, processamento de transações, recuperação, e processamento de consultas tem sua execução exposta graficamente e o estudante pode interagir para mudar comportamento ou a velocidade de execução dos mesmos.

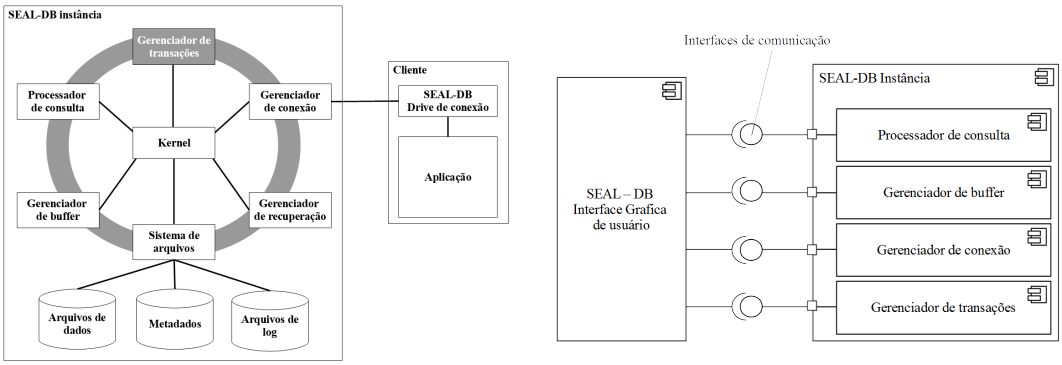
O **Seal-DB** já tem implementado a política LRU (Least Recently Used) de substituição de páginas de *buffer*, protocolo de controle de concorrência 2PL Rigoroso, e os algoritmos de junção Nested-Loop, Merge e Hash join, além do mecanismo de recuperação, após falhas de transação e de sistema. Adicionalmente, o aluno pode criar tabelas e índices, consultar e alterar dados e pesquisar os metadados usando a máquina SQL fornecida pelo **Seal-DB**.

Sendo o **Seal-DB** uma ferramenta de código aberto, desenvolvida em linguagem Java, o/a estudante tem ainda a possibilidade de substituir qualquer um dos componentes implementados por outro. Por exemplo, ele/ela poderá implementar uma outra política de alocação de páginas em *buffer* ou um outro protocolo de controle de concorrência, como o protocolo 2V2PL. A arquitetura de código interno do **Seal-DB** foi projetada para favorecer e facilitar este tipo de uso, evitando que o aluno leia muitas linhas de código para iniciar a modificação do mesmo.

3.1. Conhecendo o Seal-DB

O **Seal-DB** arquiteturalmente apresenta a conformação mostrada na Figura 1(a). O *kernel* (núcleo) é o orquestrador das interações entre os demais componentes. Por exemplo, se o componente processador de consultas desejar interagir com o componente gerenciador de transações, ele faz uma requisição ao *kernel*, que registra este pedido e abre a interação entre os dois componentes. Uma vez feito isto, os dois componentes passam a trocar mensagens diretamente entre si. Os demais componentes são já conhecidos da literatura, e comportam-se de acordo com suas funcionalidades.

Um usuário/aplicação cliente, para utilizar os serviços **Seal-DB**, deve fazer uso do *drive* de conexão do Seal-DB. Este *drive* provê toda a interface (API) para que o usuário/aplicação faça conexões com o banco, realize consultas e atualizações por meio de transações e da linguagem SQL. Ele interage com o componente gerenciador de conexão e todo o estado da conexão é mantido.

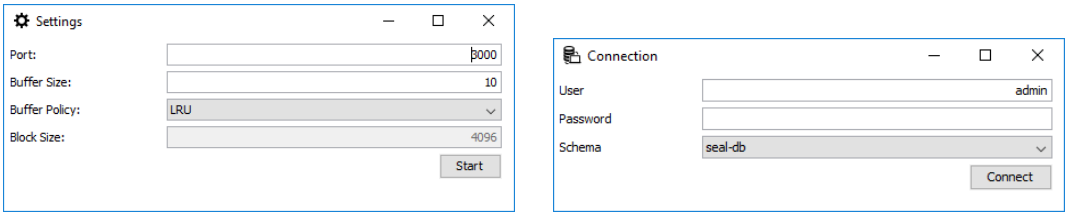


(a) Arquitetura do **Seal-DB** (b) Módulo de Interface Gráfica do **Seal-DB**

Figura 1. Arquitetura/Componentes do Seal-DB e Interface Gráfica

O módulo de interface gráfica do **Seal-DB** (Figura 1(b)) tem sido projetada de forma completamente desacoplada do SGBD. Cada componente do gerenciador fornece uma API própria a qual o módulo de interface gráfica deve-se registrar. A partir deste registro, eventos do componente podem ser capturados e mostrados graficamente. Esta decisão de projeto contribui para que se possa ter uma interface gráfica em vários ambientes, desktop ou Web. O uso do módulo de interface em sua versão desktop é mostrado nas Figuras 3 e 4.

Para inicializar o **Seal-DB** deve-se fornecer parâmetros de inicialização. Estes podem ser lidos de um arquivo de configuração ou obtidos diretamente da janela de inicialização (como na Figura 2(a)). O processo de inicialização, inicia todos os componentes, aloca memória para o buffer, realiza procedimentos de recuperação, se necessário, ativa a porta de conexão e disponibiliza o banco para uso. Em sequência, quando da interação do usuário/aplicação com o **Seal-DB**, o primeiro pedido é para realizar uma conexão (ver Figura 2(b)). Uma vez estabelecida a conexão, uma transação é aberta e o usuário/aplicação pode enviar comandos SQL. Ele pode criar tabelas e índices, alterar dados em tabelas existentes e consultar dados. Imaginemos uma consulta SQL: o comando SELECT é enviado ao **Seal-DB** para o processador de consultas. Este realiza as atividades pertinentes, como parsing do comando (usando para isto os metadados), otimização heurística e geração do plano de execução, e finalmente a execução deste plano.

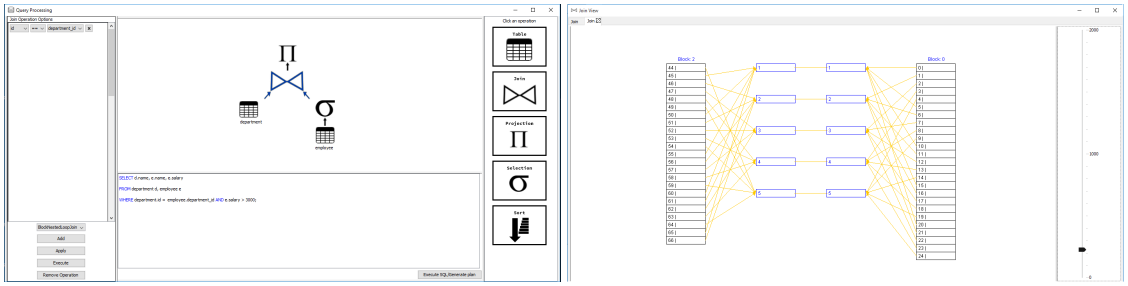


(a) Iniciando o **Seal-DB** (b) Conectando-se ao **Seal-DB**

Figura 2. Interagindo com o Seal-DB

Para efeito didático, todavia, o plano pode ser mostrado graficamente (Figura 3(a)) e o aluno tem a chance de alterá-lo, tanto em seu aspecto lógico (ordenamento e condições de junções) como em suas propriedades físicas (alterando operadores físicos). durante a

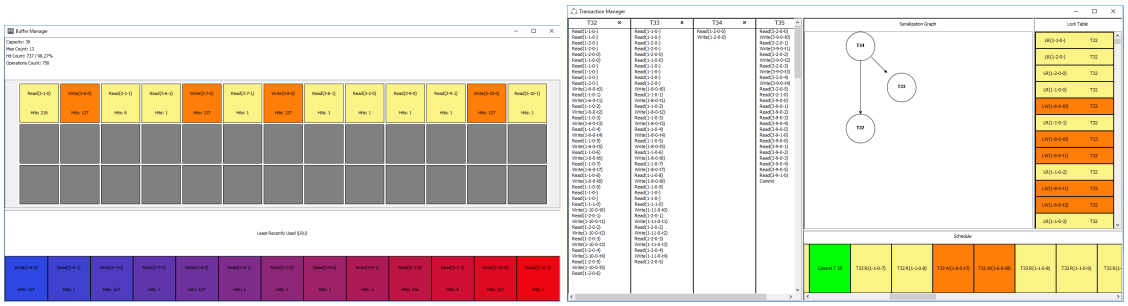
execução do plano, o aluno também tem a oportunidade de graficamente visualizar a execução de cada operação (Figura 3(b)), podendo alterar a sua velocidade de execução.



(a) Plano de Execução (b) Operador do plano em Execução

Figura 3. Plano de Consulta e sua execução

Obviamente, durante a execução de um comando SQL, haverá pedidos de bloqueios, de leitura ou de escrita, de objetos de banco (por exemplo, linhas) que serão sincronizados pelo gerenciador de transações, através do protocolo 2PL. O componente gerenciador de buffer também será demandado para buscar páginas de dados requisitadas. Para isso, este usa a LRU como política de substituição de páginas. É também fornecido ao aluno a visualização do gerenciador de buffer e de transações. Na interface visual do gerenciador de buffer, o aluno pode ver a alocação de páginas e o comportamento da política LRU incluindo os hits de páginas (Figura 4(a)). Quanto ao gerenciador de transação (Figura 4(b)), serão mostradas as operações de cada transação, o schedule produzido e o grafo de precedência. O formato das operações segue o formato $read_i(x)/write_i(x)$. Onde x assume o valor do ID do objeto sendo referenciado (ID da linha ou da página), i é o identificador da transação, e *read* ou *write* indica operação de leitura (comando SELECT do SQL) ou alteração (por exemplo, comando UPDATE do SQL).



(a) Execução do Gerenciador de Buffer (b) Execução do Gerenciador de Transações

Figura 4. Exemplo de Execução – Gerenciadores de Buffer e Transação

4. Conclusão

Neste artigo apresentamos a ferramenta **Seal-DB** de apoio ao ensino de conceitos e técnicas de banco de dados. **Seal-DB** é um gerenciador de banco de dados relacional totalmente operacional juntamente com uma interface gráfica e interativa. Um dos pontos

positivos do **Seal-DB** é prover um ambiente real de operação de banco de dados e permitir que o aluno interaja, tentando facilitar, assim, a absorção de conhecimento da área por parte dos alunos.

Referências

[Douglas and Barker 2004] Douglas, P. and Barker, S. (2004). A logic programming e-learning tool for teaching database dependency theory. In *Proceedings of the First International Workshop on Teaching Logic Programming: TeachLP*, pages 71–80, San Antonio, Texas, USA. Linköping University Electronic Press.

[Garcia-Molina et al. 2008] Garcia-Molina, H., Ullman, J. D., and Widom, J. (2008). *Database Systems: The Complete Book*. Prentice Hall Press, Upper Saddle River, NJ, USA, 2 edition.

[Haerder and Reuter 1983] Haerder, T. and Reuter, A. (1983). Concepts for implementing a centralized database management system. In Schneider, H. J., editor, *International Computing Symposium*, pages 28–6O, Nuernberg, Germany. German Chapter of ACM, B. G. Teubner, Stuttgart. (Invited Paper).

[Härder and Rahm 2001] Härder, T. and Rahm, E. (2001). *Datenbanksysteme: Konzepte und Techniken der Implementierung*. Springer, Berlin and Heidelberg, Germany, 2nd edition. doi:10.1007/978-3-642-56419-2.

[Kreie and Ernst 2013] Kreie, J. and Ernst, B. A. (2013). From database concepts to application: Use problem-based learning and oracle development tools to facilitate learning. In *Information Systems Educators Conference*, pages 1–20, San Antonio, Texas, USA. Education Special Interest Group of the AITP.

[McNeil 1990] McNeil, D. (1990). Teaching database concepts. In *International Conference on Teaching Statistics*, pages 201–208, Dunedin, New Zealand. International Association for Statistical Education.

[Mitra 2009] Mitra, P. (2009). Relational algebra learning tool. Technical report, Imperial College, London, UK.

[Murray and Guimaraes 2009] Murray, M. and Guimaraes, M. (2009). Animated courseware support for teaching database design. *Issues in Informing Science and Information Technology*, 6:201–211.

[North 2008] North, M. A. (2008). An experiment in teaching database concepts independent of software platform). *Issues in Information Systems*, IX(1):176–180.

[Sedbrook 1994] Sedbrook, T. (1994). Teaching database development with hypertext. *Journal of Information Systems Education*, 6(1):32–37.

Uma Ferramenta para Assegurar a Confidencialidade de Dados em Serviços de Armazenamento em Nuvem Sessão de Demos

Eliseu C. Branco, Roney Reis, Javam C. Machado, José Maria Monteiro
Gabriel G. Melo, Thiago de Sousa Garcia, Ricardo J. Lima
Júlio Tavares, Angelo Brayner

[1]Universidade Federal do Ceará (UFC)
Fortaleza – CE – Brazil

{eliseu,roney,javam,monteiro,gabriel,thiago,ricardo,julio,brayner}@lia.ufc.br

Abstract. *Large amounts of confidential data stored on servers in the cloud is a trend for companies looking for opportunities to reduce costs and increase the availability of their digital services. However, in cloud computing environments data control is no longer belongs to the data owner and the control belongs to the service provider, which open new challenges related to privacy, security, and confidentiality. In this context, different solutions have been proposed. Despite this, problems related to the effectiveness of these techniques in relation to the attacks, loss or theft of data have occurred in recent years. In this paper, we present a new tool, called QSM-EXTRACTION, to ensure the confidentiality of data in cloud storage services. The QSM-EXTRACTION tool is based on the fragmentation of a digital file into fragments called information objects, on the decomposition of these objects through the extraction of their characteristics and the dispersion of these characteristics in different storage services in Cloud.*

Resumo. *O armazenamento de grandes quantidades de dados em servidores na nuvem é uma tendência para as empresas que buscam oportunidades de reduzir custos e aumentar a disponibilidade de seus serviços digitais. Contudo, nestes ambientes o controle do dado deixa de ser do seu proprietário e passa a ser do provedor do serviço, o que proporciona novos desafios relacionados à privacidade, segurança e confidencialidade. Neste contexto, diferentes soluções para assegurar a confidencialidade dos dados armazenados na nuvem foram propostas. Contudo, problemas relacionados à eficácia destas soluções em relação à ataques, perda ou roubo de dados têm ocorrido nos últimos anos, causando prejuízos de milhões de dólares para empresas e clientes. Neste trabalho apresentamos uma ferramenta denominada QSM-EXTRACTION, para assegurar a confidencialidade de dados em serviços de armazenamento em nuvem. A ciência por trás dessa ferramenta utiliza conceitos da Doutrina do Ser de Hegel. A ferramenta QSM-EXTRACTION baseia-se na fragmentação de um arquivo digital em fragmentos denominados objetos de informação, na decomposição desses objetos por meio da extração de suas características (Qualidade, Quantidade e Medida) e na dispersão dessas características em diferentes serviços de armazenamento em nuvem, permitindo a posterior recuperação desses dados.*[1]

[1]Um vídeo sobre a QSM-EXTRACTION pode ser encontrado no link: `http://tiny.cc/sbbd-iobject`.

1. Introdução

Nos últimos anos, os serviços de armazenamento de dados em nuvem, tais como Dropbox, Google Drive, Amazon Cloud Drive, Box, iCloud, OneDrive, dentre outros, têm obtido grande popularidade. Assim, muitos usuários utilizam esses serviços para armazenar arquivos pessoais, tais como, fotos, documentos, além de arquivos de áudio e vídeo. Neste casos, o controle dos dados armazenados na nuvem deixa de ser do proprietário do dado e passa a ser do provedor do serviço de armazenamento. Apesar da maioria dos grandes provedores de computação em nuvem estarem passando por rigorosas auditorias para validar a conformidade de seus processos com normas e padrões de segurança, tais como ISO 27001, SSAE-16, PCI DSS, HIPAA, FedRAMP, entre outros, o problema de garantir a confidencialidade dos dados armazenados em relação ao provedor de nuvem persiste.

As premissas aplicadas ao escopo deste problema são as seguintes:

- Os usuários da nuvem devem estar identificados e autenticados para terem acesso aos dados armazenados na nuvem;
- Os provedores de nuvem devem garantir disponibilidade, confiabilidade e integridade dos dados;
- Os provedores de nuvem são considerados “honesto-curiosos", isto é, executam corretamente os protocolos de acesso aos dados, mas têm interesse de inferir e analisar dados (incluindo índices) e o fluxo de mensagens recebidas durante o protocolo de modo a aprender informações adicionais sobre os dados.

Neste trabalho, propomos uma ferramenta, denominada QSM-EXTRACTION, que assegura a confidencialidade dos dados armazenados em provedores de serviços em nuvem. A ferramenta proposta inviabiliza que estes provedores (“honesto-curiosos") tenham acesso ao conteúdo original dos dados dos clientes. De fato, o quê é armazenado nos provedores não são os dados originais do cliente, mas o resultado de uma manipulação executada sobre esses dados, de forma que é possível, posteriormente, obter o dado original a partir do que foi armazenado nos provedores. Desta forma, por meio da confidencialidade dos dados armazenados em nuvem, a ferramenta QSM-EXTRACTION assegura a privacidade dos proprietários desses dados.

2. Trabalhos Relacionados

Para tratar o problema de assegurar a confidencialidade de dados, várias abordagens foram propostas. Estas abordagens podem ser classificadas em três categorias [Samarati 2014]:

- uso de criptografia antes de enviar os dados para o servidor da nuvem;
- fragmentação vertical de dados e dispersão entre vários servidores na nuvem;
- combinação de fragmentação vertical e criptografia.

Existem questões abertas nas três abordagens previamente propostas. Em relação ao uso da criptografia, há uma troca entre a segurança dos dados e o desempenho das consultas [Kantarcıoğlu and Clifton 2005]. Além disso o uso da criptografia impõe a sobrecarga de armazenar e gerenciar chaves criptográficas. Para se utilizar a fragmentação vertical é necessário, inicialmente, definir os fragmentos, separando em fragmentos diferentes os atributos com associação sensível. Contudo, este é um problema NP-difícil [Samarati and di Vimercati 2010, Joseph et al. 2013]. Adicionalmente, consultas que envolvam fragmentos diferentes devem ser executadas no cliente, o que pode gerar uma

sobrecarga na comunicação entre o cliente e os provedores que armazenam os fragmentos. Já nas soluções mistas, que utilizam uma combinação da fragmentação vertical com a criptografia, somente os atributos sensíveis são encriptados e os atributos que possuem associação sensível são dispersos em fragmentos diferentes [Fugkeaw 2012]. Isto permite a execução de consultas sobre atributos não criptografados, mas ainda exige a decriptação dos atributos sensíveis. Além disto, esta abordagem continua com a necessidade de se determinar os fragmentos, o quê, como mencionado anteriormente, é um problema da categoria não polinomial difícil (NP-difícil) (pode ser reduzido ao problema de coloração do hipergrafo) [Aggarwal 2005].

3. A Ferramenta QSM-EXTRACTION

A ferramenta QSM-EXTRACTION baseia-se na fragmentação de um arquivo digital em fragmentos denominados objetos de informação, na decomposição desses objetos por meio da extração de suas características (Qualidade, Quantidade e Medida) e na dispersão dessas características em diferentes serviços de armazenamento em nuvem, permitindo a posterior recuperação desses dados sem perda de informação. A ideia principal da ferramenta QSM-EXTRACTION foi inspirada nas ideias do filósofo alemão Georg Wilhelm Friedrich Hegel, publicadas no livro **Enciclopédia das Ciências Filosóficas** em 1817. A Doutrina do Ser trata da Lógica do Ser aborda três conceitos principais: determinidade (qualidade), a grandeza (quantidade) e a medida.

Um objeto de informação é uma fragmento de um arquivo digital contendo 256 *bytes* sequenciais. Mais formalmente, um objeto de informação é definido da seguinte forma: Seja um arquivo F = $< b_1, b_2, \cdots, b_n >$, em que b_i é um *byte*, 1 i n e n 256. Um objeto de informação iOBJ = $< b_j, b_{j+1}, \cdots, b_{j+255} >$ onde j 1, j+255 n e iOBJ $\subset$ F.

Para exemplificar a definição de objeto de informação, considere um arquivo **F** contendo 768 *bytes*. **F** = $< b_1, b_2, \cdots, b_{768} >$, em que b_i é um *byte*. Neste caso, o arquivo **F** será fragmentado em 3 objetos de informação ($iOBJ_1$, $iOBJ_2$ e $iOBJ_3$), onde $iOBJ_1$ possui os primeiros 256 *bytes* sequenciais (de b_1 até b_{256}), $iOBJ_2$ possui os próximos 256 *bytes* sequenciais (de b_{257} até b_{512}) e $iOBJ_3$ possui os últimos 256 *bytes* sequenciais (de b_{513} até b_{612}).

Qualidade é o conjunto de *bytes* diversos que compõem um determinado objeto de informação. Seja iOBJ um objeto de informação, **Q(iOBJ)** denota a propriedade da qualidade do iOBJ. **Q(iOBJ)** é um vetor ordenado que contem m *bytes* diversos presentes em iOBJ. Mais formalmente, **Q(iOBJ)** = $\{b_1, b_2, b_3, \cdots, b_m\}$ tal que 1 b_i 256 e $i \neq j \rightarrow b_i \neq b_j$, onde b_i é um *byte* existente em iOBJ.

Quantidade contém informações sobre o número de vezes que cada *byte* diverso aparece em um determinado objeto de informação. Seja iOBJ um objeto de informação, **S(iOBJ)** denota a propriedade da quantidade (extensão) do iOBJ. **S(iOBJ)** é um vetor de 256 posições que contém, para cada *byte* diverso b_j existente em **Q(iOBJ)**, o número de vezes que b_i aparece em iOBJ. Mais formalmente, **S(iOBJ)** = $\{s_1, s_2, s_3, \cdots, s_m\}$ tal que 1 s_i 256, onde s_i representa o número de vezes que b_i aparece em iOBJ.

Seja iOBJ um objeto de informação, M(iOBJ) denota a propriedade da medida de iOBJ. **M(iOBJ[256][256])** contém, para cada *byte* diverso b_j presente em **Q(iOBJ)**, um vetor m_{b_j} que armazena as posições nas quais o *byte* b_j aparece em iOBJ. Mais formalmente, M(iOBJ) = $\{m_{b_1}, m_{b_2}, \cdots, m_{b_m}\}$, tal que, m = 256 e $1 size(m_{b_i}) 256$.

A Medida armazena as posições ocupadas pelos *bytes* no iOBJ. Essas informações estão em uma matriz esparsa de 256 linhas por 256 colunas. As linhas representam a qualidade dos *bytes* existentes no objeto de informação e as colunas representam a quantidade de posições ocupadas pelos *bytes* no objeto de informação. Caso exista apenas um byte que se repete 256 dentro do objeto de informação, os 256 elementos da linha da matriz da Medida correspondente a este byte ficarão totalmente preenchidos, e as demais linhas da matriz ficarão vazias. Na situação oposta, caso todos os 256 bytes do objeto de informação sejam diferentes uns dos outros, a primeira coluna da matriz da Medida ficará completamente preenchida, ficando vazias, as demais colunas da matriz.

Um visão geral da ferramenta QSM-EXTRACTION é apresentada na Figura 1. Neste cenário temos 5 atores: um cliente, que deseja armazenar seus arquivos pessoais na nuvem pública, três provedores de serviços de armazenamento de dados em nuvens públicas e uma Terceira Parte Confiável (TPC) que é responsável pelo processamento dos algoritmos que compõem a ferramenta QSM-EXTRACTION, os quais irão assegurar a confidencialidade dos dados, além do controle das comunicações entre o cliente e os provedores de armazenamento em nuvem.

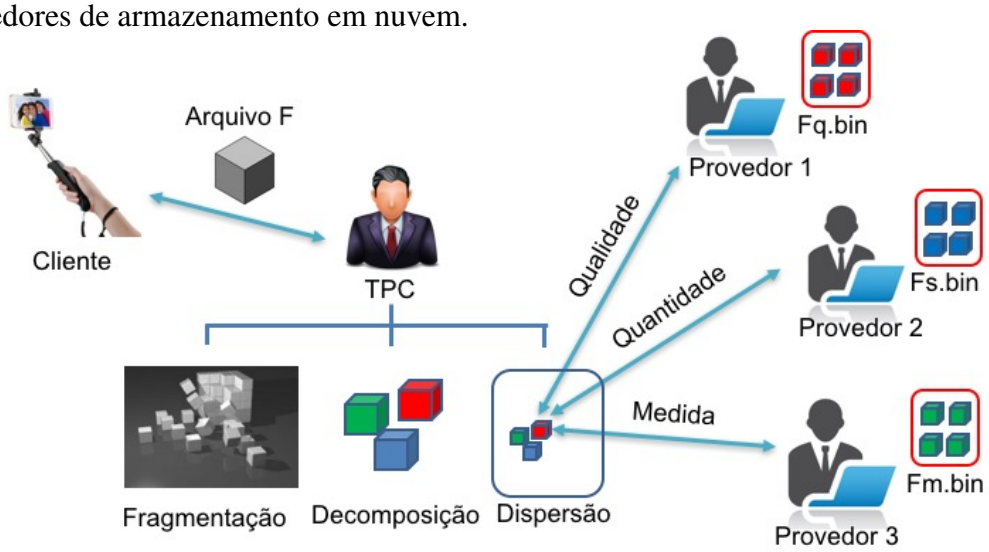


Figura 1. Visão Geral da Ferramenta QSM-EXTRACTION

As fases e etapas do processo de criação e armazenamento dos objetos de informação na infraestrutura de armazenamento da nuvem estão descritas com detalhes em [Jr. et al. 2016b, Jr. et al. 2016a].

A ferramenta QSM-EXTRACTION apresenta ainda as seguintes propriedades:

1. Generalidade: é aplicável a vários formatos de arquivos (documentos texto, som, multimídia).
2. Flexibilidade: pode ser combinada com soluções de dispersão de arquivos (ex.: Rabin, Shamir, AONTRS, etc), fragmentação vertical e criptografia (simétrica ou assimétrica).
3. Simplicidade: a confidencialidade dos dados é obtida com o uso de operações simples de substituição e transposição, que são os blocos básicos de construção de todas as técnicas criptográficas.
4. Eficiência: aplica-se a ambientes de computação em nuvem, onde as leituras são mais frequentes que as atualizações. Adicionalmente, não requer a definição de fragmentos, o que é um problema NP-difícil.

5. Gerenciabilidade: não requer o armazenamento e nem o gerenciamento de chaves criptográficas.
6. Utilidade: a recuperação das informações armazenadas na nuvem ocorre sem nenhuma perda de informação.

4. Conclusões e Trabalhos Futuros

A ferramenta QSM-EXTRACTION assegura a confidencialidade de dados em serviços de armazenamento em nuvem por meio da fragmentação de um arquivo digital em fragmentos denominados objetos de informação, na decomposição desses objetos por meio da extração de suas características (Qualidade, Quantidade e Medida) e na dispersão dessas características em diferentes serviços de armazenamento em nuvem, permitindo a posterior recuperação desses dados sem perda de informação. Desta forma, assegura-se a confidencialidade dos dados armazenados em nuvem e, por conseguinte, a privacidade dos proprietários desses dados.

Agradecimentos

Esta pesquisa foi parcialmente suportada pelo LSBD/UFC e CNPQ. Nós reconhecemos que este trabalho é um resultado parcial do projeto Gerenciamento Automático de Bancos de Dados em Nuvem suportado pelo CNPq (MCTI/CNPq 14/2014 - Universal) sob o número 446090/2014-0.

Referências

Aggarwal, C. C. (2005). On k-anonymity and the curse of dimensionality. In *Proceedings of the 31st international conference on Very large data bases*, pages 901–909. VLDB Endowment.

Fugkeaw, S. (2012). Achieving privacy and security in multi-owner data outsourcing. In *Seventh International Conference on Digital Information Management, ICDIM 2012, Macau, Macao, August 22-24, 2012*, pages 239–244.

Joseph, N. M., Daniel, E., and Vasanthi, N. A. (2013). Article: Survey on privacy-preserving methods for storage in cloud computing. *IJCA Proceedings on Amrita International Conference of Women in Computing - 2013*, AICWIC(4):1–4. Full text available.

Jr., E. C. B., Monteiro, J. M., de C. e Silva, R. R., and Machado, J. C. (2016a). A new approach to preserving data confidentiality in the cloud. In *Proceedings of the 20th International Database Engineering & Applications Symposium, IDEAS 2016, Montreal, QC, Canada, July 11-13, 2016*, pages 256–263.

Jr., E. C. B., Monteiro, J. M., Reis, R., and Machado, J. C. (2016b). A flexible mechanism for data confidentiality in cloud database scenarios. In *ICEIS 2016 - Proceedings of the 18th International Conference on Enterprise Information Systems, Volume 1, Rome, Italy, April 25-28, 2016*, pages 359–368.

Kantarcıoğlu, M. and Clifton, C. (2005). *Security Issues in Querying Encrypted Data*, pages 325–337. Springer Berlin Heidelberg, Berlin, Heidelberg.

Samarati, P. (2014). Data security and privacy in the cloud. In *Information Security Practice and Experience - 10th International Conference, ISPEC 2014, Fuzhou, China, May 5-8, 2014. Proceedings*, pages 28–41.

Samarati, P. and di Vimercati, S. D. C. (2010). Data protection in outsourcing scenarios: Issues and directions. In *Proceedings of the 5th ACM Symposium on Information, Computer and Communications Security*, ASIACCS '10, pages 1–14. ACM.

Vis4DD: A visualization system that supports Data Quality Visual Assessment

João Marcelo Borovina Josko[1], **João Eduardo Ferreira**[1]

[1]Department of Computer Science
Institute of Mathematics and Statistics (IME)
University of São Paulo (USP)
São Paulo - SP - Brazil

{jmbj,jef}@ime.usp.br

Abstract. *Data quality assessment process is essential to ensure reliable analytical outcomes. This process depends on human supervision-driven approaches since it is impossible to determine a defect based only on data. Visualization systems belong to a class of supervised tools that can make data defect pattern visible. However, their considerable design knowledge encodings and implementations provide little support design to data quality visual assessment. To cover this gap, this work reports the design approach of* $Vis4DD$ *visualization system based on patterns of data defects structures and assessment tasks. An exploratory case study used this web-based system to explore which and how visual-interactive properties facilitate visual detection of data defect.*

Key Words: Data Quality Visual Assessment, Visualization System Design, Information Visualization, Data Defect, Relational Database

1. Introduction

Data Quality Assessment process provides practical inputs to improve and keep data quality at levels required by analytical initiatives. Relevant computational models support such process, especially for data defects whose detection rules are more precise (e.g., Domain Constraint Violation [Borovina Josko et al. 2016]). Such models are based on quantitative or constraint approaches that restrict the human role in interpreting their outcomes [Dasu 2013].

On the other hand, data quality assessment process strongly depends on data context knowledge since it is impossible to confirm or refute a defect based only on data [Dasu 2013]. The context specifies the structure of meaning and relationship between data and an environment (e.g., organization departments). Hence, human supervision is essential throughout this process.

Visualization systems belong to a class of supervised approaches that combine computational capability with pattern-finding and semantic distinctions innate to human beings to permit data quality visual assessment.

Much literature has encoded design knowledge regarding visualization systems, including perceptual-driven [Ware 2004] perspectives. Related to data quality assessment, this knowledge has been encoded through certain implementations [Chen 2015] or evaluation studies [Marghescu 2007]. However, the analysis of this literature mostly reveals

concerns about *communicating* quality metrics measured on data with physical reference (e.g., a map) and little concern on how to permit *visual comprehension and assessment* of data defect structures on abstract data (e.g., sales and billing).

To address this issue, this works introduces a web visualization system (named *Visualization for Defect Detection* or $Vis4DD$) that supported an exploratory case study to identify which visual-interactive properties were more suitable for certain data defects structures on abstract data [Borovina Josko and Ferreira 2017]. Its design considered data defect structures, strategy patterns of visual assessment tasks and case study goals as inputs.

The work reported here is organized as follows: Section 2 describes requirements and design issues related to $Vis4DD$ system, while Section 3 presents its components. Section 4 conducts a comprehensive $Vis4DD$ walk-through and it briefly discusses certain case study findings. Section 5 outlines related works and Section 6 concludes this work.

2. Vis4DD Problem Domain, Requirements and Design

Data quality visual assessment denotes a nonlinear analytical process of comprehension of current data quality state mediated by visualization systems. Through interactive visual representations, data quality appraisers pursuit and correlate meanings (patterns and relationships) associated with a target defect structure until they integrate semantic evidences to confirm or refute it. Hence, absence of correspondence between a visual representation and this process goal prevents data quality appraisers from accomplishing their work.

Visualization system design is manifold since there are different techniques composition that eventually may lead to an intended result. To offer a proper support to aforementioned problem domain, most $Vis4DD$ features were based on patterns of high-level tasks. These tasks denote cognitive strategies of visual inquiry in assessing data quality according to defect structures.

The requirement analysis stage followed three steps that relied on a 6-year data quality analyst. The first step associated patterns with each data defect of case study interest according to their structure. The second step modelled and formalized high-level assessment tasks. For a complete task notation and formalization discussion, refer to [Borovina Josko and Ferreira 2017, Borovina Josko 2016]. The last step analysed all modelled tasks characteristics to identify strategy patterns in regard to data simplification, space arrangement and visual abstraction. The case study goals added another set of requirements, including color scales, homogeneous visual representation appearance and log recording.

Guided by the requirements analysis outcomes, the design stage followed three steps. The first decomposed the system into components (Section 3), while the second step selected the most appropriate interactive techniques related to each strategy pattern. For instance, in case of space arrangement pattern we selected ordering, attribute arrangement and trellis. The last step followed the case study goals to select visualization techniques of different visual variables (e.g., position, hue, saturation, size, connection) and encoding types (e.g., point, line, proportionality, directed link).

3. Vis4DD System Characteristics

Figure 1 presents $Vis4DD$ architecture style and its components communication flow. These components are based on R language due to its analytic-driven features. $Vis4DD$ visual representations used Shiny framework to compose several visualization techniques, including parallel coordinates, radial graph, heat map, scatter plot matrix and tableplot. This framework provides an easy way to build web interactive solutions through a reactive programming model. Such model permits to control how (*reactive conductors*) interface parameters (*reactive sources*) changes elements of visual representations (*reactive endpoint*).

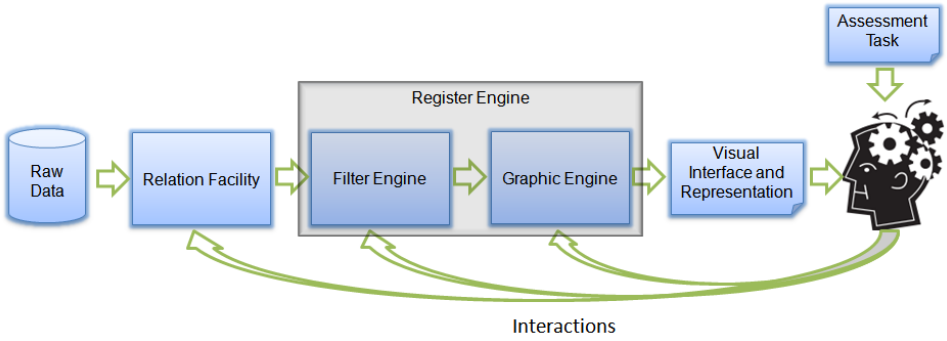


Figure 1. $Vis4DD$ **components communication (Source: Elaborated by the authors)**

The *Relation Facility* component enables managing (e.g. loading, discarding) any relation of interest in a R workspace. Relations must be first extracted from source databases as a formatted file to avoid interference in their operations and to provide a static data state for quality assessment. $Vis4DD$ provides different separators and quotes settings to load a formatted file. This operation keeps all original data values untouchable, but it executes certain structural checks (e.g., each line complies with file's header) and adjusts (e.g., convert numerical attribute into character when one of its value is not numerical).

The *Filter Engine* selects data of interest according to multiple search criteria or pointing visual items. The multiple criteria denote a set of keywords for categorical attributes or range of values for quantitative attributes. The *Graphic Engine* builds visual representations based on visualization technique, data characteristics and interactions parameters defined at visual interface. This component can handle all data or selected data regions according to Filter Engine definition. The *Register Engine* logs automatically all session interactions and their corresponding parameters. It is also in charge of taking visual representation shots when required by a data quality appraiser.

$Vis4DD$ implementation provides a rich set of visualization techniques displayed on independent visual scenes. Each scene allows certain interactions (e.g, geometric zooming, ordering, filtering, attribute arrangement, occlusion reduction) according to the visualization technique characteristics. Moreover, this implementation applies a segmented and unsegmented color scales (based on *Hue, Saturation, Lightness* model) to ensure value distinctions on dense data spaces and quantitative data isomorphism, respectively.

4. Data Quality Visual Assessment through Vis4DD

4.1. Walk-Through

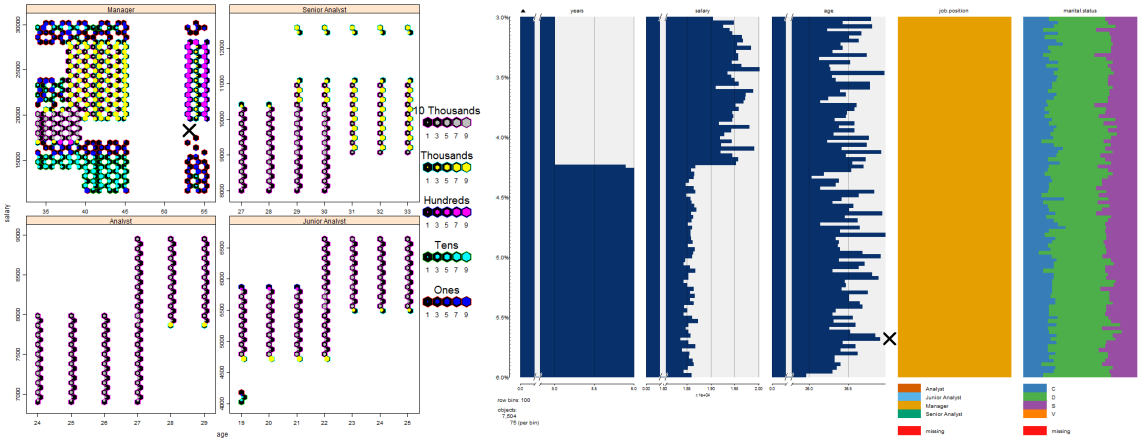
$Vis4DD$ system starts working by loading the last saved R workspace and setting global parameters. In case of an empty workspace, all visualization techniques remain unavailable until the presence of any relation.

In the early stage of data quality assessment, data quality appraisers may obtain an overall sense of all data and their patterns. They select an appropriate visualization technique to expose all data of a relation of interest. They proceed providing the corresponding target and reference attributes, and may also change default parameters of any interaction. For instance, some data quality appraiser may expose categories in different panels through trellis (e.g., Figure 2a). At the end of this setting procedure, data quality appraisers request the generation of the corresponding visual representation.

Interactions help data quality appraisers arrange data for comparison and correlation until they can isolate data regions potentially defective. In this stage, filtering and geometric zooming permit an easy and continuous refinement of data regions that are object of quality analysis. In case of strong suspicious, data quality appraisers can mark the defective data items (e.g., Figure 2b) and save the current visual representation. Otherwise, they can return to overall data view (by resetting interactions parameters) and recommences their analysis transitions until confirm or refute the presence of a data defect. At any time, a different visualization technique may be selected reusing the parameters already chosen.

4.2. Case Study Summary

Our exploratory case study used $Vis4DD$ to identify a set of relationships that exposes visual-interactive properties that permit visual assessment of different data defects. One of these data defects (atypical tuple) is outlined in this section. For a depth discussion of all data defects, refer to [Borovina Josko and Ferreira 2017, Borovina Josko 2016].



(a) Atypical tuples ($2^{\underline{nd}}$ variant) detection through compacted frequency in hue in resolution of 10^7 tuples

(b) Atypical tuples ($4^{\underline{th}}$ variant) detection through size proportional to average supported by image zooming in resolution of 10^6 tuples

Figure 2. Portions of assessment scenes of Atypical Tuple variants (Source: [Borovina Josko and Ferreira 2017, Borovina Josko 2016])

An atypical tuple deviates from the behavior of the remaining tuples of a relation for different reasons [Borovina Josko et al. 2016]. Our case study considered four atypical variants. The $1^{\underline{st}}$ and $2^{\underline{nd}}$ variants denote 0.1% and 1% of defective values in an attribute, respectively. Most visual representations permitted their assessment, but position-based visualizations were outstanding. They made easy to perceive the structures of both variants, as the atypical "manager salary" indicated in Figure 2a.

Position-based visualizations were also the best option to assess $3^{\underline{rd}}$ atypical value variant, although they required more interaction actions (e.g., filtering and point displacement). Such variant denotes atypical values interposed among data categories with certain superimposition.

The last variant ($4^{\underline{th}}$) denotes unusual combination of values considering multiples attributes. Due to its characteristics, only multidimensional visualizations permitted partial detection of atypical cases through intensive use of filter and zooming interactions. Figure 2b illustrates a $4^{\underline{th}}$ variant case involving "years", "salary" and "age" attributes.

5. Related Works

Knowledge concerning the design of visualization systems is encoded in different perspectives and depth levels. Due to the huge literature and space restrictions, this work only introduces implementation papers. For a broad discussion of such literature and its limitations in regard to data quality visual assessment, refer to [Borovina Josko and Ferreira 2017, Borovina Josko 2016].

Most implementation literature describes visualization systems based on *Quality-Aware* approach to support Data Quality Assessment [Chen 2015, Kandel et al. 2012]. Such approach optimizes visualization techniques to communicate data quality metrics (extracted by computational resources) about a particular data defect. This sort of communication is useful for those data defects that require low-moderate human supervision (e.g., Domain Constraint Violation [Borovina Josko et al. 2016]) or are visually imperceptible.

However, these optimized visualizations do not consider visual properties according to data defect structure being assessed. Hence, this nonalignment obstructs extraction and comprehension of its meanings due to the distraction effect [Ware 2004].

On the other hand, few literature describes visualization systems that support extensive use of visual exploratory analysis of meanings to determine defective data [Tennekes et al. 2013, Führing and Naumann 2007]. The supervised nature of this visual approach (named *Visual Diagnosis-Driven*) is basis for those defects whose analysis strongly depends on human supervision and contributions from computational resources (when available) are restricted. However, it is unclear *if* and *how* these aforementioned systems considered data defect structures, visual assessment tasks or backing of data quality experts to guide their design choices. Our analysis revealed a lack of proper alignment between chosen visual-interactive properties and data defects intended of assessment.

6. Conclusions

This work reports the design approach and components of $Vis4DD$ visualization system that supports quality visual assessment on abstract data. Its characteristics enabled

the analysis of which and how different visual-interactive properties facilitated (or not) the perception and comprehension of meanings in regard to data defect structures that requires high level of human supervision. Nevertheless, $Vis4DD$ neither addresses multiple coordinated views nor offers computational approaches (e.g. data mining methods) for data defects without visual evidence. As future works, it is intended to provide features to associate quality assessment outcomes to data (annotation), extract data straight from relational databases and apply creativity techniques to a broader range of data quality analysts to stimulate new ideas.

7. Acknowledgments

This work has been supported by CNPq (Brazilian National Research Council) grant number 141647/2011-6 and FAPESP (São Paulo State Research Foundation) grant number 2015/01587-0.

References

Borovina Josko, J. M. (2016). *Uso de propriedades visuais-interativas na avaliação da qualidade de dados*. PhD thesis, Universidade de São Paulo.

Borovina Josko, J. M. and Ferreira, J. E. (2017). Visualization properties for data quality visual assessment: An exploratory case study. *Information Visualization*, 16(2):93–112.

Borovina Josko, J. M., Oikawa, M. K., and Ferreira, J. E. (2016). A formal taxonomy to improve data defect description. In Gao, H., Kim, J., and Sakurai, Y., editors, *Database Systems for Advanced Applications: DASFAA 2016 International Workshops: BDMS, BDQM, MoI, and SeCoP, Dallas, TX, USA, April 16-19, 2016, Proceedings*, pages 307–320, Cham. Springer International Publishing.

Chen, C. (2015). *A system to support clerical review, correction and confirmation assertions in entity identity information management*. PhD thesis, University of Arkansas at Little Rock.

Dasu, T. (2013). Data glitches: Monsters in your data. In *Handbook of Data Quality*, pages 163–178. Springer.

Führing, P. and Naumann, F. (2007). Emergent data quality annotation and visualization. In *Proceedings of the International Conference on Information Quality (ICIQ07)*, pages 424–430, Cambridge, MA, USA.

Kandel, S., Parikh, R., Paepcke, A., Hellerstein, J. M., and Heer, J. (2012). Profiler: integrated statistical analysis and visualization for data quality assessment. In *Proceedings of the International Working Conference on Advanced Visual Interfaces*, AVI '12, pages 547–554, New York, NY, USA. ACM.

Marghescu, D. (2007). User evaluation of multidimensional data visualization techniques for financial benchmarking. In *Proceedings of the European Conference on Information Management and Evaluation*, pages 341–356. Academic Conferences Limited.

Tennekes, M., de Jonge, E., Daas, P. J., and Netherlands, S. (2013). Visualizing and inspecting large datasets with tableplots. *Journal of Data Science*, 11(1):43–58.

Ware, C. (2004). *Information Visualization: Perception for Design*. Morgan Kaufmann Publishers Inc., San Francisco, CA, USA.

32th Brazilian Symposium on Databases

wtdbd

October 2nd to 5th, 2017
Uberlândia – MG – Brazil

WORKSHOP ON THESIS AND DISSERTATIONS IN DATABASES

PROCEEDINGS

Promotion

Brazilian Computer Society – SBC
SBC Special Interest Group on Databases

Organization

Universidade Federal de Uberlândia – UFU

Workshop on Thesis and Dissertations Chair

Carina F. Dorneles

Editorial

The Workshop of Thesis and Master Dissertations in databases (WTDBD) is a traditional event co-located with the Brazilian Symposium on Databases (SBBD). This year, the event takes place in Uberlândia, Minas Gerais, gathering professors and graduate students from different Universities in Brazil to present and discuss their most recent database research results.

The WTDBD is an excellent opportunity to receive feedback upon on-going graduate work from experienced researchers. All submitted papers received, at least, three reviews. Additionally, during the Workshop, students of selected papers have the opportunity to present their work and to receive technical and scientific comments, as well as experimenting the challenge of presenting their research to an external committee. In this edition, we have nine accepted works (six masters and three doctorate works) from many different universities in Brazil.

The 2017 WTDBD Workshop chair would like to thank the students and their advisors for submitting their work to the workshop. Similarly, we are very grateful to the reviewers and the group of researchers that engaged with all their hearts in this endeavor. Their insightful comments will probably have positive impact in the development of the different research initiatives presented in the WTDBD. Finally, the WTDBD coordinator would like to thank the SBBD 2017 organizers for their outstanding support and excellent collaboration in preparing this year's edition. We wish the community an excellent workshop and success in their works.

Carina F. Dorneles, UFSC
WTDBD 2017 – CP Chair

32nd Brazilian Symposium on Databases

October 2nd to 5th, 2017
Uberlândia – MG – Brazil

Promotion

Brazilian Computer Society – SBC
SBC Special Interest Group on Databases

Organization

Universidade Federal de Uberlândia – UFU

SBBD Steering Commitee

Agma Juci Machado Traina, USP
Bernadette Lóscio, UFPE
Caetano Traina Jr., USP
Carmem Hara, UFPR
Javam Machado, UFC
Mirella M. Moro, UFMG
Vanessa Braganholo, UFF

SBBD 2017 Commitee

Steering Committee Chair
Javam Machado, UFC

Local Organization Chairs
Maria Camila N. Barioni, UFU and Humberto L. Razente, UFU

Program Committee Chair
Carmem S. Hara, UFPR

Short papers Chairs
Bernadette Lóscio, UFPE and Damires Souza, IFPB

Demos and Applications Session Chair
Daniel de Oliveira, UFF

Short Courses Chair
Vaninha Vieira, UFBA

Workshop on Thesis and Dissertations in Databases Chair
Carina Dorneles, UFSC

Tutorials Chair
Ana Carolina Salgado, UFPE

Thesis and Dissertation Contest Chair
Vânia Vidal, UFC

Workshops Chair
Fernanda Baião (UNIRIO)

Local Organization Committee

Maria Camila N. Barioni, UFU
Humberto L. Razente, UFU
José Gustavo de Souza Paiva, UFU
Marcelo Zanchetta do Nascimento, UFU
Elaine Ribeiro de Faria Paiva, UFU
João Henrique de Souza Pereira, UFU

Workshop on Thesis and Dissertations Program Committee

Altigran Soares da Silva, UFAM
Carina F. Dorneles, UFSC (Chair)
Daniel de Oliveira, UFF
Daniel Kaster, UEL
José Antonio Macêdo, UFC
Karin Becker, UFRGS
Luciano Barbosa, UFPE
Mirella M. Moro, UFMG
Renata Galante, UFRGS
Renato Fileto, UFSC
Ronaldo Mello, UFSC
Sergio Lifschitz, PUC-Rio
Valéria C. Times, UFPE
Viviane Moreira, UFRGS

Table of Contents (WTDBD)

paper:talk

Experimentação, O Terceiro Tempo...

Fábio Porto

LNCC

Produzir um resultado científico não é nada fácil. Precisa-se estudar o domínio do problema e identificar uma boa oportunidade, ainda não explorada na literatura. Em seguida, elabora-se uma estratégia e desenvolve-se uma solução. A partir dai inicia-se o terceiro tempo, a experimentação. O impulso imediatista do tipo: “já estou quase terminando pois a implementação está quase pronta" não pode estar mais longe do que precisa de fato acontecer. A fase de experimentação pode ser quase tão trabalhosa e custosa quanto suas antecessoras. Em geral, funciona em ciclos, iniciando-se pelo planejamento dos experimentos, passando pelo desenho detalhado dos alvos e processos experimentais, incluindo a obtenção de recursos, como dados e programas necessários para o experimento. Precisa-se também definir o que será medido e como será a validação. Todo o processo deve poder ser realizado por uma terceira pessoa, necessitando que seja reprodutível. Esta palestra vai explorar a fase de experimentação científica, com ênfase na área de banco de dados, e objetiva instruir os alunos de pós-graduação a prepararem satisfatoriamente a experimentação de seus trabalhos.

A middleware for storing massive RDF graphs into NoSQL

Luiz Henrique Zambom Santana , Ronaldo dos Santos Mello

[1] Federal University of Santa Catarina (UFSC)
Florianópolis - SC - Brazil

luiz.santana@posgrad.ufsc.br, r.mello@ufsc.br

Nível: Doutorado
Ingresso: Março/2015
Exame de qualificação: Junho/2017
Previsão da defesa: Março/2019

Abstract. *Governments, corporations, startups, open data initiatives and other organizations are increasingly considering RDF and SPARQL in a broad range of information management scenarios. To reduce SPARQL querying times has been the main issue for virtually all the recent RDF triplestores, yet SPARQL caching techniques have not been broadly considered. In this paper we present Rendezvous, a middleware that addresses workload-adaptive management of large RDF graphs with a caching strategy for SPARQL query results. Our middleware provides a novel RDF data partitioning approach based on a fragmentation strategy that maps RDF data into multiple NoSQL databases. Our experimental evaluation shows that the approach is promising, outperforming a recent key/value-based caching baseline.*

Resumo. *Governos, corporações, startups, iniciativas de dados abertos e outras organizações estão cada vez mais considerando RDF e SPARQL em uma ampla gama de cenários de gerenciamento de informações. Reduzir os tempos de consulta SPARQL tem sido o principal problema para praticamente todos os triplestores RDF recentes, mas as técnicas de cache SPARQL não foram amplamente consideradas. Neste artigo apresentamos Rendezvous, um middleware que aborda o gerenciamento adaptativo de carga de trabalho de gráficos RDF grandes com uma estratégia de cache para resultados de consulta SPARQL. Nosso middleware fornece uma nova abordagem de particionamento de dados RDF baseada em uma estratégia de fragmentação que mapeia dados RDF em vários bancos de dados NoSQL. Nossa avaliação experimental mostra que a abordagem é promissora, superando outra solução baseada em NoSQL chave e valor.*

1. Introduction

RDF is a standardized data model that - along with other technologies like OWL, RDFS, and SPARQL - grounds the vision of Semantic Web as an initiative to foment interlinked machine-processable information [Berners-Lee et al. 2001]. In the last decade, RDF has been increasingly used in a wide range of data management scenarios (*e.g.*, data integration, search-engine optimization, data representation, information extraction) as a resource for better understanding of complex real-world entities and their relationship. However, the current scale of data intensive applications (*e.g.*, Smart Cities, Sensor Networks, eHealth, IoT) - all of them very attractive for the Semantic Web vision -, prevents the efficient usage of existing RDF storage systems operating on a single node. In fact, such a kind of system is becoming quite a performance bottleneck giving the actual generation of massive RDF data which goes beyond its processing capacities. It raises the need for innovations in the frontier of Big Data and Semantic Web research fields.

This paper presents *Rendezvous*, a middleware that includes a novel RDF data partitioning approach with a fragmentation strategy that maps pieces of an RDF graph into NoSQL databases with different data models. Rendezvous is an RDF storage that uses the query workload to decide in which NoSQL data model is the best fit for each incoming RDF fragment. The main contributions of this work are: *(i)* a mapping of RDF data to the columnar, document and key/value data models [Sadalage and Fowler 2012]; *(ii)* a complex caching mechanism to store query results both in each server and remotely in a key/value database; *(iii)* a workload-aware partitioner based on the graph structure and, mainly, in the typical application workload; and, *(iv)* an experimental evaluation that compares our approach against a baseline (ScalaRDF [Hu et al. 2016]) by considering Apache Cassandra, MongoDB and Redis. Our high point is to process queries over large RDF graphs stored on multiple NoSQL servers with zero or a subtle amount data joining cost. An experimental evaluation shows that our middleware scales well, being able to process huge RDF datasets efficiently.

This paper is organized as follows: Section 2 presents the problem statement and explains the relevance of this PhD thesis; Section 3 presents the PhD thesis proposal, called Rendezvous; Section 4 presents our preliminary results; and Section 5 presents the methodology for reaching our objectives and concludes this paper.

2. Problem statement and relevance

The central pillar of this work is the Semantic Web, as envisioned by Tim Berners-Lee in 2001 [Berners-Lee et al. 2001]. The Semantic Web offers, as practical value, the development of applications that can handle complex human queries based not only on simple matches of raw data but also on its meaning. When Semantic Web was presented, the exponential increase of information quantity could not be foreseen by most of the specialists, but the need for data integration was already argued as one of its fundamental purposes. Thus, in the recent years, the effort of developing the Semantic Web was harvested mainly in the form of well-established standards for expressing shared meaning, defined by WWW Consortium (W3C), like Resource Description Framework (RDF) and the Simple Protocol and RDF Query Language (SPARQL).

The massive RDF is a natural transposition of Big Data on to the Semantic Web concepts. In this sense, managing big RDF graphs is gaining momentum, essentially

due to the fact that this can represent the baseline for big data analytics. As a consequence, recent surveys have highlighted the joint usage of RDF and NoSQL by Big Data applications [Ma et al. 2016]. NoSQL databases, as defined by Sadalage and Fowler [Sadalage and Fowler 2012], means database systems that use new types of storage not compatible with the traditional relational databases. NoSQL databases are usually organized into the following categories w.r.t. their data models, in this work we used the models *document*, *columnar* and *key-value*. The document data model is suitable to store semistructured data in one of the formats considered by current computational systems (*e.g.*, XML and JSON). *MongoDB* is the most popular NoSQL document database, being already tested for storing RDF as JSON documents [Ma et al. 2016]. The columnar data model aims at storing data that do not respect a rigid schema but belong to the same domain of data, *i.e.*, data instances that usually hold a standard set of properties (columns), but may have a different number of columns. Apache Cassandra is the most popular NoSQL columnar database, and it is also used for RDF storage in the CumulusRDF approach [Ma et al. 2016]. Finally, the Key/value databases are also helping the RDF solutions to scale, for instance, *ScalaRDF* [Hu et al. 2016] is a triple store that stands out for persisting data on the key/value database Redis as a distributed memory storage to speed up query performance.

There are many works proposed on this subject, denoting that scalable RDF data management is currently a very hot topic [Ma et al. 2016]. For instance, ScalaRDF [Hu et al. 2016] is a recent and relevant work that proposes a distributed in-memory RDF triplestore using Redis in a fault-tolerant store and query mechanism. In Section 4 we show that compared with ScalaRDF, *Rendezvous* polyglot capabilities for data storage, along with the *n-hop* fragmentation scheme, the workload-awareness and complex caching solution, makes our approach more suitable to dynamic query workload and offers interactive querying response time over large RDF graphs.

3. Rendezvous

Rendezvous is a middleware for partitioning and storing RDF data in multiple NoSQL database nodes. It provides a mixed-model layer, relying on a set of diverse and heterogeneous data stores, in order to provide increased performance for the applications using this layer. Figure 1 presents an overview of *Rendezvous* architecture. A *RDF-based application* issues storing or querying requests to *Rendezvous*, that is normally deployed into multiple dedicated physical node. Thus, one could integrate several applications on top of *Rendezvous* using RDF as a common data model. Another idea that is fundamental to Rendezvous is the development of a fragment-based storage which is entirely transparent to the client applications. The data flow in *Rendezvous* is most of the time in the format of fragments. As defined in the following, a fragment is essentially a part of the RDF graph to be stored and retrieved into/from NoSQL databases. Formally, an **RDF Fragment** is an RDF triple $t_{RDF} = (s, p, o)$ where $t_{RDF}.s$ is the subject, $t_{RDF}.p$ is the predicate and $t_{RDF}.o$ is the object, an RDF *fragment* is a set $F_{RDFi} = \{t_{RDF}\}$ of triples whose content may overlap with other fragment F_{RDFj}.

The data mapping is the most prominent *Rendezvous* task during a *storing* process. As stated before, our proposal is based on RDF fragmentation, so we first define our fragmentation strategy and the supported types of fragments. An RDF fragment is created when a new RDF triple to be stored (called *core triple*) is expanded with all of

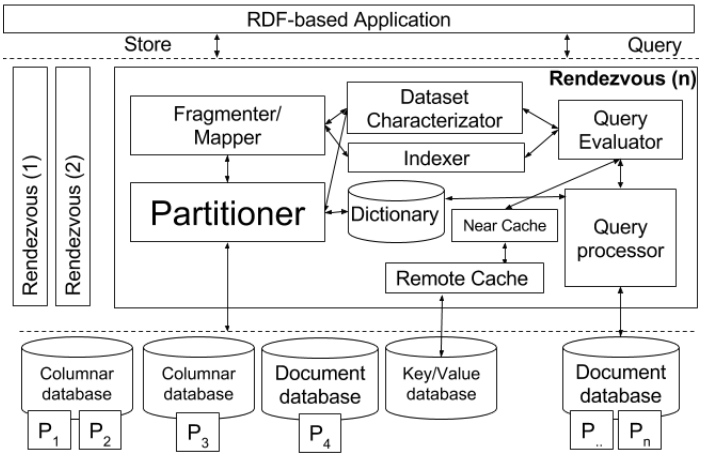


Figure 1. Rendezvous Architecture

its neighbors according to a *n-hop replication horizon*. The parameter *n* is the number of predicates from the *core triple* that our fragment expansion process considers.

The main reasoning for our fragmentation strategy is to maximize the NoSQL query capability and to minimize the cost of joining data in the *Rendezvous* node. For instance, if a core triple have to be stored, we invoke the component *Dataset Characterizator*, responsible for keeping track of the typical query workload. It manages two *in-memory hashmaps* (ii)): one for the *star-shaped* queries, where the key is a subject or a object (for subject-subject and object-object joins, respectively, and another for the *chain-shaped* queries, where the key is the predicate. The *Dataset Characterizator* also decides the size of the fragment (*n*-hop) for the star-shaped case (*n* is the longest chain among the queries in the hashmap), and for the chain-shaped case (*n* is the size of the biggest query in the hashmap). If the subject of the new triple is defined in the star-shaped hashmap, the triple is converted into a document fragment, or an existing document fragment is updated in the NoSQL document database. Otherwise, if the predicate of the new triple is defined in the chain-shaped hashmap, we create and store a columnar fragment in the columnar NoSQL database, or we update the subject and/or object of this triple in an existing column family. The same fragment can be mapped to both document and columnar fragments if the subject or object of the core triple of this fragment is in the star-shaped hashmap and its predicate is in the chain-shaped at the same time. If an RDF fragment is translated to a *document fragment* we have a mapping to a JSON document[1], which is the standard format for NoSQL document databases. Formally, a **Document Fragment** is a tuple $f_{doc} = (k_d, A)$ where $f_{doc}.k_d$ is the JSON document key and $f_{doc}.A$ is a set of attributes (key-value pairs) $f_{doc}.A = \{(k_\alpha : v)\}$, being k_α the attribute key and v a value whose domain can be atomic, a list, a set, or a tuple. If an RDF fragment is translated to a *columnar fragment*, we have a mapping to a column family which is the typical logical structuring for NoSQL column databases. Formally a **Columnar Fragment** is a tuple $f_{cf} = (k_cf, C)$ where $f_{cf}.k_cf$ is the name (key) of the column family and $f_{cf}.C$ is a set of columns (key-value pairs) $f_{cf}.C = \{(n_c : v)\}$, being n_c the column name (or column key) and v an atomic value.

[1]https://www.w3.org/TR/json-ld/

The primary purpose of *Rendezvous* is to store large RDF graphs. In such scenario, the number of RDF triples can easily surpass the performance capacity (*e.g.*, disk, memory, CPU) of a single server. When it occurs, *Rendezvous* distributes the RDF fragments among potentially many NoSQL nodes. Notice that a fragment is our smallest grain of distribution, *i.e.*, during the partitioning process we deal with fragments instead of triples. In *Rendezvous*, as defined in the following, an RDF partition is a set of fragments stored in the same physical NoSQL node, and a fragment can be replicated in multiple partitions. A **RDF Partition** P_m of an RDF graph G, such that G $\subseteq P_1 \cup P_2 \cup ...P_n$, is a set of RDF fragments $P_m = \{F_{RDFi}\}$, being not required that $P_m \cap P_t = \emptyset$, for m $\neq$ t. Nevertheless, a query can eventually access data in multiple partitions, forcing *Rendezvous* to join the data from different NoSQL nodes. Since a join operation is very costly, we try to avoid join processes by replicating fragments that are potentially part of a join. As defined in the following, if the *typical workload* for a fragment spans more than one partition, our partition scheme replicates the boundary fragments of the partition. Given SP = $\{ P_1, P_2, ..., P_n\}$ the set of RDF partitions, the **Partition Boundary** B_{P_i} of a partition $P_i \subset SP$ is the set of RDF fragments $B_{P_i} = Fb_{P_1} \cup Fb_{P_2} ... \cup Fb_{P_n}$, where $Fb_{P_k} \subset P_k$ for any k. Each $Fb_{P_i} \in B_{P_i}$ has one or more RDF triples $t_i F_{P_i} = (s_i, p_i, o_i)$ where $o_i = s_j$ where s_j is the subject of any other triple $t_j F_{P_j}$ of partition P_j where $t_j F_{P_j} = (s_j, p_j, o_j)$.

Another important task accomplished by *Rendezvous* is the query decomposition. The input SPARQL queries are analyzed by the *Query Evaluator*. It, in turn, classifies a query into *simple*, *star-shaped*, *chain-shaped* or *complex*. The *Query Evaluator* then reports the *Dataset Characterizator* in order to keep the workload metrics up-to-date, and accesses the *Dictionary* to get the partitions where the triples for these queries are located. The *star-shaped* queries (object-object or subject-subject joins) are converted to queries over NoSQL document databases. The *chain-shaped* queries (object-subject and subject-object joins) are converted to queries over NoSQL columnar databases. The processing of joins occurs when a query as a whole cannot be executed on a single partition, and it needs to be decomposed into a set of subqueries, being each subquery evaluated separately and joined at one of the *Rendezvous* nodes.

Rendezvous also provides caching management during the query decomposition. Basically, the *Cache* component verifies, during a query processing, if a fragment (or even a query result) is maintained in the *Rendezvous* cache. The cache is organized as key/value fragments, as defined in the following. **Key/Value Fragment** is a tuple $f_{kv} = (k_k v, V)$ where $f_{kv}.k_k v$ is the name of the key with the form $t_{core}.s : t_{core}.p : t_{core}.o$ (the concatenation of the t_{core} components), and the value $f_{kv}.V$ is a set $f_{kv}.V = \{(t_{1-hop})\}$, being each $t_{1-hop} \in f_{kv}.V$ a triple with a 1-hop distance of t_{core}. Architecturally, *Rendezvous* holds two types of cache: the *Near Cache*, designed as an in-memory *TreeMap* located on each *Rendezvous* server, and a *Remote Cache* maintained as a remote key-value NoSQL database (see Figure 1). When a query is issued against a *Rendezvous* server, it firstly checks its *near cache* to get all the fragments that are already available in the server. Then, the server accesses the *remote cache* to get all the missing fragments, and finally queries the document and columnar databases. When the cache is almost full, *Rendezvous* automatically evicts some keys. *Rendezvous* currently implements a cache eviction policy for *Most Recently Used (MRU)* as well as *Most Frequently Used (MFU)*.

4. Experimental Evaluation

This section presents an experimental evaluation of Rendezvous. The considered dataset comes from the *Lehigh University Benchmark (LUBM)* [Guo et al. 2005]. LUBM features an ontology for the University domain, synthetic RDF data scalable to any size, and 14 extensional queries representing a variety of properties. In our experiments, we generate a dataset with 4000 universities. The dataset size is around 100 GB and contains around 500 million triples. Regarding query complexity, we have twelve queries with joins, all of them have at least one subject-subject join, and six of them also have at least one subject-object join. We ran experiments for data loading and querying to test the performance and scalability of *Rendezvous*. The total dataset size, the loading time, and the average querying time are shown in Figure 2 (a) to (c), respectively. In Figure 2 (a) and Figure 2 (b), we notice that the dataset and the loading time grow exponentially with the number of nodes and the n-hop, ramping up from around 102 GB, loaded around 20 minutes in the 2 nodes with 2-hop configuration, to more than 500 GB loaded in more than 60 minutes in the 10 nodes with 10-hop configuration. These results can make *Rendezvous* very costly in cloud environments that charge per storage usage. We are investigating compression techniques to mitigate this problem. In Figure 2 (c), we studied both the not cached configuration and cached configuration. In the not cached configuration, the best response time was achieved with 10 nodes and a 10-hop. In such a configuration, we did not register any join outside of the fragment - with this significant fragment size all the queries could be solved within a unique NoSQL access - and each server CPU and Memory load were very small. The cached configuration presents the same response time (around 40 ms in average) regardless the hop size. The results show that the fragmentation and partition solution of *Rendezvous* is scalable and if the tradeoff for the dataset size is acceptable, the average response time can be subtle, and that cache is a good solution for scalability. In Figure 2 (d) and (e) we compared different settings of *Rendezvous*. In Figure 2 (e), we show that the systematic replication of fragments in the boundary of each partition (”b” parameter) increases the speed on the query response, without a big impact in the total size of the dataset and the data load time. This is because the size of the boundaries' triples is not very significant in such big dataset. This result is motivating new studies on the optimal boundary replication size to accelerate the query response. In Figure 2 (d) we compared our partitioning solution to the NoSQL database partition solutions. We analyzed here *Rendezvous* accessing each NoSQL database server separately (as a partition), as well as accessing the servers as a cluster (delegating the data partition to the NoSQL database). The results show that, especially for Cassandra, the graph awareness of the proposed schema plus the replication boundary lead to better performance. For MongoDB, we can conclude that the most important factor is the size of the fragment (n-hop) since a bigger fragment will typically lead to a smaller number of database accesses. Finally,in Figure 2 (f) we compared the performance of *Rendezvous* 5-hop - cached and not cached - with the recent related work *ScalaRDF*[2]. Our cached solution is 30 percent faster on average. This results is mainly due our Near Cache component - which is not present in ScalaRDF - and avoid network latency between the Rendezvous server and Redis. The downsides of this comparison are the loading time - is almost twice slower - and the dataset size - almost five times bigger.

[2]The code for ScalaRDF was found in https://github.com/xinghuayu007/ScalaRDF/

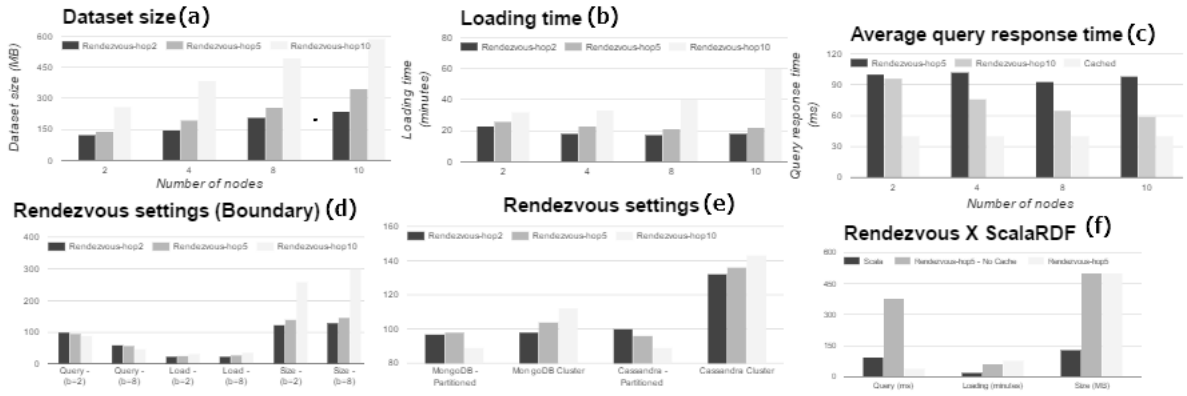


Figure 2. Summary of the Experimental Evaluation

5. Methodology and plan

This paper presented *Rendezvous*, a novel workload-aware RDF partitioning approach for the persistence of RDF data into NoSQL Databases. Our methodology can be summarized as running performance experiments to guaranty performance and scalability in large graphs. For instance our latest experiments revealed that a bigger replication boundary can accelerate the queries without a negative impact regarding storage space and load time. Besides, *Rendezvous* outperformed a recent disk-based baseline, denoting that our proposal is promising. In general, *Rendezvous* is a contribution to the problem of efficient mapping of the RDF data model to NoSQL data models. This PhD thesis is the middle of its development, so we have some future works in mind , like implementing algorithm for triples compression. The lack of this feature makes *Rendezvous* uses exponentially more storage space as the n-hop horizon grows. We also intend to consider update and deletion operations, other NoSQL types in the *Rendezvous* architecture as well as cluster capabilities in the *Rendezvous* server. With these improvements, the purpose will be to disseminate the results of this PhD thesis in important venues, like the *Conference on Innovative Data Systems Research (CIDR)*, a biennial A2 event that aims to discuss innovative research topics in the database area, and the *Special Interest Group On Management of Data (SIGMOD)* and *Very Large Databases (VLDB)*, the two most relevant conferences in the Database area.

References

Berners-Lee, T., Hendler, J., Lassila, O., et al. (2001). The semantic web. *Scientific american*, 284(5):28–37.

Guo, Y., Pan, Z., and Heflin, J. (2005). Lubm: A benchmark for owl knowledge base systems. *Web Semantics: Science, S. and Agents on the WWW*, 3(2):158–182.

Hu, C., Wang, X., Yang, R., and Wo, T. (2016). Scalardf: a distributed, elastic and scalable in-memory rdf triple store.

Ma, Z., Capretz, M. A., and Yan, L. (2016). Storing massive resource description framework (rdf) data: a survey. *The Knowledge Engineering Review*, 31(4):391–413.

Sadalage, P. J. and Fowler, M. (2012). *NoSQL distilled: a brief guide to the emerging world of polyglot persistence*. Pearson Education.

Uma Abordagem para Processamento em Memória de Operações de Seleção em Sistemas de Bancos de Dados

Diego G. Tome [1], **Marco A. Z. Alves**[1], **Eduardo C. de Almeida**[1]

[1]Programa de Pós-Graduação em Informática (PPGINF)
Universidade Federal do Paraná (UFPR) – Curitiba – PR – Brasil

{dgtome,mazalves,eduardo}@inf.ufpr.br

Nível: Mestrado **Data de admissão:** 02/2016
Data de defesa da proposta: 05/2017 **Data esperada para defesa:** 02/2018
Passos concluídos:

– Implementação das estratégias tuple/column/vector-at-a-time.
– Resultados dos cenários x86/HMC.
– Publicação: Operand size reconfiguration for big data processing in memory. Design Automation and Test in Europe (DATE), 2017.

Próximos passos:

– Implementação da novas melhorias arquiteturais.
– Resultados da proposta final.
– Escrita da dissertação de mestrado.

Abstract. *A considerable portion of the time spent during databases operation processing consists in moving data around the memory hierarchy rather than actually processing it. The emergence of smart memories, as the new Hybrid Memory Cube (HMC) allows mitigating this memory wall problem, by executing instructions directly inside the memory, reducing data movement. In this work, we discuss the processing of databases inside the HMC. We focus on the select scan operator, because the scanning of columns moves large amounts of data prior to other costly operations like joins (i.e., push-down optimization). Our preliminary results with the tuple / column / vector-at-a-time query engines show performance gains of 30% for tuple-at-a-time, 11% column-at-a-time and 6× in the vector-at-a-time execution when compared to x86.*

Resumo. *Uma considerável porção do tempo gasto no processamento de operações de banco de dados consiste na movimentação de dados pela hierarquia de memória ao invés de um real processamento. O surgimento de memórias inteligentes, como o novo Cubo de Memória Híbrido (HMC) permite mitigar esse problema de "memory wall", executando instruções diretamente dentro da memória, reduzindo a movimentação de dados. Neste trabalho, é discutido o processamento de banco de dados dentro do HMC. O foco foi no operador de seleção, já que esta operação move grandes quantidades de dados antes de outras operações dispendiosas, como junções (ou seja, otimização push-down). Os resultados preliminares com mecanismos de consulta tuple/column/vector-at-a-time apresentaram ganhos de desempenho de 30% para tuple-at-a-time, 11% em column-at-a-time e 6× na execução vector-at-a-time quando comparados com o x86.*

1. Introdução

Nas últimas décadas, a disparidade entre o desempenho do processador e a latência da memória principal cresceu fortemente, um problema bem conhecido chamado *"memory wall"* [Wulf and McKee 1995]. Esta lacuna crescente apresenta impacto direto no processamento de dados em larga escala, especialmente em bancos de dados em memória. Ao longo dos anos, os bancos de dados em memória, tornaram-se populares devido a queda no custo da DRAM e ao crescimento de sua capacidade de megabytes para terabytes. No entanto, quando aplicações com comportamento de *streaming* movem grande quantidade de dados através da hierarquia de memória, elas sofrem penalidades pela latência das interconexões e da cache.

Para atenuar o problema de movimentação de dados, a abordagem de processamento em memória (PIM - Processing-in-Memory) [Kautz 1969] inverte o fluxo de processamento de dados, movendo as instruções para onde os dados residem. Recentemente, o lançamento do Cubo de Memória Híbrida (HMC - Hybrid Memory Cube) tornou o PIM tangível para aplicações orientada a dados [Balasubramonian et al. 2014]. O HMC utiliza tecnologia de integração 3D, unindo 4 ou 8 camadas de DRAM a uma camada de lógica usando interconexões através do silício (TSVs - Through-Silicon Vias) [Beyne et al. 2008]. Essa memória é dividida em 32 partições independentes chamadas de *vaults*. A camada lógica controla os bancos DRAM e também permite a execução de instruções com alto nível de paralelismo entre os *vaults*.

O HMC pode ser usado como uma memória principal simples (substituindo as DDR 3), proporcionando em média 10× melhor desempenho e 70% menos consumo de energia. Além disso, também apresenta um conjunto de instruções de atualização formadas por operações de leitura-operação ou leitura-modificação-escrita sobre dados de 8 ou 16 bytes [Jeddeloh and Keeth 2012]. No entanto, o atual conjunto de instruções não é otimizado para operar sobre grandes volumes de dados, como os presentes em bancos de dados [Santos et al. 2017]. As memórias DRAM presentes no HMC utilizam linhas (também chamadas de páginas) de 256 bytes com política de página fechada, a qual fecha a linha após a mesma receber o último acesso, ou seja, não houver nenhum outro acesso à uma mesma linha dentro dos buffers de requisição. Isso significa que o HMC favorece acessos aleatórios (linhas diferentes). No entanto, os acessos posteriores para a mesma linha sofrerão alta latência para reabrir a linha fechada recentemente. Tais acessos tardios geralmente acontecem ao executar várias instruções de 16 bytes de dados por vez. Tais instruções podem chegar tardiamente devido a contenções nos buffers e interconexões existentes entre o processador e a memória principal.

Neste trabalho, é investigado o conjunto de instruções do HMC para executar as operações de seleção em banco de dados com uma abordagem de processamento próximo a memória [Khoram et al. 2017]. Ademais, são apresentadas as seguintes contribuições:

1. Extensão do conjunto de instruções do HMC para aproveitar ao máximo a arquitetura DRAM interna do HMC ao processar operações de seleção.
2. Análise de desempenho de execução das operações de seleção dentro do HMC.
3. Avaliação os prós e contras das extensões propostas ao executar os principais mecanismos de consulta: *tuple-at-a-time, column-at-a-time, vector-at-a-time*.

O restante deste artigo está organizado da seguinte forma: A Seção 2 apresenta os

trabalhos relacionados. A Seção 3 apresenta a proposta. A seção 4 descreve a metodologia e os resultados parciais. A seção 5 descreve as próximas etapas deste trabalho.

2. Trabalhos Relacionados

O problema de *"memory wall"* motivou diversos trabalhos nas últimas décadas com foco em algoritmos para melhor explorar os benefícios da cache [Wulf and McKee 1995, Boncz et al. 1999].

No contexto do processamento de dados próximo a memória, [Xi et al. 2015] apresenta o acelerador para SGBD externo a DRAM chamado JAFAR, o qual envia as operações de seleção de 64 bits para processamento em memórias DDR 3. Já com a utilização do HMC, o processador envia as instruções para a camada lógica sem necessidade de hardware externo, além de operar sobre vetores de até 256 bytes por instrução.

[Mirzadeh et al. 2015] apresenta outra abordagem para colocar um acelerador dentro da camada lógica do HMC, porém com o objetivo de suportar algoritmos de junção. Este trabalho remodela os algoritmos de junção de *hash* e de *merge* para minimizar o acesso de uma única palavra (por exemplo, 16 bytes) evitando o re-acesso da linha de DRAM. Por outro lado, não considera as modificações necessárias no HMC para executar tais operações para SGBDs do tipo *row-store*.

Em outros trabalhos recentes, foi considerado o uso de grandes unidades funcionais vetoriais com bancos de registradores dentro do HMC [Alves et al. 2016], estáticas ou com capacidade de reconfiguração [Santos et al. 2017]. No entanto, esse design requer um controle fino do programador para escolher o melhor tamanho do operando. Estes trabalhos consistem em uma modelagem extremamente cara em termos de hardware, porque requerem muita lógica extra para suportar processamento de vetores de 8 KB.

Diante disso, na presente dissertação propõe-se uma abordagem que estende as instruções do HMC para operar sobre dados maiores reduzindo as operações de abertura de linha DRAM e melhor explorando a camada lógica disponível.

3. Processamento de predicados no HMC

Nesta seção, é discutida a extensão proposta para a camada lógica do HMC com o objetivo de executar as operações de seleção. Primeiro, é feita uma avaliação em torno da migração do processamento da seleção para HMC e em seguida é apresentada a modificação arquitetural proposta. Para essa investigação foi implementada a consulta 06 do TPC-H em linguagem C e depois traduzida para o assembly a ser executado, como mostrado na figura 1. A consulta 06 executa a seleção em três atributos da tabela *lineitem*.

O processamento de predicados consiste em operações de comparação de registros como mostrado na Figura 1 representando a primeira comparação do atributo *l_shipdate* na consulta 06. Neste caso, é apresentado uma remodelagem no fluxo de processo da seleção, substituindo a instrução de comparação (*CMP*) x86 pela instrução HMC recíproca (*HMC_CMP*), a qual envia os bits de status da operação de volta para o processador x86. Nesta abordagem de PIM, o processador continua a buscar e desencadear as instruções, mas a execução e o acesso aos dados dependem da camada lógica do HMC, sem sofrer da latência imposta pela hierarquia de cache.

```
static double tpch_query6(int lineitemSize, double *__restrict__ p_quantity,
                          double *__restrict__ p_extendedprice,
                          double *__restrict__ p_discount,
                          int *__restrict__ p_shipdate,
                          double  sum_revenue) {
  for (int i = 0; i < lineitemSize; i++) {
    if (p_shipdate[i] >= 19940101 && p_shipdate[i] < 19950101) {
      if (p_discount[i] >= 0.05 && p_discount[i] <= 0.07) {
        if (p_quantity[i] < 24) {
          sum_revenue += p_extendedprice[i] * p_discount[i];
  }}}}
  return sum_revenue;
}
```

x86

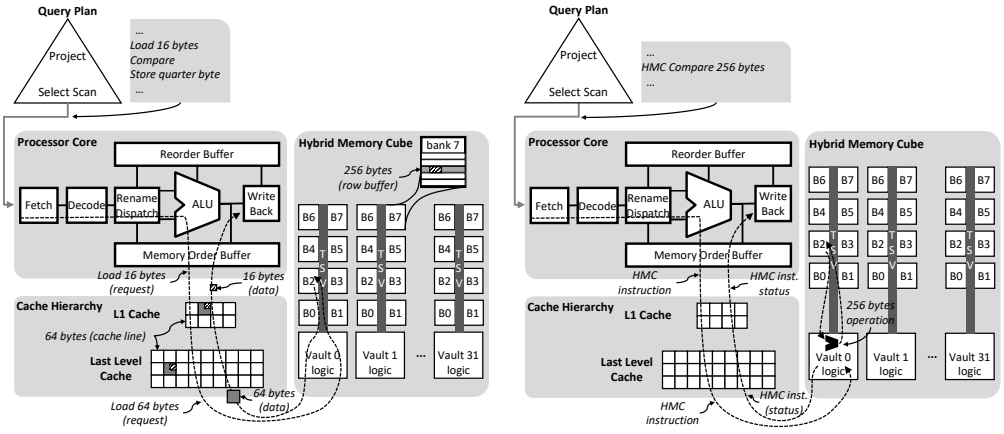
MOV	$p_shipdate, R1
CMP	$19940101, R1
JL	LOOP_START

HMC

HMC_CMP	$19940101, $p_shipdate
JL	LOOP_START

Figura 1. Implementação em linguagem C da consulta 06 do TPC-H e trecho de código assembly correspondente ao primeiro predicado.

A Figura 2(a) ilustra o processamento de predicados em uma operação de seleção com as instruções da arquitetura x86 atuais e utilizando o HMC como memória principal (reiterando que o HMC trabalha como uma pilha de DRAMs). Neste cenário de processamento, as operações subjacentes do SGBD continuam sem modificação, com o mesmo modelo de processamento de consulta. A operação de seleção envia um predicado para qualificar os dados. Dessa forma, as instruções subjacentes do x86 são alocadas no *pipeline* do processador. Na arquitetura x86 atual com instruções AVX-128, cada operação irá trabalhar sobre 16 bytes de dados. No primeiro acesso a cada linha de cache, uma falta de dados nas caches L1, L2 e L3 irá requerer um acesso à memória, já que a hierarquia de cache sempre trabalha com linhas de cache (64 bytes). Após a memória retornar a linha de cache, o processador poderá operar sobre 16 bytes a cada instrução.



(a) Processamento x86. (b) Processamento HMC.

Figura 2. Processamento de predicados com 86 vs. HMC

A Figura 2(b) descreve a proposta de seleção para operar em toda a linha da DRAM de 256 bytes. Nessa abordagem, quando uma instrução HMC é identificada no pipeline do processador, essa instrução é enviada para ser executada no HMC (sem sofrer latência da hierarquia de cache). A camada lógica solicita até 256 bytes por requisição ao banco DRAM para executar cada instrução. Uma vez que a instrução é processada, um status é enviado para o processador. A operação de seleção termina quando o processador armazena a saída usando instruções x86 seguindo o plano de consulta tradicional.

4. Experimentos

Nesta seção são apresentados o ambiente de simulação, a metodologia experimental e os resultados com a implementação da operação de seleção sobre os mecanismos de consulta *tuple/column/vector-at-a-time*.

4.1. Metodologia e Configuração

Para avaliar a proposta foi utilizado um simulador com precisão de ciclos chamado SiNUCA [Alves et al. 2015]. O SiNUCA permite modelar o nosso tamanho de operação personalizado até 256 bytes dentro do HMC, para assim, entendermos o comportamento arquitetural ao executar a operação de seleção.

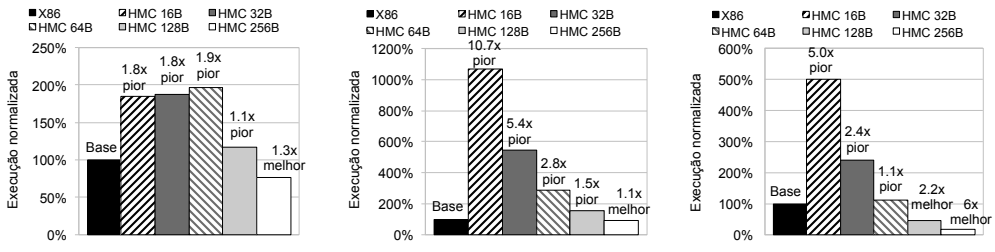
A arquitetura do cenário base é inspirada na micro-arquitetura do processador Intel Sandy Bridge referente ao x86. O Sandy Bridge foi configurado com 16 cores com conjunto de instruções AVX-128 modelando parâmetros iguais aos utilizados pelo autor na validação do simulador [Alves et al. 2015]. Os parâmetros para memória principal utilizada foram baseados na versão 2.0 do HMC [HMC-Consortium 2017]. Nos experimentos do presente trabalho, foi gerado um banco de dados TPC-H de 1 GB, além disso, foi selecionado um micro-benchmark executando a consulta 06 do TPC-H. A consulta 06 implementa expressões booleanas complexas, o que resulta na possibilidade de empurrar para baixo os predicados mais seletivos no plano de consulta.

A abordagem proposta neste trabalho utiliza o conjunto atual de instruções do HMC estendendo apenas, o tamanho dos operadores de 16 bytes para tamanhos de até 256 bytes. Neste caso, estamos executando as instruções *load* e *compare* dentro do HMC entrelaçado com as instruções x86 tradicionais.

4.2. Resultados Preliminares

Os resultados de desempenho para o cenário base x86 e HMC executando a consulta 06 são apresentados na figura 3. O HMC foi avaliado utilizando cinco tamanhos de instrução diferentes, 16, 32, 64, 128 e 256 bytes, enquanto o x86 foi configurado para 16 bytes (ou seja, o maior tamanho de instrução x86 suportado). As instruções de *store* são executadas com assistência da cache nos dois casos (x86 e HMC), enquanto as instruções *load-compare* são feitas na camada lógica do HMC quando não utilizamos o x86 puro.

Tuple-at-a-time: A figura 3(a) apresenta os resultados para a execução *tuple-at-a-time* no modelo de armazenamento *row-store*. Ao executar operações de 16 bytes de



(a) Tuple-at-a-time. (b) Column-at-a-time. (c) Vector-at-a-time

Figura 3. Tempo de execução normalizado para a operação de seleção da consulta 06 do TPC-H

largura no HMC, o tempo para processar a seleção aumentou em 84% em relação ao x86. Observou-se o mesmo resultado ao executar com operações de tamanho 32 e 64 bytes. Estas operações aumentaram o tempo de resposta em 88% e 97%, respectivamente, em comparação com o x86. Como esperado, o aumento do tempo de resposta foi devido à quantidade de acessos à DRAM para abrir e fechar a linha de dados ou seja, a política de página fechada. Além disso, observa-se que cada instrução de *load* x86 obtém uma linha de cache (64 Bytes) da DRAM e a relação entre o tamanho da tupla e o tamanho da operação faz com que cada operação processe apenas uma tupla de 64 bytes. Considerando a operação HMC de 128 bytes, o tempo de execução aumenta apenas em 16%, pois duas tuplas de 64 bytes são avaliadas por operação. O melhor cenário ocorre quando o tamanho da instrução está configurado para 256 bytes. O tempo de execução caiu em 30% em comparação com cenário base x86, já que a seleção pode processar 4 tuplas contíguas por operação sem sofrer a latência da hierarquia de cache.

Column-at-a-time: A figura 3(b) apresenta os resultados para a execução *column-at-a-time* no modelo de armazenamento *column-store*. A execução *column-at-a-time* é mais eficiente em uso de CPU do que a *tuple-at-a-time*, porque não há desperdício em *load* de colunas não utilizadas, ou seja, dados de uma mesma coluna agora estão contíguos na memória. Além disso, apenas o primeiro predicado processa todos os valores de uma determinada coluna. Os predicados remanescentes apenas processam as correspondências da coluna anterior evitando muitos acessos a DRAM.

O tempo de execução em comparação com x86 aumentou em 10,7×, 5,44×, 2,84× e 1,55× ao processar no HMC com operações de 16, 32, 64 e 128 bytes respectivamente, dado o número de re-acessos a mesma linha na DRAM. O HMC demonstrou melhorias de desempenho de 11% em relação ao cenário base x86 com o tamanho da operação em 256 bytes para tirar proveito do tamanho da linha de DRAM disponível e mitigar o impacto da política de página fechada. Observa-se que após o processamento da primeira coluna, o processador precisa buscar a máscara de bits gerada anteriormente para decidir as porções da segunda coluna que precisam ser processadas. Gerando assim, uma dependência de dados e atrasando o envio de novas instruções para o HMC.

Vector-at-a-time: A figura 3(c) apresenta os resultados para a estratégia *vector-at-a-time* no modelo de armazenamento *column-store*. A estratégia *Vector-at-a-time* é a mais eficiente em uso de CPU e com o melhor uso de cache segundo a literatura. A avaliação com o HMC de 16, 32 e 64 bytes aumentou o tempo de execução em 5×, 2,41× e 1,11×, respectivamente, em comparação com o x86. Na execução x86 atual, a cache é preenchida com muitos dados que são pouco utilizados durante o processamento de diferentes predicados. Mesmo com os esforços para melhorar a localidade espacial de cache com o uso da estratégia *vector-at-a-time*, a localidade temporal baixa para as diferentes colunas faz com que a latência devido as faltas em cache cause aumento no tempo de execução. As execuções usando operações de 128 e 256 bytes superaram o cenário base x86 em 2,19× e 6,00× respectivamente. Diferentemente da execução *column-at-a-time*, a *vector-at-a-time* avalia os predicados executando apenas *loads* que filtram os valores. Tal modelo, armazena o resultado do *bitmap* somente após o processamento do último predicado. Uma vez que a presente proposta substitui apenas as instruções de *load-compare*, menos instruções x86 são intercaladas com o HMC, permitindo que o HMC processe todos os valores de cada vetor de coluna e apenas armazene o resultado final.

5. Trabalhos Futuros

O cronograma para finalizar essa dissertação inclui uma modelagem de registradores na camada lógica do HMC para oferecer operações de *load*, *operation* e *store* em endereços distintos de memória, uma vez que atualmente apenas operações de atualização sobre o mesmo endereço são suportadas. Além disso, pretendemos incluir operações de predicação transferindo parte do controle de execução para o HMC. Por fim, pretendemos apresentar resultados de execução com outras consultas do TPC-H incluindo variações da seletividade dessas consultas.

Agradecimentos

Este trabalho foi parcialmente financiado pelo CAPES e CNPq, concedido por 441944/2016-0.

Referências

Alves, M. A. Z., Diener, M., Santos, P. C., and Carro, L. (2016). Large vector extensions inside the hmc. In *DATE*, pages 1249–1254.

Alves, M. A. Z., Villavieja, C., Diener, M., and t al. (2015). Sinuca: A validated micro-architecture simulator. *HPCC*, pages 605–610.

Balasubramonian, R., Chang, J., Manning, T., and et al. (2014). Near-data processing: Insights from a MICRO-46 workshop. *IEEE Micro*, pages 36–42.

Beyne, E., Moor, P. D., Ruythooren, W., and et al. (2008). Through-silicon via and die stacking technologies for microsystems-integration. *IEDM*, page 1–4.

Boncz, P. A., Manegold, S., and Kersten, M. L. (1999). Database architecture optimized for the new bottleneck: Memory access. In *VLDB*, pages 54–65.

HMC-Consortium (2017). Hmc specification 2.1.

Jeddeloh, J. and Keeth, B. (2012). Hybrid memory cube new dram architecture increases density and performance. In *(VLSI)*, pages 87–88.

Kautz, W. H. (1969). Cellular logic-in-memory arrays. *IEEE Trans. Comput.*, pages 719–727.

Khoram, S., Zha, Y., Zhang, J., and Li, J. (2017). Challenges and opportunities: From near-memory computing to in-memory computing. In *ISPD*, pages 43–46.

Mirzadeh, N. S., Kocberber, O., Falsafi, B., and Grot, B. (2015). Sort vs. hash join revisited for near-memory execution. In *ASBD*.

Santos, P. C., Oliveira, G. F., Tome, D. G., and et al. (2017). Operand size reconfiguration for big data processing in memory. In *DATE*, pages 710–715.

Wulf, W. A. and McKee, S. A. (1995). Hitting the memory wall: Implications of the obvious. *SIGARCH*, 23(1):20–24.

Xi, S. L., Babarinsa, O., Athanassoulis, M., and Idreos, S. (2015). Beyond the wall: Near-data processing for databases. In *DAMON*, pages 2:1–2:10.

Processamento eficiente de consultas sobre grandes volumes de dados usando arquiteturas multi-core

Frank W. R. da Silva[†,¥], Victor T. de Almeida[†,§], Vanessa Braganholo[†]
[†]Instituto de Computação, Universidade Federal Fluminense (UFF)
[¥]Universidade do Estado de Mato Grosso (UNEMAT)
[§]Petrobras S.A.
{frankwrs, valmeida, vanessa}@ic.uff.br
Programa de Pós-graduação em Ciência da Computação – IC/UFF
Nível: Mestrado
Ingresso: Março/2016
Previsão de Conclusão: Fevereiro/2018

Abstract. *Big Data Management Systems usually manage each machine as one node in the parallel query processing pipeline. In multi-core architectures, they leave several processor cores aside that could contribute to speed-up query processing. In this context, this dissertation contributes by exploring the use of all available processor cores, assessing the query processing performance in several scenarios. In particular, we use the concept of worker nodes (which are allocated in cores without disk access) and data nodes (which are allocated in cores with disk access) in the same machine using the MyriaX engine as a base platform that supports this concept. We evaluate several cluster configurations varying the amount of data and worker nodes to process queries and workloads with and without data replication factors. Preliminary results show that increasing the I/O parallelism in terms of data nodes is not always the most effective strategy, but adding worker nodes in available processing cores do improve speed-up up to certain levels. This reinforces the idea of using worker nodes in the query processing pipeline.*

1. Introdução

A necessidade de análise de grandes volumes de dados continua a crescer na indústria, governo e ciência. Isto levou empresas e organizações a buscar alternativas com melhor custo/benefício, pois sistemas tradicionais de banco de dados tornaram-se opções pouco atrativas. Motivadas pela emergência de plataformas de nuvem que dependem de *clusters* de centenas ou milhares de máquinas de propósito geral e também pelo aumento do poder de computação com a tecnologia *multi-core*, estas organizações contribuíram para o surgimento de diversos sistemas de gerência de Big Data.

Estes sistemas, em sua maioria, utilizam *clusters* de máquinas de propósito geral para processamento massivamente paralelo de consultas. As arquiteturas utilizadas por eles variam. Alguns exigem que cada máquina tenha um disco de dados acoplado, tais como ElasTras [Das et al., 2013], Impala [Bittorf et al., 2015], HadoopDB [Abouzeid et al., 2009], Apache Spark[1], Pregel [Malewicz et al., 2010], Dryad [Isard et al., 2007], Asterix [Alsubaiee et al., 2012], Presto[2], MyriaX [Wang et al., 2017] e Vertica[3]. Já o SnowFlake [Dageville et al., 2016] utiliza-se de uma base de dados compartilhada onde todos os nós acessam de forma concorrente. Outros sistemas exploram a tecnologia *multi-core* alocando vários nós em uma mesma máquina, tais como Nephele [Warneke and Kao, 2009], Amazon Redshift [Gupta et al., 2015] e Greenplum[4]. Contudo, em tais sistemas o recurso de armazenamento é fatiado ou compartilhado entre os nós, o que implica em concorrência no acesso ao disco.

A maioria destes sistemas trata cada máquina como sendo um único nó. Mesmo que cada máquina possua diversos processadores, cada um com diversos núcleos, e vários discos de dados, estes são normalmente gerenciados pelo sistema como um único recurso. No entanto, sabe-se que o uso de mais núcleos implica um melhor potencial de escalabilidade para operadores relacionais, tais como junções [Kim et al., 2009]. Contudo, em nossa pesquisa bibliográfica, não encontramos qualquer estratégia que explore totalmente os recursos disponíveis de armazenamento e processamento de forma independente em uma mesma máquina, sendo incapazes de explorar núcleos ociosos de processadores para atuar no processamento de operadores da consulta.

Com a finalidade de melhorar a eficiência do processamento paralelo de consultas em sistemas de gerência de Big Data, esta dissertação investiga o impacto da exploração de arquiteturas *multi-core* e replicação de dados no processamento paralelo de consultas. Em especial, investigamos: (i) o impacto da utilização de todos os recursos de processamento (núcleos dos processadores) disponíveis por máquina no tempo de processamento de consultas; (ii) o impacto do uso de replicação de dados, comparando o desempenho do cenário (i) com e sem replicação; e (iii) o impacto do uso das técnicas dos cenários (i) e (ii) em execução de conjuntos de consultas (*workloads*). Em nossas análises, a exploração de todos os recursos de processamento é feita por meio da alocação, em uma mesma máquina, de *worker nodes*, que realizam processamento, mas não possuem acesso a disco, e *data nodes*, que processam e acessam dados em disco. A partir da avaliação experimental desta abordagem, pretende-se definir um conjunto de

[1] https://spark.apache.org/
[2] http://prestodb.io/
[3] https://www.vertica.com/
[4] http://greenplum.org/

heurísticas a serem usadas para alocação de recursos no processamento paralelo de consultas de alto custo em sistemas de gerência de Big Data.

O restante deste artigo apresenta a abordagem proposta (Seção 2), descreve os experimentos (Seção 3), resultados e avaliações preliminares (Seção 4) e finalmente discute algumas considerações finais (Seção 5).

2. Abordagem Proposta

Para avaliar o impacto do uso de *worker nodes (wn)* e *data nodes (dn)*, explorando recursos ociosos de processamento, no desempenho de consultas, utilizamos o mecanismo de execução de consultas relacional, *shared-nothing* e paralelo MyriaX, componente do sistema de gerência de Big Data Myria [Wang et al., 2017]. Sua arquitetura é composta por um *master node*, por *worker nodes* e *data nodes*. Esta escolha foi motivada pelo fato deste mecanismo permitir o uso de *worker nodes* e *data nodes* de forma independente. MyriaX gerencia cada nó como sendo de um único tipo (*worker node*). A diferenciação entre *data node* e *worker node* acontece na atribuição de tarefas para cada nó. Nós que recebem operações de leitura e/ou escrita de dados atuam como *data nodes*. Por outro lado, nós que não recebem tarefas de leitura e/ou escrita de dados atuam como *worker nodes*. Desse modo, o MyriaX viabiliza a alocação de *data nodes* e *worker nodes* em uma mesma máquina, apesar disto nunca ter sido explorado pelo Myria. A Figura 1 ilustra uma implantação deste mecanismo com nove máquinas, sendo quatro *data nodes* e quatro *worker nodes*, além do *master node*.

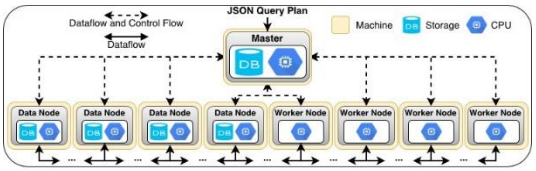


Figura 1. Arquitetura MyriaX com nove máquinas

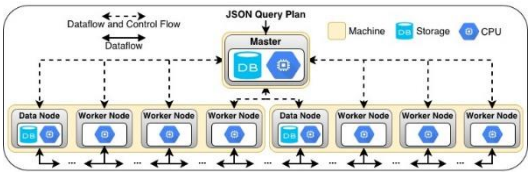


Figura 2. Arquitetura MyriaX com três máquinas

A Figura 1 ilustra o possível cenário *m:8_dn:4_wn:4* e a Figura 2 o cenário *m:2_dn:2_wn:6*. O cenário da Figura 1 utiliza 8 máquinas (*m:8*) para processamento, organizadas em 4 *data nodes* (*dn:4*) e 4 *worker nodes (wn:4)*, além do *master node*. Como os dados são particionados entre os data nodes usando uma estratégia de *round-robin* [Mehta and DeWitt, 1997], este cenário oferece um maior particionamento dos dados em relação ao cenário da Figura 2, que conta com 2 *data nodes* e 6 *worker nodes*, além do *master node*. Este último cenário, por sua vez, sofre menor influência de tráfego de rede na transmissão de dados entre nós, pois contém 2 máquinas com 1 *data node* e 3 *worker nodes* cada, aumentando assim o paralelismo intra-máquina. Ambos os cenários, após concluídas as operações de leitura de dados (realizada apenas por *data nodes*), terão 8 nós (2 *dn* + 6 *wn*) disponíveis para o processamento paralelo da consulta.

Em nossa avaliação experimental, planejamos a realização de três experimentos atacando diferentes pontos importantes ao desempenho do processamento paralelo de consultas analíticas. O primeiro (i) aplica diversas configurações e alocações de *data nodes* e *worker nodes* para processamento de consultas. O segundo experimento (ii) utiliza as configurações com melhor desempenho no primeiro e aplica diferentes fatores de replicação de dados entre *data nodes*. Por fim, o terceiro experimento (iii) utiliza *workloads* aplicadas às configurações com melhor desempenho do primeiro

experimento utilizando variações no fator de replicação de dados. Os detalhes do planejamento dos experimentos são apresentados na próxima seção.

3. Descrição dos Experimentos

Como mencionado na introdução, máquinas utilizadas no processamento paralelo de consultas podem conter recursos subutilizados, como núcleos de processadores com e sem discos acoplados, que podem ser utilizados como *worker nodes* e *data nodes* adicionais no processamento de consultas. Com base nisto, esta seção descreve o planejamento da avaliação experimental desta dissertação.

3.1. Experimento (i): explorar núcleos de processamento

Para validar a hipótese de que *"é possível diminuir o tempo de processamento de consultas através da adição de worker nodes em núcleos ociosos de processadores*", realizamos diversos experimentos preliminares em um *cluster*. O *cluster* utilizado é composto de 42 máquinas, cada uma com 2 processadores Intel Xeon *quadcore* (8 núcleos), 16 GB de RAM e 160 GB de HD. A rede é interligada com *switch Gigabit Ethernet*. O sistema operacional é o Red Hat Enterprise Linux Server versão 5.3.

Nesses experimentos, utilizamos um algoritmo que, a partir de uma quantidade de *worker nodes* e *data nodes*, gera vários cenários de distribuição e alocação de *data nodes* e *worker nodes* em máquinas do *cluster*. O script realiza 5 rodadas de consultas para cada cenário gerado. Cada rodada de consultas é executada para todos os cenários gerados antes de iniciar a rodada seguinte, ao invés de todas as 5 rodadas de consultas serem executadas de forma sequencial para cada cenário. Com esta lógica, evita-se que um cenário seja totalmente prejudicado por alguma atividade de manutenção do sistema operacional que fuja ao nosso controle durante a execução das consultas.

Cada cenário é composto por uma quantidade de máquinas (*m*), *data nodes* (*dn*) e *worker nodes* (*wn*), onde a quantidade de *worker nodes*, quando maior que zero, é dividida igualmente para cada máquina do cenário. A heurística para criar os cenários se baseou em valores de potência de 2 (iniciando em 2) para a quantidade de nós em cada implantação do serviço Myria em *cluster*, tendo como limite a quantidade máxima de núcleos de processadores disponibilizados, que para esse experimento foi um processador com 8 núcleos. Em cada implantação o algoritmo determina a quantidade para cada tipo de nó. Para *data nodes*, também foram utilizados valores de potência de 2 (iniciando em 2), tendo como limite a quantidade de máquinas, pois cada máquina possui uma única controladora de disco de dados. Isso evita a concorrência no acesso a disco. Os demais nós são alocados como *worker nodes*. Vale ressaltar que nos experimentos, *data nodes* também atuam como *worker nodes* no processamento da consulta. Em todos os cenários, uma das máquinas é reservada para atuar como *master node* e, portanto, apenas as demais 8 máquinas são consideradas pelos cenários. Cabe ressaltar que os dados são particionados entre os *data nodes* usando a estratégia de *round-robin* [Mehta and DeWitt, 1997], de forma que todos os *data nodes* acessam partições de mesmo tamanho.

A base de dados utilizada para o experimento preliminar foi a do Twitter que contém uma tabela com duas colunas (*follower* e *followee*) com valores de identificadores de usuários e representa relações entre usuários seguidores (*follower*) e seguidos (*followee*). Esta base contém 4.532.185 de tuplas e foi escolhida por ser uma base de dados real com grande volume de dados.

Foram utilizadas duas consultas referentes à base de dados do Twitter. A primeira consulta (C1) realiza auto-junção entre as colunas citadas da base, com a finalidade de identificar relações entre usuários onde ambos seguem um ao outro. A segunda consulta (C2) conta com uma operação de junção a mais e identifica triângulos entre usuários. Um triângulo se forma quando um usuário *A* segue um usuário *B*, que por sua vez segue um usuário *C*, que segue o usuário *A*. De fato, esta é uma consulta que ainda apresenta desafios para a comunidade científica quando os dados não cabem na memória [Hu et al., 2013]. Na base de dados utilizada no experimento, a primeira e segunda junções retornam 2.045.216.395 e 89.084.893 de tuplas, respectivamente. Durante as submissões das consultas, o algoritmo captura o tempo que cada consulta leva para executar, elimina o maior e menor tempo, e calcula a média dos tempos restantes. Isso é feito para cada cenário.

Para dar sequência aos experimentos discutidos nesta seção, planejamos utilizar *workloads* para avaliar o comportamento da abordagem proposta com execução de consultas que utilizam um número maior de tabelas e relações.

3.2. Experimento (ii): aplicar fator de replicação de dados

Planos de execução de consultas analíticas, em sua maioria, envolvem múltiplas operações de leitura de dados. Em nossa abordagem, essas operações são executadas por *data nodes*. Neste sentido, para validar a hipótese de que "*é possível diminuir o tempo de processamento de consultas que realizam mais de uma leitura de dados na mesma tabela, separando esta leitura em data nodes distintos, através da replicação de dados, aumentando o paralelismo de I/O*", planejamos reproduzir a execução das consultas utilizadas no experimento (i) utilizando diversos fatores de replicação de dados. Para tal, planejamos utilizar cenários com melhores desempenhos identificados no experimento (i). Os resultados serão avaliados e comparados com os resultados do primeiro experimento, onde não usamos replicação de dados.

3.3. Experimento (iii): utilizar *workloads*

Sistemas de gerência de Big Data atuam diretamente com *workloads*. Tais *workloads* envolvem múltiplas consultas diferentes executando em paralelo, algumas na mesma máquina, sobre uma massiva quantidade de dados. Neste sentido, o objetivo deste experimento é validar a hipótese de que "*é possível obter melhor eficiência no processamento de workloads utilizando configurações de worker nodes e data nodes e replicação de dados adequadas"*. Para tal, planejamos utilizar cenários de destaque do experimento (i) com fatores de replicação de dados identificados no experimento (ii) para execuções em *cluster*. Será realizado um estudo prévio para determinar qual *workload* será utilizada neste experimento.

4. Resultados e Avaliações Preliminares

Nesta seção são apresentados e avaliados os resultados preliminares obtidos com o experimento (i), pois os experimentos (ii) e (iii) estão em fase de planejamento. Por questões de espaço, os resultados obtidos com a consulta C2 não são apresentados neste artigo (apresentamos apenas uma breve avaliação dos resultados). Os resultados e avaliação completa são apresentados em [Silva et al., 2017]. Para fins de identificação, cenários padrão do MyriaX estão nomeados como *Baseline* e cenários que se baseiam na hipótese desse trabalho são nomeados *Avaliação*.

Consulta C1 (auto-junção). A Figura 3 apresenta os resultados para a consulta C1, utilizando cenários com até 8 máquinas agrupados pela quantidade total de nós (8, 16, 32 e 64). Os tempos são medidos em segundos.

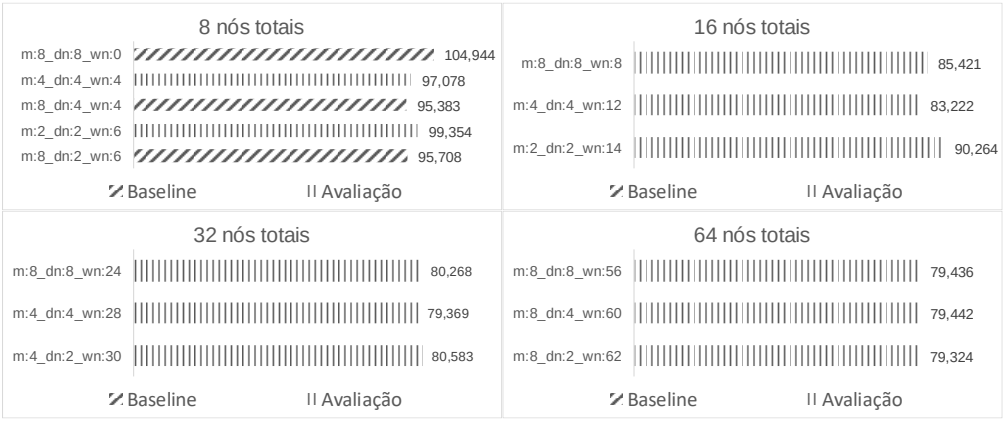


Figura 3. Tempo médio (em segundos) da consulta C1 em cenários com até 8 máquinas agrupados pela quantidade total de nós

Como apresenta a Figura 3, cenários com 8 nós totais apresentaram o maior tempo médio para execução da consulta C1. O cenário que não utiliza *worker nodes* apresentou o maior tempo médio dentre estes cenários, justificado pela execução sequencial dos operadores da consulta. Os demais cenários com quantidade totais de 16, 32 e 64 nós apresentam melhora gradativa no tempo médio do processamento da consulta C1, respectivamente, e apresentam resultados semelhantes dentro destas quantidades de nós totais. Isto porque estes cenários utilizam de maior quantidade de *worker nodes* que passam a processar pequenas quantidades de dados e aumentam o tráfego de rede. Desta forma, a Figura 3 evidencia que, para a consulta C1, é possível atingir um determinado desempenho com menor quantidade de máquinas dimensionando *workers nodes* e *data nodes* e explorando recursos de núcleos de processamento e discos de dados de uma mesma máquina. Por exemplo, o cenário *m:8_dn:4_wn:4* utiliza 8 máquinas e apresentou aceleração de apenas 1,05x e 1,2x em relação aos cenários *m:2_dn:2_wn:14* e *m:4_dn:4_wn:28* que utilizam 2 e 4 máquinas, respectivamente. É importante acrescentar que em experimentos adicionais, para os cenários com até 2 máquinas *m:2_dn:2_wn:0* e *m:2_dn:2_wn:14*, a simples adição de *worker nodes* implicou em aceleração de até 2,92x quando se explora todos os núcleos de processamento disponíveis.

Consulta C2 (triângulos). Para esta consulta, o uso de todo o recurso disponível em dada quantidade máquinas piora o desempenho de processamento. Por outro lado, cenários *Avaliação* com quantidades iguais de *data nodes* e *worker nodes* apresentaram melhores desempenhos, com aceleração de até 1,07x.

5. Considerações Finais

Esta dissertação avalia o uso de *worker nodes*, que participam do *pipeline* de processamento da consulta e não acessam dados em disco, e *data nodes*, que acessam dados em disco e também atuam como *worker nodes*, alocados em uma mesma máquina. O mecanismo de execução de consulta relacional, *shared-nothing* e paralelo MyriaX foi utilizado como plataforma base dos experimentos preliminares por permitir o uso desta abordagem. Em especial, nesse trabalho pretendemos investigar: (i) o impacto da utilização de todos os recursos de processamento disponíveis por máquina

no tempo de processamento de consultas através da alocação de *worker nodes*; (ii) o impacto do uso de replicação de dados, comparando o desempenho do cenário (i) com e sem replicação; e (iii) o impacto do uso das técnicas dos cenários (i) e (ii) em execução de *workloads*. Cada experimento da fase de avaliação experimental será descrito e avaliado em um ou mais artigos a serem submetidos, contribuindo para comunidade de pesquisa em bancos de dados paralelos. O experimento (i), como mostram as Seções 3 e 4, foi executado e avaliado. Os resultados deste foram descritos em um artigo completo [Silva et al., 2017] aceito para publicação no Simpósio Brasileiro de Banco de Dados (SBBD) 2017. A partir da avaliação experimental, pretende-se definir um conjunto de heurísticas a serem usadas para alocação de recursos no processamento paralelo de consultas de alto custo em sistemas de gerência de Big Data.

Referências

Abouzeid, A., Bajda-Pawlikowski, K., Abadi, D., Silberschatz, A. and Rasin, A. (2009). HadoopDB: An Architectural Hybrid of MapReduce and DBMS Technologies for Analytical Workloads. *Proceedings of the VLDB Endowment (PVLDB)*, v. 2, n. 1, p. 922–933.

Alsubaiee, S., Behm, A., Grover, R., et al. (2012). ASTERIX: Scalable Warehouse-Style Web Data Integration. *International Workshop on Information Integration on the Web*, p. 1–4.

Bittorf, M., Bobrovytsky, T., Erickson, C. C. A. C. J., et al. (2015). Impala: A Modern, Open-Source SQL Engine for Hadoop. In *Conference on Innovative Data Systems Research (CIDR)*.

Dageville, B., Cruanes, T., Zukowski, M., et al. (2016). The Snowflake Elastic Data Warehouse. *International Conference on Management of Data (SIGMOD)*, p. 215–226.

Das, S., Agrawal, D. and El Abbadi, A. (2013). ElasTraS: An Elastic, Scalable, and Self-Managing Transactional Database for the Cloud. *Transactions on Database Systems (TODS)*, v. 38, n. 1, p. 5.

Gupta, A., Agarwal, D., Tan, D., et al. (2015). Amazon Redshift and the Case for Simpler Data Warehouses. In *International Conference on Management of Data (SIGMOD)*.

Hu, X., Tao, Y. and Chung, C.-W. (2013). Massive Graph Triangulation. In *SIGMOD*.

Isard, M., Budiu, M., Yu, Y., Birrell, A. and Fetterly, D. (2007). Dryad: Distributed Data-parallel Programs from Sequential Building Blocks. In *European Conference on Computer Systems (EuroSys)*.

Kim, C., Kaldewey, T., Lee, V. W., et al. (2009). Sort vs. Hash Revisited: Fast Join Implementation on Modern Multi-core CPUs. *Proceedings of the VLDB Endowment (PVLDB)*, v. 2, n. 2, p. 1378–1389.

Malewicz, G., Austern, M. H., Bik, A. J., et al. (2010). Pregel: A System for Large-scale Graph Processing. In *International Conference on Management of Data (SIGMOD)*.

Mehta, M. and DeWitt, D. J. (1997). Data Placement in Shared-nothing Parallel Database Systems. *The VLDB Journal*, v. 6, n. 1, p. 53–72.

Silva, F. W. R., Almeida, V. T. and Braganholo, V. (2017). Explorando Arquiteturas Multi-core para Processamento Eficiente de Consultas em Sistemas de Gerência de Big Data. In *Simpósio Brasileiro de Banco de Dados (SBBD)*.

Wang, J., Baker, T., Balazinska, M., et al. (2017). The Myria Big Data Management and Analytics System and Cloud Service. In *Conference on Innovative Data Systems Research (CIDR)*.

Warneke, D. and Kao, O. (2009). Nephele: Efficient Parallel Data Processing in the Cloud. In *Many-Task Computing on Grids and Supercomputers (MTAGS)*.

Interoperabilidade entre DaaS e DbaaS heterogêneos

Marcelo Aires Vieira, Daniela Barreiro Claro

FORMAS - Grupo de Pesquisa sobre Formalismos e Aplicações Semânticas
Instituto de Matemática e Estatística – Universidade Federal da Bahia (UFBA)
Avenida Adhemar de Barros, s/n – 40.170-110 – Salvador – BA – Brazil

marceloav@ufba.br, dclaro@ufba.br

Nível: Mestrado

Mês e ano de ingresso: Julho de 2016

Mês e ano previstos para defesa: Julho de 2018

Etapas concluídas:

- Desenvolvimento inicial do método e experimentos parciais;
- Conclusão de exame de qualificação: 29 de junho de 2017.

Publicações:

VIEIRA, M. ; RIBEIRO, E. ; ROCHA, W. S. ; MANE, B. ; CLARO, D. B. ; SANTOS, J. ; LIMA, E. . Enhancing MIDAS Towards a Transparent Interoperability Between SaaS and DaaS. In: Simpósio Brasileiro de Sistemas de Informação, 2017, Lavras. Anais do XII Simpósio Brasileiro de Sistemas de Informação. Porto Alegre: SBC, 2017. v. 1.

Interoperabilidade entre DaaS e DbaaS heterogêneos

Marcelo Aires Vieira, Daniela Barreiro Claro

FORMAS - Grupo de Pesquisa sobre Formalismos e Aplicações Semânticas
Instituto de Matemática e Estatística – Universidade Federal da Bahia (UFBA)
Avenida Adhemar de Barros, s/n – 40.170-110 – Salvador – BA – Brazil

marceloav@ufba.br, dclaro@ufba.br

Abstract. *Organizations are using cloud services to persist, consume and provide their data. These services are known as Database as a Service (DbaaS) and Data as a Service (DaaS). Often these data and services are heterogeneous and the applications require additional efforts to access and join them. This work provides a transparent interoperability between DaaS and DbaaS, which ensures data access and join independently of its model and supplier.*

Resumo. *Organizações estão utilizando serviços em nuvem para persistir, consumir e disponibilizar seus dados. Estes serviços são conhecidos como Database as a Service (DbaaS) e Data as a Service (DaaS). Frequentemente, estes dados e serviços são heterogêneos e as aplicações necessitam de esforços adicionais para acessá-los e juntá-los. Com isso, este trabalho fornece uma interoperabilidade transparente entre DaaS e DbaaS, na qual garante o acesso e a junção dos dados, independente do seu modelo e fornecedor.*

1. Introdução

A computação em nuvem é um modelo para permitir uma rede ubíqua e sob demanda, constituída por aplicativos, plataformas e hardwares que podem ser fornecidos como serviços por provedores de nuvem [Armbrust et al. 2010]. Estes serviços são organizados em níveis e ofertados no modelo de *pay-per-use*. A arquitetura mais difundida é composta por 3 níveis: (i) Infraestrutura como Serviço (IaaS); (ii) Plataforma como Serviço (PaaS); e (iii) Software como Serviço (SaaS) [Armbrust et al. 2010]. A partir destes níveis, os provedores podem fornecer outros modelos de serviços. tais como Bancos de dados como Serviço (DbaaS) e Dados como Serviço (DaaS).

Embora, às vezes considerados sinônimos, os conceitos de DaaS e DbaaS são diferentes. O DaaS baseia-se no conceito dos dados serem fornecidos sob demanda como serviços aos consumidores independentemente da localização geográfica do fornecedor e do consumidor [Truong and Dustdar 2009]. Exemplos de DaaS são: Portal Brasileiro de Dados Abertos[1] e DATA.GOV[2]. Por outro lado, o DbaaS fornece Sistemas Gerenciadores de Banco de Dados (SGBDs) com mecanismos contínuos para que as organizações armazenem, acessem e manipulem seus bancos de dados [Hacigumus et al. 2002]. Alguns exemplos de DbaaS são: ClearDB[3], ElephantSQL[4] e BD Cosmos[5].

[1]http://dados.gov.br/
[2]https://www.data.gov/
[3]http://w2.cleardb.net/
[4]https://www.elephantsql.com/
[5]https://azure.microsoft.com/services/cosmos-db/

Empresas, instituições e governos usam os modelos DbaaS e DaaS como uma forma de persistir e disponibilizar seus dados para usuários públicos e privados. No entanto, acessar e juntar os dados de diferentes organizações ainda é um desafio. A maioria dos dados e dos serviços fornecidos possuem heterogeneidades, que incluem: pluralidade dos modelos de dados e diferenças de acesso aos DbaaS e APIs (*Application Programming Interface*) personalizadas por cada provedor. Essas heterogeneidades aumentam os esforços das aplicações ao acessar os dados e se agrava quando há a necessidade de juntar dados de duas ou mais fontes para ter uma visão integrada e transparente. Estes problemas ocorrem devido à falta de interoperabilidade entre as aplicações e as fontes de dados.

Como exemplo, suponha que duas organizações forneçam dados em formatos e sistemas diferentes, um DaaS e um DbaaS que fornecem XML e JSON, respectivamente. Para que a manipulação e junção desses dados seja possível, os usuários teriam que adaptar suas aplicações para reconhecer os padrões de consultas e os modelos de dados. Assim, o presente trabalho propõe um método que permite interoperar DaaS e DbaaS, garantindo a junção e formatação das fontes de dados heterogêneas.

O método proposto foi incorporado no *Middleware Interoperability between DaaS e SaaS* (MIDAS 1.6) [Vieira et al. 2017] e contém três funcionalidades principais: (i) acesso e obtenção de dados em DbaaS e DaaS heterogêneos; (ii) junção de dados advindos de fontes de dados heterogêneas; e (iii) adequação do resultado ao requisitado pelo SaaS. Para avaliá-lo, três experimentos foram realizados, considerando quatro fatores importantes: funcional, tempo de execução, sobrecarga e escalabilidade. Os experimentos permitiram observar a viabilidade da abordagem e a corretude dos resultados.

2. Trabalhos Relacionados

Algumas soluções próximas da abordagem deste trabalho foram propostos para solucionar a falta de interoperabilidade. No domínio médico, [Park and Moon 2015] propõe uma solução para DBaaS heterogêneos que compartilham dados médicos entre diferentes instituições. Contudo, esta abordagem manipula dados que seguem as normas de padronização *Health Level Seven* (HL7), minimizando os esforços referente à heterogeneidade.

Em outro domínio, os autores de [Igamberdiev et al. 2016] apresentaram um *framework* para resolver problemas em sistemas *Big Data* na área de petróleo e gás. O objetivo é automatizar a transferência de informações entre projetos, identificando semelhanças e diferenças. Porém, o *framework* manipula apenas uma fonte de dados por consulta, não possibilitando a junção de dados de mais de uma fonte distinta.

Para solução de interoperabilidade sem domínio específico, [Sellami et al. 2014] e [Xu et al. 2016] fornecem uma API e um *middleware*, respectivamente, que permitem gerenciar diferentes bancos de dados relacionais e NoSQL. Entretanto, estas propostas ainda não satisfazem todos os tipos de NoSQL, tais como baseado em grafo e coluna, e nem NewSQL. Além disso, manipulam fontes de dados sem efetuar junções e não trabalham com dados fornecidos em DaaS, uma das principais vantagens desta proposta.

Por fim, a abordagem mais próxima desta proposta é o trabalho de Marinho [Marinho et al. 2016], pois fornece uma interoperabilidade entre SaaS e DaaS por meio de um *middleware*. No entanto, este *middleware* possui algumas limitações: (i) trabalha

exclusivamente com dados de DaaS e apenas no formato JSON; (ii) retorna ao SaaS somente dados no formato JSON; e (iii) acessa somente uma fonte de dados por solicitação, não possibilitando junções de fontes heterogêneas.

3. Proposta

O método proposto visa interoperar diferentes DaaS e DbaaS, tornando transparente o processo de acesso, junção e formatação dos dados. Especificamente, este trabalho tem como objetivo minimizar as heterogeneidades no processo de obtenção e junção dos dados persistidos em DaaS e DbaaS. Para isso, este método é capaz de:

- identificar as fontes de dados, a partir de informações recebidas;
- construir as requisições a serem enviadas aos DaaS e DbaaS, possibilitando enviar para duas ou mais fontes de dados em uma única consulta;
- reconhecer os dados de diferentes DbaaS e DaaS, independente do modelo; e
- juntar os dados e formatar os dados recebidos, encaminhando o resultado no formato solicitado pelo SaaS, tais como JSON, XML e CSV.

Para alcançar estes objetivos, o método proposto, incorporado no MIDAS 1.6 [Vieira et al. 2017], é composto por 6 componentes (ilustrados na Figura 1): (i) Query Decomposer, que recebe uma consulta do SaaS e mapeia a instrução em um array; (ii) Query Builder, que recebe a consulta decomposta e constrói uma requisição ao DaaS ou DbaaS; (iii) Data Mapping, que identifica e obtém os dados dos diferentes DbaaS; (iv) Dataset Information Storage (DIS), que consiste no conjunto de dados no qual são persistidas as informações da API do DaaS; (v) Crawler, que mantém os dados do DIS atualizados; e (vi) Result Formatter, que formata, junta e seleciona os dados antes de retornar ao SaaS.

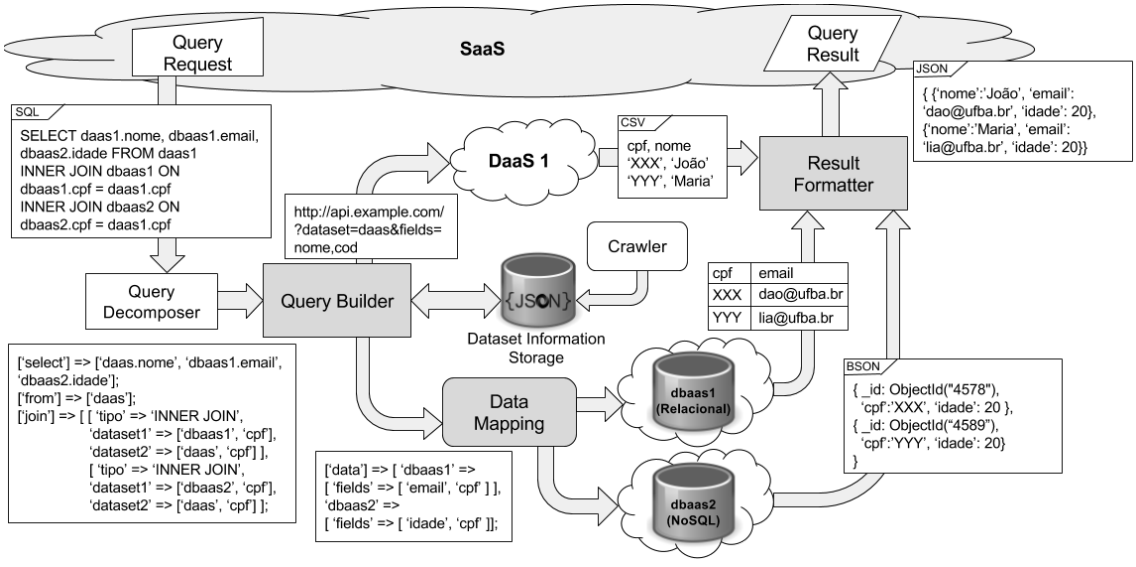


Figura 1. Sequência de execução do MIDAS entre um DaaS e dois DbaaS através de instruções de Join

Os componentes incorporados pelo método proposto neste trabalho são: Data Mapping, Query Builder e Result Formatter, responsáveis pela comunicação com os DaaS e DbaaS. O Data Mapping identifica em qual DbaaS os dados se encontram, realiza o

acesso e obtém os dados solicitados. Estes DbaaS podem ser baseados em tabelas, colunas, grafos, chave-valor ou documentos. Pelo Query Builder, é possível acessar vários DaaS em uma mesma consulta, caso a consulta possua uma instrução de junção (como *Join* do SQL ou *lookup* do MongoDB). Por outro lado, o Result Formatter recebe os dados dos DaaS e DbaaS e, realiza a junção e formatação dos dados, independente do modelo.

A Figura 1 ilustra a sequência de execução do MIDAS para uma consulta SQL com a instrução *Join* que acessa um DaaS e dois DbaaS. Neste exemplo, o SaaS envia uma consulta SQL ao MIDAS, que executa a decomposição pelo Query Decomposer e encaminha para o Query Builder. O Query Builder acessa o DIS e identifica que os dados se encontram no daas1 e em dois DbaaS (dbaas1 e dbaas2). Com isso, o Query Builder constrói a requisição ao DaaS e solicita ao Data Mapping que conecte-se aos dois DbaaS para obter o restante dos dados. Em seguida, cada provedor executa a solicitação e retorna o resultado ao Result Formatter (daas1, dbaas1 e dbaas2 retornam nos formatos CSV, em tabela e em documento, respectivamente). O Result Formatter recebe os dados, efetua a junção e formata o retorno solicitado (JSON) e encaminha ao SaaS.

4. Experimentos e Resultados Parciais

Para avaliar o trabalho até o momento foram realizados três experimentos isolados. O primeiro executa consultas sem junção de dados em um DaaS e o segundo com instrução de junção entre dois DaaS distintos. O terceiro experimento realiza consultas em três DbaaS distintos: um Relacional, um NoSQL e um NewSQL. Para cada experimento, foi analisado o tempo de execução, a sobrecarga, a escalabilidade e a eficácia. As ferramentas utilizadas nos experimentos foram o Apache JMeter[6] e o Hurl.it[7].

Nos experimentos, foram utilizados quatro conjuntos de dados distintos, fornecidos pelo NYC Open Data[8] e DATA.GOV: (i) *DOB Job Application Filings*, com 294 mil instâncias e 82 atributos; (ii) *Health and Hospitals Corporation (HHC) Facilities*, com 78 instâncias e 6 atributos; (iii) *Borough Enrollment Offices*, com 13 instâncias e 6 atributos; e (iv) *Hospital General Information*, com 4805 instâncias e 13 atributos.

4.1. Experimento 1: consultas sem instrução de junção

O primeiro experimento é executado sem junção de dados, cujo objetivo é analisar a sobrecarga causada pelo MIDAS. Foram executadas 100 consultas SQL sucessivamente para recuperar 100, 1000 e 5000 instâncias de dados no DaaS, sendo dividido em duas tarefas: 100 consultas SQL diretamente para o provedor DaaS e 100 consultas SQL executadas ao provedor DaaS através do MIDAS. As médias de tempo são apresentadas na Tabela 1.

Tabela 1. Resultado do primeiro experimento (tempos em milissegundos)

Consultas	100 instâncias	1000 instâncias	5000 instâncias
sem MIDAS	761.31 ± 212.00	2333.66 ± 237.50	11321.19 ± 3722.50
com MIDAS	1073.49 ± 228.00	4525.92 ± 410.50	31653.98 ± 4147.00

[6]http://jmeter.apache.org/
[7]https://www.hurl.it/
[8]https://opendata.cityofnewyork.us/

4.2. Experimento 2: consultas com junções de dois DaaS

Neste experimento, avaliou-se a abordagem com a cláusula de junção, executando a consulta em dois DaaS distintos através do MIDAS. O objetivo deste experimento é avaliar a escalabilidade e o desempenho ao utilizar diferentes linguagens de consulta (SQL e MongoDB) com instruções de junção entre dois DaaS distintos. Neste experimento são executadas 20 consultas SQL e 20 consultas MongoDB com instrução de junção em dois DaaS distintos através do MIDAS. O resultado das médias dos tempos de execução são: 201.36 ± 26 ms para consultas SQL e 240.7 ± 31.36 ms para consultas MongoDB.

4.3. Experimento 3: consultas em DbaaS distintos

O terceiro experimento, realizado em três DbaaS (MySQL, MongoDB e MemSQL) em um ambiente local, permitiu analisar a sobrecarga causada pelo componente Data Mapping. Os três DbaaS foram populados com o conjunto de dados *Hospital General Information*. Em seguida, foram executadas 1000 consultas em cada banco de dados, sendo 1000 diretas e 1000 através do componente. Os resultados são apresentados na Tabela 2.

Tabela 2. Resultado das consultas diretas e através do MIDAS aos DbaaS

		Média	Desvio Padrão	% de sobrecarga
MySQL	Direto	28 ms	4.55 ms	-
	Data Mapping	46 ms	8.69 ms	64%
MongoDB	Direto[9]	132 ms	97.78 ms	-
	Data Mapping	55 ms	11.05 ms	-42%
MemSQL	Direto	34 ms	5.23 ms	-
	Data Mapping	50 ms	10.75 ms	47%

5. Conclusões parciais

Este trabalho propõe um método que fornece uma interoperabilidade entre DaaS e DbaaS, na qual minimiza as heterogeneidades dos dados e de como obtê-los. Considerando a complexidade do problema de interoperabilidade entre os serviços e os dados heterogêneos, este método, incorporado ao MIDAS, requer uma adaptação mínima das aplicações SaaS e promove a obtenção e junção de dados em DaaS e DbaaS distintos.

Ao analisar os resultados dos experimentos, percebe-se que apesar da sobrecarga, o desempenho mostrou-se satisfatório, pois os benefícios oferecidos pelo MIDAS facilitam a interoperabilidade de diferentes fontes de dados.

6. Trabalhos futuros

Como trabalhos futuros, pretende-se executar as seguintes atividades: (i) formalizar as diferenças dos modelos de dados e serviços; (ii) melhorar o método para fornecer suporte aos principais modelos de dados; (iii) avaliar o método, comparando-o com outros métodos e publicar os resultados; e (iv) melhorar e adequar a avaliação, permitindo uma maior interoperabilidade para Big Data.

Considerando os trabalhos futuros descritos, define-se um cronograma com as atividades a serem desenvolvidas para a conclusão desta dissertação (ver Tabela 3).

[9]O *overhead* no MongoDB foi causado pela ferramenta Apache JMeter, pois não possui suporte nativo a este tipo de SGBD.

Tabela 3. Cronograma de atividades

Atividades	08/2017	09/2017	10/2017	11/2017	12/2017
Formalizar as diferenças dos modelos de dados	X	X	X	X	X
Melhorar o método para suportar outros modelos de dados			X	X	X
Avaliar o método, comparando-o com outros métodos				X	X
Atividades	01/2018	02/2018	03/2018	04/2018	05/2018
Avaliar o método, comparando-o com outros métodos	X	X	X	X	X
Acompanhamento de oportunidades de publicação	X	X	X	X	X
Elaboração dos artigos para publicação dos resultados	X	X	X	X	X
Revisão e submissão dos artigos	X	X	X	X	X
Escrita e correção da dissertação		X	X	X	X

Referências

Armbrust, M., Fox, A., Griffith, R., Joseph, A. D., Katz, R., Konwinski, A., Lee, G., Patterson, D., Rabkin, A., Stoica, I., et al. (2010). A view of cloud computing. *Communications of the ACM*, 53(4):50–58.

Hacigumus, H., Iyer, B., and Mehrotra, S. (2002). Providing database as a service. In *Proceedings 18th International Conference on Data Engineering*, pages 29–38.

Igamberdiev, M., Grossmann, G., Selway, M., and Stumptner, M. (2016). An integrated multi-level modeling approach for industrial-scale data interoperability. *Software & Systems Modeling*, pages 1–26.

Marinho, T., Cidreira, V., Claro, D. B., and Mane, B. (2016). Midas: A middleware to provide interoperability between saas and daas. In *Proceedings of the XII Brazilian Symposium on Information Systems on Brazilian Symposium on Information Systems: Information Systems in the Cloud Computing Era-Volume 1*, page 53. Brazilian Computer Society.

Park, H.-K. and Moon, S.-J. (2015). Dbaas using hl7 based on xmdr-dai for medical information sharing in cloud. *International Journal of Multimedia and Ubiquitous Engineering*, 10(9):111–120.

Sellami, R., Bhiri, S., and Defude, B. (2014). Odbapi: a unified rest api for relational and nosql data stores. In *Big Data (BigData Congress), 2014 IEEE International Congress on*, pages 653–660. IEEE.

Truong, H. L. and Dustdar, S. (2009). On analyzing and specifying concerns for data as a service. In *2009 IEEE Asia-Pacific Services Computing Conference (APSCC)*, pages 87–94.

Vieira, M. A., Ribeiro, E. L., Rocha, W. S., Mane, B., Claro, D. B., Oliveira, J. S., and Lima, E. (2017). Enhancing midas towards a transparent interoperability between saas and daas. In *XIII Brazilian Symposium on Information Systems (SBSI2017)*, volume 2017, pages 356–363.

Xu, J., Shi, M., Chen, C., Zhang, Z., Fu, J., and Liu, C. H. (2016). Zql: A unified middleware bridging both relational and nosql databases. In *Dependable, Autonomic and Secure Computing, 14th Intl Conf on Pervasive Intelligence and Computing, 2nd Intl Conf on Big Data Intelligence and Computing and Cyber Science and Technology Congress (DASC/PiCom/DataCom/CyberSciTech), 2016 IEEE 14th Intl C*, pages 730–737. IEEE.

MetisIDX - From Adaptive to Predictive Data Indexing

Elvis Teixeira[1], **Javam Machado**[1]

[1]Laboratório de Sistemas e Banco de Dados (LSBD) – Universidade Federal do Ceará (UFC) – CEP 60440-900 – Fortaleza – CE – Brazil

{elvis.teixeira,javam.machado}@lsbd.ufc.br

Resumo. *Indexação adaptativa é uma estratégia adequada à exploração de dados, cenário no qual a ausência de conhecimento sobre a carga de trabalho e de tempo para a criação de índices completos são desafios constantes. Uma característica de índices adaptativos é a sua construção através de operações incrementais de baixo custo enquanto o índice já é utilizado. Este trabalho explora a flexibilidade das estruturas de dados para atuar de forma preditiva. Os intervalos de chave são indexados antecipadando-se às consultas usando técnicas de aprendizagem de máquina, com base na carga de trabalho recentemente processada. Máquinas de aprendizado extremo (ELM) são discutidas e avaliados neste contexto e as implicações para outros componentes do SGBD, tais como gerenciador de buffer, são também consideradas.*

A presente proposta de trabalho de mestrado, foi elaborada pelo aluno Elvis Marques Teixeira, ingresso no programa de mestrado e doutorado em ciência da computação da Universidade Federal do Ceará (MDCC-UFC) and Março de 2016. Foi aprovada como proposta em Março de 2017 e a previsão de defesa é Abril de 2018. A pesquisa é desenvolvida sob a orientação do professor Dr. Javam de Castro Machado.

Abstract. *Adaptive indexing is a distinctive strategy for approaching data exploration, in which lack of workload knowledge and time for full index building are constant issues. At the heart of the adaptive techniques is the use of partial indexes and the ability to tune them by means of small and lightweight actions, while the index is already being used. This work explores the flexibility of the data structures used in adaptive indexing to introduce a predictive physical design tuning, using machine learning techniques based on the recent workload. The key ranges predicted to be queried next are indexed in advance. Extreme learning machines (ELM) are discussed and evaluated in this context. Implications to other DBMS components, such as buffer managers are also considered.*

1. Introduction

Important modern database applications do not have a known workload pattern. This means that data subsets which are focus of query attention may change, and that ad hoc queries should be expected. Examples of this kind of application are found in scientific work or in exploratory analysis, which is an increasingly common daily task in many areas of business today. No assumptions should be made about the workload, and database systems should still be able to answer the queries from these dynamic workloads efficiently, searching through and updating suitable data structures to speed up query processing.

To accomplish this challenging task, adaptive indexing was introduced. It enables the system to change according to the dynamic workload by adapting its internal structures to quickly answer queries that follow the current trends. The fundamental idea is that each time data is scanned to answer a user query a further step is performed to refine or provide an index structure, that will permit subsequent scans to prune the search space and perform better. Such operations should be simple in order to avoid adding a prohibitive overhead to query execution [Athanassoulis et al. 2016] while still being useful. A simple partitioning action may do the job, as pointed out in [Kersten and Manegold 2005].

Changing database physical layout in response to the workload can be powerful if used properly. For instance, consider the situation in which, while trying to follow the query trends, the system changes layout in response to queries which deviate from the workload pattern. This wastes time and compromises performance. Additionally, using only the current query may not be enough to figure out the best key range to index in order to enable the next queries to benefit from the structure. More contextual information, or a workload model, would perform better. These issues are similar the problems of overfitting and generalization in pattern recognition and machine learning tasks.

This work develops this analogy further by using an actual machine learning technique. It leverages an adaptive structure, and adds a predictive behavior to the index building steps based on a dynamical model of the workload. By indexing data expected to be requested next, our index builder is less sensitive no anomalies in the workload, and avoids the effort of indexing uninteresting records. Such intelligent access structures fit naturally in the context the emerging self driving systems [Arulraj et al. 2016], which promise to be able to handle highly dynamical and hybrid workloads while requiring even less manual operation than traditional systems.

2. Related work

MetisIDX shares traits with previous adaptive indexing techniques, such as incrementally building layouts. It also inherits ideas from approaches which are not strictly adaptive, for instance, it adds indexing procedures outside of select operator execution. We now revisit the previous work that influenced our design choices.

2.1. Database Cracking

Database Cracking [Kersten and Manegold 2005] [Idreos et al. 2007] is one of the first and most extensively tested adaptive indexing techniques. It was designed to operate in the context of column stores and showed promising results by accelerating the queries from the level of full linear scans to nearly that of full index scan after a number of queries. The main advantage of such technique is the fact that it does not require any index building time, in which the system will not be able to answer user queries. Instead, it starts answering the first query without any index support and performs indexing steps at each query processing operation. Further queries in the same key range are answered via partial index scans and improve the structure further.

The data structure used by Database Cracking is composed by a copy of the data column being indexed, usually called the cracking column, and an AVL tree used to keep track of the positions of the partitions made in the column. The indexing process consists of a quick sort like partitioning, done at each scan time. The pivots used in the process are the values in the query predicate. After the query is answered, the structure of the column is such that all of the values smaller than the query predicate will be placed above the partition point, in unspecified order, and all of the values greater than or equal to the partition point will appear below it. The partition point is stored in the tree structure to enable future queries to prune the search space instead of scanning the entire column.

As more queries are processed the column gets further partitioned and approaches a sorted column. The search performance [Schuhknecht et al. 2016] improves continually from the level of a linear scan to that of a binary search. Another optimization given by Database Cracking is the fact that only key ranges that are actually queried get their physical design optimized, and no effort is wasted indexing cold data i. e. ranges that will not be focus of query activity any time soon.

2.2. Adaptive Merging

While database cracking is an efficient strategy for column stores and thus for analytic workloads, distinct work has been done in the context of tuple-based storage. Adaptive Merging [Graefe and Kuno 2010] consists of using a partitioned B tree as the indexing structure and performing merge operations in the key ranges touched by queries until it reaches the form of an index that is not partitioned.

During the processing of the very first query, adaptive merging read the raw data in chunks called runs and then writes them back to secondary storage sorted. Each sorted run is assigned a sequential artificial leading column (ALC) so that the data is globally sorted, with respect to the compound key formed by the ALC and the user defined key. This procedure makes it possible to construct a B-Tree structure with $O(\log n)$ I/O operations in contrast with at least $O(n \log n)$ required to build a full B-tree index. This process is similar to the first steps in a bulk loading index build operation but with less I/O.

After this partitioned index is built, subsequent queries will already be faster than equivalent ones over the unsorted data. How much faster depends mainly on the number of runs or partitions created, since for any given query it is necessary to traverse the tree as many times as there are partitions. Adaptive index optimization is performed at query processing time, just like in Database Cracking, by merging the queried key ranges into a final, not partitioned, index. As more data is queried its physical layout achieves the final form, and if queried again, these key ranges will return the response in the minimum time.

2.3. Holistic Indexing

On top of the promising innovations represented by the adaptive techniques, such as Database Cracking and Adaptive Merging, some refinements were attempted. These efforts were aimed at accelerating layout optimization, handle anomalous requests or make use of spare computing resources.

The one single most important addition to adaptive indexing is Holistic indexing [Petraki 2012] [Petraki et al. 2015]. This technique adds indexing operations to those in database cracking exploiting the parallelism capabilities of modern processors. In this system, a monitoring thread is always active looking for available CPU cores to perform cracking operations along with those triggered by the queries.

Holistic indexing advocates the use of random pivots for performing cracking operations and presents a measure of distance from an optimal index to an actual one, this measure can be used to select the indexes to be further cracked. It is argued, however, that a random choice of index is also preferable. These design choices make holistic indexing intrinsically stochastic and deviates from one of the core philosophical statements of adaptive indexing, which suggest that indexing efforts should only be devoted to data regions actually requested by user queries. This is not a weakness of the technique, since the premise is spare CPU resources, but marks a distinction from strictly adaptive systems.

Holistic indexing, as well as database cracking were implemented inside column based relational system MonetDB, with the assumption that all data is main memory resident. Performance gains over plain database cracking have been shown to be accomplished by holistic indexing, both in the context of standard benchmarks such as TPC-H and of synthetic workloads specifically created to test the capabilities of the strategy.

3. MetisIDX

We introduce our new indexing scheme by pointing out its differences to the previous related work. In the process, the differences and the new challenges addressed are explained, and the advantages are explored by means of a proposed experimental evaluation.

3.1. Motivation

MetisIDX proposes an indexing mechanism for relational data that targets secondary storage (HDDs, SSDs, etc). The data structure used, a partitioned B+ tree, and some of the index creation and maintenance procedures are similar to those developed for adaptive merging. A B-tree based structure was chosen to exploit it's intrinsically paged data transfer naturally performed with secondary storage devices, and it permits meaningful data blocks to be exchanged between the levels of a memory hierarchy. However, differently from the Adaptive Merging approach, MetisIDX indexes data before it is requested

and is less sensitive to workload anomalies since it leverages the information provided by multiple requests rather than the current query alone.

It has been shown that improvements to adaptive indexing can be achieved by indexing key ranges not strictly equal to those of the query responses, possibly adding a stochastic component to the indexing process [Halim et al. 2012]. Other possibility is using periods of time when computing resources are not being exhaustively used to index key ranges not yet touched [Petraki et al. 2015]. The main advantage of these approaches is the possibility of indexing a region of data that will be queried in the near future. When this happens, the response time will be optimal and the effort of indexing can be done independently of query processing, thus not incurring any extra overhead to it.

3.2. Indexing process

The first query triggers a full scan over the unindexed data. During this first scan's execution, data is read from secondary storage in chunks, usually called runs in the context of adaptive merging. Each run is then sorted, and the records that belong to the query's response set are collected and returned to the user. Sorted runs are written back to disk as the leaves of a B+ tree. Global ordering is achieved by introducing an artificial leading attribute whose value is the run's creation order. This partitioned index is already able to speed up the processing of subsequent queries but, in order to achieve optimal read access, the partitions must be merged into a single one so that the index becomes a regular B+ tree. In past adaptive techniques it is the job of the select operator to marge the partitions.

In adaptive merging, the transition from the partitioned structure to the final index is accomplished by collecting the records that belong to the response set of each query and merging them into a final partition, which becomes the full index after some number of queries. All the partitions compose a single tree structure, including the final one. In MetisIDX two separate structures are used, one for the partial index, the partitioned tree, and a second structure for the final index. This second index structure is the final index, unpartitioned, created by merging key ranges from the partial index. This design was chosen to keep the partial index as a read only structure.

The decision of which key ranges to merge and the actual merging operation are done independently and in parallel to query processing. Such decision process comes from the extreme learning machine that is continuously trained in background by a dedicated thread. That same thread is used to perform merges at the end of each training mini-batch. After merging is done, a new model update is started. An important difference between a predictive system and a strictly adaptive one is the fact that, by indexing a key range before it is queried, the records in the indexed range will be brought up to cache. This means that predictive indexing is also a predictive cache prefetching.

3.3. Workload modeling and query forecasting

As the goal is to update a model of the workload continuously and use it to perform merge operations in parallel with query processing, the learning algorithm must be lightweight enough to be executed in time frames that do not exceed the order of magnitude of that required to answer a single query. This constraint makes it impractical to use many state of the art machine learning techniques such as massive deep neural networks, since using these methods would result in a training time long enough to make the resulting model already outdated, in the sense that it reflects a workload pattern that is already gone.

In addition to having a lightweight training process, the ideal regression mechanism should be able to update the model using new available data, new queries in this context, without discarding the previous model version altogether. A suitable technique that fulfills these requirements is the Extreme Learning Machine [Liang et al. 2006], a class of neural networks which always have a single hidden layer, and only the weights of the connections between the hidden neurons and the output neurons must be trained.

The training data used in model updates is the sequence of key range boundaries from the last N queries processed by the system. N is a hyper parameter that has to be chosen by hand. The trained network is then the current model of the workload. In the next training step it is updated but not discarded. After that, the model is used to forecast the key range to be queried next, and a new merging operation triggered. The process of training and triggering merges is carried out cyclically.

3.4. Performance evaluation

We will compare the performances of MetisIDX and Adaptive Merging, both implemented in the context of a custom storage and access engine loaded with a synthetic dataset. Since MetisIDX is not a strictly adaptive technique, and it adds indexing steps which are not part of query processing, it would make sense to compare it with some variant of Holistic Indexing, which has similar attributes. Holistic Indexing, however is based on Database Cracking, so it is designed to work in the context of main memory column stores, while our approach targets tuple-based systems in secondary storage.

Response times will be used as a performance metric for the purpose of comparison. We use this metric instead of the more usual I/O operations count because this quantity is not as directly linked to response times in MetisIDX as it is in Adaptive Merging. The reason is that, as we index data before query processing occurs and the indexing process must bring data up to the main memory cache, the select operator is expected to find its response in cache. In other words, it does not matter how many disk accesses were performed if the data is in cache by the time one needs it. Other performance metric used in the experiments is therefore cache hit rates, which reflect the error rates in the query forecasting process.

4. Current status and future work

Our own implementation of adaptive merging is complete. We had to write this baseline for comparison since no publicly available implementation was found. On top of this structure our predictive component is in development process. The whole software system to be used in the experiments is composed of a LRU-based cache manager, an expression library for processing fixed length tuples, and a framework for custom extension operators.

The adaptive merging setup yields results similar to those found in the original work, and the addition of the neural network and training thread are in progress. The system is being developed, in modern C++ (11/14), with the goal of being released as set of modular open source libraries for building database components or standalone experiments. The final results of this work are planned for late October 2017, at this time we plan to submit a paper to a high impact conference.

There are plenty of possibilities for future work in this field. One of the promising ones is the exploration of predictive models to address concurrency. For example, if more than one client is performing read operations in an unindexed base, then the index builder may want to index and cache key ranges that will be requested by multiple transactions. The discussion and evaluation of the extra features required to make predictions in the context of concurrency is a valuable research problem, and has been under investigation in our adaptive database systems research group.

Referências

Arulraj, J., Pavlo, A., and Menon, P. (2016). Bridging the archipelago between row-stores and column-stores for hybrid workloads. In *Proceedings of the 2016 International Conference on Management of Data, SIGMOD Conference 2016, San Francisco, CA, USA, June 26 - July 01, 2016*, pages 583–598.

Athanassoulis, M., Kester, M. S., Maas, L. M., Stoica, R., Idreos, S., Ailamaki, A., and Callaghan, M. (2016). Designing access methods: The RUM conjecture. In *Proceedings of the 19th International Conference on Extending Database Technology, EDBT 2016, Bordeaux, France, March 15-16, 2016, Bordeaux, France, March 15-16, 2016.*, pages 461–466.

Graefe, G. and Kuno, H. A. (2010). Self-selecting, self-tuning, incrementally optimized indexes. In *EDBT 2010, 13th International Conference on Extending Database Technology, Lausanne, Switzerland, March 22-26, 2010, Proceedings*, pages 371–381.

Halim, F., Idreos, S., Karras, P., and Yap, R. H. C. (2012). Stochastic database cracking: Towards robust adaptive indexing in main-memory column-stores. *PVLDB*, 5(6):502–513.

Idreos, S., Kersten, M. L., and Manegold, S. (2007). Database cracking. In *CIDR 2007, Third Biennial Conference on Innovative Data Systems Research, Asilomar, CA, USA, January 7-10, 2007, Online Proceedings*, pages 68–78.

Kersten, M. L. and Manegold, S. (2005). Cracking the database store. In *CIDR*, pages 213–224.

Liang, N., Huang, G., Saratchandran, P., and Sundararajan, N. (2006). A fast and accurate online sequential learning algorithm for feedforward networks. *IEEE Trans. Neural Networks*, 17(6):1411–1423.

Petraki, E. (2012). Holistic indexing: offline, online and adaptive indexing in the same kernel. In *Proceedings of the ACM SIGMOD/PODS PhD Symposium 2012, Scottsdale, AZ, USA, May 20, 2012*, pages 15–20.

Petraki, E., Idreos, S., and Manegold, S. (2015). Holistic indexing in main-memory column-stores. In *Proceedings of the 2015 ACM SIGMOD International Conference on Management of Data, Melbourne, Victoria, Australia, May 31 - June 4, 2015*, pages 1153–1166.

Schuhknecht, F. M., Jindal, A., and Dittrich, J. (2016). An experimental evaluation and analysis of database cracking. *VLDB J.*, 25(1):27–52.

Mecanismo de Inferência de Diagnóstico Baseado na Classificação de Sinais ECG

Priscila Rocha Ferreira Rodrigues[1], José Maria da Silva M. Filho[1]

[1]Programa de Mestrado e Doutorado em Ciencia da Computação (MDCC)

Universidade Federal do Ceará (UFC)

Campus do Pici – Bloco 910 – 60.455 – 760- Fortaleza– CE – Brasil

priscila.rfr@alu.ufc.com, jose.macedo@dc.ufc.br

Nível: Mestrado
Ingresso: Fevereiro 2017
Previsão de Término: Fevereiro 2019
Etapas já concluídas: Revisão Bibliográfica, Definição do Problema
Defesa da Pré-Proposta: Outubro 2017
Defesa da Proposta: Fevereiro 2018

Abstract. *Due to the increasing number of individuals presenting with cardiopathies, numerous surveys have sought to extract patterns from ECG databases. However, the vast majority of these initiatives focus on the classification of an ECG signal into a set of previously known arrhythmias. However, such classifiers do not involve the causes that trigger the cardiopathies, and it is up to the medical professional to investigate and diagnose these causes. In this sense, with the objective of assisting medical professionals in this diagnostic stage, this work proposes an ECG signal classifier that besides classifying the signal according to the type of arrhythmia, will indicate, based on a degree of probability, Its possible cause, directing and facilitating the investigation carried out by the medical professional.*

Keywords: *Pattern Detection, Classification, ECG*

Resumo. *Devido ao número crescente de indivíduos que apresentam cardiopatias, inúmeras pesquisas têm procurado extrair padrões a partir de bases de dados eletrocardiograma (ECG). Porém, a grande maioria dessas iniciativas concentra-se na classificação de um sinal de ECG em um conjunto de arritmias previamente conhecidas. No entanto, tais classificadores não envolvem as causas que desencadearam as cardiopatias, cabendo ao profissional de medicina investigar e diagnosticar essas causas. Neste sentido, com o objetivo de auxiliar os profissionais de medicina nesta etapa de diagnóstico, este trabalho propõe um classificador de sinais de ECG que além de classificar o sinal de acordo com o tipo de arritmia, irá indicar, com base em um grau de probabilidade, a sua possível causa, direcionando e facilitando a investigação realizada pelo profissional de medicina.*

Palavras-chave: *Detecção de Padrões, Classificação, ECG*

1. Introdução

A atividade elétrica do coração pode ser registrada por meio de eletrodos conectados a pontos específicos do corpo e que capturam impulsos elétricos em forma de sinais de eletrocardiograma (ECG). É possível detectar anormalidades cardíacas de forma não-invasiva por meio das variações do ECG [Kohler et al 2002]. Recentemente, devido ao número crescente de indivíduos que apresentam cardiopatias [Mozaffarian et al 2016], inúmeras pesquisas têm procurado extrair padrões a partir de bases de dados de ECG, de maneira que doenças cardiovasculares, como arritmias e insuficiências cardíacas, possam ser identificadas de maneira eficaz [Kavitha et al 2014], [Jambukia et al 2015], [Shadmand et al 2016] e [Elhaj et al 2016].

No entanto, a grande maioria dessas iniciativas concentra-se na classificação de um sinal de ECG em um conjunto de arritmias previamente conhecidas. Além disso, tais classificadores não envolvem as causas que desencadearam as cardiopatias, cabendo ao profissional de medicina investigar e diagnosticar essas causas. Dentre as principais causas, destacam-se: hipertensão, diabetes, tabagismo e colesterol alto [Mackay et al 2004]. Neste sentido, com o objetivo de auxiliar os profissionais de medicina nesta etapa de diagnóstico, este trabalho propõe um classificador de sinais de ECG que além de classificar o sinal de acordo com o tipo de arritmia, irá indicar, a sua possível causa, direcionando e facilitando a investigação realizada pelo profissional de medicina. A certeza de que compreender a natureza motivadora da doença é mais importante do que meramente identificá-la, foi o que impulsionou o tema em estudo.

Este trabalho concentra-se na identificação de padrões em sinais de ECG, mais especificamente, buscando associar um determinado sinal de ECG a certos tipos pré-definidos de arritmias, apontando também as prováveis causas da arritmia. Para isso, iremos explorar uma grande base de dados de sinais fisiológicos denominada *PhysioBank* [Goldberger et al 2000]. Tal base, além de disponibilizar os sinais, fornece informações do diagnóstico geral do indivíduo, o que permite o estudo dos sinais agrupados pelas causas que provocaram determinada doença cardíaca. Uma vez identificados os padrões desses sinais com base nas causas das cardiopatias, um classificador ECG é proposto. Logo, além de identificar a doença cardiovascular propriamente, o classificador infere, com certo grau de certeza probabilística, qual a causa associada a anormalidade encontrada. Essa abordagem é relevante pois auxiliará o profissional de medicina a identificar mais rapidamente o diagnóstico, otimizando tempo e recursos, uma vez que ele poderá direcionar sua investigação a uma causa específica, apontada pelo classificador, não mais investigando o que ocasionou a anormalidade de maneira geral.

Na Seção 2, é descrita a fundamentação teórica e a definição do problema. Na Seção 3 é apresentada a caracterização da contribuição. Os trabalhos relacionados estão descritos na seção 4. Por fim, a Seção 5 apresenta as conclusões e as direções futuras desta pesquisa.

2. Fundamentação Teórica

A seguir serão descritos importantes conceitos para a compreensão deste trabalho. Além disso, será definido formalmente o modelo de cenário imaginado para o problema em questão.

2.1 Problemas Cardiovasculares e sinais ECG

As doenças cardiovasculares são aquelas que afetam o coração e as artérias, cuja principal característica é a presença da aterosclerose, acúmulo de placas de gorduras nas artérias ao longo dos anos que impede a passagem do sangue. Como exemplo podemos citar o infarto, acidente vascular cerebral, arritmias cardíacas, isquemias ou anginas.

Tais doenças tornaram-se uma das principais causas de morte no mundo. São aproximadamente 17,5 milhões de óbitos registrados anualmente decorrentes de doenças cardiovasculares [Mozaffarian et al 2016]. Ainda segundo [Mozaffarian et al 2016], o aumento nas taxas de mortalidade por doenças cardiovasculares está associado a transição epidemiológica para estilos de vida não saudáveis, sendo as principais causas o tabagismo, hipertensão, níveis anormais de glicose e colesterol alto.

O coração é um músculo que se contrai de forma rítmica, bombeando sangue para todo o corpo. Essa contração se inicia no nodo sinoatrial (estrutura anatômica do coração responsável por regular os batimentos cardíacos) e se propaga para todo o músculo, atuando assim, como um marcapasso natural. Essa propagação elétrica do sinal segue um padrão [Spach e Kootsey, 1983]. Como resultado desta atividade, correntes elétricas são geradas na superfície do corpo, provocando variações no potencial elétrico da superfície da pele. Esses sinais podem ser capturados ou medidos com o auxílio de eletrodos conectados a pontos específicos do corpo. O registro gráfico das atividades elétricas do coração, resultado deste processo de captura, é chamado de eletrocardiograma (ECG). A Figura 1 apresenta a representação diagramática dos diferentes componentes de um sinal ECG.

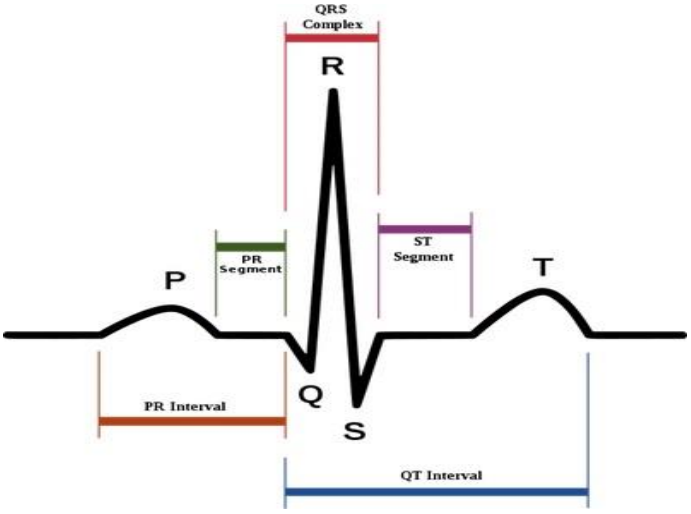


Figura 1 - Componentes do sinal ECG. Fonte: EKG Academy

Esses sinais são essencialmente formados por ondas P, um complexo QRS e ondas T. Todas as irregularidades na onda P, complexo QRS, componentes da onda T e intervalos que vão de R a R indicam uma doença no coração **[**Kohler et al 2002**]**. É possível extrair diferentes recursos do sinal, tais como duração, forma, altitude, pontos de pico, dentre outros. O ECG pode ser usado para auferir a velocidade e o ritmo dos

batimentos cardíacos, a presença de qualquer dano às células do músculo cardíaco, os efeitos de medicamentos e até a função de marca-passos implantados.

Nesse contexto, bases de dados que contém sinais ECG vêm sendo exploradas e inúmeros métodos de extração de características desses sinais têm sido propostos, com o intuito de aperfeiçoar a classificação e detecção de doenças cardiovasculares.

2.2 *PhysioBank*

PhysioBank é um grande *dataset* de sinais fisiológicos, criado em 2000 pelo *National Institutes of Health* dos EUA para uso da comunidade de pesquisa biomédica. Esse *dataset* possui acesso público e contém atualmente mais de 90.000 registros, aproximadamente 4 *terabytes* de sinais fisiológicos digitalizados em séries temporais e organizado em mais de 80 sub-*datasets*. Muitos dos *datasets* integrados ao *PhysioBank* foram desenvolvidos no *MIT (Massachusetts Institute of Technology)* e no *Beth Israel Hospital* de *Boston - EUA*. Inclui sinais cardiopulmonares, neurais e outros sinais biomédicos que vão desde de indivíduos saudáveis a pacientes com uma variedade de condições com implicações importantes para a saúde pública, incluindo arritmias com risco de vida, insuficiência cardíaca congestiva, apneia do sono, distúrbios neurológicos e envelhecimento [Goldberger et al 2000].

Esse *dataset* é um dos mais referenciados na literatura no contexto de pesquisa em sinais ECG [Belle et al 2015]. Novas bases de dados são integradas a ele constantemente, sendo um dataset ainda crescente e por esses motivos escolhido para ser a fonte de exploração do trabalho em questão.

2.3 Definição do Problema

Verifica-se a existência de inúmeras pesquisas em bases de dados com sinais ECG, que propõem técnicas eficazes de classificação desses sinais para identificação de doenças cardiovasculares. No entanto, uma vez que o profissional de medicina realiza o exame no paciente e detecta a anormalidade cardíaca, para fornecer um diagnóstico preciso, o mesmo necessita investigar as causas que encadearam tal doença [Mackay et al 2004]. Essa investigação para elaboração do diagnóstico consiste em uma etapa que demanda uma série de esforços, uma vez que atualmente o médico necessita realizar, de maneira geral, inúmeros exames no paciente para chegar ao ponto específico que está encadeando a doença.

Sabendo disso, objetiva-se minimizar tempo e recursos demandados na etapa do diagnóstico, sendo proposto um classificador que além de identificar a doença cardiovascular, irá indicar, com base em um grau de probabilidade, a possível causa. Tal classificador não tem o intuito de substituir o diagnóstico médico, mas direcionar sua investigação.

3. Caracterização da Contribuição

O presente trabalho tem por finalidade propor um classificador de sinais ECG tal que, dado o ECG de um paciente qualquer, o classificador seja capaz de identificar cardiopatias associadas a ele, se houver, e apresentar probabilisticamente a causa que desencadeou tal doença.

Os passos empregados para alcançar o objetivo proposto (Figura 2), são os descritos a seguir. A primeira etapa consiste em agrupar os sinais de uma base de dados do *PhysioBank*, separando-os de acordo com as causas detectadas nos

respectivos diagnósticos, e associando-os às cardiopatias. Para isso, será utilizado um *script* implementado na linguagem *Python*.

Uma vez realizado o agrupamento e divididos os *datasets*, a etapa seguinte consiste em detectar os padrões de cada *dataset* por meio da extração de características do sinal ECG, aplicando essas características a uma rede neural artificial para reconhecimento das cardiopatias e causas que implicam nas mesmas. Para extração das características do sinal, será utilizado o modelo matemático de previsão por comportamento de curvas, chamado de auto-regressivo (AR) [Liu et al 2008], onde será utilizado o passado histórico recente da curva para determinar o próximo ponto. A rede neural de topologia *perceptron* multicamadas com algoritmo de treinamento *backpropagation* **[**Widrow et al 1990**]** serão aplicados para o reconhecimento de padrões. Optou-se pelo uso de uma rede neural artificial devido a sua grande capacidade de generalização, funcionalidade importante ao tratarmos de sinais biomédicos, por serem dados com alto índice de variabilidade e com grande volume de informações. Ademais, estudos que comparam algoritmos de aprendizagem automática no domínio de classificação de arritmias, tem demonstrado que as redes neurais artificiais apresentam melhores resultados em termos de precisão, sensibilidade, predição e taxas de falsos positivos, sendo inferior apenas em tempo de execução [Moavenian e Khorrami, 2010], [Luz et al 2016].

Identificados os padrões dos sinais, um novo mecanismo de classificação de ECG será desenvolvido. Para isso, serão definidas medidas de similaridade baseadas nas causas anteriormente classificadas. O classificador ECG receberá como entrada os sinais ECG de um determinado indivíduo e fornecerá como saída a cardiopatia apresentada, associada a possível causa da qual a doença decorre. Para validar o mecanismo proposto serão realizados experimentos em sinais ECG, associados à seus respectivos diagnósticos, disponibilizados na base de dados *PhysioBannk.*

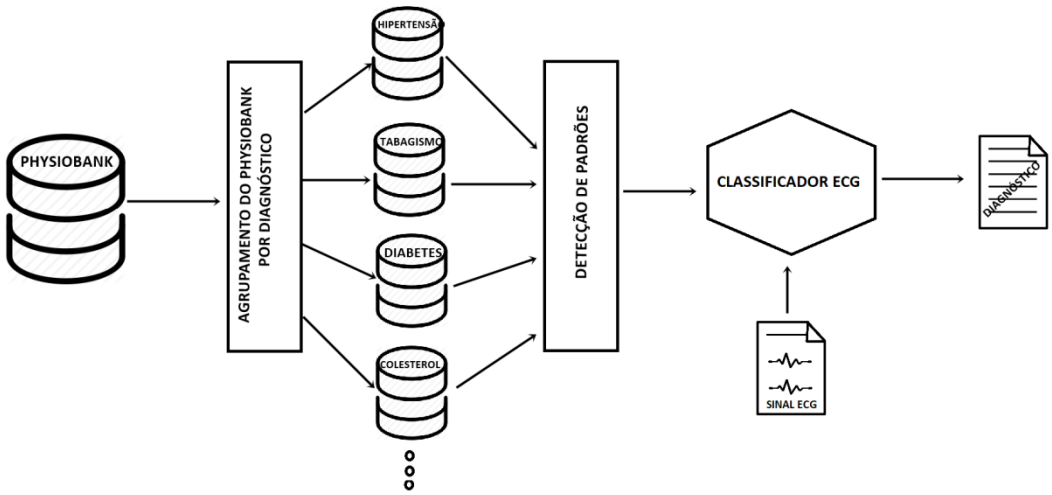


Figura 2 - Arquitetura da Solução Proposta

4. Trabalhos Relacionados

[Belle et al 2015], investiga a relevância da análise dos grandes volumes de dados produzidos pelos sistemas de saúde atuais. Os autores analisam alguns *datasets* disponíveis, dentre eles o *PhysioBank*, concluindo que o desenvolvimento da pesquisa neste espaço, embora relevante e tendo a capacidade de proporcionar um impacto significativo em seu campo de atuação, ainda é tido como desafiador tendo em vista os paradigmas inerentes aos dados.

Em [Elhaj et al 2016] são propostas técnicas que melhoram a classificação de sinais ECG, por meio da extração de significativas informações escondidas no sinal ECG em condições ruidosas. No entanto, tal pesquisa é direcionada apenas para o contexto de identificação de arritmias. Por fim, [Shadmand et al 2016] classifica os batimentos cardíacos do ECG, utilizando uma rede neural como classificador. Semelhante à este trabalho, a construção geral da rede é determinada pela movimentação de sinais através dos blocos. A entrada na rede neural é um vetor onde seus elementos são as características extraídas dos sinais de ECG.

Apesar de apresentarem melhorias significativas para a precisão da classificação de sinais ECG, de maneira geral, os trabalhos relacionados focam fortemente em aspectos que aperfeiçoam a classificação de arritmias previamente conhecidas. Tendo isso em vista, o trabalho proposto vislumbra uma contribuição que, aliada aos métodos já existentes, represente benefícios concretos aos diagnósticos de doenças cardíacas, abordando também as causas que desencadearam as cardiopatias identificadas.

5. Conclusões

Este trabalho aborda o problema de identificação de padrões em sinais ECG, com o intuito de identificar possíveis doenças relacionadas ao coração e suas possíveis causas. A solução proposta consiste na elaboração de um classificador ECG que além de identificar a cardiopatia, será capaz de indicar à causa associada a doença com base em uma análise probabilística decorrente das características do sinal.

Atualmente, o agrupamento da base de dados por diagnóstico encontra-se em andamento. As etapas subsequentes consistem em: (i) detectar os padrões de cada *dataset* fazendo uso de uma rede neural artificial; (ii) implementar o classificador capaz de definir a arritmia, fazendo uso de uma abordagem probabilística para direcionar a causa; (iii) validar o classificador proposto, utilizando sinais ECG extraídos da base de dados *PhysioBank,* verificando se os resultados apresentados estão de acordo com o diagnóstico real previamente registrado pelo profissional de medicina no *PhysioBank*. Além disso, pretende-se comparar os resultados obtidos decorrentes das arritmias identificadas com classificadores já existentes.

Vale ressaltar que não encontramos na literatura abordagens que empreguem classificadores a fim de identificar padrões que relacionem a cardiopatia à uma causa. No entanto, a estratégia apresenta-se bastante útil no direcionamento do diagnóstico do profissional de medicina, que otimizará tempo e recursos com o emprego do mecanismo.

Referências

Belle, Ashwin, et al. "Big data analytics in healthcare." BioMed research international 2015 (2015).

Elhaj, Fatin A., et al. "Arrhythmia recognition and classification using combined linear and nonlinear features of ECG signals." Computer methods and programs in biomedicine 127 (2016): 52-63.

Goldberger, Ary L., et al. "Physiobank, physiotoolkit, and physionet." Circulation 101.23 (2000): e215-e220.

Jambukia, Shweta H., Vipul K. Dabhi, and Harshadkumar B. Prajapati. "Classification of ECG signals using machine learning techniques: A survey." Computer Engineering and Applications (ICACEA), 2015 International Conference on Advances in. IEEE, 2015.

Kavitha, R., and T. Christopher. "A Study on ECG Signal Classification Techniques." International Journal of Computer Applications 86.14 (2014).

Kohler, B-U., Carsten Hennig, and Reinhold Orglmeister. "The principles of software QRS detection." IEEE Engineering in Medicine and Biology Magazine 21.1 (2002): 42-57.

Liu, Weifeng, Puskal P. Pokharel, and Jose C. Principe. "The kernel least-mean-square algorithm." IEEE Transactions on Signal Processing 56.2 (2008): 543-554.

Luz, Eduardo José da S., et al. "ECG-based heartbeat classification for arrhythmia detection: A survey." Computer methods and programs in biomedicine 127 (2016): 144-164.

Mackay, Judith, et al. The atlas of heart disease and stroke. World Health Organization, 2004.

Moavenian, Majid, and Hamid Khorrami. "A qualitative comparison of artificial neural networks and support vector machines in ECG arrhythmias classification." Expert Systems with Applications 37.4 (2010): 3088-3093.

Mozaffarian, Dariush, et al. "Heart disease and stroke statistics—2016 update." Circulation 133.4 (2016): e38-e360.

Shadmand, Shirin, and Behbood Mashoufi. "A new personalized ECG signal classification algorithm using block-based neural network and particle swarm optimization." Biomedical Signal Processing and Control 25 (2016): 12-23.

Spach, Madison S. e J. Mailen Kootsey. "A natureza da propagação elétrica no músculo cardíaco". American Journal of Physiology - Heart and Circulatory Physiology 244.1 (1983): H3-H22.

Widrow, Bernard, and Michael A. Lehr. "30 years of adaptive neural networks: perceptron, madaline, and backpropagation." Proceedings of the IEEE 78.9 (1990): 1415-1442.

Metadata Curation Framework for Supporting Data Ecosystems

Marcelo Iury S. Oliveira[1,2], Bernadette Farias Lóscio (advisor)[1]

[1]Programa de Pós-Graduação em Ciências da Computação – Centro de Informática – Universidade Federal de Pernambuco (UFPE) - Recife – PE – Brasil

[2]Unidade Acadêmica de Serra Talhada -- Universidade Federal Rural de Pernambuco

marcelo.iury@ufrpe.br, {miso, bfl}@cin.ufpe.br

Level: Doctoral Degree
Admission: April 2014
Expected Conclusion: July 2018

Concluded stages: Literature Review; Preliminary definition of the metadata curation framework; Data Ecosystem Analysis; Preliminary development of core meta-model

Future stages: Formalization and Improvement of the metadata curation framework; Extension of meta-model; Validation of the proposed solutions.

Resumo. *Existe um consenso geral quanto ao papel crucial que os metadados podem desempenhar na sustentabilidade de Ecossistemas de Dados. No entanto, na maioria dos casos, o gerenciamento de metadados é apresentado de forma superficial ou não é implementado pelos trabalhos científicos da área. O emprego de uma estratégia de curadoria de metadados pode trazer sustentabilidade a um ecossistema e ainda garantir a realização dos objetivos dos atores envolvidos em Ecossistemas de Dados. Neste trabalho, nossa contribuição é dupla: um framework de curadoria de metadados e um meta-modelo de metadados. O primeiro pretende propor um arcabouço de curadoria de metadados que oriente os atores de forma a garantir a disponibilidade de metadados que atendam a um conjunto de requisitos de qualidade desejáveis. A promessa é o emprego de uma estratégia de curadoria bem concebida e eficiente para metadados. Este trabalho também pretende propor um meta-modelo de metadados que descreve os elementos essenciais de um Ecossistema de Dados.*

Abstract. *There is a general consensus as to the crucial role metadata can play on the Data Ecosystem. However, in most cases, the metadata management is underspecified, if not unaddressed at all. The employment of a metadata curation strategy can bring an ecosystem success and further ensure realization of Data Ecosystem actors' purposes. In this work, our contribution is two-fold: metadata curation framework and core meta-model. The former aims to propose a metadata curation framework that guides actors towards a available metadata that address a set of desirable quality requirements. The promise is the employment of a well-conceived, efficient curation strategy for metadata. This work also aims to propose a core metadata meta-model for describing Data Ecosystem elements.*

Keywords*: Data Ecosystem, metadata management, metadata curation.*

1. Introduction and Motivation

The highly rapid development of networks, devices, and Web related technologies in recent years are opening up various forms of capturing, storing, and analyzing collections of data [Khan et al. 2014]. The promise is to enable Data Ecosystems (DEs), where different actors produce and consume collections of data. Typically, a DE relies on a vast set of actors [Heimstädt 2014]. These actors are heterogeneous and autonomous, each one with different properties, quality and functional requirements.

This distributed, heterogeneous and dynamic DE landscape turns difficult, or in some cases hinders, the sustainability of ecosystems. Metadata have been seen as critical to the continued success of DE initiatives [Dinter et al. 2015]. For instance, metadata are the foundation for harnessing the vast and diverse amounts of data before they become unmanageable [Dinter et al. 2015]. When metadata are available, the objects (*e.g.*, actors and data ecosystem resources) that they describe can be rapidly located and accessed for new applications [Dinter et al. 2015]. In this sense, metadata need to be maintained and preserved as well as to be highly available over long-time in order to be properly discovered and used by the DE actors.

Most of the researches on DE assume the existence of catalogues or repositories that are used to store and to manage metadata [Dinter et al. 2015, Belhajjame et al. 2008]. However, in most cases, the metadata management is underspecified, if not unaddressed at all [Dinter et al. 2015, Belhajjame et al. 2008]. The lack of a well-defined strategy for organizing, preserving, and provisioning metadata for the long term has resulted in valuable information becoming lost or discarded [Dinter et al. 2015]

A promising solution is the employment of a well-conceived curation strategy for metadata. Metadata curation is the continuous process of managing, improving and enhancing the metadata and its use [Freitas e Curry 2016]. Furthermore, the metadata curation aims to ensure that the metadata meets a defined set of quality requirements, such as integrity constraints or metadata availability expectations. One of the major challenges towards achieving efficient and continuous curation of metadata is creating a methodology to structure curation activities as well as specifying the set of tools and techniques conceived to support the curation activities.

A metadata curation needs to give those who use and contribute to the metadata a sense of ownership and control [Goble et al. 2008]. Data Ecosystem actors should be involved with the capture and the collection of metadata, the preservation of metadata, the analysis arising from metadata everyday use, and other tasks related to the metadata maintenance.

The systematic curation of metadata for supporting DE requires an underlying metadata model to describe all the aspects related to the underlying DE. Hence, in order to move towards sustainable DE, the metadata maintained should embrace the whole DE by detailing, conceptualizing and interrelating their actors and resources. DE metadata should include information about data processing activities, descriptive and technical information about ecosystem's resources [Dinter et al. 2015]. Although explicit and comprehensive representation of metadata about DE has been identified as essential, most commercial and scientific approaches only provide limited solutions ignoring important types of metadata. In particular, the metadata about the processes used to consume and produce data has not yet been investigated sufficiently.

In this work, our contribution is two-fold: metadata curation framework and common metadata meta-model. The former aims to propose a metadata curation framework that guides DE actors towards available metadata that address a set of desirable quality requirements. The promise is the employment of a well-conceived, efficient curation strategy for metadata. This work also aims to propose a core metadata meta-model for describing DE elements. It is composed of different layers that deal, on the one hand, with data and, on the other hand, with metadata from activities used to monitor and control the DE functioning.

This paper is organized as follows: Section 2 discusses related work. Section 3 presents our proposal. Section 4 depicts our research method. Finally, Section 5 discusses future work.

2. Related Work

Curation has traditionally been associated with archivists, librarians, scientists and historians. However, curation of digital resources is a relatively new research field. For many authors, the curation of data and metadata are sub-aspects of digital curation [Higgins 2008]. Another similar tendency is understating data curation as a digital curation applied to datasets. In fact, most of the digital curation works have been performed entirely on digital data.

Through the literature, a broad variety of models and frameworks used for the curation of data were proposed, such as the works [Higgins 2008, DDI 2012, Lee 2010, Burton e Treloar 2009]. These works focus on the specification of stages, steps and/or methods related to the curation of data. The DCC Curation Lifecycle Model [Higgins 2008] proposes a full lifecycle of data curation tasks, intended as a planning tool for data producers, curators and data consumers. The Data Documentation Initiative (DDI) Combined Lifecycle Model [DDI 2012] is another more linear curation model for research data, particularly social science data. Historically, DDI was focused on data archiving. A more complete curation framework is the Open Archival Information System (OAIS) Reference Model [Lee 2010]. The OAIS model is a conceptual framework for building a complete data curation organization, consisting of an organization of people and systems, which has accepted the responsibility to preserve information and make it available for a target community. It describes functions, roles and responsibilities of data repositories.

With a different purpose, the Australian National Data Service (ANDS) developed vocabulary, called Data Sharing Verbs, which describes the entire data curation tasks [Burton e Treloar 2009]. They are used as a structuring base for operational planning and an advocacy tool for both data producers and data consumers. The current list of Data Verbs contains Create, Store, Identify, Describe, Register, Discover, Access, and Exploit. However, these verbs are not meant to cover all functions related to research data.

In fact, a number of communities have developed curation processes for digital resources, such as multimedia content and research data [Ball 2012]. Most of these curation processes serve as a kind of framework for construct a detailed planning of curation lifecycle tailored for particular domains [Ball 2012]. In short, they emphasize two primary functions for a curation process: (i) to preserve curated resource (*i.e.*, to secure the long-term persistence of digital resources) and (ii) to provide access to the curated resource, in a manner consistent with the needs of their users [Ball 2012].

However, the problems of managing metadata are moving to a new level. Despite important, none of the presented works are fully devoted to the curation of metadata. It is, therefore, unsurprising that metadata curation in a generalist context is still an immature discipline. Moreover, very often data users have relied on ad-hoc methods (spreadsheets, documents or descriptive files) to manage and maintain metadata [Dinter et al. 2015]. Since the amount of metadata to be treated may be very large, because the potential growing number of actors and data in a Data Ecosystems, *ad-hoc* curation methods do not scale at all.

3. Proposed Solution

This work proposes a Metadata Curation Framework for Data Ecosystems, called Louvre, which provides a generic enough model that can be used for structuring metadata curation strategy to be applied to different contexts and serve different Data Ecosystems. It allows describing DE elements, offers modeling facilities, address quality requirements, and provides a method that supports curation activities.

Moreover, due to the lack of a devoting individuals dedicated to data management in Data Ecosystems, the Louvre also aims creating an environment in which all the actors of a Data Ecosystem can realize the curation of their correspond metadata. We believe that engagement of actors is crucial in order to develop successful and effective curation environment. In this sense, any actor is encouraged and guided to contribute with the creation, organization, assessment, enrichment, sharing and preservation of metadata.

Instead of provide a ready to use curation method, the Louvre framework is generic in nature. The Louvre framework identifies curation actions which are applicable across the whole metadata lifecycle. By using the proposed framework, actors perform metadata curation applying well-defined procedures to ensure that the metadata they manage meet quality standards and, ultimately, adding value by making metadata more discoverable and easier to access for potential reuse. This is accomplished with knowledge about Data Ecosystem properties, the use of meta-modeling as well as the development of modeling notations, an investigation of technical specifications for Data Ecosystem descriptions, and by establishing metadata discreet layers.

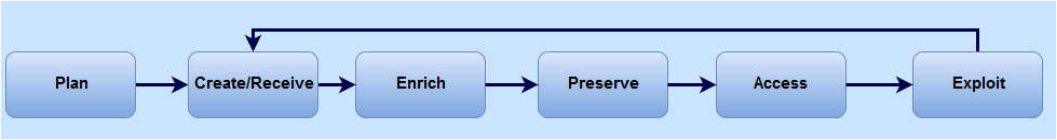


Figure 1: Metadata Curation Framework Lifecycle

The Louvre design is based on the Method Engineering, which is a theory about the development of methods to support the formalization of knowledge and to share it among practitioners [Brinkkemper 1996, Scheithauer 2009]. In our context, the knowledge represents the intended metadata. The Louvre embodies meta-models for result metadata specification, activities to guide the curation process, role definitions, tools, and techniques. Activities comprise best practices about which steps are to be performed in order to curate metadata items. A role defines a specific set of skills and capabilities which are needed for an activity. Techniques describe helpful theories to create and to assess metadata, ranging from data modeling to quality evaluation methodologies, for instance. Tools can be used to support the application of activities and techniques. Meta-model

specifies how the metadata is formalized, thereby guaranteeing the consistency of the entire produced metadata.

As can be seen in Figure 1, the Louvre framework is divided into some different stages, which are described below:

- Plan: Involves the confection of a metadata management planning, including the definition of standards and rules to guarantee the quality and preservation of metadata as well as involves the definition of activities, techniques, tools and meta-models used;
- Creation and Receive: Refers to the tasks used to reliably create, capture and collect metadata in a way that facilitates preservation and reuse. The metadata can be generated and recorded by the DE actors as well as pre-existing metadata can be collected from other sources. This step considers both manual and automatic metadata harvest methods;
- Enrich: Involves the analyzing, refining, cleaning, formatting and transformation of the metadata. This stage can also perform some quality assurance and quality control;
- Preserve: Involves the different ways in which metadata are stored to ensure long-term access to them. It also implies the ensuring that the metadata can be verified, replicated and actively curated over time. A suitable metadata storage should have indexing, replication, distribution and backup features, among other services;
- Access: Involves the tasks focused on keeping the metadata accessible. To guarantee the appropriate access to metadata, it is advised to provide a suitable search engine to retrieve these data. It also may involve controlling the access to metadata;
- Exploit: Comprises series of actions and methods performed on metadata. In fact, it is the actual use of the metadata by Data Ecosystem actors;

Moreover, for establishing a common way to describe metadata, the aforementioned DE properties must follow a shared understanding. This is realized by using meta-models that define the Data Ecosystem's fundamental elements and their inter-relationships. The core metadata meta-model is developed with different abstraction layers for describing a Data Ecosystem as whole. Each layer formalizes a valid aspect of functioning and management of ecosystem. Following that, a set of metadata elements is derived from the data ecosystem domain as well as from a literature research.

4. Research Method

This work, from its objective perspective, is exploratory, for providing a better understanding on the proposed problem and also allowing the construction of hypotheses [Yin 2013]. More specifically, this work aims to investigate and development of a metadata curation framework for supporting Data Ecosystems. Our research design can be defined as a pragmatic research paradigm that aims to create novel and innovative artifacts to solve real-world problems [Von Alan et al. 2004]. It involves creation and evaluation of artifacts in order to solve a specific problem. Based on our goals and the pragmatic paradigm, we divided the research design into four phases.

The first phase aims understanding Data Ecosystem as well as identifying metadata elements that constitute DEs. The research strategy adopted is the literature review that allowed the identification in the literature of the constituent conceptual elements of DEs. The second phase of research aims to realize the goal of building a common metadata model for DEs in establishing concepts and terminologies well founded in the form of a core meta-model. The research strategy adopted is the research-design science.

The third phase aims proposing the metadata curation framework. This research phase will formalize the problem to be solved by defining the features and requirements to be implemented. As one of the research goals is the development of the metadata curation framework itself, so build it is an essential methodological component.

Both meta-model and metadata curation framework development guided by an iterative development methodology, where each iteration of the process will be delivered new components to the developed artifacts. The fourth phase of research aims to evaluate developed artifacts in the earlier phases. The research strategy adopted is Case Study and Focus Group.

5. Future Work

In this work the metadata curation problem was presented. While state-of-the-art works had already proven the benefit of digital curation, the majority of the works simply preserves datasets and fails to explore others related to Data Ecosystems. In addition, existing curation processes, besides not specifying how metadata curation should be mantained, are not thought to support the autonomous and displaced context of the actors.

Therefore, this work is inserted in this database area to propose a framework for metadata curation to support the operation of Data Ecosystems. Preliminary investigations have already identified the important metadata for Data Ecosystems and how they relate. We have also studied how to structure and support the curatorial activities in order to guarantee the availability and adequacy of the metadata for contemporary and future use.

As future steps, we intend to: (i) finalize the meta-model conception; (ii) finalize the specification of the metadata curation framework; and (iii) evaluate and evolve the proposed solutions.

Parts of the investigation about Data Ecosystem have been published at the works [Oliveira 2016][Oliveira 2017]. Our preliminary planning for further publications involves the following works and venues: (i) "*Investigation about Data Ecosystems: a Systematic Mapping Study*" to Knowledge and Information Systems journal; (ii) "*Investigations about Data on the Web Publication and Consumption: A Systematic Mapping Study*" to WWW journal; (iii) "*Towards a Meta-model for Data Ecosystems*" to SAC; (iv) "*Evaluating a Metadata Curation Framework with Survey Based on Expert Opinion*" to International Journal of Metadata, Semantics and Ontologies.

Acknowledgments

This work was partially supported by funds from the following Brazilian funding agencies: FACEPE and INES. Marcelo Iury also gives thanks to the CNPq and CAPES, for their granting of a PHD fellowship. The authors would also like to thank the colleagues of the Aladin research group for their input for this paper.

References

Ball, A. (2012). Review of the State of the Art of the Digital Curation of Research Data. University of Bath, 2012.

Belhajjame, K., Wolstencroft, K., Corcho, O., Oinn, T., Tanoh, F., William, A., & Goble, C. (2008, May). Metadata management in the taverna workflow system. In Proceedings of CCGRID'08. (pp. 651-656).

Brinkkemper, S. (1996). Method engineering: engineering of information systems development methods and tools. Information and software technology, 38(4), 275-280.

Burton, A., & Treloar, A. (2009). Designing for Discovery and Re-Use: the ANDS Data Sharing Verbs Approach to Service Decomposition. International Journal of Digital Curation, 4(3), 44-56.

DDI (2012). Data documentation initiative specification. Data Documentation Initiative. Available at http://www. ddialliance.org/Specification.

Dinter, B., Gluchowski, P., & Schieder, C. (2015). A Stakeholder Lens on Metadata Management in Business Intelligence and Big Data–Results of an Empirical Investigation. AMCIS 2015 Proceedings, 2015.

Freitas, A., & Curry, E. (2016). Big Data Curation. In New Horizons for a Data-Driven Economy (pp. 87-118). Springer International Publishing.

Goble, C., Stevens, R., Hull, D., Wolstencroft, K., & Lopez, R. (2008). Data curation+ process curation= data integration+ science. Briefings in bioinformatics, 9(6), 506-517.

Heimstädt, M., Saunderson, F., & Heath, T. (2014, May). Conceptualizing Open Data ecosystems: A timeline analysis of Open Data development in the UK. In *Proceedings of the International Conference for E-Democracy and Open Government* (pp. 245-255).

Higgins, S. (2008). The DCC curation lifecycle model. International Journal of Digital Curation, 3(1), 134-140.

Khan, N., Yaqoob, I., Hashem, I. A. T., Inayat, Z., Mahmoud Ali, W. K., Alam, M., ... & Gani, A. (2014). Big data: survey, technologies, opportunities, and challenges. The Scientific World Journal 2014 (2014).

Lee, C. A. (2010). Open archival information system (OAIS) reference model. Encyclopedia of Library and Information Sciences, 4020-4030.

Oliveira, L. A., Oliveira, M. I. S. & Lóscio, B. F (2017, October). Um Survey sobre Soluções para Publicação de Dados na Web sob a Perspectiva das Boas Práticas do W3C. In *Proceeding of SBBD*, 2017

Oliveira, M. I. S., de Oliveira, H. R., Oliveira, L. A., & Lóscio, B. F. (2016, June). Open Government Data Portals Analysis: The Brazilian Case. In *Proceedings of the 17th DGO* (pp. 415-424). ACM.

Scheithauer, G. (2009). Business Service Description Methodology for Service Ecosystems. Proceedings of the 16th CAiSE-DC, 9, 9-10.

Von Alan, R. H., March, S. T., Park, J., & Ram, S. (2004). Design science in information systems research. MIS quarterly, 28(1), 75-105.

Yin, R. K. (2013). Case study research: Design and methods. Sage publications.

An External Memory Approach for de Bruijn Graph Construction

Author: Elvismary M. de Armas[1]
Advisor: Sérgio Lifschitz[1]

[1]Departamento de Informática
Pontifícia Universidade Católica do Rio de Janeiro (PUC-Rio)

{earmas, sergio}@inf.puc-rio.br

Status:

- Ingresso no doutorado: Março 2014
- Exame de qualificação: Dezembro 2016
- Defesa de proposta de tese: Maio 2017
- Data prevista para defesa de tese: Fevereiro 2018

Related publications:

- "K-mer Mapping and De Bruijn Graphs: the case for Velvet Fragment Assembly". Armas, Elvismary M., Lifschitz, Sérgio, Haeusler, E. H. ; Holanda, Maristela T. ; Ferreira, P. C. G. ; Silva, W. M. C. Proceedings in IEEE International Conference on Bioinformatics and Biomedicine (BIBM 2016), 2016. p. 882-889.
- "Sugarcane transcriptome analysis in response to infection caused by *Acidovorax avenae subsp. avenae*". Ferreira, P. C. G. ; Brigida, A. B. S. ; Rojas, C. ; Grativol, C. ; Armas, E. M. ; Entenza, J. O. ; Thiebaut, F. ; Lima, M. ; Farrinelli, L. ; Hemerly, A. ; Lifschitz, Sérgio . PLOS One 11(12) e0166473, 2016.

Resumo. *A montagem de fragmentos de genoma é um problema fundamental na bioinformática. Na montagem de novo, onde não existe uma cadeia de referencia, é usada a estrutura de dados do grafo de Bruijn para realizar o processamento computacional. No entanto, a construção do grafo de Bruijn foi identificada como o subprocesso computacional com maior consumo de memoria principal. Poucas soluções computacionais foram desenvolvidas para executar essa tarefa no modelo de memória externa, mas elas geram todos os k-mers com alta redundância, aumentando o número de dados externalizados e, consequentemente, o número de operações de I/O. Este trabalho expõe uma nova abordagem para a construção do grafo Bruijn sem a necessidade de gerar todos os k-mers em um modelo de memória externa, com o objetivo de reduzir os requisitos computacionais e aumentar o desempenho.*

1. Introduction

The computational fragment assembly [El-Metwally et al. 2013] is a fundamental problem for bioinformatics. As DNA sequencing technologies cannot read whole genomes in a single run, one needs to reconstruct the original sequences considering the large volume of short *reads*. Indeed, Next-Generation Sequencing (NGS) projects breaks the genome randomly at several places and generates several small fragments, the so called *reads* of the genome. This process generates some level of redundancy. There is an additional challenge when dealing with *de-novo* [Schatz et al. 2010] assembling methods, since there is no reference genomes to guide the assembly procedure.

Some genome assemblers have been implemented based on a *de Bruijn* graph (dBG) data structure, which helps to compute assembly overlaps (see [Bradnam et al. 2013] and Gage [Salzberg et al. 2012]). The *de Bruijn* graph approach has a fundamental computational drawback: it requires an enormous amount of memory for its construction and processing. For example, one may check the supplementary results for Assemblathon2 [Bradnam et al. 2013].

In order to build a *de Bruijn* graph, the set of short reads $R = \{r_i\}$ are firstly decomposed into *k*-mers (substrings with specific *k* length). A short read *r* is a string over the alphabet $\Sigma = \{A, T, C, G\}$, with $|\Sigma| = 4$. The short read length m depends on the sequence technology used, varying from 100 to 150bp (base pairs) with low error rates, using Illumina Technology, up to 10 to 15 kbp with higher error rates, when using Pacific Biosciences Technology. We need to identify all distinct *k*-mers (V set) and map all duplicate *k*-mers into the unique corresponding node in the *de Bruijn* graph. The total number of *k*-mers present in one read (not only distinct *k*-mers) is equal to $m - k + 1$, while the total number of *k*-mers present in n reads is $(m-k+1) * n$. The distinct ***k-mers*** space for *k* value is 4^k.

An edge between two nodes is created whenever their corresponding *k*-mers are adjacent in at least one short read. In other words, an edge is create between two *k*-mers if they have an exact suffix-prefix overlap of length *k-1* that occurs in at least one read. Therefore, a way to get the set of edges E, once the set of vertices is build, is by creating them through scanning all the reads in the short sequences file, and querying nodes in V.

The creation and manipulation of the *de Bruijn* graph has been identified as the step with most memory and runtime consumption for some assembling experiments [Li et al. 2009][Cook and Zilles 2009][Li et al. 2013]. These high memory requirements is caused by the fact that a huge number of *k*-mers need to be processed, in addition to the high level of redundancy present in those *k*-mers, evaluated as $\mathcal{O}(k \times n)$. Thus, there is a clear need for solutions that reduce the memory footprint for dBG construction.

This text is structured as follows: we will formalize next (Section 2) our technical challenges and scientific problem. Then, Section 3 contains a description of some related works within the context of this research work. Our novel approach is briefly described in Section 4. A summary of the chosen methodology and the current state of the research is given in Section 5. Finally, in Section 6, we list the expected contributions.

2. Problem formulation

To formalize our problem, we define the memory requirements for each process involved with dBG construction:

Identification of the set of distinct *k*-mers *V* is a process that, starting from a collection of all *k*-mers, C_v, such $|C_v| = (m-k+1) \times n$, it gets the set of distinct *k*-mers V, such as $V \subseteq C_v$ and $|V| < |C_v|$. There are two common ways to do that: the first one is a *sorting approach* in which $|C_v|$ *k*-mers are sorted and all duplicates are removed. The memory cost $M_{C_v}(m, n, k)$ for this solution corresponds to the needed memory for each *k*-mer times $|C_v|$. The second solution, called *on-line approach*, involves the use of specialized data structures to maintain the set V. Each *k*-mer is generated one at time and, if it is not already present in the graph structure, it is inserted. A priori, the memory cost for on-line approach M_{DS}, for a specific data structured, may be smaller than the sorting approach, because only a subset of *k*-mers are needed to be stored. However, the memory needed to the structure self-maintenance must be taken into account. The number of distinct *k*-mers would be so high that even the on-line approach fails when a RAM-only processing is executed. The memory needed to get V is defined as $M_V = minimum(M_{C_v}, M_{DS})$

Identification of the set of edges *E* is the process that from a C_e collection of all binary relation $u \rightarrow v$ between two *k*-mers such as $|C_e| = (m-k) \times n \rightarrow$ relations, it gets the set E of distinct relations corresponded with V. There are also several approaches here: one starts with $|C_e|$ elements, includes sorting and duplicates elimination, which needs $M_{C_e}(m, n, k)$ memory (including the vertices's data); the other approach generates, given a previously calculated V set, the edges scanning the Sequence reads file and builds the adjacency list representation of the graph. The last approach, which uses $M_L(m, n, k)$ memory units, including vertices, is cheaper regarding memory consumption than the sorting approach. Let $M_E = minimum(M_{C_e}, M_L)$ be the memory required for obtaining E. $M_{dBG}(m, n, k) = M_E$ is the needed memory to construct the *de Bruijn* graph.

The **external memory** model [Aggarwal and Vitter 1988] is also called the "I/O Model" or the "Disk Access Model" (DAM). An external memory model is commonly applied in algorithms developed to manage huge amount of data, such as those used in Data Base Management Systems. It simplifies the memory hierarchy to just two levels. The CPU is connected to a fast cache of size M; this cache, in turn, is connected to a much slower disk of effectively infinite size. Both cache and disk are divided into blocks of size B, so there are M blocks in the cache. Transferring one block from cache to disk B (or vice versa) costs 1 unit. Memory operations on blocks resident in the cache are free. Thus, the natural goal is to minimize the number of transfers between cache and disk. The external memory model is a good approximation to the slowest connection in the memory hierarchy. For a large database, the "cache" could be system RAM and "disk" could be the hard disk. [Massachusetts Institute of Technology 2012]

Based on above definitions, we have identified in this research work the following **problem**:
Let M be the main memory available on a computer and let $M_{dBG}(m, n, k)$ be the memory needed for building the de Bruijn graph:

- If $M \geq M_{dBG}(m, n, k)$, there are RAM-only solutions.
- If $M < M_{dBG}(m, n, k)$, an external memory solution must be used.

Then, our **scientific problem** may be defined as: *Is it possible to build a de Bruijn graph efficiently by reducing memory requirements?*

3. Related works

Among the dBG construction strategies, we have found only a few approaches that consider external memory processing. These fit into two strategies: one based on external memory sorting and the other based on *k*-mers partitioning and disk distribution.

External memory sorting is implemented in the solution proposed by [Kundeti et al. 2010]. In this work the authors present an efficient parallel strategy for constructing large *de Bruijn* graphs, also extensible to the out-of-core model. After the generation of all canonical edges, they detect duplicate *k*-mers by sorting all the edges using radix sort algorithms in an external implementation. There are some questions unanswered about how this approach manages the possibility of insufficient memory for the construction of the adjacency list structure for the graph.

The distributed processing over disk partitions presents in general three steps: distribution, process and merging. Firstly, they distribute all *k*-mers into disk partitions (non disjoint partition for all cases), process individually each partition in main memory and, later, merge them with others to build a dBG. The Minimum Substring Partition (MSP) approach [Li et al. 2013] shows a solution that breaks the short reads into multiple small disjoint partitions based on the minimum *p*-substring of the *k*-mers, allowing consecutive *k*-mers to be distributed in the same partition, decreasing the number of I/O operations. The number of distributed elements is the number of all *k-mers* generated from the sequence $((m - k + 1) \times n)$.

The approaches presented in [Chikhi et al. 2014] and [Chikhi et al. 2016] are also focused on distributing the process over disk partitions but with the direct construction of a compressed graph. Given a set of short reads and a value of k, the compacted graph $G_c(S, k)$ is a graph obtained from $G(S, k)$ by compression of all its maximal non-branching paths.The work in [Chikhi et al. 2014] proposes a new pipeline using first a DSK *k*-mer counter as an input of a new algorithm to enumerate all the maximal simple paths (called BCALM) and represent them using a new data structured called DBGFM using a FM index. Despite its excellent results, DSK assumes that the set of distinct *k*-mer values can be uniformly partitioned by this hash function, which could incur in a imbalancing of the partition files size affecting the I/O throughput. This approach takes the output of DSK and distribute distinct *k*-mer sets by minimizers frequency, while trying to compact consecutive *k*-mers that constitute a simple path in the graph. DSK + BCALM perform two cycles of disk distribution: DSK distributes all *k*-mers in partition files and process them, while BCALM distributes all distinct *k*-mers, which count for I/O operations. A sort of parallel version for the BCALM algorithm, called BCALM2, is presented in [Chikhi et al. 2016]. It differs from BCALM because it does not use DSK. All *k*-mers are distributed to the disk partitions using the same concept of frequency of $\mathcal{L}$-minimizer. However, it might be possible that the same *k*-mer is distributed to two file partitions in the beginning. As many other approaches, BCALM2 has as a negative aspect that all the *k*-mers are generated and distributed, which is added to the fact that the same *k*-mer could be distributed (copied) to two distinct files, increasing the number of I/O operations.

4. A Novel External Memory Approach

The existing external memory dBG construction approaches relies on the use of external memory sorting or on disk partitioning. Both works from the beginning considering the total number of *k*-mers, maintaining a high level of redundancy and, consequently, a greater number of I/O operations. This research has as an hypothesis that it is possible to reduce runtime and memory requirements for dBG construction by reducing the number of elements to process from the beginning. It only uses an external memory processing for the final steps, for a smaller number of elements and when the number of presented elements are strictly unavoidable.

Given the fact that the objective of dBG is to find read overlaps with a minimum size *k*, we propose a novel approach for dBG construction for genome fragment assembly, with the following principles:

a) Find overlaps regions greater than k earlier can save the corresponding memory to store the redundant information for each *k*-mer and redundant information for consecutive *k*-mer chains are duplicated.

In the following we analyze the amount of elements that we would avoid to process with the new approach. Let be $R = \{r_1, r_2, .., r_n\}$ a set m length reads, if we have a substring of m, s_i, such as $|s_i| = s$, $k < s$, then:

- the number of *k*-mers in R is $(m - k + 1) \times n$ (k length strings)
- the number of characters in R as *k*-mers is $(m - k + 1) \times n \times k$
- the number of *k*-mers in s is $(s - k + 1)$ (k length strings)

If there is a substrings of m, s_1, such as $|s_1| = s$ and $k < s$ and there is at least another substrings of m, s_i such as $s_1 = s_i$, then:

- $(s - k + 1) \times frequency(s_1)$ *k*-length strings will not be processed.
- $(s - k + 1) \times k \times frequency(s_1) - s$ characters will not be processed.

If there is a substring l_1 of s_1, such that $|l_1| = l$, $k \leq l < s$ and l_1 has external copies, then:

- $(s - k + 1) \times (frequency(s_1) - 1)$ *k*-length strings will not be processed.
- $(s - k + 1) \times k \times (frequency(s_1) - 1)$ characters will not be processed.

i.e. although we have to decompose s_1 to search overlaps of length l, we still have a set of important elements that we will not have to be processed.

b) Avoid generate all k-mers using the following steps:

- Search overlaps regions of size k_1, $k \leq k_1 < m$, generating vertices called k_1-chars and apply the suffix-prefix overlap of length $(k - 1)$ criteria for edges. This generates what we call an extra-compacted de Bruijn Graph $G_{k1}(V, E)$ with $k \leq k_1 < m$.
- Search overlaps regions of size k_2, $k \leq k_2 < k_1$. Decompose each element in V from $G_{k2}(V, E)$ in k_2-chars vertices and update the edges. This generates an extra-compacted de Bruijn Graph $G_{k2}(V, E)$.
- Search for minor overlaps in each round $(k \leq k_n < ... < k_3 < k_2 < k_1)$ until all overlaps of k size are found.

Since we avoid to process a large amount of redundant *k*-mers from the beginning, it is likely that we need to use a disk distribution approach only at the last rounds of the algorithm, avoiding to externalize an important volume of data into disk, decreasing the number of I/O operations.

5. Methods

The thesis subject arose from a demand of *de-novo* assembling for sugarcane sequences as a part of a research cooperation between PUC-Rio's BioBD Laboratory and the Laboratory of Molecular Biology of Plants (IBqM), Institute of Medical Biochemistry at UFRJ. One goal is to study the sugarcane genome into Brazilian species, which is very complex as there are high rates of heterozygosity and repetitions, demanding a high availability of main memory so that assembly programs can complete their executions.

An initial bibliographical survey was carried out, where the works related to this thesis subject were examined, looking for ad-hoc data structures, computer processing and variables that impact the dBG construction memory footprint. Also, two assembling tools Velvet [Zerbino 2016] and SOAPdenovo[Luo et al. 2012] were thoroughly studied, in order to understand how the *k*-mers are generated, and which programming structures are used for implementing the distinct *k*-mers identification subroutine. Moreover, two approaches were implemented for Velvet([Zerbino 2016] in that process. The first one was an strategy based on a buffer manager and persists parts of a hash table in disk, when the available memory is not enough to fit the hash table. The second approach [de Armas, E. M. et al. 2016] was the implementation of the *k*-mer mapping process (identify distinct *k*-mer and map duplicate ones) as functions over a DBMS), evaluating the impact of different indexes in its performance. All of these activities allows one to have a deeper knowledge and understanding of the problem ,and define the variables involved besides some bounds and limits.

6. Contributions

This approach, as far as we know, is unique in the sense that it combines two principles for dBG construction:

- the reduction of the total number of *k*-mers to be analyzed, with positive runtime impact in both RAM-only and external memory model processing.
- less delay for the external memory processing (if its needed) for the last steps of the algorithm, reducing the total number of *k*-mers to be externalized and, consequently, the number of I/O operations.

Furthermore, this thesis proposal brings a set of additional contributions, as listed below:

- Identification and formalization of the main variables that impact the dBG construction.
- A new disk distribution strategy.
- A survey of dBGs approaches, emphasizing the data structures, algorithms and disk distribution algorithms used.
- An actual implementation of the approach.

References

Aggarwal, A. and Vitter, Jeffrey, S. (1988). The input/output complexity of sorting and related problems. *Commun. ACM*, 31(9):1116–1127.

Bradnam, K. R. et al. (2013). Assemblathon 2: evaluating de novo methods of genome assembly in three vertebrate species. *GigaScience*, 2(1):1–31.

Chikhi, R., Limasset, A., Jackman, S., Simpson, J. T., and Medvedev, P. (2014). *On the Representation of de Bruijn Graphs*, pages 35–55. Springer International Publishing, Cham.

Chikhi, R., Limasset, A., and Medvedev, P. (2016). Compacting de bruijn graphs from sequencing data quickly and in low memory. *Bioinformatics*, 32(12):i201.

Cook, J. J. and Zilles, C. (2009). Characterizing and optimizing the memory footprint of de novo short read dna sequence assembly. In *Performance Analysis of Systems and Software, 2009. ISPASS 2009. IEEE International Symposium on*, pages 143–152.

de Armas, E. M., Haeusler, E. H., Lifschitz, S., de Holanda, M. T., da Silva, W. M. C., and Ferreira, P. C. G. (2016). K-mer mapping and de bruijn graphs: The case for velvet fragment assembly. *Procedings IEEE International Conference on Bioinformatics and Biomedicine (BIBM)*, pages 882–889.

El-Metwally, S., Hamza, T., Zakaria, M., and Helmy, M. (2013). Next-generation sequence assembly: Four stages of data processing and computational challenges. *PLoS Comput Biol*, 9(12):1–19.

Kundeti, V., Rajasekaran, S., and Dinh, H. (2010). Efficient Parallel and Out of Core Algorithms for Constructing Large Bi-directed de Bruijn Graphs. *ArXiv e-prints*.

Li, R. et al. (2009). De novo assembly of human genomes with massively parallel short read sequencing. *Genome Research*.

Li, Y., Kamousi, P., Han, F., Yang, S., Yan, X., and Suri, S. (2013). Memory efficient minimum substring partitioning. *Proc. VLDB Endow.*, 6(3):169–180.

Luo, R. et al. (2012). Soapdenovo2: an empirically improved memory-efficient short-read de novo assembler. *GigaScience*, 1(1):1–6.

Massachusetts Institute of Technology, E. D. (2012). Lecture notes in Advanced Data Structures, MIT course number 6.851.

Salzberg, S. L. et al. (2012). Gage: A critical evaluation of genome assemblies and assembly algorithms. *Genome Research*, 22(3):557–567.

Schatz, M. C., Delcher, A. L., and Salzberg, S. L. (2010). Assembly of large genomes using second-generation sequencing. *Genome Research*, 20(9):1165–1173.

Zerbino, D. (2016). Velvet software. embl-ebi.

Uma Abordagem para Criação e Uso de Perfis de Conjuntos de Dados com Metadados Enriquecidos Semanticamente

Aluna: Natacha Targino Rodrigues Simões Brasileiro

E-mail: ntrsb@cin.ufpe.br

Orientadora: Ana Carolina Brandão Salgado

E-mail: acs@cin.ufpe.br

Co-orientadora: Damires Yluska de Souza Fernandes

E-mail: damires@ifpb.edu.br

Nível: Mestrado

Universidade Federal de Pernambuco – UFPE

Programa de Pós Graduação em Ciência da Computação – Centro de Informática

Ingresso: março/2016

Conclusão prevista: fevereiro/2018

Etapas Concluídas: Créditos em disciplinas, Referencial Bibliográfico, Definição do Problema, Especificação, Implementação da Primeira Versão da Abordagem Proposta.

Etapas Futuras: Finalização da Especificação e Implementação, Realização de Experimentos, Escrita da Dissertação.

Publicações Relacionadas: Targino, N., Souza, D., Salgado, A. C. (2017) "Uma Proposta de Perfil de Conjuntos de Dados na Web com Enriquecimento Semântico" To appear in: Anais do 32th Simpósio Brasileiro de Banco de Dados (SBBD).

Abstract. *Most datasets published on the Web do not have enough metadata to describe them, which makes it difficult to locate and access them using search engines or applications that can use them. Providing a dataset profile facilitates communication between publishers and consumers and the integrated use of datasets. In this light, this work proposes an approach that describes datasets on the Web through the generation of a profile composed of descriptive, structural and quality semantic enriched metadata. This work presents the proposal of the approach, its main characteristics and some results obtained until now.*

Keywords: *Web Datasets, Dataset Profile, Metadata enrichment*

1. Introdução e Motivação

A variedade de conjuntos de dados disponibilizados na Web possibilita um cenário ilimitado de informações, e combinações dessas informações podem trazer descobertas importantes. Entretanto, quase nunca os produtores de dados e seus consumidores se conhecem. Logo, é necessário fornecer informações sobre os conjuntos de dados que contribuam para a comunicação entre eles, facilitando sua compreensão e reutilização.

Como uma boa prática de publicação de dados na Web, o *World Wide Web Consortium* (W3C)[1] recomenda o uso de metadados para o fornecimento de informações que ajudem usuários e aplicações a entender os dados, bem como outros aspectos importantes que descrevam um conjunto de dados ou sua forma de disponibilização. O W3C também recomenda que os dados publicados sejam enriquecidos sempre que possível, por meio de um conjunto de processos que podem ser utilizados para aumentar, refinar ou melhorar dados brutos ou processados anteriormente [Lóscio et al. 2017]. Essa atividade também pode ser realizada com relação aos metadados, de modo a ajudar na atribuição de significado e melhoria de suas descrições, como também complementando informações que promovam a compreensão e facilitem o processamento dos dados pelos consumidores.

Neste panorama o problema alvo deste trabalho é definido como segue: *Dado um ecossistema de produção e consumo de dados na Web, como gerar e manter metadados que forneçam informações descritivas sobre o conteúdo, estrutura, qualidade e que possibilitem a compreensão e processamento dos dados?*

Como hipótese de solução, este trabalho apresenta uma proposta de abordagem que descreve conjuntos de dados publicados na Web por meio de um perfil composto de metadados descritivos, estruturais e de qualidade que são enriquecidos semanticamente. São gerados metadados descritivos enriquecidos com a identificação do domínio de conhecimento ao qual o conjunto de dados pertence (e.g., saúde, música). Os metadados estruturais são gerados a partir da identificação da composição estrutural (propriedades) do conjunto de dados, e enriquecidos com a recomendação de vocabulários de domínio que podem referenciá-los. Já os metadados de qualidade estão relacionados aos aspectos do conjunto de dados e são enriquecidos a partir do *feedback* de consumidores. O trabalho vem sendo especificado, foi implementada uma primeira versão da abordagem, que inclui alguns dos metadados descritivos e estruturais. Experimentos realizados verificaram a relevância do enriquecimento, em termos do domínio identificado e dos vocabulários recomendados.

Este trabalho está organizado como segue: a Seção 2 introduz alguns conceitos; a Seção 3 descreve a caracterização da contribuição; a Seção 4 apresenta e avalia os resultados obtidos até o momento; a Seção 5 discute alguns trabalhos relacionados, e a Seção 6 tece considerações, indicando os próximos passos para sua conclusão.

2. Fundamentação Teórica

Um conjunto de dados pode ser definido como uma coleção de dados, publicados ou curados por um agente e disponível para acesso ou *download* em um ou mais formatos

[1] https://www.w3.org/

[Maali et al. 2014]. A utilização de metadados que descrevem esses conjuntos de dados possibilita aos consumidores o acesso a informações adicionais para o entendimento de seu significado e de sua estrutura. Para viabilizar a estruturação dos metadados de conjuntos de dados, alguns autores propuseram a criação de perfis. Abele (2016) define o perfil de um conjunto de dados como o grupo de informações descritivas e estatísticas a seu respeito. Segundo [Ellefi et al. 2014], a criação de um perfil ajuda na identificação de conjuntos de dados, podendo ser definido como um conjunto de características, tanto semânticas quanto estatísticas, que permitem sua melhor descrição.

Metadados que indiquem a qualidade do conjunto de dados também podem ser adicionados ao perfil. Os critérios de qualidade devem prover a melhor representação dos aspectos de um conjunto de dados, possibilitando avaliar sua adequação para uma determinada tarefa. Existem vários critérios de qualidade que podem ser utilizados para abranger os mais diferentes aspectos de um conjunto de dados como, por exemplo, exatidão, completude, objetividade, relevância e compreensibilidade [Naumann et al. 2000]. Como os critérios de qualidade podem variar em função do tempo e do contexto, é possível que esses metadados sejam atualizados a partir do *feedback* de consumidores de dados, o que também pode prover uma interação maior entre publicadores e consumidores.

3. Caracterização da Contribuição

Considerando a literatura sobre perfis de conjuntos de dados [Abele 2016; Assaf et al. 2015] e, de acordo com as indicações de boas práticas para publicação de dados na Web do W3C [Lóscio et al. 2017], define-se, a seguir, o conceito de Perfil de Conjunto de Dados.

Definição. Perfil de Conjunto de Dados (PCD). Um Perfil de Conjunto de Dados pode ser compreendido como uma anotação específica que contém metadados descritivos, estruturais e de qualidade referentes a um determinado conjunto de dados publicado na Web.

A abordagem proposta compreende os processos relacionados ao ciclo de vida do PCD (Figura 1). No ecossistema ao qual a abordagem está inserida existem diferentes atores, podendo ter papéis de publicadores, que estão relacionados à publicação, e/ou de consumidores, que estão relacionados ao consumo desses conjuntos de dados publicados. Os publicadores de dados podem criar um novo PCD a partir de um conjunto de dados existente ou atualizar um PCD conforme a atualização do conjunto de dados correspondente. Após a criação ou atualização, o perfil é publicado e então estará disponível para o consumo. Os consumidores também podem dar um *feedback* sobre o conjunto de dados correspondente ao perfil. Com o *feedback,* os metadados de qualidade do PCD serão enriquecidos de forma automática e o perfil é novamente publicado. Para referenciar esses metadados, pretende-se utilizar vocabulários recomendados e disponíveis.

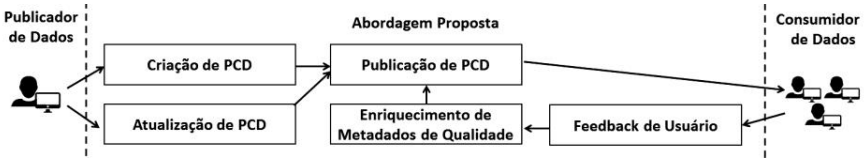


Figura 1. Atividades e Atores relacionados à abordagem.

A Figura 2 apresenta o processo de criação do PCD até a sua disponibilização para o consumo, de acordo com a abordagem proposta. A criação é iniciada a partir de um conjunto de dados de um publicador. São executadas as seguintes etapas para a criação do perfil: (i) geração de metadados descritivos: extração de alguns metadados descritivos e o enriquecimento com a identificação de domínio; (ii) geração de metadados estruturais: são identificadas as propriedades do conjunto de dados e enriquecidas com a recomendação de vocabulários de domínio; e (iii) geração de metadados de qualidade: são extraídos e calculados alguns metadados relacionados aos aspectos do conjunto de dados e são enriquecidos a partir do *feedback* de usuários. O perfil é gerado de forma automática, mas, caso o produtor deseje realizar alguns ajustes, é possível editá-lo. Uma vez gerados os metadados, o PCD é publicado e disponibilizado para os consumidores de dados.

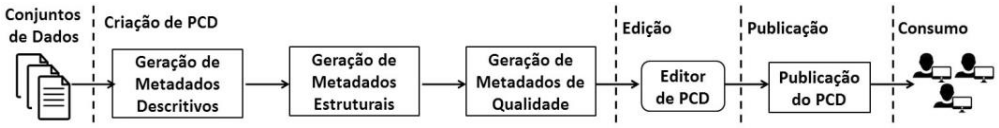


Figura 2. Processos relacionados à criação e disponibilização do PCD.

3.1 Metadados Descritivos

Esses metadados descrevem os recursos gerais dos conjuntos de dados e sua forma de disponibilização, permitindo sua compreensão e descoberta automática [Lóscio et al. 2017]. Entre os metadados descritivos escolhidos para a composição do perfil (Tabela 1), alguns fazem parte do esquema proposto pelo W3C, que são: título, palavras-chave, publicador, data de publicação, data da última modificação, domínio (tema), formato (tipo de mídia).

Tabela 1: Metadados Descritivos

Metadados Descritivos	Tipo de dado recebido	Descrição
Título	Literal	Título do conjunto de dados
Palavras-chave	Literal	Palavras-chave identificadas no conjunto de dados
Domínio (Tema)	Literal	Domínio de conhecimento do conjunto de dados
Vocabulário de domínio	URI	Vocabulário de domínio recomendado para referenciar as propriedades do conjunto de dados.
Endereço do Conjunto de Dados	URL	Endereço em que o conjunto de dados está disponível
Data da última atualização	Data	Data da última atualização do conjunto de dados
Data de publicação	Data	Data da criação do conjunto de dados
Publicador	Literal	Nome do publicador do conjunto de dados
Formato	Literal	Formato em que o conjunto de dados está disponível
Tamanho	Literal	Tamanho do conjunto de dados
Versão	Numeral	Versão do conjunto de dados

3.2 Metadados Estruturais

Os metadados estruturais descrevem a estrutura interna do conjunto de dados. Além de uma visão geral dos conjuntos de dados, facilitam o uso por aplicações de consumo. Conforme descrito na Tabela 2, o PCD proposto terá os seguintes metadados estruturais:

Tabela 2: Metadados Estruturais

Metadados Descritivos	Tipo de dado recebido	Descrição
Qtd_Propriedades	Numeral	Quantidade de propriedades identificadas no conjunto de dados
Propriedade	Array	Para cada propriedade é incluído um array com o Nome e o Tipo
Propriedade - Nome	Literal	Nome da propriedade no conjunto de dados
Propriedade - Tipo	Literal	Tipo de dado que a propriedade recebe no conjunto de dados

3.3 Metadados de Qualidade

A qualidade da informação é um dos aspectos mais importantes para os consumidores e usuários da internet [Naumann et al. 2000]. Entre os critérios de qualidade que podem ser utilizados sobre os diferentes aspectos de um conjunto de dados, pretende-se utilizar a completude, disponibilidade, atualidade e compreensibilidade, por estarem relacionados ao consumo de conjunto de dados e ser possível enriquecê-los a partir do *feedback* do usuário.

Em Silva Neto (2016) a completude de dados é definida como o grau em que uma fonte de dados não possui dados nulos ou faltantes, já a disponibilidade é definida como o grau em que as fontes de dados estão disponíveis para o consumo de seus dados. Naumann et al. (2000) definiram a compreensibilidade como o grau em que a informação está em conformidade com a capacidade técnica do consumidor; e a atualidade está relacionada à idade da informação. Para que esses metadados correspondam à realidade dos conjuntos de dados, serão enriquecidos com o *feedback* de usuários. Essa integração entre publicadores e consumidores de dados torna possível a melhora da qualidade dos dados publicados.

4. Avaliação dos Resultados e Estado Atual do Trabalho

Foi implementada a primeira versão para a geração de forma automática do PCD, que abrange alguns dos metadados descritivos e estruturais. O perfil é gerado em RDF[2], por esse ser o modelo recomendado pelo W3C para a geração de metadados. A Figura 3 mostra um exemplo de perfil gerado, referente a um conjunto de dados sobre música. Neste exemplo, são exibidos os prefixos dos vocabulários utilizados, juntamente com os metadados descritivos e estruturais. Para a identificação do título, caso o conjunto de dados não possua alguma propriedade referente, será retornado seu nome de arquivo. No exemplo, o conjunto de dados possui a propriedade title:Wanderlust. As palavras-chave são identificadas utilizando a função TF-IDF sobre o conjunto de dados indexados. Conforme apresentado em alguns trabalhos, como em Ouksili (2014) e Abele (2016), o TF-IDF pode ser utilizado para identificar os termos mais importantes de um documento. Em seguida, é possível observar o domínio (tema) e vocabulário de domínio que receberam os valores encontrados durante a execução da aplicação. Também é incluída uma descrição do esquema do conjunto de dados, com os nomes e tipos das propriedades identificadas.

Alguns experimentos foram realizados com o intuito de avaliar a abordagem proposta [Targino et al. 2017]. Dois objetivos foram identificados: avaliar a recomendação de vocabulários de domínio do conjunto de dados; e avaliar o domínio identificado de acordo com o conjunto de dados. Com esses objetivos, foram definidos *gold standards* para

[2] https://www.w3.org/RDF/

os conjuntos de dados utilizados no experimento. Foram calculadas as métricas de precisão, cobertura e F-measure para a identificação do domínio e para recomendação de vocabulários de domínio dos conjuntos de dados selecionados. Com os resultados obtidos foi observado que os conjuntos de dados pertencentes a áreas mais específicas, por suas palavras serem bastante especializadas, é mais provável alcançar os resultados esperados.

```
@prefix skos:   <http://www.w3.org/2004/02/skos/core# >.
@prefix void:   <http://rdfs.org/ns/void# >.
@prefix dct:    <http://purl.org/dc/terms/> .
@prefix dcat:   <http://www.w3.org/ns/dcat#:> .
@prefix rdfs: <http://www.w3.org/2000/01/rdf-schema#>.
@prefix schema: <http://schema.org/>.

[] a      dcat:Dataset ;
   dct:title "Wanderlust" ;
   dcat:keyword "singer" , "artist" , "name" ;
   dcat:theme
         [ rdfs:label "Musical Work";
           void:uriSpace "http://dbpedia.org/ontology/MusicalWork"
         ] ;
   void:vocabulary <http://purl.org/ontology/mo/> ;
   skos:inScheme
         [ void:properties "3" ;
           void:property
                 [ schema:name "id" ;
                   dct:type "string"
                 ] ;
           void:property
                 [ schema:name "title" ;
                   dct:type "string"
                 ];
           void:property
                 [ schema:name "name" ;
                   dct:type "string"
                 ]
         ] .
```

Figura 3. Perfil exemplo gerado no formato RDF/Turtle

5. Trabalhos Relacionados

Com relação à recomendação de vocabulários, um dos trabalhos é apresentado por Ellefi et al. (2015) onde é proposto um sistema de recomendação de vocabulários baseado no repositório LOV[3]. Neste trabalho, propõe-se a recomendação para cada propriedade existente no conjunto de dados em questão, contanto que o vocabulário esteja ativo. Isso é original comparando-se com os trabalhos que encontramos.

Com relação à identificação do domínio, o trabalho de Ouksili et al. (2014) apresenta uma abordagem que possibilita a identificação de temas de um determinado conjunto de dados. Para isso, é necessário que o conjunto de dados esteja no formato RDF, pois é utilizada uma combinação de critérios estruturais e semânticos para o agrupamento dos grafos, onde cada agrupamento corresponde a um tema. Neste trabalho, são extraídas as palavras-chave de um conjunto de dados para identificação do domínio, que é realizado por meio de um mecanismo com informações sobre uma grande quantidade de domínios de conhecimento de forma bem estruturada. No trabalho de Silva Neto (2016) é proposto um Perfil de Qualidade, composto de um conjunto de metadados sobre a qualidade de uma fonte de dados dinâmica, que é atualizado de acordo com os resultados obtidos em avaliação contínua. Em comparação com este trabalho, o PCD proposto não possui apenas metadados de qualidade e, além de abordar diferentes aspectos, o *feedback* do usuário será utilizado como forma direta e automática de enriquecimento dos metadados de qualidade.

[3] http://lov.okfn.org/dataset/lov/

Em termos de geração de perfis, como exemplo, o trabalho apresentado por Abele (2016) propõe uma abordagem para a descrição detalhada de conjuntos de dados utilizando metadados para prover informações gerais sobre os mesmos como, por exemplo, descrição, data de atualização e informações de licença. Diferentemente do trabalho citado, propomos a geração de um perfil de conjunto de dados que pode estar em qualquer formato de dados, não apenas no formato RDF e, em nosso perfil, além de metadados descritivos, abordamos metadados estruturais e de qualidade com enriquecimento semântico.

6. Considerações e Desenvolvimento Necessário para Conclusão

Este trabalho apresentou uma proposta de abordagem para a geração de um perfil de conjuntos de dados publicados na Web com enriquecimento semântico, a qual foi parcialmente especificada, e uma primeira versão foi implementada. Atualmente, estão sendo especificadas as definições dos metadados de qualidade e as estratégias necessárias para sua medição.

Referências

Abele, A. (2016) "Linked Data Profiling: Identifying the Domain of Datasets Based on Data Content and Metadata", In: 25th International Conference Companion on World Wide Web. Canada, p. 287-291.

Assaf, A., Troncy, R. and Senart, A. (2015) "Roomba: An extensible framework to validate and build dataset profiles", In: 24th International Conference on World Wide Web. Italy, p. 159-162.

Ellefi, M. B., Bellahsene, Z., Scharffe, F. and Todorov, K. (2014) "Towards semantic dataset profiling" In: International Workshop on Dataset Profiling & Federated Search for Linked Data co-located with the 11th Extended Semantic Web Conference. Greece.

Ellefi, M. B., Bellahsene, Z. and Todorov, K. (2015) "Datavore: a vocabulary recommender tool assisting Linked Data modeling", In: 14th International Semantic Web Conference.

Lóscio, B. F., Burle, C., Calegari, N. (2017) "Data on the web best practices. The World Wide Web Consortium", https://www.w3.org/TR/dwbp/ Acesso: 20 de maio de 2017.

Maali, F., Erickson, J., and Archer, P. (2014). "Data catalog vocabulary. The World Wide Web Consortium", https://www.w3.org/TR/vocab-dcat/ Acesso: 20 de maio de 2017.

Naumann, F.; Rolker, C. (2000) "Assessment methods for information quality criteria" In: 5th Conference on International Quality (IQ). United States, p. 148-162

Ouksili, H., Kedad, Z., Lopes, S. (2014) "Theme Identification in RDF Graphs", In: International Conference on Model and Data Engineering (MEDI). Cyprus, p. 321-329.

Silva Neto, E. C. (2016) "Um Perfil de Qualidade para Fontes de Dados Dinâmicas" - Dissertação de Mestrado Universidade Federal de Pernambuco. Brasil.

Targino, N., Souza, D., Salgado, A. C. (2017) "Uma Proposta de Perfil de Conjuntos de Dados na Web com Enriquecimento Semântico" To appear in: Anais do 32th Simpósio Brasileiro de Banco de Dados (SBBD).

32th Brazilian Symposium on Databases

ctd

October 2nd to 5th, 2017
Uberlândia – MG – Brazil

THESIS AND DISSERTATIONS CONTEST

PROCEEDINGS

Promotion

Brazilian Computer Society – SBC
SBC Special Interest Group on Databases

Organization

Universidade Federal de Uberlândia – UFU

CTD Program Chair

Vânia Maria Ponte Vidal, UFC

Editorial

The SBBD Thesis and Dissertation Contest (CTD) aims to disseminate and award the best doctoral theses and master's dissertations in the database area, approved by the candidate's university between January 1^{st}, 2015 and December 31^{st}, 2016.

The competition process was divided into two phases. In the first phase, the committee selected the two best doctoral theses and the six best master's dissertations, according to their scientific and technological contributions, as well as to their potential impact on the society and on the state of the art in the Database area. Every submitted manuscript received at least three reviews from selected members of CTD's evaluation committee. Great responsibility fell to the preliminary evaluation committee, as they had to choose the theses and dissertations that would be given an opportunity to compete in the final round of the competition. The committee consisted of the academic staff representatives, who read and evaluated the theses and dissertations, and gave their recommendation. A deep bow and praise to all the members of the preliminary evaluation committee for the work done!

During the second phase, held during SBBD 2017 in Uberlândia-MG, the students presented their work and answered questions from the committee members and audience. From the works selected in the first phase, the committee chose the best doctoral thesis and the three best master's dissertations. The high quality and diversity of the submitted works made the selection process both highly challenging and rewarding. They are a portrait of the high-quality research, in the database area, developed in our graduate programs. I would like to thank the committee members for their endeavor in providing high quality reviews for the submitted papers. I am also very grateful to the students and to their advisors for the willingness of submitting their work to CTD. Finally, we thank the local organization committee and the symposium chairs who worked hard to guarantee an outstanding symposium. I wish an excellent event to the whole database community of both CTD and SBBD and wish to see you all in Uberlândia, MG.

Vânia Maria Ponte Vidal, UFC
CTD Program Chair

32nd Brazilian Symposium on Databases

October 2nd to 5th, 2017
Uberlândia – MG – Brazil

Promotion

Brazilian Computer Society – SBC
SBC Special Interest Group on Databases

Organization

Universidade Federal de Uberlândia – UFU

SBBD Steering Commitee

Agma Juci Machado Traina, USP
Bernadette Lóscio, UFPE
Caetano Traina Jr., USP
Carmem Hara, UFPR
Javam Machado, UFC
Mirella M. Moro, UFMG
Vanessa Braganholo, UFF

SBBD 2017 Commitee

Steering Committee Chair
Javam Machado, UFC

Local Organization Chairs
Maria Camila N. Barioni, UFU and Humberto L. Razente, UFU

Program Committee Chair
Carmem S. Hara, UFPR

Short papers Chairs
Bernadette Lóscio, UFPE and Damires Souza, IFPB

Demos and Applications Session Chair
Daniel de Oliveira, UFF

Short Courses Chair
Vaninha Vieira, UFBA

Workshop on Thesis and Dissertations in Databases Chair
Carina Dorneles, UFSC

Tutorials Chair
Ana Carolina Salgado, UFPE

Thesis and Dissertation Contest Chair
Vânia Vidal, UFC

Workshops Chair
Fernanda Baião (UNIRIO)

Local Organization Committee

Maria Camila N. Barioni, UFU
Humberto L. Razente, UFU
José Gustavo de Souza Paiva, UFU
Marcelo Zanchetta do Nascimento, UFU
Elaine Ribeiro de Faria Paiva, UFU
João Henrique de Souza Pereira, UFU

Thesis and Dissertations Contest Program Committee

Agma Traina (ICMC-USP)
Altigran Soares da Silva (UFAM)
Ana Carolina Salgado (UFPE)
Angelo Brayner (UFC)
Caetano Traina Júnior (ICMC-USP)
Cristina Ciferri (USP)
Fabio Andre Porto (LNCC)
Fernanda Baião (UNIRIO)
Javam Machado (UFC)
José Maria Monteiro (UFC)
José Antônio Macêdo (UFC)
José Palazzo Moreira de Oliveira (UFRGS)
Luiz André Paes Leme (UFF)
Marco Antônio Casanova (PUC-RIO)
Mirella M. Moro (UFMG)
Renata Galante (UFRGS)
Renato Fileto (UFSC)
Sergio Lifschitz (PUC-RIO)
Valéria C. Times (UFPE)

Table of Contents (CTD)

A Multi-View Approach for Assessing the Quality of Collaboratively Created Content on the Web 2.0

Daniel Hasan Dalip[1], Advisor: Marcos A. Gonçalves[1], Co-advisor: Marco Cristo[2]

[1] Dpto. de Computação – Universidade Federal de Minas Gerais
hasan@decom.cefetmg.br mgoncalv@dcc.ufmg.br
[2] Instituto de Computação – Universidade Federal do Amazonas
marco.cristo@icomp.ufam.edu.br

Categories and Subject Descriptors: H.3 [**Information Storage and Retrieval**]: Miscellaneous; I.7 [**Document and Text Processing**]: Miscellaneous

Keywords: Quality Assessment, Wiki, Q&A Forums, Machine Learning, Information Quality

1. INTRODUCTION

A key characteristic that explains the success of the Web 2.0 is the advent of user-generated content (UGC). UGC deeply changed the way content is created and consumed. Based on it, many new media technologies were introduced to provide facilities for regular Web users to publish their own content. As a consequence, new kinds of knowledge repositories, to which anyone can freely contribute, have emerged. Examples of these repositories include web-based encyclopedias. Wikipedia[1], for instance, contains currently more than thirty million articles, written in hundreds of different languages[2]. We have also the Wikia[3], which has grown from one hundred to several thousands of collections of wikis in just a few years. Another example is question and answer forums, where users can collaborate asking and answering questions regarding topics such as programming, math, and English language, among others. A popular forum, Stack Overflow[4], contains more than 3 millions questions and almost 7 millions answers about programming.

Problem statement: If, on one hand, free edition enables the rapid expansion of the knowledge in these sites, on the other hand, it raises a tough question: *how can we determine the quality of information resulting from this process?* It is clear that, given the growth and dissemination of collaboratively created content, mechanisms to assess the quality and trust of this type of material must be provided. The traditional approach to deal with this problem consists of adopting quality control mechanisms that rely on users to determine content quality and appropriateness, as well as editor reputation. Unfortunately, this kind of manual strategy not only does not scale, but it is also subject to human bias, which can be influenced by varying backgrounds, expertise, and even a tendency for abuse. As a consequence, several strategies to automatically estimate the quality of collaborative UGC have been proposed in the last few years.

Such estimates could be used, in the case of a collaborative encyclopedia, as an indicator of which

[1] http://www.wikipedia.org
[2] https://en.wikipedia.org/wiki/Wikipedia
[3] http://www.wikia.org
[4] http://stackoverflow.com

This work is supported by the INWeb (MCT/CNPq grant 57.3871/2008-6), MasWeb, Google, CNPq, CAPES and FAPEMIG.

documents need revision, to detect vandalism or inadequate revision methods, or even to recommend articles based on their estimated quality and reliability level. Similarly, in Q&A forums, these predictions could be helpful in detecting which question does not have a good answer yet, sort the answers to a question, to identify uncorrelated answers and spam, and to recommend answers to a question posted in others services such as search engines.

To better understand such strategies, it is worth to properly define *information quality* (IQ). According to [Wang and Strong 1996], IQ is the information that is fit for use by consumers. As observed in [Ge and Helfert 2007], since information consumers are not very capable of finding errors in information and altering the way they use it, an alternative definition of IQ takes a data perspective, i.e., IQ is the information that meets the specifications or requirements. In this sense, quality assessment algorithms attempt to estimate quality by means of the combination of statistical indicators that try to measure how well the information meets different requirements. For instance, it is expected that a good article has a length large enough to properly discuss a topic and it is cited as it provides reputable information. Thus, it is possible to learn how to combine these indicators to predict a quality level. Also, from a theoretical point of view, as highlighted by [Wang and Strong 1996; Tejay et al. 2006; Ge and Helfert 2007], quality synthesizes the measurement of *various dimensions*.

Our method: This view of quality as a multi-dimensional concept suggests that it should be thought of as a combination of independent assessments where, as before, each assessment can be estimated from several statistical indicators. For example, the quality of a textual document can be viewed as a composition of dimensions such as clarity, factual accuracy and importance. In other words, quality is a multifaceted concept, in which each facet corresponds to an aspect that can be individually analyzed by an automated "expert." The "opinions" of these experts can then be combined for a final decision. We refer to this method as a *multi-view approach*. In such an approach, each view (the expert opinion) corresponds to a partition of the set of indicators where the indicators of a particular view are naturally seen as a group. Further, defined as partitions, the views cannot share indicators which implies that views are designed to be as much independent as possible from each other. In Machine Learning, this approach is somewhat similar to an ensemble technique of learning using multiple experts in two levels [Wolpert 1992].

However, multi-view learning is more complex than simple classifier ensembles, since, in this technique, we need to know how to partition features into views in the "best possible way" for the domain at hand. In other words, there is a non-trivial *feature engineering* process that needs to be carried out before any classifier combination can be performed. In this context, we here propose a general approach for combining statistical indicators into views to assess content quality in collections created collaboratively. Our approach gives guidelines regarding how to optimize such partitioning in a natural way for the studied domains. Furthermore, we indicate how to adapt the approach for similar problems in other domains and other applications related to quality of collaborative content. We evaluated our method using indicators extracted from two domains, namely, collaborative encyclopedias (hereafter called Wikis, for short) and Q&A forums. These indicators were grouped into specific sets of views that, when combined, led to improvements in quality evaluation. In the following, we discuss our contributions as well as the main conclusions and future work.

Our contributions are fourfold: (1) a feature representation and its impact for each tested domain; (2) a general multi-view approach and an in-depth study of the views; (3) the exploitation of a feature selection approach in order not only to reduce the number of features without losing performance but also to do a feature analysis; (4) an application that helps the user to infer the quality in Wikipedia, namely GreenWiki. Following we describe each of these contributions.

Our publications include three journals [Dalip et al. 2011; Dalip et al. 2012; Dalip et al. 2016], among them an ACM Journal on Data and Information Quality and a Journal of the Association for Information Science and Technology (JASIST), one of the main journals on this research area (impact factor 2.23). We also published in four conferences among the most important of the field such as

ACM SIGIR, ACM/IEEE JCDL, and TPDL [Dalip et al. 2011; Dalip et al. 2012; 2013; Dalip et al. 2014]. Particularly, the work presented in JCDL 2014 won the Best Student Paper Award in that edition. Furthermore, this thesis has influenced other studies such as search query expansion [Brandão et al. 2014], to infer detractors and evangelists on Twitter [Bigonha et al. 2010], polarity detection on foursquare tips [Moraes et al. 2013] and sentiment analysis [Gonçalves et al. 2016]. Our work has also been inspiration for some Master dissertations: (i) one, already defended, analyzes the impact of using GreenWiki to help the user to evaluate the quality of the content available on Wikipedia [Lara 2011]; (ii) another, which will be defended soon, tries to automatize the procedure of generating the views.

The potential impact of this thesis on society is significant from an educational point of view (e.g. evaluation of learning repositories, distance education systems) as well as for the construction of better information services such as recommendation systems and search engines which take into account the quality of the content. Such potential was recognized by Google Inc. which honored this research with the Google Focused Research Award, after strong competition with hundreds of projects throughout Brazil (only six projects were funded nationwide). Moreover, the work also received media coverage[5]. Furthermore, this thesis was top 3 in the CTDIAC 2016 (Concurso de Teses e Dissertações em Inteligência Artificial e Computacional).

2. CONTRIBUTIONS

2.1 Feature proposal and its impact

We first studied the impact of features and quality indicators in Wikis and Q&A Forums domains. To accomplish this, we performed a thorough analysis of the capability of an automatic method to estimate content quality in Wikis. First, we extended our previous work [Dalip et al. 2009] which assessed the quality in Wikipedia, to assess the quality of two others Wikis, namely *Wookieepedia*[6], about the Star Wars universe, and *Muppet*[7], regarding the TV series "The Muppet Show". Our consistent results throughout a large body of experiments and analyses allow us to make more generalizable conclusions than any previous work.

From this analysis, we observed that the most useful feature group was the one associated with the text Structure (e.g. citation, images and section count). Interestingly, these features are also those easiest to extract from freely available collaborative encyclopedias. We also noted that best results are achieved when Structure features are combined with Network and Revision features. This work were published in ACM Journal of Information and Data Quality [Dalip et al. 2011].

In Q&A Forums we studied which, out of the 68 features previously used [Dalip et al. 2009; 2011], could be useful in Q&A Forums together with others features previously proposed in literature for this domain [Agichtein et al. 2008; Shah and Pomerantz 2010], and 89 new proposed features, totaling 186 features. Using our proposed approach, we were able to outperform a state of the art baseline with gains of up to 12% in NDCG, a metric used to evaluate rankings. We also conducted a comprehensive study of the features showing that, user and review features are the most important in the Q&A Forums domain. This work was published in the main Information Retrieval conference of the world, ACM Conference on Research and Development in Information Retrieval (SIGIR 2013, Qualis A1) [Dalip et al. 2013].

In the thesis, we detail the features (from Wiki and Q&A Forum) in Chapter 3. Some results of those work (adapted for multi-view) are presented in Chapter 5, for Wikis, and in Chapter 6 for Q&A Forums.

[5] `http://blogs.estadao.com.br/link/google-financia-pesquisas-no-brasil/`
[6] `http://starwars.wikia.com/`
[7] `http://muppet.wikia.com/`

2.2 General multi-view approach and view analysis

In order to study better ways to combine the used features, we proposed an approach to assess the quality of collaboratively created content by organizing quality indicators into semantically related views and combining these views by means of meta-learning. With that, we did an in-depth analysis of this approach and of the impact of the views on quality assessment of collaborative content in Wikis and Q&A Forums. Our experimental results show that the proposed meta-learning approach is able to improve quality assessment over a state-of-the-art approach in five out of the six tested collections, with gains of up to 30.9% (cf. thesis Table 5.1). Through correlation and performance analysis, we observed that our approach was either capable of selecting the best single view or to combine them when they contained complementary information.

In addition, we were able to reach more generalized conclusions and a better understanding from a qualitative point of view regarding why some features performed well and others not. This was essential to better comprehend certain theoretical aspects of multi-view learning (e.g., when and why it is supposed to work) applied to quality estimation. Furthermore, we propose a general multi-view approach that takes advantage of groups (i.e., views) of quality indicators – this new approach generalizes and allows to better comprehend several previous solutions including some proposed by ourselves.

Preliminary results were published in the *Journal of Information and Data Management* (Qualis B3) [Dalip et al. 2012] and a more detailed explanation of the approach together with an in-depth analysis of it were presented at the *2012 International Conference on Theory and Practice of Digital Libraries (TPDL)* (Qualis B1) [Dalip et al. 2012]. The general framework together with a more generalized conclusions of it were published in the *Journal of The American Society for Information Science and Technology (JASIST)* (Qualis A1) [Dalip et al. 2016].

In the thesis, we detail the approach in the Chapter 3, results of these work are presented in the Chapter 5, for Wikis, and Chapter 6 for Q&A Forums.

2.3 Feature selection and analysis

We also studied the impact of feature selection on our multi-view approach for assessing quality in all the studied collections. We were motivated not only by the possibility of decreasing the complexity of the learned models but also by the opportunity of analyzing the importance of views and features. To accomplish this, we modeled the problem as a multi-objective search (using genetic algorithms) for the *smallest* set of features that is able to simultaneously *reduce* the quality assessment error. Results show that we can reduce the feature set to a fraction of 15% through 25% of the original set, while obtaining error rates comparable to the state of the art. We also investigated the impact and redundancy of different features and views for the Wikis domain. This part of the work was published at the ACM/IEEE Joint Conference on Digital Libraries 2014 (JCDL) (Qualis A2), the main conference in the field, where it received the **Best Student Paper Award** [Dalip et al. 2014]. Chapter 2 presents the feature selection approach and results are presented in Chapter 5 and 6.

2.4 Implemented Tool

We were able to propose a tool, called GreenWiki[8], using some of the proposed metrics. This is a Wiki with some articles collected from Wikipedia and a panel of quality indicators about the article being read. Note that GreenWiki does not intend to evaluate the quality of an article, but rather, its goal is to present indicators that will help users get to their own conclusions about its quality. This work was

[8] `http://www.hasan.com.br/greenwiki`

published as a Demo at ACM/IEEE Joint Conference on Digital Libraries 2011 (Qualis A2) [Dalip et al. 2011].

3. CONCLUSIONS

In this thesis, we have introduced a general quality evaluation approach for user-generated content based on the idea that quality is a multi-faceted concept. Accordingly, the quality assessment of an collaborative document must consider the different quality views applicable to that item. In our approach, we propose the use of views that are related to several quality dimensions and sources. These views are: (1) general enough to be applicable to many kinds of items as long as they have a predetermined and rational quality evaluation criteria, (2) capable to highlight quality dimensions where quality indicators are lacking, (3) lead to a natural combination strategy to obtain a summarized quality estimate, and (4) useful to organize and assess the importance of quality views and indicators.

We show the generality of our proposed approach by applying it to six datasets belonging to two different domains: Wikis and Q&A Forums. We were able to extract quality indicators related to all views and dimensions in both domains. We were also able to improve different performance criteria specific for each domain, such as numeric assessments in Wikis and ranking order in Q&A Forums. With that, we were able to see which indicators we need to take into account more in order to assess the quality of content.

In our work, we observed that usually Edit History features are good predictors of the content quality. At the same time, we observed poor effectiveness of Readability features, which indicates they may have to be adapted to each specific collection. Moreover, for scenarios where quality has to be assessed from very short and informal messages, e.g. posts in microblogs, such as Twitter[9], these features may have to be completely rethought. For instance, instead of a static equation, we could use machine learning to combine different aspects of the text to infer the readability of a post. Using inverse reasoning, lack of quality may also indicate other types of problems such as spam or the presence of attacks such as vandalism. Thus, methods for detecting these problems could benefit from our proposal.

Finally, as future work we also intend to apply our approach to other domains. In particular, we are interested in the product review domain, where the users want to read the best quality reviews. In such scenarios, some dimensions are clearly relevant such as usefulness, reliability and relevance. We also intend to suggest means to improve quality assessments in applications such as web search. In addition, using our method, we can propose tools to assist users with the process of quality assessment and visualization. Other future work includes the analysis and assessment of the evolution of an article's quality over time.

REFERENCES

Agichtein, E., Castillo, C., Donato, D., Gionis, A., and Mishne, G. Finding High-Quality Content in Social Media. In *Proc. of the WSDM'08*. pp. 183–194, 2008.

Bigonha, C., Cardoso, T. N., Moro, M. M., Almeida, V., and GonÃğalves, M. A. Detecting Evangelists and Detractors on Twitter. In *Webmedia 2010*. Belo Horizonte, MG, Brazil, 2010.

Brandão, W. C., Santos, R. L., Ziviani, N., Moura, E. S., and Silva, A. S. Learning to Expand Queries using Entities. *JASIST* 65 (9): 1870–1883, 2014.

Dalip, D. H., Cardoso, T., Gonçalves, M., Cristo, M. U., and Calado, P. A Multi-view Approach for the Quality Assessment of Wiki Articles. *JIDM* 3 (1), 2012.

Dalip, D. H., Gonçalves, M. A., Cristo, M., and Calado, P. A general multiview framework for assessing the quality of collaboratively created content on web 2.0. *Journal of the Association for Information Science and Technology*, 2016.

[9]For example, users trying to influence the opinions of others may have a tendency to write better written messages [Bigonha et al. 2010].

Dalip, D. H., Gonçalves, M. A., Cristo, M., and Calado, P. Automatic Quality Assessment of Content Created Collaboratively by Web Communities: a Case Study of Wikipedia. In *Proceedings of the 2009 JCDL*. Austin, TX, USA, pp. 295–304, 2009.

Dalip, D. H., Gonçalves, M. A., Cristo, M., and Calado, P. Automatic Assessment of Document Quality in Web Collaborative Digital Libraries. *Journal of Data and Information Quality* 2 (3): 1–30, 2011.

Dalip, D. H., Gonçalves, M. A., Cristo, M., and Calado, P. On MultiView-Based Meta-Learning for Automatic Quality Assessment of Wiki Articles . In *Proceedings of the 2012 International Conference on Theory and Practice of Digital Libraries*. Paphos, Cyprus, 2012.

Dalip, D. H., Gonçalves, M. A., Cristo, M., and Calado, P. Exploiting User Feedback to Learn to Rank Answers in Q&A Forums: A Case Study with Stack Overflow. In *Proceedings of the 36th International SIGIR*. Dublin, Ireland, pp. 543–552, 2013.

Dalip, D. H., Lima, H., Gonçalves, M. A., Cristo, M., and Calado, P. Quality Assessment of Collaborative Content with Minimal Information. In *Proceedings of the 2014 JCDL*, 2014.

Dalip, D. H., Santos, R. L., Oliveira, D. R. R., Amaral, V. F., Gonçalves, M. A., Prates, R. O., Minardi, R. C. M., and Almeida, J. M. D. GreenWiki - A Tool to Support Users âĂŹ Assessment of the Quality of Wikipedia Articles. In *Proc. of the 2011 JCDL*. pp. 469–470, 2011.

Ge, M. and Helfert, M. A Review of Information Quality Research - Develop a Research Agenda. In *International Conference on Information Quality* (2010-07-12). pp. 76–91, 2007.

Gonçalves, P., Dalip, D. H., Costa, H., Gonçalves, M. A., and Benevenuto, F. On the combination of off-the-shelf sentiment analysis methods. In *Proc. of the 31st ACM SAC*, 2016.

Lara, R. *Qualidade de Artigos na Wikipedia para seus Usuários - Análise e Proposta de Interação* . M.S. thesis, Universidade Federal de Minas Gerais, Brazil, 2011.

Moraes, F., Vasconcelos, M., Prado, P., Dalip, D., Almeida, J., and GonÃğalves, M. Polarity Detection of Foursquare Tips. In *Social Informatics*. LNCS, vol. 8238. Springer, 2013.

Shah, C. and Pomerantz, J. Evaluating and Predicting Answer Quality in Community Q&A. In *Proceedings of the 19th SIGIR*. Geneva, Switzerland, 2010.

Tejay, G., Dhillon, G., and Chin, A. G. Data Quality Dimensions for Information Systems Security: A Theoretical Exposition. In *Security Management, Integrity, and Internal Control in Information Systems*. Springer, pp. 21–39, 2006.

Wang, R. Y. and Strong, D. M. Beyond accuracy: What data quality means to data consumers. *Journal of Management Information Systems* 12 (4): 5–34, 1996.

Wolpert, D. Stacked Generalization. *Neural Networks* 5 (2): 241–259, 1992.

Data Classification in Complex Networks via Pattern Conformation, Data Importance and Structural Optimization

Murillo G. Carneiro[1,2], **Liang Zhao**[1,3]

[1]Instituto de Ciências Matemáticas e de Computação, Universidade de São Paulo (USP)
13566-590 – São Carlos – SP – Brazil

[2]Faculdade de Computação, Universidade Federal de Uberlândia (UFU)
38400-902 – Uberlândia – MG – Brazil

[3]Departamento de Computação e Matemática, Universidade de São Paulo (USP)
14040-901 – Ribeirão Preto – SP – Brazil

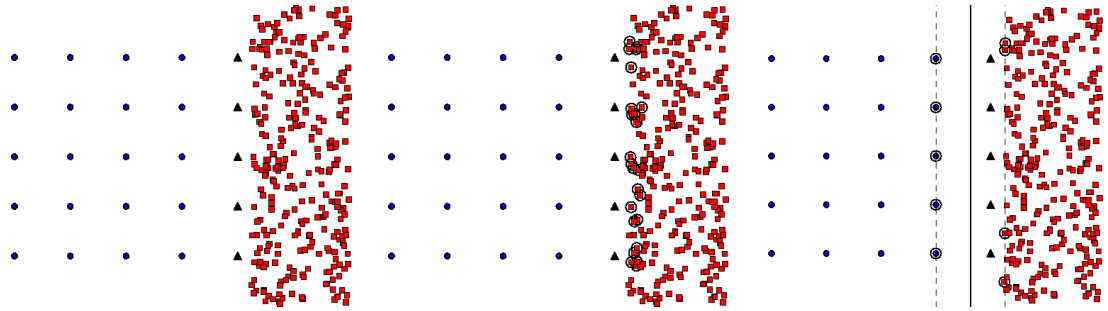
mgcarneiro@ufu.br, zhao@usp.br

Abstract. *This thesis focuses on the development of classification methods based on complex networks. Differently from most data classification techniques which rely only on the physical features of the data (e.g., distance or distribution), the proposed methods consider not only physical features but also structural and dynamical properties of the data from the network representation. This salient feature enables the detection of intrinsic and semantic relations among data items, as has been demonstrated through extensive experiments against representative state-of-the-art methods over a wide range of artificial and real data sets, including applications in domains such as heart disease diagnosis, semantic role labeling and image classification.*

1. Overview

This article describes the main contributions of the doctoral research presented in [Carneiro 2016]. The thesis has complex networks and data classification as major topics. *Complex networks* are known by providing a set of efficient and robust tools to model and analyze networked data. In the context of machine learning and data mining, network-based techniques are widely employed to unsupervised and semi-supervised tasks, e.g., community detection (or data clustering), label propagation and dimension reduction [Chapelle et al. 2006, Fortunato 2010]. Although *data classification* is a largely investigated task, the development of supervised learning methods based on complex networks is also a barely explored topic as there is little space for any label propagation due to the few unlabeled data items. Besides covering this lack, the following issues are also addressed by the investigations presented in the thesis:

Problem: Given its practical importance, the literature contains many data classification techniques. Typically, these techniques define decision boundaries in the data space according to the physical features of a training set and a new data item is classified by verifying its relative position to the boundaries. Such kind of classification, which is only based on the physical attributes of the data (e.g., similarity, distance or distribution), has difficulty detecting intrinsic and semantic relations among the data items such as the pattern formation, for instance. Let us consider Fig. 1(a), which shows a simple data set with two classes denoted by the circle and the square data items. The triangle data items



(a) Two-class data set (b) k-NN (**Nearest neighbors**) (c) SVM (**Support vectors**)

Figure 1. Analysis of the classification process of traditional techniques in a simple two class data set where circle data items denote a clear pattern and triangle data items need to be classified. Such techniques fail to consider the semantic structure of the data and they consequently label triangle data items as belonging to square/red class.

represents test instances that need to be classified. Figs. 1(b) and 1(c) shows the classification process behind the traditional techniques k-nearest neighbors (k-NN) and Support Vector Machine (SVM). For example, k-NN classifies a test instance by verifying the label of its k nearest neighbors (circulated in Fig. 1(b)), and SVM takes into account the support vectors (circulated in Fig. 1(c)) to approximate each class with a convex hull and to find the best separating hyperplane between the classes. One can see both techniques fail to identify semantic patterns formed by the data. By contrast, the usage of complex networks is a promising way to capture spatial, topological and functional relationships of the data, as the network representation unifies structure, dynamic and functions of the networked system [Newman 2010].

Pattern conformation: In supervised data classification, the first attempt to use complex networks in order to consider the semantic relations among the data items is the hybrid framework for high-level classification proposed in [Silva and Zhao 2012]. Such framework combines the associations produced by traditional and network-based techniques. The network-based technique uses complex network measures to estimate the membership of a test item according to the data pattern formation. However, given the high number of parameters in such framework (e.g., network formation, network measures variation, parameters of the traditional technique, convex combination, and so on), a simplified framework for high-level classification which employs a unique network to provide both physical and complex-network based associations is proposed in the thesis.

Data importance: Although pattern conformation has been employed by high-level classification as a new classification concept, other concepts can also be derived from complex networks. For example, structural and dynamical properties of the networked data can provide additional layers of information by defining quantitatively the importance of each data item. Despite it has been a common practice in data classification to assume that all data samples have the same relevance, such an assumption is obviously not compliant to the natural classification performed by the human brain. Moreover, neglecting the individual importance of each data sample may change the understanding of the whole data set. Such point is addressed in [Carneiro and Zhao 2017] with the development of a new method based on the importance concept of complex networks.

Network optimization: A common characteristic among network-based methods is the construction of the underlying data graph. In machine learning, such graph is usually formed from the input vector data, where each data item is represented as a node and the edges are defined from the affinity (or similarity) among the data items, for example, by connecting each data item to its k nearest neighbors. Although the network formation is a crucial step for good performance, little attention has been devoted to this topic [Newman 2010]. Such a situation is confirmed by the common usage of the simple kNN network construction method in literature. kNN method considers only local data relationships and it is a general-purposed one, i.e., the constructed networks can be used for any machine learning or data mining tasks. By contrast, sophisticated methods consider both local and global relationships of the input data, but they are restricted for specific purposes. In order to fill this gap, an optimization framework, which is responsible to construct an "optimal" network regarding a given processing goal, is presented in [Carneiro et al. 2016a, Carneiro et al. 2017a].

In summary, this doctoral research investigated whether the structural and dynamical features derived from network representation can provide efficient computational methods to sort out the issues discussed above. The main contributions derived from this thesis are briefly discussed in the next section.

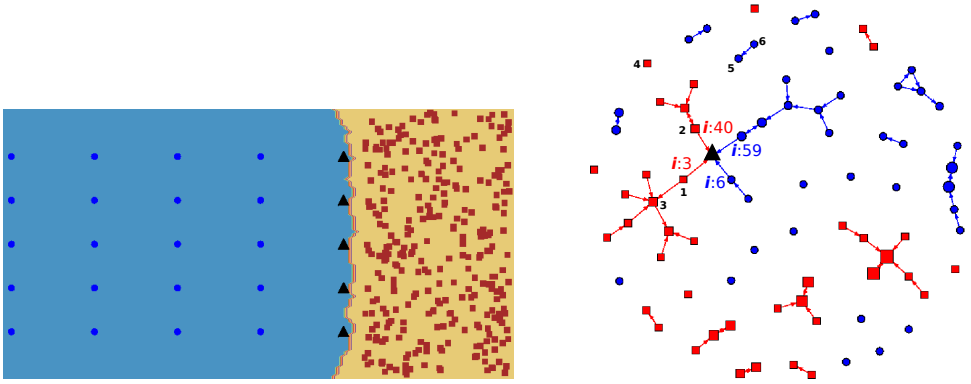
2. Thesis Contributions

Most of the doctorate research has been focused on data classification via importance concept in complex networks and network structural optimization. Following both contributions are briefly discussed.

Classification Based on Data Importance: This investigation proposed a new data classification concept based on the importance concept of complex networks. Instead of data space division as having been done in traditional techniques or pattern conformation as having been done in high-level classification techniques, the classification based on the data importance considers the individual importance of each data item in order to classify an unlabeled item into that class where it has the highest importance. In the developed technique, the concept of importance is derived from PageRank, the ranking measure operating behind the universal search engine of Google. In addition, the technique captures spatial and structural properties of the networked data from a new network measure created, named spatio-structural differential efficiency. The good performance of the proposed method is attested by comparisons against state-of-the-art methods over a wide range of artificial and real-world data sets, including applications in domains such as heart disease diagnosis [Carneiro and Zhao 2017].

Taking into account the artificial data set presented by Fig. 1(a), in which k-NN and SVM are unable to label the test data items correctly, Fig. 2(a) demonstrates that the importance-based method accurately detect the pattern formation of the data. A real-world example is also presented in Fig. 2(b), which illustrates the classification process of a patient (represented as $\triangle$/black data item) in terms of heart disease diagnosis. In the figure, patients diagnosed with heart disease are represented by blue/$\circ$ markers; otherwise by red/$\square$ markers. Despite the importance-based technique diagnoses the heart disease of the patient correctly, traditional techniques, such as k-NN, SVM, random forest, etc., fail in such a task by considering only the physical features of the data. For example,

in the same figure, the nearest neighbors of the new patient is showed. One can see k-NN classifies the patient to the red/□ class, i.e., without heart disease. Thus, both examples in Fig. 2 show that the proposed technique contributes to the data classification task by considering the organizational structure of the data beyond the physical features. Moreover, the experimental results also revealed the low computational cost of the new technique in comparison to other traditional techniques widely used in literature.



(a) Classification of the data in Fig. 1(a)

(b) Heart disease diagnosis of a patient

Figure 2. Analysis of the classifications provided by the importance-based data classification over misleading cases for traditional techniques.

Bioinspired Network Optimization. In the thesis, it is also proposed a bioinspired optimization framework, which is expected to build up the network while conducting the optimization of a task-oriented quality function. Two quality functions are evaluated in the experiments: high-level and importance-based classification. Results on artificial and real-world data sets (including semantic role labeling) reveal the network provided through the structural optimization presents statistically better results than those generated by the most used network formation methods in literature, especially in higher complexity of class configuration (such as the mixture among different classes). In addition, they also performed well in comparison with widely used traditional classification techniques, such as SVM and logistic regression [Carneiro et al. 2016a]. Moreover, the proposed framework can also be adapted to perform structural optimization for other graph-based learning tasks, such as dimension reduction [Carneiro et al. 2017a].

Following we also list other relevant contributions discussed in the thesis:

- **A simplified framework for high-level classification:** in the thesis, the high-level classification is simplified in a proposed hybrid technique where physical and complex-network based associations are produced from the same network, reducing considerably the number of parameters [Carneiro et al. 2014b, Carneiro and Zhao 2013]. Experimental results show that a larger portion of the high-level association is required to get correct classification when there is a complex-formed and well-defined pattern in the data set. They also demonstrate that the proposed technique presents competitive performance against state-of-the-art methods (e.g., SVM) and it outperforms typical data classification techniques (e.g., classification and regression trees).
- **Graph-based semantic role labeling (SRL):** the Brazilian Portuguese SRL is taken into account in [Carneiro et al. 2016b] and [Carneiro et al. 2017b]. The former pro-

posed a semi-supervised framework based on label propagation in order to investigate the diffusion of semantic roles for that language; and the latter presented a high-level system for the classification of semantic roles in sentences.

- **Parameter-free graph-based dimension reduction (DR):** in [Carneiro et al. 2014a, Cupertino et al. 2013], it is proposed a parameter-free graph-embedding DR method which results are competitive compared to classical network approaches (e.g., kNN) and widely used DR methods (e.g., principal component analysis).

3. Publications

In this section, we present the list of the main articles related to the doctoral research and their corresponding Qualis according to the latest version released by CAPES*. We also present the Impact Factor (IF) measure of the journals classified as Qualis A1.

- **[Qualis A1, IF: 7.4]** Carneiro, M. G., Zhao, L., and Jin, Y. Bio-inspired structural optimization for network-based data classification (under review). *IEEE Trans. Cybern.*.
- **[Q.A1, IF: 3.9]** Cupertino, T. H., Carneiro, M. G., Zheng, Q., Zhang, J., and Zhao, L. A scheme for high level data classification using random walk and network measures (under review). *Expert Syst. Appl.*.
- [Qualis B3] Carneiro, M. G., Cupertino, T. H., Zhao, L., and Rosa, J. L. G. Semi-supervised semantic role labeling for brazilian portuguese (under review). *JIDM*.
- [Qualis B1] Carneiro, M. G., Cupertino, T. H., Cheng, R., Jin, Y., and Zhao, L. (2017a). Nature-inspired graph optimization for dimensionality reduction (accepted). In *IEEE ICTAI*.
- **[Q. A1, IF: 6.1]** Carneiro, M. G. and Zhao, L. (2017). Organizational data classification based on the importance concept of complex networks. *IEEE Trans. Neural Netw. and Learn. Syst.*, PP(99):1–13.
- [Qualis B1] Carneiro, M. G., Rosa, J. L. G., Zheng, Q., Liu, X., and Zhao, L. (2017b). Improving semantic role labeling using high-level classification in complex networks. In *FSKD*, pages 2185–2191.
- **[Qualis A1]** Carneiro, M. G., Zhao, L., Cheng, R., and Jin, Y. (2016a). Network structural optimization based on swarm intelligence for highlevel classification. In *IEEE IJCNN*, pages 3737–3744.
- [Qualis B5] Carneiro, M. G., Zhao, L., and Rosa, J. L. G. (2016b). Graph-based semi-supervised learning for semantic role diffusion. In *KDMiLe*, pages 108–115.
- **[Qualis A1, IF: 3.3]** Cupertino, T. H., Zhao, L., and Carneiro, M. G. (2015). Network-based supervised data classification by using an heuristic of ease of access. *Neurocomputing*, 149:86–92 .
- [Q.B1] Carneiro, M. G., Rosa, J. L. G., Lopes, A. A., and Zhao, L. (2014b). Network-based data classification: combining k-associated optimal graphs and high-level prediction. *J. Braz. Comp. Soc*, 20(1):1–14.
- **[Qualis A1]** Carneiro, M. G., Cupertino, T. H., and Zhao, L. (2014a). K-associated optimal network for graph embedding dimensionality reduction. In *IEEE IJCNN*, pages 1660–1666.
- Cupertino, T. H., Carneiro, M. G., and Zhao, L. (2013). Dimensionality reduction with the k-associated optimal graph applied to image classification. In *IEEE IST*, pages 366–371.
- Carneiro, M. G. and Zhao, L. (2013). High level classification totally based on complex networks. In *BRICS-CCI*, pages 507–514.

In summary, the main articles derived from the thesis have been published in well-reputed venues, such as IEEE TNNLS which is a top international journal in the Machine Learning and Neural Networks areas (6.1 impact factor). According to *Microsoft Academic Research*†, IEEE TNNLS ranks 2nd when taking into account the journals in the Machine Learning field. According to *Google Scholar Metrics*, it also figures at the top positions of the ranking in comparison with other very important journals from Machine Learning, Pattern Recognition, Computer Vision and Data Mining areas. Besides the top publications, the *30th Thesis and Dissertations Contest of the Brazilian Computer Society* also recognized the quality of our research by classifying the work among the eleven best PhD thesis defended in Brazil in 2016.

*http://www.capes.gov.br/images/documentos/Qualis_periodicos_2016/Qualis_conferencia_ccomp.pdf

†http://academic.research.microsoft.com/

References

Carneiro, M. G. (2016). *Redes complexas para classificação de dados via conformidade de padrão, caracterização de importância e otimização estrutural*. PhD thesis, Universidade de São Paulo.

Carneiro, M. G., Cupertino, T. H., Cheng, R., Jin, Y., and Zhao, L. (2017a). Nature-inspired graph optimization for dimensionality reduction (accepted). In *IEEE International Conference on Tools with Artificial Intelligence (ICTAI)*.

Carneiro, M. G., Cupertino, T. H., and Zhao, L. (2014a). K-associated optimal network for graph embedding dimensionality reduction. In *IEEE International Joint Conference on Neural Networks*, pages 1660–1666.

Carneiro, M. G., Rosa, J. L. G., Lopes, A. A., and Zhao, L. (2014b). Network-based data classification: combining k-associated optimal graphs and high-level prediction. *Journal of the Brazilian Computer Society*, 20(1):1–14.

Carneiro, M. G., Rosa, J. L. G., Zheng, Q., Liu, X., and Zhao, L. (2017b). Improving semantic role labeling using high-level classification in complex networks. In *International Conference on Natural Computation, Fuzzy Systems and Knowledge Discovery*, pages 2185–2191.

Carneiro, M. G. and Zhao, L. (2013). High level classification totally based on complex networks. In *IEEE BRICS Congress on Computational Intelligence*, pages 507–514.

Carneiro, M. G. and Zhao, L. (2017). Organizational data classification based on the importance concept of complex networks. *IEEE Transactions on Neural Networks and Learning Systems*, PP(99):1–13. doi:10.1109/TNNLS.2017.2726082.

Carneiro, M. G., Zhao, L., Cheng, R., and Jin, Y. (2016a). Network structural optimization based on swarm intelligence for highlevel classification. In *IEEE International Joint Conference on Neural Networks*, pages 3737–3744.

Carneiro, M. G., Zhao, L., and Rosa, J. L. G. (2016b). Graph-based semi-supervised learning for semantic role diffusion. In *Symposium on Knowledge Discovery, Mining and Learning (KDMiLe)*, pages 108–115.

Chapelle, O., Scholkopf, B., and Zien, A. (2006). *Semi-Supervised Learning*. MIT Press.

Cupertino, T. H., Carneiro, M. G., and Zhao, L. (2013). Dimensionality reduction with the k-associated optimal graph applied to image classification. In *IEEE International Conference on Imaging Systems and Techniques*, pages 366–371.

Cupertino, T. H., Zhao, L., and Carneiro, M. G. (2015). Network-based supervised data classification by using an heuristic of ease of access. *Neurocomputing*, 149:86–92.

Fortunato, S. (2010). Community detection in graphs. *Physics Reports*, 486(3-5):75–174.

Newman, M. (2010). *Networks: An Introduction*. Oxford University Press, Inc.

Silva, T. C. and Zhao, L. (2012). Network-based high level data classification. *IEEE Transactions on Neural Networks and Learning Systems*, 23(6):954–970.

ANALYSIS OF ACADEMIC SOCIAL NETWORKS CONSIDERING SOCIAL CAPITAL

Thiago H. P. Silva, Mirella M. Moro (advisor), Ana Paula Couto da Silva (co-advisor)

[1]Departamento de Computação
Universidade Federal de Minas Gerais – Belo Horizonte, MG – Brazil

{thps,mirella,ana.coutosilva}@dcc.ufmg.br

Abstract. *Our goal is measuring influence and evaluating productivity in science based on social capital, i.e., the advantage of a good location in a social structure. Specifically, we analyze people that act as bridges (brokerage) with well-defined closed groups (closure) in social academic networks. We characterize the networks, propose ranking strategies, validate and evaluate such strategies by using real datasets, official ground-truth rankings and state-of-art indicators. Overall, our results show our proposed strategies contribute for a more robust and pluralistic productivity evaluation of academic social networks.*

Introduction

With the growth of research productivity (e.g., given by the increasing number of publication venues), there have never been so many tools and techniques to automate scientific performance evaluation. Yet, few of those have analyzed the patterns underlying scientific communities based on *social* relationships. Specifically, an important concept on social relationship is *Social Capital*: the advantage created by a good location in a social structure. For example, Burt [2004] calls **brokerage** the activity of people who act between two parts of network (at the intersection) and **closure** the tightening of coordination in a closed network of people. Similar studies reinforce that people make a network stronger by building social capital or bringing new ideas to a different group [Lima et al 2013].

In such a context, here we explore the idea that structural properties of scientific networks may help assessing the quality of the work produced or the influence of the people involved. Thus, we investigate if social characteristics of a researcher or a group of researchers can contribute in the productivity evaluation of academic networks. We do so by proposing metrics on three levels of networks. Figure 1 summarizes the levels and our contributions as well as maps them to the dissertation's chapters. At the end, our metrics may help evaluating research productivity on the individual and group levels.

Researchers productivity evaluated in social networks (defined as communities of individuals) given by (*i*) conferences/journals: we define new endogamy indicators for evaluating influence (*C-Endo* and *Comb*) - Chapter 4, and analyze authorship contributions dynamics with algorithms based on four new properties (*permanency*, *migration*, *exclusivity* and *plurality*) - Chapter 5; (*ii*) interest groups: we study researchers' contribution and propose a new temporal-community index (*tc-index)* - Chapter 6, and analyze the contribution across communities with a new index for knowledge transfer evaluation (*3c-index*) - Chapter 7; (*iii*) graduate programs: we characterize (Appendix D) and propose a performance indicator based on international publication profiles - Chapter 8.

Figure 1. Dissertation's organization and summary of contributions

Ranking Strategies based on Social Capital

We explore the **brokerage** and **closure** concepts to analyze how social influence impacts on academic networks by proposing evaluation strategies to different levels of networks (given by communities of individuals). Specifically, we propose four ranking strategies based on Social Capital (C-endo, tc-index, 3c-index and volume similarity) and four algorithms to characterize mobility behaviors based on contributions dynamics (permanency, migration, exclusivity and plurality). Given that detailing all of them is impossible due to space constraints, Table 2 summarizes the formulations of such new proposed metrics as well as the evaluation metrics.

Next, we first present the experimental methodology used throughout our work. Then, we briefly discuss the insights underlying each proposed strategy and report a couple of results (the dissertation has a bigger set of results).

Experimental Methodology[1]. We validate and evaluate our strategies by using real datasets (collected from DBLP, Google Scholar, ACM SIGs, and graduate programs website), official ground-truth rankings (ERA and Qualis venue rankings; graduate program ranking from CAPES; and researchers awarded by ACM SIGs), and by comparing to existing approaches (e.g., [Montolio et al. 2013, Lima et al 2013]) and state-of-art indicators (number of citations received, number of publications and h-index), as well as performing random sampling analysis. All data consider Computer Science researchers[2].

Community-based Endogamy. We first verify if the concept of social capital can assess the quality of research. The research endogamy proposed by [Montolio et al. 2013] follows the intuition that cooperating with new researchers is very likely to introduce new ideas to a research community. However, the endogamy computation ($Endo$) explores the *degree* of new collaborations, but not the *importance* of relationship between authors and their communities. For instance, researchers who tend to publish in few venues with a selected set of authors are considered very important to the community because, often, such authors act as hubs due to high expertise, thus making the community stronger as a whole by following the aforementioned concepts of bridges (brokerage). Therefore, we explore this social concept and create a new community-based endogamy metric (*C-Endo*) and a combined version (*Comb*). Table 2 shows the agreement results for two relevant quality venues rankings. The best results are for our proposed metrics and, thus, we can successfully apply social concepts to assess the research quality.

Authorship Contribution Dynamics. As the researchers' behavior impacts on the venue quality, we also analyze the dynamics of authors within communities given by publication venues. We propose algorithms for analyzing the contribution profile (*Exclusivity* and *Plurality*) and contribution dynamics (*Permanency* and *Migration*). Overall, results show researchers in Computer Science do move their authorship contribution around venues. Their focus is on the high quality venues, with conferences having more diverse contributions and journals a more defined core.

[1]The databases building processes (e.g., collecting and matching), distributions of scores, sensibility of variables and statistical tests (e.g., Spearman's, ANOVA, Tukey's test and Kendall tau) are not discussed in this summary, as statistics and further information are presented in Chapter 3 of the dissertation.

[2]Without loss of generality, all metrics may be applied to networks created by researchers from any area, as long as the researchers may be grouped according to the community type.

Table 1: Overview of the evaluation metrics and the new proposed metrics.

	Item	Formula
Evaluation	Discounted Cumulative Gain	$DCG@k = \sum_{i=1}^{k} \frac{2^{rel_i}-1}{log_2(i+1)}$, where rel_i is the relevance of the item at the i-th position
	Agreement	$P = 100\frac{p}{p+f}$, where p and f are # of concordant and discordant pairs
Symbols	Author, work and venue	a, w and v
	Set of all authors, works and venues	$\mathcal{A}$, $\mathcal{W}$ and $\mathcal{V}$
	Set of authors with similar interests	Community c
	Time window and the set of all time window	t, $\mathcal{T}$
	Set of authors of w	A_w
	Set with all works of a	$\mathcal{W}_a$. At time window t is $\mathcal{W}_a(t)$
	Set with all works in v	$\mathcal{W}_v$. At time window t is $\mathcal{W}_v(t)$
	Set with all venues published by a	$\mathcal{V}_a$
Community Endogamy	Co-authored works of a set authors P in v	$l_v(P)$
	Subsets of authors of size i in w	$L_i(w)$
	Subsets with at least two authors in w	$L(w) = \bigcup_{i=2}^{i=\|A_w\|} L_i$
	Community Endogamy of a set of authors P at v	$C\text{-}Endo(\mathcal{P}, v) = \frac{\|l_v(P)\|}{\|\cup_{a\in P}\, l_v(\{a\})\|}$
	Community Endogamy of w at v	$Endo(w, v) = \frac{\sum_{x\in L(w)} C\text{-}Endo(x, v)}{\|L(w)\|}$
	Community Endogamy of v at c	$Endo(v, c) = \frac{1}{\|\mathcal{W}_v\|} \sum_{w\in\mathcal{W}_v} C\text{-}Endo(w, v)$
Authorship Contribution	Set of authors with at least one work at t	$\mathcal{A}(t)$
	Authorship Contribution to w	$C_w = \frac{1}{\|A_w\|}$
	Temporal Contribution	$C_a(t) = \frac{1}{\|\mathcal{A}(t)\|} \sum_{\forall w\in W_a(t)} C_w$
	Authorship Contribution of a at v	$C_{av}(t) = \frac{1}{C_a(t)} \sum_{\forall w\in\{W_a(t)\cap W_v(t)\}} C_w$
	Authorship Contribution Dynamics	$C_{av}(T) = \{C_{av}(1), C_{av}(2), \cdots, C_{av}(T)\}$
	Authorship Contribution at v	$C_v(t) = \frac{1}{\|\mathcal{A}(t)\|} \sum_{\forall a\in\mathcal{A}(t)} C_{av}(t)$
	Permanency	$P_v(t) = \frac{1}{C_v(t)} \sum_{\forall a\in\mathcal{A}(t)} \min(C_{av}(t-1), C_{av}(t))$
	Migration of an author	$M_a(t) = \{(v_i, v_j, \delta_{ij}) \mid \forall (v_i, v_j) \in \{V_a(t-1) \times V_a(t)\}\}$
	Migration between two venues	$M_{(v_i,v_j)}(t) = \frac{1}{\|\mathcal{A}_{v_i}(t)\|} \sum_{\forall a\in\mathcal{A}(t)} M_a(t)$
	Exclusivity	$E_v(t) = \frac{1}{C_v(t)} \sum_{\forall a\in\{\mathcal{A}(t)\mid C_a(t)=C_{av}(t)\}} C_a(t)$
	Plurality	$Pl(v_i, v_j, t) = \frac{1}{\|\mathcal{A}(t)\|} \sum_{\forall a\in\mathcal{A}(t)} min(C_{av_i}(t), C_{av_j}(t)), \forall\,(v_i, v_j)$
tc-index	Score of w and a threshold of a set of works W	$score(w)$ and $score_{lim}(W)$
	tc-index of a	$tc\text{-}index(a) = \sum_{\forall v\in\mathcal{V}} \sum_{\forall t\in\mathcal{T}} \sum_{\forall w\in\mathcal{W}_v(t)} \mathbb{1}(.)$, where $\mathbb{1}(.) = \begin{cases} 1, & \text{if } w \in \mathcal{W}_a \text{ and } score(w) \geq \text{score}_{lim}(\mathcal{W}_v(t)) \\ 0, & \text{otherwise} \end{cases}$
3c-index	Percentile ranking of a in c	$p_a^c = \frac{l_a^c + 0.5e_a^c}{N^c}$, where N^c is the number of authors in c, l_a^c and e_a^c the number of authors with rankings values lower or equal than to the a
	Base Community of a	$b_a = \text{argmax}_{\forall c\in\mathcal{C}} p_a^c$
	Degree of influence of a in c	$infd_a = b_a - p_a^c$
	Projection function of percentile x at b_a	$f_b(x)$
	3c-index of a	$3c\text{-}index(a) = f_b(b_a) + \sum_{\forall c\in\mathcal{C}} (b_a - p_a^c) f_b(p_a^c)$
Volume Intensity	Set of venues published by community c	$\mathcal{U}_c : \{\forall a \in c : \bigcup V_a\}$
	Set of all publications of community c	$A_c = \{\forall a \in c : \bigcup \mathcal{W}_a\}$
	Cosine of two communities	$Cosine_{ij} = \frac{\sum_{\forall v\in\mathcal{V}} \|A_i \cap \mathcal{W}_v\|\|A_j \cap \mathcal{W}_v\|}{\sqrt{\sum_{\forall v\in\mathcal{V}} \|A_i \cap \mathcal{W}_v\|^2} \sqrt{\sum_{\forall v\in\mathcal{V}} \|A_j \cap \mathcal{W}_v\|^2}}$.
	Jaccard of two communities	$Jaccard_{ij} = \frac{\mathcal{U}_i \cap \mathcal{U}_j}{\mathcal{U}_i \cup \mathcal{U}_j}$.
	Volume Intensity of two communities	$Volume\ Intensity_{ij} = 2 \sum_{\forall v\in\mathcal{V}} \frac{min(\|\mathcal{W}_v \cap A_i\|, \|\mathcal{W}_v \cap A_j\|)}{\|A_i \cup A_j\|}$

Table 2. Means of agreements for conferences and journals with random samples of venues per tier by ERA and Qualis rankings.

	Conferences		Journals	
Metric	**ERA**	**Qualis**	**ERA**	**Qualis**
Endo	73.1	76.7	59.6	59.2
C-Endo	69.5	77.9	**69.1**	**77.3**
Comb	**75.4**	**81.3**	68.7	69.9

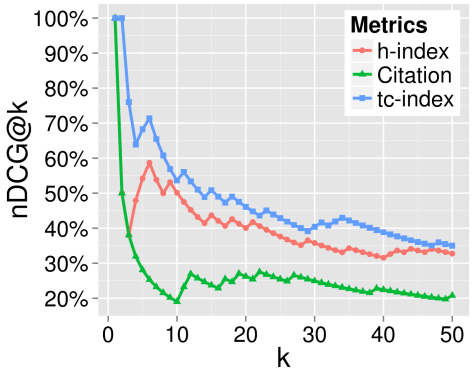


Figure 2. Comparison among h-index, citation count and tc-index rankings by and nDCG@k.

Research Productivity Index based on Evolving Communities. Here, the focus is exploring the link between a researcher and his/her community, i.e., it explores the closure concept with a fairer strategy. Recent studies have considered the performance of each researcher individually, as well as measured the degree of influence among research areas (i.e., index based on one's own publications). In contrast, we define the *tc-index* (temporal community-based index) that assesses researchers' productivity based on the evolution of communities by comparing each publication with similar ones over a *threshold* of community contribution (i.e., exploring closure by comparing within of a closed group). Thus, the proposed strategy is fairer because it enables to compare recent works to other recent works (same temporal visibility) and, at same time, it avoids comparing works with different publication patterns. Figure 2 shows that tc-index consistently outperforms both h-index and citation metrics: it ranks relevant researchers early. Overall, tc-index successfully measures the impact of researchers' contributions and identifies outstanding researchers in their communities.

Research Contribution Across Communities. Given that knowledge transfer among communities is crucial to the progress of Science, we propose the *3c-index* (cross-community contribution) that aims to identify and quantify such transfer by considering two social capital concepts: *closure* as the potential knowledge acquired and *brokerage* as potential of sharing information. Then, **3c-index** measures the influence of researchers according to their specialties on other contexts. Figure 3 shows the results of the ranking produced by 3c-index against traditional metrics (left) and cross-area index proposed by [Lima et al 2013] (right), applying as function score the h-index. The 3c-index outperforms all indices by ranking relevant researchers in early positions, then endorsing the use of the metric as a complementary tool for performing a better productivity evaluation.

Publication Profile Similarity. Here, we quantify the scientific performance of a community (group of researchers) based on the similarity between its publication profile and a reference community's publication profile. The idea is to explore the link *researcher-venue* to define *closed groups*, then measuring the overlapping between them as a performance indicator. In other words, our objective focuses on the plural perspective (instead of individual one) of evaluating graduate programs. We then propose a performance indicator (called **volume intensity**) for academic communities based on international publication profiles, which applies only researchers' publication records (i.e., inexpensive, highly available information). Figure 4 (left) shows that the strategy based on volume in-

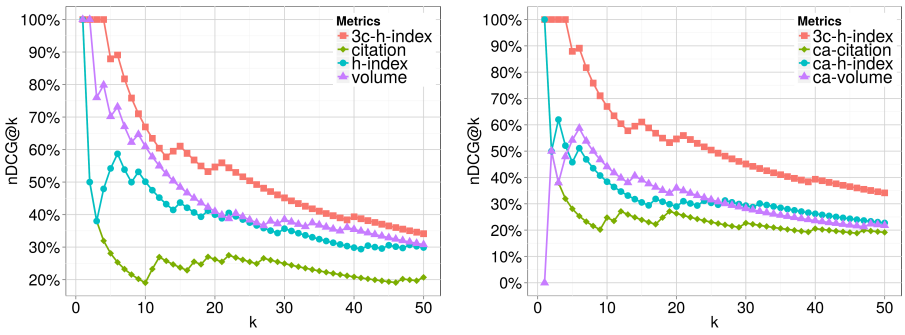


Figure 3. Comparison between 3c-index and the *baselines* (left) and ca-index (right) according to the nDCG.

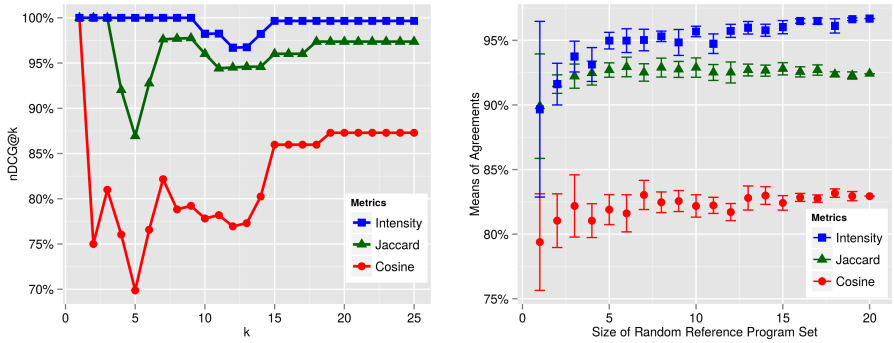


Figure 4. Comparison among publication profile similarity strategies (left) and for random samples by varying the size of reference program set (right).

tensity retrieves the relevant communities earlier. Figure 4 (right) shows that the volume intensity is consistent when the set of programs chose is random, reinforcing the use of reference publication profile as a performance indicator.

Concluding Remarks

Overall, our contributions and respective publications are summarized in three fronts:

- Community-based ranking strategies:
  - We proposed endogamy metrics based on the communities given by closure concept. They outperformed the existing endogamy computation [Montolio et al. 2013] and were published in the ACM/IEEE JCDL [Silva et al. 2014a].
  - By exploring the knowledge acquired (closure) and sharing information (brokerage) potentials, we propose a new indicator for measuring the influence through communities. The experimental evaluation results were better than a previous metric proposed by our research group [Lima et al 2013]. This study was published as a full paper in the main Brazilian event on databases [Silva et al. 2015c] and was invited for an extended version in a journal [Silva et al. 2014b].
- Evaluation metrics from original perspectives:
  - We introduced the concept of *authorship contribution* by providing new insights and shedding some light on researchers' dynamics behaviors and how they move their contributions on the networks. This part was published in the ACM SAC [Silva et al. 2015a].
  - We proposed a temporal community index for assessing the researchers productivity by comparing the impact of his/her work with a group formed by similar works. This part was published in the TPDL [Silva et al. 2015b].

  - We proposed an approach for quantifying the scientific performance of a group of researchers based on the similarity between its publication profile and a reference group's publication profile. This part was published in the Scientometrics [Silva et al. 2016].
- Characterization Studies:
  - We created a spatial-temporal tool for analyzing the Brazilian Computer Science programs in terms of where their members have obtained their academic degrees. The tool was published as a Demo [Silva et al 2015]. There is also a video available at `http://dcc.ufmg.br/~mirella/Tools/CollabViz/`.
  - We performed an analysis of the Brazilian Computer Science graduate programs based on venues' quality, collaborations, students mentored and career length. This analysis has been submitted to Scientometrics (*A Profile Analysis of Brazilian Computer Science Graduate Programs - Thiago H. P. Silva, Clodoveu Davis, Alberto Laender, Mirella M. Moro and Ana P. C. da Silva*).

Although there are good results, we understand that quality is defined by the work content and not solely by bibliometric indices. Hence, we are not just assessing the quality, but providing new performance metrics and strategies based on social capital to be used with multiple indicators for a more robust and plural scientific production evaluation. Our studies may be used for providing new insights on the overall production of the researchers as, for instance, to expand researchers' perspective by building new collaborations with the objective of become his/her networks stronger.

As for future work, ideas for extending and improving this work include (but are not limited to): consider different scientific areas, as our metrics may be applied to any area of science given proper datasets; expand the model from Chapter 8 to a perspective based on authors' profile; and check if our indicators can be used in other contexts such as academic recommendation systems.

References

Burt, R. S. (2004). Structural Holes and Good Ideas. *Am. J. Sociology*, 110(2):349–399.

Lima et al, H. (2013). Aggregating Productivity Indices for Ranking Researchers Across Multiple Areas. In *JCDL*, pages 97–106, USA.

Montolio, S. L., Dominguez-Sal, D., and Larriba-Pey, J. L. (2013). Research Endogamy as an Indicator of Conference Quality. *SIGMOD Rec.*, 42(1):11–16.

Silva, T. H., Penha, G., da Silva, A. P. C., and Moro, M. M. (2016). A performance indicator for academic communities based on external publication profiles. *Scientometrics*, 107(3):1389–1403.

Silva, T. H. P., Moro, M. M., and Silva, A. P. C. (2015a). Authorship Contribution Dynamics on Publication Venues in Computer Science: an Aggregated Quality Analysis. In *SAC*, pages 1142–1147, Spain.

Silva, T. H. P., Moro, M. M., and Silva, A. P. C. (2015b). tc-index: a New Research Productivity index based on Evolving Communities. In *TPDL*, pages 209–221, Poland.

Silva, T. H. P., Moro, M. M., Silva, A. P. C., Meira Jr., W., and Laender, A. H. F. (2014a). Community-based Endogamy as an Influence Indicator. In *JCDL*, pages 67–76, UK.

Silva, T. H. P., Rocha, L. M. A., Moro, M. M., and Silva, A. P. C. (2014b). Research Contribution across Communities as an Influence Indicator. *JIDM*, 6(3):192–205.

Silva, T. H. P., Rocha, L. M. A., Moro, M. M., and Silva, A. P. C. (2015c). Contribuição de Pesquisa entre Comunidades como um Indicador de Influência. In *SBBD*, Brazil.

Silva et al, T. H. P. (2015). Análise Espaço-Temporal da Colaboração Acadêmica em Programas de Pós-Graduação em Computação. In *SBBD*, Brazil.

DISCRETIZADOR HEURÍSTICO PARA O CONTEXTO DE CLASSIFICAÇÃO HIERÁRQUICA

Leandro Ribeiro Galvão[1]
orientador: Luiz Henrique de Campos Merschmann[2]

[1]Departamento de Computação – Universidade Federal de Ouro Preto

[2]Departamento de Ciência da Computação – Universidade Federal de Lavras

leandrodvmg@gmail.com, luiz.hcm@dcc.ufla.br

1. Introdução

O KDD (*Knowledge Discovery from Data*) é um processo de descoberta de conhecimento a partir de bases de dados composto por três etapas principais: o pré-processamento dos dados, a mineração de dados e a validação dos resultados.

O processo de discretização pode ser definido como uma etapa do pré-processamento de dados, cujo objetivo é transformar atributos contínuos em atributos discretos [Yang et al. 2005]. Os métodos de discretização podem ser categorizados como supervisionados ou não-supervisionados [Garcia et al. 2013]. Os métodos que utilizam a informação presente no atributo classe durante o processo de discretização de um atributo são categorizados como supervisionados, enquanto que os métodos que desconsideram o atributo classe são categorizados como não-supervisionados.

Na área de classificação há algoritmos que requerem que todos os atributos da base de dados sejam discretos, como por exemplo, o ID3 [Quinlan 1986]. Mas existem também aqueles algoritmos que, apesar de conseguirem lidar com atributos contínuos, produzem modelos preditivos mais precisos quando os atributos estão discretizados, como é o caso do *Naive Bayes* [Richeldi and Rossotto 1995, Chlebus and Nguyen 1998]. Desse modo, fica evidente a importância das técnicas de discretização como um pré-processamento para a etapa de mineração de dados.

A classificação é uma tarefa de mineração de dados que busca identificar a classe de um determinado objeto a partir de suas características [Han and Kamber 2011]. Diferentes tipos de problemas de classificação podem ser encontrados na literatura, cada qual com o seu nível de complexidade [Silla Jr and Freitas 2011]. Em problemas de classificação plana, cada instância é associada a uma ou mais classes pertencentes a um conjunto finito de classes, sendo que não há qualquer tipo de relacionamento entre elas. Porém, existem problemas de classificação mais complexos, conhecidos como problemas de classificação hierárquica, nos quais as classes encontram-se organizadas em uma hierarquia.

Diversas áreas de pesquisa e aplicação, tais como categorização de textos [Dollah and Aono 2011], predição de funções de proteínas [Merschmann and Freitas 2013, Valentini 2014], classificação de gêneros musicais [Silla and Freitas 2009] e classificação de imagens [Kramer et al. 2012], já se beneficiaram com a utilização de técnicas de classificação hierárquica, uma vez que as classes a serem preditas encontravam-se naturalmente organizadas em uma hierarquia.

[1]Os autores agradecem à CAPES, CNPq, FAPEMIG e UFOP pelo apoio financeiro concedido.

Os métodos de discretização supervisionados disponíveis na literatura não foram projetados para levar em consideração os relacionamentos entre classes existentes em problemas de classificação hierárquica. Dessa forma, pesquisas na área de classificação hierárquica que necessitaram de atributos discretos, tais como [Silla and Kaestner 2013] e [Merschmann and Freitas 2013], utilizaram métodos de discretização não-supervisionados.

A hipótese levantada neste trabalho é que métodos de discretização supervisionados, pelo fato de levarem em consideração o atributo classe no momento da discretização, poderiam proporcionar melhoria no desempenho preditivo de classificadores hierárquicos. Portanto, a inexistência de métodos de discretização supervisionados adequados para serem usados em conjunto com métodos de classificação hierárquica motivou o desenvolvimento desta pesquisa.

Os principais objetivos deste trabalho foram propor, implementar e avaliar um método de discretização supervisionado capaz de lidar com o relacionamento entre classes existente em problemas de classificação hierárquica. O método aqui proposto corresponde a uma heurística que foi denominada *Agglomerative Discretization Heuristic for Hierarchical Classification* - ADH2C.

Devido à inexistência de métodos de discretização supervisionados para o contexto de classificação hierárquica e dado que os métodos de discretização não-supervisionados *EqualWidth* (EW) e *EqualFrequency* (EF) vêm sendo utilizados em trabalhos de classificação hierárquica, esses dois métodos não-supervisionados foram adotados como base de comparação com a heurística proposta neste trabalho.

Confirmando a hipótese levantada inicialmente, para a maioria das bases de dados utilizadas nos experimentos, o desempenho preditivo de um classificador hierárquico, quando a base de dados foi discretizada pela heurística ADH2C, foi superior àquele obtido quando a mesma base de dados foi discretizada pelos métodos não-supervisionados EW e EF.

2. Método Proposto

Para discretizar atributos contínuos de bases de dados utilizadas em problemas de classificação hierárquica monorrótulo, a heurística proposta neste trabalho, denominada *Agglomerative Discretization Heuristic for Hierarchical Classification* - ADH2C, utiliza uma adaptação do cálculo de distância entre classes apresentado em [Blockeel et al. 2002]. A heurística ADH2C pode ser dividida em três fases principais: inicialização, construção das soluções candidatas e seleção da melhor solução.

A fase de inicialização começa com a ordenação dos valores do atributo contínuo em ordem crescente. Em seguida, são criadas as partições puras, ou seja, cada valor (ou conjunto de valores) distinto do atributo contínuo é atribuído a uma partição. Por fim, as partições adjacentes são avaliadas duas a duas e, sempre que todos os valores contínuos do par de partições estão associados a uma mesma classe, essas partições são fundidas.

A segunda fase da heurística ADH2C é responsável pela construção de uma lista de soluções candidatas. Uma solução candidata é composta por conjunto de pontos de corte que definem um esquema de discretização dos dados. Os pontos de corte são definidos como a média dos valores localizados nas fronteiras das partições de dados.

A execução da segunda fase do método requer que tenham sido produzidas pelo menos 4 partições na fase de inicialização. Desse modo, se essa restrição não for atendida, o processo de discretização é finalizado retornando o conjunto de pontos de cortes obtido na fase de inicialização. Essa restrição garante que o critério de seleção da melhor solução candidata, explicado mais adiante, seja executado corretamente. Supondo que tenham sido geradas N partições na fase de inicialização, sendo N $\geq$ 4, o número de soluções candidatas geradas nessa segunda fase é igual N$-$1. Isso ocorre porque a cada iteração realizada nessa fase, um par de partições adjacentes é fundido produzindo uma nova solução candidata. Esse processo de fusões continua até restar apenas uma partição.

Em cada iteração da segunda fase, ou seja, da fase de construção de soluções candidatas, a seleção do par de partições que sofrerão a fusão é realizada de acordo com a distância entre as partições, ou seja, o par de partições adjacentes que possuir a menor distância é escolhido para o processo de fusão. O cálculo da distância entre partições é feito utilizando uma adaptação que foi proposta a partir do cálculo de distância entre classes apresentado em [Blockeel et al. 2002].

A última fase de execução da heurística é responsável pela seleção da solução (esquema de discretização) que será apresentada como resultado. Como cada solução candidata, gerada em cada uma das iterações da segunda fase, está associada à distância entre o par de partições adjacentes que ao serem fundidas deram origem à solução em questão, a seleção da solução apresentada como resultado final é realizada com base nessa distância.

Dentre as soluções candidatas obtidas na segunda fase, somente as denominadas soluções candidatas viáveis são consideradas no processo de escolha da melhor solução. Uma solução S contendo N partições é considerada viável quando a distância associada à S é menor do que aquelas associadas às soluções vizinhas, ou seja, soluções contendo N-1 e N+1 partições. Identificadas as soluções viáveis, deve-se quantificar a qualidade de cada solução viável. Finalmente, a solução candidata viável de melhor qualidade é escolhida e o conjunto de pontos de cortes associado a ela (esquema de discretização) é retornado como solução final. Caso não haja nenhuma solução candidata viável, a heurística retorna como solução o conjunto de pontos de corte das partições geradas na fase de inicialização.

3. Experimentos

Nos experimentos deste trabalho foram utilizadas importantes bases de bioinformática disponibilizadas por [Clare and King 2003] que possuem atributos contínuos. Essas bases de dados são multirrótulo, ou seja, cada instância encontra-se associada a uma ou mais classes da hierarquia. Como este trabalho lida com o cenário monorrótulo (*single path of labels*), essas bases de dados foram transformadas para conter uma única classe por instância. Essa transformação foi realizada selecionando-se, para cada instância, a classe mais frequente na base de dados original. Em caso de empate na frequência das classes, seleciona-se de forma aleatória uma das classes.

A partir das bases de dados monorrótulo, um pré-processamento foi realizado para substituição dos valores ausentes de atributos nessas bases. A heurística HSIM foi adotada para a substituição dos valores ausentes [Galvão and Merschmann 2016].

Os experimentos executados neste trabalho permitiram a comparação da qualidade da discretização obtida por meio da heurística proposta com aquela alcançada a partir dos

métodos de discretização não supervisionados EW e EF. Esses métodos foram escolhidos como base de comparação pelo fato de, devido à inexistência de métodos de discretização supervisionados capazes de lidar com as hierarquias de classes, eles já terem sido utilizados em trabalhos na área de classificação hierárquica, tais como [Silla and Kaestner 2013] e [Merschmann and Freitas 2013].

Os métodos de discretização EW e EF possuem o parâmetro k que representa o número de intervalos discretos que devem ser gerados ao final do processo de discretização. Nos experimentos realizados neste trabalho, visando promover uma comparação justa, o número *k* de partições geradas pelos métodos EW e EF para cada atributo de cada base de dados foi exatamente igual ao gerado pela heurística ADH2C. Para exemplificar esse processo, suponha que a heurística ADH2C tenha gerado 6 partições para o primeiro atributo da base de dados *Church*. Logo, os métodos EW e EF irão discretizar o primeiro atributo dessa mesma base utilizando o parâmetro *k* igual a 6.

Após a discretização dos atributos das bases de dados a partir da heurística ADH2C e dos métodos utilizados como base de comparação, essas bases são utilizadas pelo classificador hierárquico *Global Model Naive Bayes* (GMNB) [Silla and Freitas 2009] e o desempenho preditivo do mesmo é avaliado através da técnica 10-validação cruzada. A medida de avaliação *F-measure hierárquica* (hF), que corresponde a uma adaptação da tradicional medida *F-measure*, foi adotada para mensurar o desempenho preditivo do classificador GMNB.

4. Resultados e Conclusão

A Tabela 1 apresenta o desempenho preditivo obtido pelo classificador GMNB utilizando as bases de dados pré-processadas pela heurística ADH2C e pelos métodos EW e EF. O parâmetro k adotado para os métodos EW e EF variou de acordo com o número de partições geradas pela heurística ADH2C para cada atributo das bases de dados. A primeira coluna da Tabela 1 contém o nome das bases de dados utilizadas nos experimentos. A partir da segunda até a última coluna são apresentados os valores médios de hF (com desvio-padrão entre parênteses) obtidos pelo classificador GMNB utilizando as bases de dados pré-processadas pelos métodos EW, EF e pela heurística ADH2C, respectivamente.

Os resultados computacionais obtidos foram submetidos ao teste estatístico *Wilcoxon Signed-Rank Test* (*two sided test*) para avaliar se a diferença no desempenho do classificador GMNB ao utilizar as bases de dados pré-processadas pela heurística ADH2C e pelos métodos usados na comparação é estatisticamente significativa [Japkowicz and Shah 2011]. O teste estatístico foi aplicado utilizando-se um nível de confiança de 95%. As duas últimas linhas da Tabela 1 apresentam uma sumarização dos resultados desse teste estatístico, ou seja, a quantidade de vezes em que a heurística ADH2C foi estatisticamente superior (Resultados Superiores) ou inferior (Resultados Inferiores) ao método não-supervisionado representado na coluna correspondente. Além disso, para cada base de dados, o maior valor médio de hF está destacado em negrito.

A partir dos resultados apresentados na Tabela 1 observa-se que, para 6 das 9 bases de dados, o classificador GMNB alcançou o maior hF médio ao utilizar as bases de dados pré-processadas pela heurística ADH2C. Além disso, para as mesmas 6 bases de dados, a heurística apresentou desempenho estatisticamente superior aos demais métodos utilizados nas comparações. Vale ressaltar que em apenas 1 das 9 base de dados

Tabela 1. Valores médios de hF obtidos pelo GMNB após a discretização utilizando-se os métodos EW, EF e ADH2C.

Bases de dados	EW + GMNB *hF* (desvio padrão)	EF + GMNB *hF* (desvio padrão)	ADH2C + GMNB *hF* (desvio padrão)
CellCycle	22,15 (2,83) +	29,64 (3,08) +	**34,43 (2,68)**
Church	17,35 (2,21) +	17,85 (1,34) +	**21,42 (1,49)**
Derisi	12,43 (0,90)	**12,47 (1,19)**	12,41 (1,10)
Eisen	21,26 (2,50) +	21,18 (1,82) +	**22,93 (1,75)**
Expr	38,81 (1,49) +	44,64 (2,53) +	**47,98 (2,19)**
Gasch1	20,69 (2,11) +	24,42 (2,32) +	**27,33 (2,61)**
Gasch2	18,15 (1,48) +	21,03 (1,66) +	**22,67 (2,67)**
Sequence	**21,55 (1,21) -**	18,87 (0,63)	18,23 (1,20)
SPO	**13,89 (1,55)**	13,65 (1,22)	13,76 (1,37)
Resultados Superiores	6	6	
Resultados Inferiores	1	0	

(Sequence) a heurística ADH2C apresentou desempenho estatisticamente inferior em relação a um dos métodos de comparação (EW). Os resultados mostrados nas duas últimas linhas da Tabela 1 confirmam a superioridade da heurística ADH2C em relação a cada um dos métodos de discretização utilizados na avaliação comparativa.

Portanto, é possível concluir que a heurística ADH2C mostrou-se adequada para a discretização de dados relacionados ao contexto de classificação hierárquica, proporcionando ao classificador hierárquico GMNB um desempenho preditivo (em termos de hF) superior àquele obtido ao se utilizar métodos de discretização não-supervisionados (EF e EW) no pré-processamento das bases de dados.

O classificador hierárquico global GMNB foi escolhido para avaliação experimental pelo fato de a literatura já ter mostrado que o classificador Naive Bayes se beneficia da discretização de dados. No entanto, qualquer outro classificador hierárquico global pode ser utilizado em conjunto com o discretizador proposto neste trabalho.

Como trabalhos futuros, pode-se investigar o desempenho da heurística ADH2C com outros classificadores hierárquicos globais e implementar outros métodos de discretização supervisionados para serem comparados com a heurística aqui proposta.

Subprodutos do Trabalho

Como parte da pesquisa desenvolvida ao longo do mestrado, um método supervisionado de imputação de valores ausentes para o contexto de classificação hierárquica foi proposto, implementado e avaliado. A proposta desse método foi apresentada e resultou na publicação de um artigo em uma conferência internacional da área de mineração de dados (*19th International Conference on Discovery Science* – DS 2016). O artigo aceito nessa conferência foi publicado no *Lecture Notes in Artificial Intelligence* (LNAI).

Referências

Blockeel, H., Bruynooghe, M., Dzeroski, S., Ramon, J., and Struyf, J. (2002). Hierarchical multi-classification. In *Proceedings of the ACM SIGKDD Workshop on Multi-*

relational Data Mining, pages 21–35.

Chlebus, B. S. and Nguyen, S. H. (1998). On finding optimal discretizations for two attributes. In *Rough Sets and Current Trends in Computing*, pages 537–544. Springer.

Clare, A. and King, R. D. (2003). Predicting gene function in saccharomyces cerevisiae. *Bioinformatics*, 19(suppl 2):ii42–ii49.

Dollah, R. B. and Aono, M. (2011). Classifying biomedical text abstracts based on hierarchical 'concept'structure. *World Academy of Science, Engineering and Technology, International Journal of Computer, Electrical, Automation, Control and Information Engineering*, 5(2):178–183.

Galvão, L. R. and Merschmann, L. H. (2016). Hsim: A supervised imputation method for hierarchical classification scenario. In *Proc. of the International Conference on Discovery Science (LNAI)*, pages 134–148.

Garcia, S., Luengo, J., Sáez, J. A., López, V., and Herrera, F. (2013). A survey of discretization techniques: Taxonomy and empirical analysis in supervised learning. *IEEE Transactions on Knowledge and Data Engineering*, 25(4):734–750.

Han, J. and Kamber, M. (2011). *Data Mining: Concepts and Techniques: Concepts and Techniques*. Elsevier.

Japkowicz, N. and Shah, M. (2011). *Evaluating Learning Algorithms*. Cambridge University Press.

Kramer, G., Bouma, G., Hendriksen, D., and Homminga, M. (2012). Classifying image galleries into a taxonomy using metadata and wikipedia. In *Natural Language Processing and Information Systems*, pages 191–196. Springer.

Merschmann, L. H. d. C. and Freitas, A. A. (2013). An extended local hierarchical classifier for prediction of protein and gene functions. In *Proc. of the Data Warehousing and Knowledge Discovery*, pages 159–171.

Quinlan, J. R. (1986). Induction of decision trees. *Machine learning*, 1(1):81–106.

Richeldi, M. and Rossotto, M. (1995). Class-driven statistical discretization of continuous attributes. In *Machine Learning: ECML-95*, pages 335–338. Springer.

Silla, C. N. and Freitas, A. A. (2009). Novel top-down approaches for hierarchical classification and their application to automatic music genre classification. In *International Conference on Systems, Man and Cybernetics*, pages 3499–3504.

Silla, C. N. and Kaestner, C. A. (2013). Hierarchical classification of bird species using their audio recorded songs. In *Systems, Man, and Cybernetics (SMC), 2013 IEEE International Conference on*, pages 1895–1900. IEEE.

Silla Jr, C. N. and Freitas, A. A. (2011). A survey of hierarchical classification across different application domains. *Data Mining and Knowledge Discovery*, 22(1):31–72.

Valentini, G. (2014). Hierarchical ensemble methods for protein function prediction. *ISRN bioinformatics*, 2014.

Yang, Y., Webb, G. I., and Wu, X. (2005). Discretization methods. In *Data mining and knowledge discovery handbook*, pages 113–130. Springer.

Efficiently Computing Geometric Composition Patterns in Big Data

[1]**Author**: Amir Khatibi, Dexl Lab, LNCC, Petropolis
Advisor: Prof. Fabio Porto, Dexl Lab, LNCC, Petropolis
Co-Advisor: Prof. Eduardo Ogasawara, Computer Science department, CEFET-RJ

Abstract. *Big data processing is expected to empower decision-making as more information becomes accessible to analytical tools. In this work [1], we argue that the data deluge produced by the Big Data phenomenon blurs, amongst billions of dataset elements, high-level objects that can only be perceived once adequate composition models are in place. We argue that identifying such objects is relevant for various disciplines and we focus on Geometric Composition model using an example in astronomy. This work formulates the problem of Efficiently Computing Geometric Composition Patterns in Big Data (GCP). We present a novel technique with some pruning strategies to find these objects while we show the robustness and efficiency of our technique using a SPARK implementation over Hadoop parallel architecture.*
Keywords: *Big Data, Geometric Query Processing, Spatial Indexing, Astronomy Sample Query , Parallel Processing*

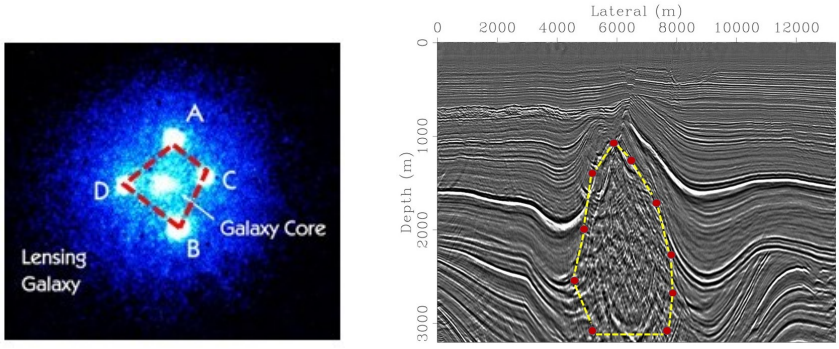
1. Motivation

In recent years, Big Data has become a new ubiquitous buzzword. The availability of large datasets in science, web and mobile applications enables new interpretations about natural phenomena and human behavior. From inferring popular touristic sites based on comments in social media to the existence of dark matter inferred from multiple occurrences of quasars in Sky surveys[Overbye 2015], new knowledge emerges whenever individual observations are combined allowing for queries on patterns. This paper formulates and processes a type of pattern queries in spatial databases that we refer to as Geometric Composition Patterns (GCP). In order to illustrate the scenarios in which GCP arises, consider the following two use cases:

Scenario 1. An astronomy catalog is a table holding billions of sky objects from a region of the sky, captured by telespcopes. An astronomer may be interested in identifying the effects of *gravitational lensing* in quasars, as predicted by Einstein's General Theory of Relativity [Einstein 2015]. According to this theory, massive objects like galaxies bend light rays that travel near them just as a glass lens does. Due to this phenomenon, one would observe two or more virtual images of the lensed quasar leading to a composed new object (Figure 1.a), such as the Einstein cross [Overbye 2015].

Scenario 2. In seismic studies [Brown 2004], a huge dataset holds billions of seismic traces, which, for each position in space, present a list of amplitudes of a seismic wave at various depths (i.e. seismic traces). A seismic interpreter tries to extract meaning

[1]This research is partially funded by the DELL-EMC Brazil, Brazilian National Institute of Science and Technology for Web Research (MCT/CNPq/INCT) and FAPERJ.



(a) (b)

Figura 1. (a) Einstein Cross identification from astronomic catalogs [NASA's Hubble Space Telescope 1990], (b) Salt Dome identification from seismic datasets [Sethian 2006]

out of such datasets by finding higher-level seismic objects (Figure 1.b) such as: faults [Ciarlini et al. 2015], salt domes, etc. Those *features* may be obtained from the seismic dataset through a spatial composition of seismic traces. Indeed, such compositions conveys meaning to the user in terms of real seismic objects of interest.

In the scenarios above, GCP, such as the Einstein cross or the salt dome, are obtained from compositions of individual elements in large datasets. Extracting GCP from large datasets entails the discovery of geometric patterns obtained from the composition of individual data points whose spatial position forms an object with a particular shape satisfying a geometric constraint. Thus, consider a dataset D and a number k of elements that forms such object, an upper bound for candidate combinations of elements is roughly $|D|^k$.

Our main contribution in this work is identifying and modeling the problem of Efficiently Computing Geometric Composition Patterns in Big Data. In GCP, the problem of large permutations within elements of dataset ($|D|^{|Q|}$) produces a huge search space to answer the GCP query. Thus, we suggest reducing the processing time by cutting both the sizes of $|D|$ and $|Q|$. In order to tackle the above challenges, we discuss some novel techniques including **pre-processing the dataset** and **pre-processing the query**. Once the query is processed and the dataset has been preprocessed, the GCP process applies a composition and a neighboring elements filtering steps. In fact, the query elapsed-time is dominated by the former. Two classes of composition algorithms are introduced: Bucket Nested Loop (Bucket_NL) and **Matrix Multiplication**. We have implemented the complete processing strategy for **Computing GCP in Spark** and conducted a thorough experimentation.

2. Research challenges

The notion of finding objects of interest in datasets is a widespread area that has been discussed under different titles like *Object Identification* [Singla and Domingos 2005], *Graph Queries* [Zou et al. 2011], *Pattern Matching* and *Pattern Recognition* [Bishop 2006]. In **Object Recognition** (or object identification) the goal is to determine whether different

observations correspond to a query object. In other words, object recognition assumes that it is provided dataset of observations so that there is no need to compose objects to produce the candidate combinations (observations) from the dataset. Most approaches in this field are variants of the original Fellegi-Sunter [Fellegi and Sunter 1969] model, in which object recognition is viewed as a classification problem: given a vector of similarity scores between the attributes of two observations, classify it as 'match' or 'non-match'.

Pattern Recognition researches focus on the identifying patterns and regularities in data. Graphs are commonly used in pattern recognition due to its flexibility in working with structural geometric and relational descriptions for concepts, such as pixels, predicates, and objects [Jolion 2001, Conte et al. 2004]. In this way, problems are commonly mapped as a graph query problem, such as subgraph search, shortest-path query, reachability verification, and pattern match. Among these, pattern match query is well related to our work.

When comparing GCP with **Graph Pattern Matching**, GCP is more general, which makes it applicable to many areas of Big Data and not just over graph databases. Another main difference is related to the search space. In graph databases, they find structures in a defined graph, which commonly reduces the size of permutations within elements of dataset. For example, in [Zou et al. 2011], the $|V(G)|$ is about 10K to 100K vertices in their experiments. In [Tong et al. 2007], for their experimental results, they use graphs with about 356K nodes and 1.9M edges. In GCP, the order of matching elements does matter and we have the problem of large permutations within elements in the dataset. This problem usually does not occur in graph queries, since the number of permutations of elements is much smaller due to the structure of the graph. Table 1 summarizes the major characteristics of the above works with GCP.

Tabela 1. Comparison of Approaches

Approach Criteria	Object Composition	Problem of large permutations
GCP	Unconstrained	$O(\frac{\|D\|!}{(\|D\|-\|Q\|)!}) \approx \|D\|^{\|Q\|}$
Graph Pattern Matching	Constrained	$O(\|D\| * \|E\| + \|E_q\| * \|D\|^2 \|Q\| * \|D\|) \approx \|D\|^3$
Object Recognition	No	O(1)

3. GCP Processing

In this section, we introduce the GCP processing strategy. It involves producing an algebraic expression that can be mapped to a traditional iterator based [Graefe 1990] query execution operator plan or to a job in a Big data dataflow system, such as Apache Spark.

In a GCP scenario, such as the Einstein cross, the dataset *D* presents a schema $D = (x, y, atr_1, \ldots, atr_n)$, where attributes x, y are spatial coordinates. Similarly, Q_k holds elements with schema $Q = (x, y, atr_1, \ldots, atr_n)$. The Euclidean distances among elements in Q_k induces its shape. Let us consider a $Q\prime = (q_0, < q_1, d_1 >, < q_2, d_2 > , \ldots, < q_k, d_k >)$, where q_0 corresponds to an anchor element in Q_k and d_i is the distance between a query element q_i and q_0. Such description is depicted in Figure 2.b.

The processing of GCP includes three main steps. (1) Query Preprocessing, (2) Dataset Preprocessing, (3) Composing dataset elements and its neighbors, according to distance constraints.

3.1. Query Preprocessing

In GCP, the number of possible compositions increases exponentially with the query size k, ($N \approx |D|^k$). One may intuitively suggest reducing the size k of a GCP in order to save computation. As it turns out, elements in a full query Q_k may induce redundant constraints, specially those located very close to each other. In this context, $Q_{k\prime}$ can be build from subset M of elements of Q_k that only includes elements that are candidates for defining the geometric shape induced by $Q_{k\prime}$ ($k\prime \leq k$). As an example, $Q_{k\prime}$ may only include elements that are at a certain distance apart. Once $Q_{k\prime}$ has been fixed, an anchor element q_0 is picked and pairwise distances, from it to every remaining element $Q_{k\prime} \setminus q_0$ are computed.

3.2. Dataset Preprocessing

The second opportunity to reduce the complexity of GCP is to reduce the size of D. Additionally, we want to look for elements to build compositions that are candidates for producing shapes geometrically close to that of Q_k. The element q_0 from Q_k becomes a key to such reduction. It enables to fix an anchor for building compositions both with respect to attribute values and to shape constraints. Regarding the former, q_0 reduces the size of D to $|\sigma_{f(e_i)}(D)|$. In other words, we only test for compositions that hold a similar anchor element as q_0 in Q_k. Secondly, as we scan D, looking for anchor elements, we store each element in a PH_Tree [Zäschke et al. 2014]. The latter is used in the sequel to search for candidate elements in the neighborhood of selected e_i within a radius ρ, corresponding to the distance of the furthest query element q_i to q_0 plus ϵ, as depicted in Figure 2.a. The possible candidate solutions having e as anchor are within this set.

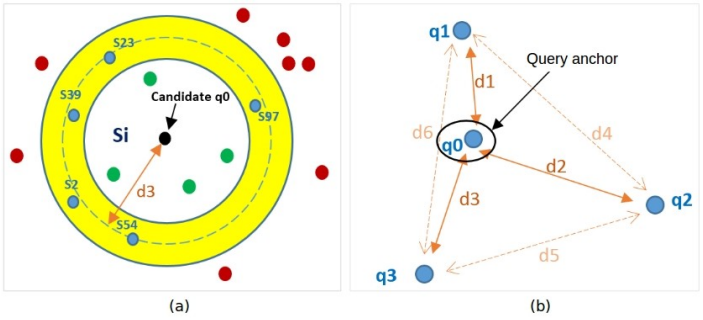


Figura 2. (a) candidate anchor and neighboring ring elements and (b) geometric query

The GCP based on an anchor e includes the neighbors within radius ρ whose distances to e match the distance d_i of some query element in Q_k. For a GCP with k query elements, we produce $k-1$ buckets holding neighbors of anchor e. The matching candidates are the sets of k-1 neighbors of e with one element from each bucket and having pairwise distances matching the corresponding pairwise distances of elements in

Q_k. The pre-processing of D produces an intermediary relation $D\prime$, substantially smaller than D, with schema $D\prime = (e : Dom, list\ of\ neighbors < n_k, d_k >)$, where e refers to an anchor element in D, and n_k is a neighbor of e in D with distance less than the largest distance (q_i, q_0), for all q_i in Q_k. An interesting side-effect of computing $D\prime$ is that it fosters the parallelization of the GCP processing by enabling a balanced distribution of $D\prime$ tuples over a cluster of machines to be evaluated by a Big Data processing framework, such as Spark.

3.3. Composition Algorithms

We have implemented the GCP execution as a Spark job running on the preprocessed dataset having each anchor with its set of neighbors. The job evaluates the efficiency of the composition algorithms, once the attribute value selection function presents insignificant costs. As each dataset entry includes the anchor element and its neighbors, distribution becomes straight forward and can benefit from the balanced blocking of HDFS [Shvachko et al. 2010]. Moreover, the composition algorithms are implemented as *Map* functions in the Spark framework. The experiment involves running a GCP based on the Einstein cross elements and assessing the elapsed-time of different composition algorithms. The system has been deployed at the GRID5000 Cluster[2] consisting of 30 Shared-Noting Machines in 2 sites with about 264 running cores. The results of the execution can be seen in Figure 3.

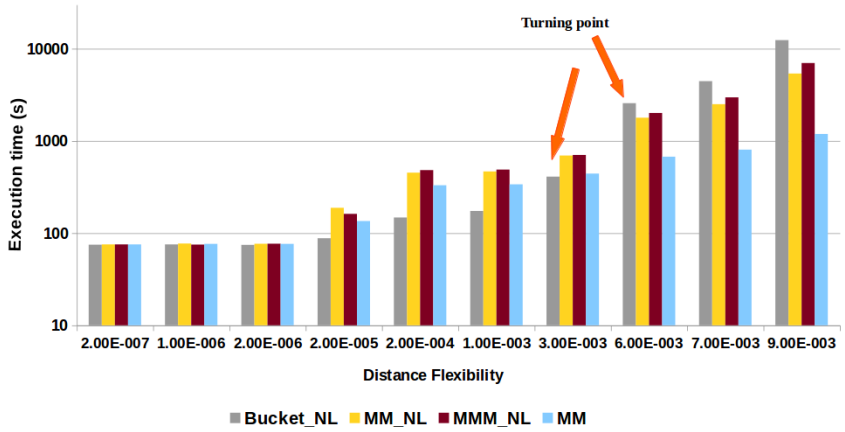


Figura 3. Effect of Distance Flexibility over composition algorithms

Our observations conclude that query modification is tough as it may affect query answering quality. Moreover, the Bucket_NL [3] composition algorithm may lead to performance gains of up to 240% for queries with low to medium distance flexibility (less than $6 * 10^{-}3$) . For larger flexibility, the advantage of quickly identifying non productive query elements make matrix multiplication based strategy to beat Bucket_NL by up to

[2]Experiments presented in this section were carried out using the Grid'5000 testbed, supported by a scientific interest group hosted by Inria and including CNRS, RENATER and several Universities as well as other organizations (see https://www.grid5000.fr)

[3]The complete explanation of the algorithms with their algorithms can be found in chapter 5 of the dissertation

45.6% (the turning point in Figure 3). Finally, in the context of larger shape flexibility, the matrix multiplication approach to answer existential constellation queries exceeds all the others with gains of up to 82% with respect to Bucket_NL.

Referências

Bishop, C. M. (2006). Pattern recognition and machine learning. *Springer*, page 7.

Brown, A. R. (2004). Interpretation of three-dimensional seismic data.

Ciarlini, A., Porto, F., Khatibi, A., and Dias, J. (2015). Methods and apparatus for parallel evaluation of pattern queries over large n-dimensional datasets to identify features of interest.

Conte, D., Foggia, P., Sansone, C., and Vento, M. (2004). Thirty years of graph matching in pattern recognition. *International Journal of Pattern Recognition and Artificial Intelligence Vol. 18*.

Einstein, A. (2015). Relativity: The special and the general theory.

Fellegi, I. and Sunter, A. (1969). A theory for record linkage. *American Statistical Association*, pages 1183–1210.

Graefe, G. (1990). Encapsulation of parallelism in the volcano query processing system. In *Proceedings of the 1990 ACM SIGMOD International Conference on Management of Data*, SIGMOD '90, pages 102–111, New York, NY, USA. ACM.

Jolion, J. (2001). Graph matching : what are we really talking about. *Proceedings of the 3rd IAPR Workshop on Graph-Based Representations in Pattern Recognition*.

NASA's Hubble Space Telescope (1990). First ESA Faint Object Camera Science Images the Gravitational Lens G2237 + 0305.

Overbye, D. (2015). Astronomers observe supernova and find they're watching reruns. New York Times, USA.

Sethian, J. (2006). Seismic velocity estimation.

Shvachko, K., Kuang, H., Radia, S., and Chansler, R. (2010). The hadoop distributed file system. In *2010 IEEE 26th Symposium on Mass Storage Systems and Technologies (MSST)*, pages 1–10.

Singla, P. and Domingos, P. (2005). Object identification with attribute-mediated dependences. *Proceedings of the 9th European conference on Principles and Practice of Knowledge Discovery in Databases*.

Tong, H., Gallagher, B., Faloutsos, C., and EliassiRad, T. (2007). Fast best-effort pattern matching in large attributed graphs. *Proceedings of the 13th ACM SIGKDD international conference on Knowledge discovery and data mining*, pages 737–746.

Zäschke, T., Zimmerli, C., and Norrie, M. C. (2014). The ph-tree: A space-efficient storage structure and multi-dimensional index. In *Proceedings of the 2014 ACM SIGMOD International Conference on Management of Data*, SIGMOD '14, pages 397–408, New York, NY, USA. ACM.

Zou, L., Chen, L., Özsu, M. T., and Zhao, D. (2011). Answering pattern match queries in large graph databases via graph embedding. *The VLDB Journal*, 21:97–120.

paper:CTD6

Join Operators for Asymmetric Media*

Neusa Liberato Evangelista[1], José de Aguiar Moraes Filho[1], Angelo Brayner[2]

[1]University of Fortaleza (UNIFOR)
Department of Computer Science (PPGIA), Fortaleza, CE, Brazil

[2]Federal University of Ceará (UFC)
Department of Computer Science, Fortaleza, CE, Brazil

Abstract. *Nowadays the use of Solid State Drive (SSD) is a reality for storing large databases. SSDs are capable to provide random access time up to three orders of magnitude lower than that delivered by magnetic hard disk drives (HDD). Nonetheless, SSDs presents time asymmetry for executing read/write operations, since write operations consumes more time than read operations. Moreover, SSD lifetime is determined by the amount of write operations on it. Such features pose challenges on database technology, for existing database management systems (DBMS) have been designed assuming that databases are stored on HDDs, in which there is no read/write asymmetry and the lifetime is independent of the number of write operations. The join operation is a query operator which requires the highest amount of secondary memory accesses (read/write operations) for being executed. This research presents new join operators namely, Bt-Join and Dict-Join. Their key goal is to minimize the amount of write operations during the execution of a join operation while keeping low response times. Bt-Join and Dict-Join have been empirically compared to FlashJoin and Hybrid-hash join. The results show that the proposed join operators can be at least 30% faster than FlashJoin and reduces significantly the number of write operations.*
Keywords: Databases, Solid State Memory, Query Operator, Join Algorithm, Materialization Strategies, Join index.

1. Introduction

Magnetic Hard Disk Drives (HDDs) have been, for decades, the main storage media for persisting data. Nonetheless, while the speed of processors has exponentially raised, the Input/Output Operations Per Second (IOPS) rates afforded by magnetic Hard Disk Drive (HDD)s has only increased marginally. Therefore, there is a significant gap between the amount of time to access data in HDDs and the time processors may consume data. This has been the main motivation for the development of asymmetric media. Asymmetric media presents no mechanical parts and a write operation consumes more time and energy than a read operation. Observe that HDDs may be considered symmetric storage device, since the time for executing read and write operations is practically the same. Solid state memory devices are an example of asymmetric media.

Database Management System (DBMS) were designed presupposing the usage of HDDs for storing data. DBMS components (e.g., query engine and buffer manager) have

*Dissertation text available at `http://uolp.unifor.br/oul/ObraBdtdSiteTrazer.do?method=trazer&obraCodigo=99890&programaCodigo=83&ns=true#`.
Advisor:José de Aguiar Moraes Filho. Co-advisor: Angelo Brayner

been optimized based on performance characteristics of HDDs. Simply replacing HDDs by faster SSDs may yield DBMS performance gains. However, such a strategy does not fully exploit benefits provided by SSDs. The asymmetric read/write latency and the lifetime dependent on the amount of write operations pose challenges to database technology. Write-intensive components of database systems (e.g., query engine and logging components) may negatively impact write-operation bandwidth SSDs.

Regarding query engine, join operator requires the highest amount of accesses (read and write operations) to secondary memory. For instance, to process a grace hash join operator between tables R and S, the query engine has to access the secondary memory $2(P_R + P_S)$ times to build the partitions of R and S to be used during the probe phase, where P_R and P_S represent the size (in pages) of R and S, respectively. From the $2(P_R + P_S)$ disk accesses, there are $P_R + P_S$ write operations to store the partitions on secondary memory. Taking into account that a SSD is being used, such a number of writes may impair the media lifetime and performance of join operator. Therefore, a SSD-aware join operator should be built taking into account the read-write asymmetry of asymmetric devices in order to improve the performance of database systems running on SSDs. For that reason, some SSD-aware join operators have been published - RARE join ([Shah et al. 2008]) and FlashJoin ([Tsirogiannis et al. 2009]), for example.

This research proposes two SSD-aware join operators, namely *Bt-Join* and *Dict-Join* which are able to use different materialization strategies. *Bt-Join* and *Dict-Join* join operators have empirically been evaluated and compared against published SSD-aware join algorithms. We highlight the following contributions: *(i)* design and implement new SSD-aware join algorithms; (ii) propose a new materialization strategy, called MIX Materialization Strategy (MIX), and *(iii)* design a flexible join index structure which accommodates different materialization strategies and enables a reduction of writes in SSD.

2. *Bt-Join* and *Dict-Join*: new SSD-aware Join Operators

Bt-Join and *Dict-Join* use the concept of *Join Result by Value* (for short, JRV). JRV has been conceived as a structure to collect in-memory as much information as possible to process a join, enabling, thus, a decrease of number of writes in SSD. Different from conventional join index structure [Valduriez 1987], JRV is built during the execution of a given join operation and each JRV entry is composed of a join attribute value and a rowIDList, which is a list of corresponding row IDs of the join operands. Additionally, it presents an optional component, denoted extension, to be filled depending on the materialization strategy used (see Figure 1).

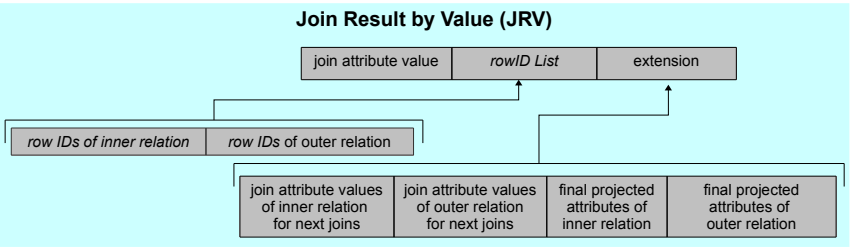


Figure 1. Structure of a JRV entry

2.1. Bt-Join

Bt-Join ([Evangelista et al. 2015]) implements the concept of JRV as an extended version of the B^+-tree ([Graefe 2011]), called B^{+Bt}-tree. The extension takes place in leaf nodes in which each leaf node contains just join attribute values and a pointer to its respective *JRV entry*. *Bt-Join* executes a join operation in two phases. Figure 2 illustrates the *Bt-Join* process for a $R \bowtie S$ join through a simplified B^{+Bt}-tree. In the first phase - *Tree-build phase* - the B^{+Bt}-tree is built upon the inner relation (S) and, for each read tuple, the join attribute value is inserted in the B^{+Bt}-tree and an corresponding *JRV entry* is created with the tuple information. In the second phase - *Join phase* - the outer relation (R) is read and, for each tuple, a search for the join attribute value is done in B^{+Bt}-tree. If the join attribute value is present in the B^{+Bt}-tree, the correspondent *JRV entry* is updated to receive the tuple information of outer relation. If value is not found, the tuple of outer relation is discarded.

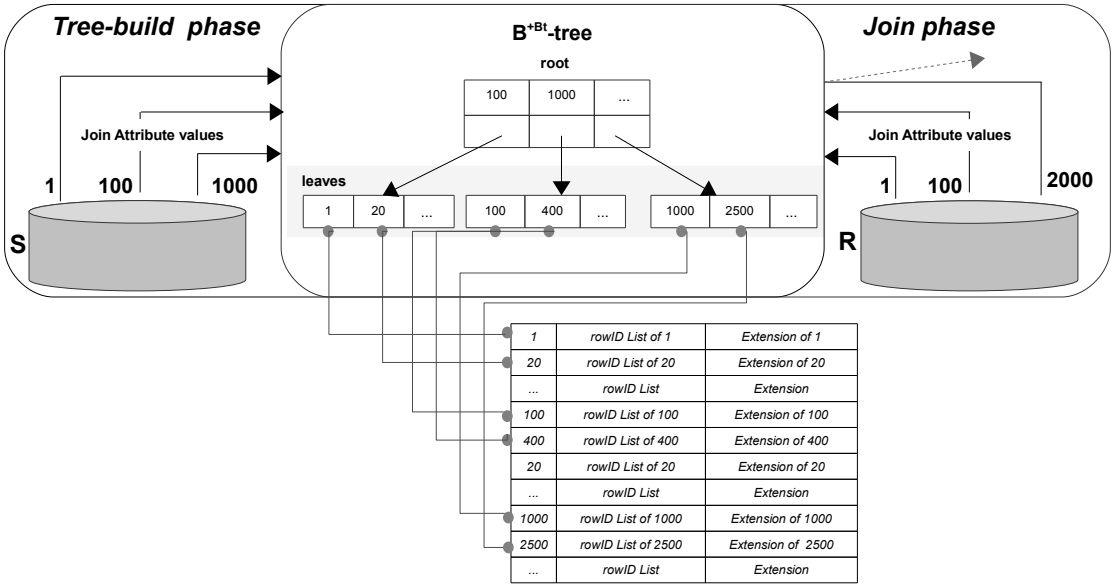


Figure 2. *Bt-Join* phases.

2.2. Dict-Join

Dict-Join [Evangelista et al. 2017] exploits the notion of dictionary to implement the JRV concept. Dictionary is a generic and abstract concept which encompasses any structure which maps keys to values. Every key is associated with at most one value and any non null object can be used as a key and as a value. Combining the dictionary concept with the fast lookup of a hash structure, our dictionary structure, called *Dict-Struct*, maps a hash value to a list of tuple information. Therefore, *Dict-Struct* entry is composed of a *Dict key*, which is determined by applying a hash function to the join attribute value, and a *Dict value*, which is a list of *JRV* entries.

The *Dict-Join* has two phases, each one deals with one of the relations involved in the join operation. Figure 3 illustrates the *Dict-Join* process for a $R \bowtie S$ join. In the first phase, denoted *Dictionary-build phase*, the *Dict-Struct* is built upon the inner relation (S) and, for each tuple read, The function h is applied to the join attribute value to determine

the *Dict-Struct* entry in which the corresponding tuple information (JRV entries) should be allocated. In the second phase, called *Join phase*, the outer relation (R) is read and, for each tuple, the same hash function used in *Dictionary-build phase* is applied to the join attribute value, and if the *entry* with *Dict key* equal to the obtained hash value exists and this join attribute value is found in it, the *entry* is updated to receive the respective *EJI entry*. If *entry* does not exist or the join attribute value is not found into the existing *entry*, the tuple of outer relation is discarded.

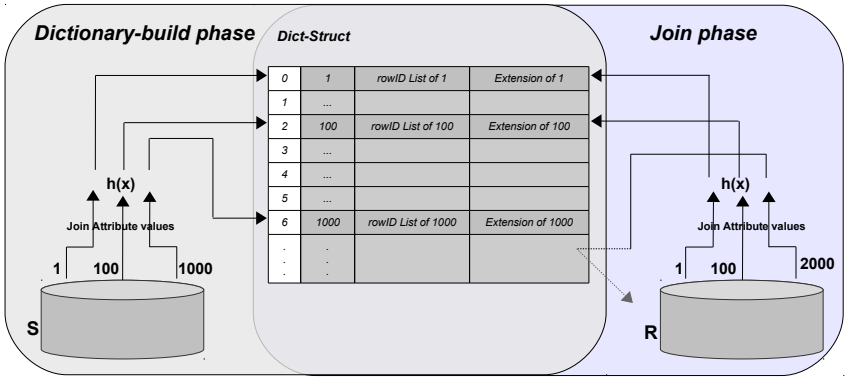


Figure 3. *Dict-Join* phases.

Besides the Early Materialization Strategy (EMS) and Late Materialization Strategy (LMS) ([Abadi et al. 2007]), for both join operators proposed, we devise a new materialization strategy, named MIX Materialization Strategy (MIX). The idea is to combine the best of the two existing strategies. Thus, MIX uses EMS for join attributes and LMS for final projection attributes. MIX may represent a gain in the plan execution time of *n-way join* queries because no rereads are performed for join operations. Furthermore, it reduces the size of intermediate results, since final projection attributes are not stored. In other words, MIX represents a trade-off between memory size needed to intermediate results and the processing costs for reread operations. Whenever the selectivity decreases along the query execution, a small number of tuples is needed to be reread for accessing final projection attributes. Such a property represents another benefit of implementing the MIX strategy.

Bt-Join and *Dict-Join* algorithms are available in the dissertation text (Chapter 6 and Chapter 7, respectively) and all materialization strategies, for each algorithm, are depicted through running examples (see [Evangelista 2016]).

3. Experimental Setup

We have comparatively evaluated our join operators against our implementations of FlashJoin and Hybdrid-hash join using the same TPCH factor 10 workload. We have elected 5 queries from TPCH and used 2 synthetic queries distributed in 2 classes: 2-way and n-way join queries. The chosen queries represent relevant join scenarios in a decision support workload as explained in the dissertation text ([Evangelista 2016], Chapter 9). The performance has been measured in terms of response time and number of writes in SSD media. Each join operator has been assessed limiting the memory space available for join in 50MB, 100MB and 200MB. For *Bt-Join* and *Dict-Join*, the three materialization strategies (LMS, EMS and MIX) have been experimented. FlashJoin and Hybrid-hash

join uses only LMS and EMS, respectively. For B^{+Bt}-trees, we have defined the maximum height of 4 for all utilized queries. For space restriction, we show only results of some TPCH queries. The results and SQL expressions for all queries can be found in [Evangelista 2016].

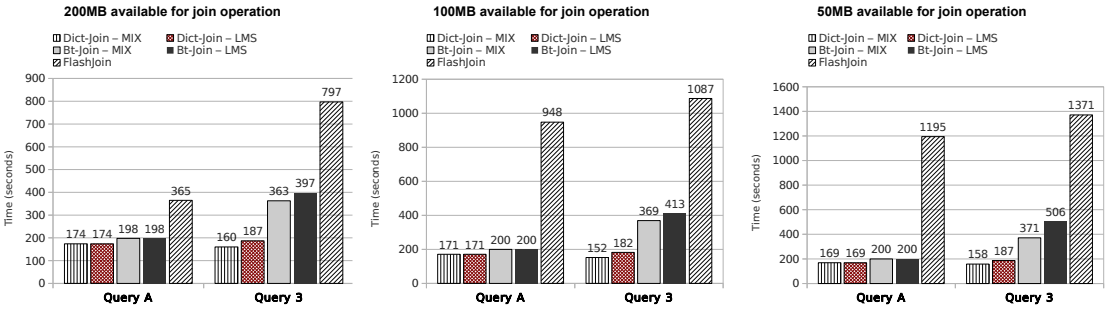


Figure 4. Response time comparison

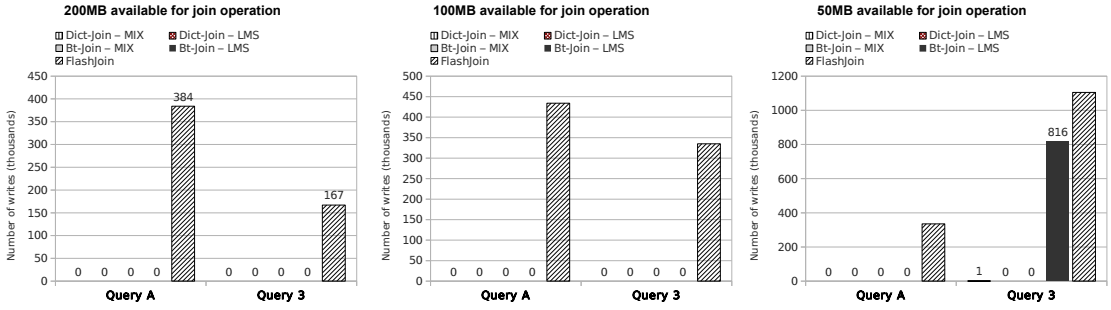


Figure 5. Number of write operations

TPCH's *Query A* (select o_orderkey, o_orderdate, l_extendedprice, l_discount from orders, lineitem where lineitem.l_orderkey = orders.o_orderkey and orders.o_orderdate $\geq$ 1993-10-01 and orders.o_orderdate $<$ 1994-01-01 and lineitem.l_returnflag=R) is a 2-way join involving the two largest tables in TPCH . The usage of MIX in *Query A* works exactly as LMS, once a second join does not exist. On the other hand, in TPCH's *Query 3* (select o_orderkey, l_extendedprice, l_discount, o_orderdate, o_shippriority from customer, orders, lineitem where order.o_custkey = customer.c_custkey and lineitem.o_orderkey = orders.orderkey and orders.o_orderdate $<$ 1995-03-15 and lineitem.l_shipdate $>$ 1995-03-15 and customer.c_mktsegment=BUILDING), a N-way join with the largest tables in TPCH, the best response times have been reached with MIX strategy. Our proposal of a MIX strategy and its application in this *N-way join* query have been beneficial because it has saved the rereads among join operations.

For all memory configurations, ours algorithms have been faster than FlashJoin (see Figure 4). Considering the response times for 200MB and using LMS strategy - the configuration with smaller differences - for *Query A*, the *Dict-Join* LMS has reached the response time of 174 seconds and *Bt-Join* LMS has reached 198 seconds against 365 seconds of FlashJoin, but for 50MB, FlashJoin has been seven times slower than *Dict-Join* LMS. For *Query 3*, *Dict-Join* was 23% of FlashJoin response time (187 seconds of *Dict-Join* against 797 seconds of FlashJoin) and *Bt-Join* response time was 50% of FlashJoin response time (310 seconds of *Bt-Join* against 797 seconds of FlashJoin).

Regarding the number of writes (see Figure 5), for *Query A*, our algorithms have

performed no writes and for *Query 3*, *Dict-Join* performed less than one thousand writes while *Bt-Join* performed 816,222 writes with LMS when only 50MB is available for join.

For the scenario of chosen queries, i.e., scenarios with low selectivity factors, our proposed algorithms have reached faster response times and have performed less writes than that of competitors. These goals have been reached at the expense of additional reads performed when LMS takes place. Additionally, compared to FlashJoin, a well-known join operator proposed to be deployed in SSDs, the results achieved by our algorithms for experimented queries are better than those of FlashJoin.

4. Conclusion

We have designed two SSD-aware join operators, namely *Bt-Join* and *Dict-Struct* and empirically compared them to published SSD-join algorithm. The empirical result seems to show that the use of a unified structure, B^{+Bt}-tree in *Bt-Join* and *Dict-Struct* in *Dict-Join*, which gathers the input of both source relations in a join, allows more intermediate result to be kept in main memory, consequently reducing the occurrence of writes and query response time.

References

Abadi, D. J., Myers, D. S., DeWitt, D. J., and Madden, S. R. (2007). Materialization strategies in a column-oriented dbms. In *Proceeding of the 2007 IEEE 23rd International Conference on Data Engineering*, ICDE '07, pages 466–475.

Evangelista, N. L. (2016). Join operators for asymmetric media. master thesis, University of Fortaleza, Washington Soares, 1321. available online at `http://uolp.unifor.br/oul/ObraBdtdSiteTrazer.do?method=trazer&obraCodigo=99890&programaCodigo=83&ns=true`.

Evangelista, N. L., de Aguiar M. Filho, J., and Brayner, A. (2017). Dict-join: A join operator for asymmetric storage device. *Information Systems*, 69. submitted to publication.

Evangelista, N. L., de Aguiar M. Filho, J., Brayner, A., and Alencar, N. (2015). Bt-join: A join operator for asymmetric storage device. In *Proceedings of the 30th Annual ACM Symposium on Applied Computing*, SAC '15, pages 988–993, New York, NY, USA. ACM.

Graefe, G. (2011). Modern b-tree techniques. *Found. Trends databases*, 3(4):203–402.

Shah, M. A., Harizopoulos, S., Wiener, J. L., and Graefe, G. (2008). Fast scans and joins using flash drives. In *Proceedings of the 4th International Workshop on Data Management on New Hardware*, DaMoN '08, pages 17–24, New York, NY, USA. ACM.

Tsirogiannis, D., Harizopoulos, S., Shah, M. A., Wiener, J. L., and Graefe, G. (2009). Query processing techniques for solid state drives. In *Proceedings of the 2009 ACM SIGMOD International Conference on Management of Data*, SIGMOD '09, pages 59–72, New York, NY, USA. ACM.

Valduriez, P. (1987). Join indices. *ACM Trans. Database Syst.*, 12(2):218–246.

Parallel Execution of Workflows driven by Distributed Database Techniques

Renan Souza, Marta Mattoso

COPPE – Universidade Federal do Rio de Janeiro (UFRJ)

{renanfs,marta}@cos.ufrj.br

Abstract. *Many-Task Computing (MTC) workflow executions involve thousands of parallel tasks that consume and produce large amounts of data and are scheduled on multiple nodes in a large HPC cluster. A complete run may last for weeks. Users have to analyze and steer the dataflow at runtime. This introduces several challenges for efficient data management without jeopardizing performance. This dissertation combines distributed data management techniques (ACID transactions, concurrency control, and database design) to propose a scalable solution for MTC workflows. Domain data, dataflow provenance, and workflow execution data are managed together in an in-memory distributed DBMS. As a result, a distributed scheduling via transactions in this database attains high scalability in a 1,008-cores HPC cluster, while maintaining runtime data analytical capabilities.*

1. Introduction

Large-scale scientific computations in a wide variety of domains are often modeled as Many-Task Computing (MTC) workflows, with thousands of parallel tasks that run on a High Performance Computing (HPC) cluster. The tasks produce data elements to be consumed by other tasks in a coherent dataflow. A complete workflow execution may last for days in a large cluster and process tera or petabytes of complex scientific data. Meanwhile, users cannot wait for the workflow to finish so their result data analysis can start. They need to correlate input with output, check correctness of their hypotheses, modify simulation parameters, debug, visualize the flowing data elements, and steer the computation at runtime, maintaining high parallel efficiency of the HPC simulation. Since these processes are data-centric, an efficient data management that allows for data parallelism in MTC and runtime data analysis without jeopardizing performance is of utmost importance.

Parallel Scientific Workflow Management Systems (SWMSs) have been employed to orchestrate workflow executions in HPC [1]. Pegasus [3] and Swift [16] are two well-known SWMSs and are the most scalable ones. To allow for result data analysis, they collect distributed data provenance and store in multiple typically unstructured log files, which are much harder to query at runtime. Alternatively, they load the log files (through an ETL process) into a database for *post-mortem* analytical queries. Additionally, task scheduling data (information about each task, which node each task ran, CPU and memory consumed by each task) are managed completely separated from dataflow provenance and domain data stores. This highly limits analytical capabilities because users need to analyze the data integrating domain, execution, and provenance [10,14]. To cope with this, a data-oriented SWMS solution, called Chiron, was built [6]. Chiron adopts a DBMS to manage its scheduling data by updating it while processing parallel tasks and storing provenance data in this same database, which we call Work-

flow Database (wf-Database). Thus, users can query the same database being used by the SWMS scheduler, which has been shown to be very beneficial [7,13].

Nevertheless, all this runtime analytical support comes with a price. This data-oriented SWMS solution relies on a centralized task scheduling data management, using a master-workers design where the centralized master node is the only able to access the centralized DBMS. This introduces a significant performance overhead at the master node [14] highly limiting the system scalability to about 300 cores. Additionally, two single points of failure are introduced: the master node and the centralized DBMS, limiting the solution's fault tolerance. Therefore, we are not aware of a SWMS solution that both allows for analytical queries integrating domain data, provenance, and execution at runtime and is highly scalable.

In this dissertation, we make extensive use of distributed data management techniques (ACID transactions, concurrency control, and database design) to propose SchalaDB: a scalable data-oriented distributed task scheduling solution for SWMSs. In SchalaDB, all workers directly query and update the wf-Database through SQL. There is no centralized master to which workers need to communicate via message passing during scheduling. To accommodate multiple workers concurrently querying the wf-Database, the solution uses an in-memory distributed DBMS (DDBMS) with distributed ACID transactions. We implemented SchalaDB by completely redesigning Chiron's traditional centralized scheduling, and we call it d-Chiron. We evaluated it using synthetic benchmarks, and real case studies in oil and gas and bioinformatics in Grid5000 (www.grid5000.fr) clusters with up to 1,008 cores. As a result, this is the first SWMS that achieves high scalability in an HPC cluster of this size while maintaining support for rich data analysis at runtime. These are the main contributions of this dissertation:

- A scalable design for MTC workflow scheduling driven by a DDBMS. We specify how task scheduling and parallel data placement are done to maximize system performance, improve availability, and reduce load imbalance.
- A concrete implementation of this design in d-Chiron SWMS and performance tests on a 1,008-cores cluster.
- For reproducibility, all executables, instructions to use d-Chiron on an HPC cluster, pre-configured workflows, and sample analytical queries are on GitHub [4].

2. A Scalable Architecture for Scheduling MTC Workflows Driven by Distributed Data Management

This dissertation aims at providing high scalability while maintaining runtime data analytical support in a SWMS. Data analysis is supported via queries in the wf-Database. It follows PROV-Wf, an entity-relationship diagram that models workflow general concepts relevant for provenance data collection. PROV-Wf is based on a W3C recommendation, which facilitates integration and queries using the representation for provenance. For example, data from the wf-Database can be published on the semantic web to be consumed by different research groups, using different SWMSs [2]. An implementation of PROV-Wf in a concrete database schema is on [4]. Since the wf-Database is continuously populated at runtime, it can potentially be queried for data generation tracking, monitoring, and other advanced data analyses [7,10,13,14]. Also, a SWMS engine can use this data-oriented approach for runtime optimizations [6] or adaptivity [8].

To allow for these features, the SWMS engine updates the wf-Database as the tasks are created, scheduled, executed, and completed. It has to store all fine-grained task-related data in the wf-Database for each task. Thus, both users and the SWMS engine have the most up-to-date possible data available for structured queries. However, in MTC, there can be thousands of parallel tasks running, each taking few seconds to minutes. When the task scheduler is centralized, there are points of contention and failure, negatively impacting performance. SchalaDB, however, has a decentralized scheduling design and uses an in-memory DDBMS. Although some DDBMSs, like MySQL Cluster, are well-known for their efficiency in processing OLTP while being able to run OLAP queries [9], their use in MTC scheduling has not been experienced before.

Using a DDBMS in an MTC scheduler has several advantages beyond ACID transactions controlling multiple concurrent updates during task scheduling. Particularly, DDBMSs that allow for ACID transactions are useful to facilitate data consistency control when multiple concurrent updates occur in the task-related data during task scheduling. Moreover, DDBMSs enable robust parallel cache memory data management. Also, there are DDBMSs that run exclusively in the cluster main memory, avoiding intense disk I/O operations. Data replication and partitioning into multiple nodes are also well studied and implemented in many DDBMSs. Considering that a DDBMS already implements most of these mechanisms usually very efficiently [9], it can be an integrant part of an SWMS architecture and alleviate the effort on developing such complex controls inside the SWMS engine's source code. In this way, the SWMS developers can focus on specific concerns related to dataflow management (*e.g.*, data dependencies between tasks) instead of implementing distributed data management algorithms and dealing with sophisticated distributed concurrency issues and contention at scheduling queues. As centralized DBMSs, DDBMS also has query interfaces through which users can query the continuously populated wf-Database. Therefore, SchalaDB controls the parallel execution of workflows with a distributed task scheduling driven by a DDBMS.

Architecture details. For execution control, the main supporting relation is the Work Queue (WQ), which has the list of tasks to be scheduled and their data. SchalaDB distributes the WQ data across *D data nodes*. The data nodes are responsible for managing the distributed data partitions playing the role of multiple masters. SchalaDB uses database drivers to implement the *connectors*. Figure 1 illustrates SchalaDB.

It is the main data provider for the SWMS engine's distributed scheduler so that worker nodes can submit queries to retrieve the data to be used in a scheduling decision and a task execution. Instead of having workers requesting tasks to a master through regular message passing, like in traditional task scheduling implementations, workers send structured queries to the DDBMS. Instead of having a master to receive the worker request; get the next ready tasks; and send them to the worker, the DDBMS uses its distributed data nodes to respond the multiple concurrent requests from the workers, diminishing contention. Regarding availability, the DDBMS can use replication to replicate all relations, including the WQ relation. Since the wf-Database stores workflow control data, input and output domain data composing the dataflow, and paths to large scientific files stored on disk [6], it is not large and replication in the main-memory is viable considering a cluster with multiple nodes, each with at least few gigabytes of RAM. If a node hosting a WQ partition fails, there is still at least one extra replica that may be utilized. To increase availability in the system, each worker may connect and

query the DDBMS via two different database connector communications: the main communication, represented by full gray lines in Figure 1 and the secondary communication by dashed gray lines. If one connector fails, workers connected to it just need to connect to their secondary database connector. In addition, the secondary supervisor node removes the single point of failure at the supervisor node. Data node's availability is outsourced to the DDBMS.

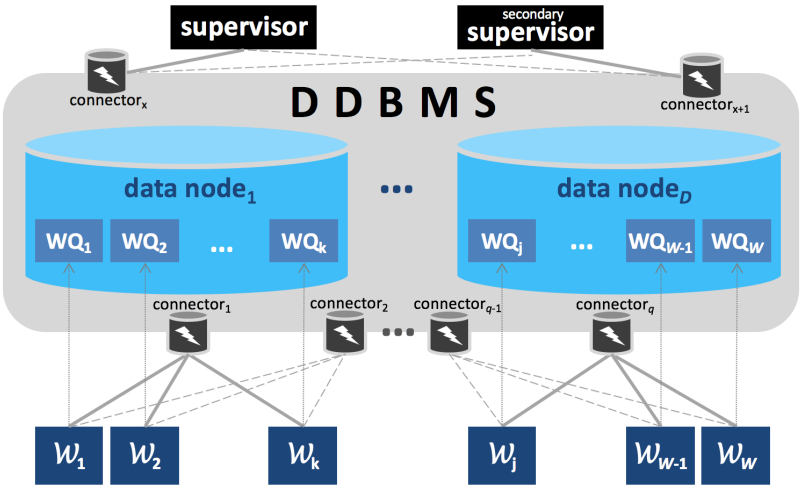


Figure 1. SchalaDB design. W workers directly accessing the DDBMS composed of D data nodes.

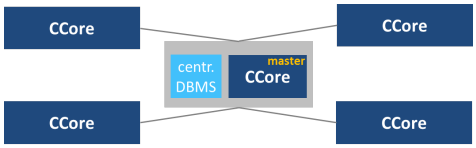


Figure 2. Chiron's centralized architecture relying on a centralized DBMS.

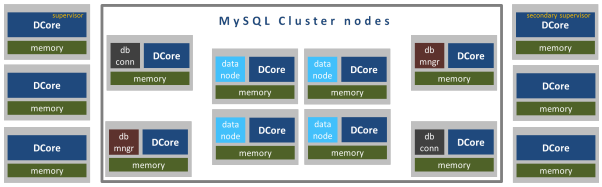


Figure 3. d-Chiron architecture. The gray boxes represent physical nodes in an HPC cluster.

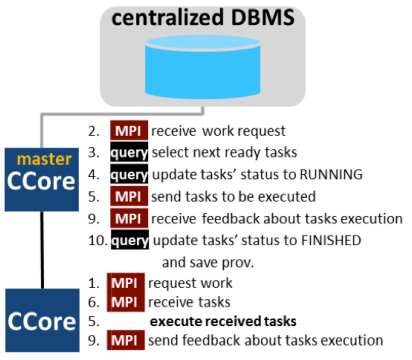


Figure 4. Centralized scheduling with a centralized DBMS.

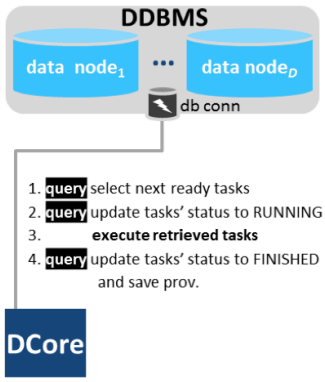


Figure 5. DDBMS-driven scheduling.

Distributed Data Design. The WQ is typically the largest data structure for the scheduler in terms of number of elements. A distributed database design requires a partitioning strategy that matches the desirable number of partitions, and the placement of the partitions [9]. To reduce load imbalance, SchalaDB distributes the WQ partitions across the data nodes. The number of partitions is equal to the number W of workers. Thus, each worker has its own WQ partition to improve data parallelism by using different memory spaces in parallel for each partition. Local processing is also improved because

task lookup for each worker goes straight to its partition instead querying a large WQ. This also reduces race condition among workers, which otherwise would be competing for the entire WQ. Each worker w_i only accesses its own WQ_i partition using queries like "select/update the next ready tasks in the WQ where $partition = WQ_i$".

With respect to implementation, after trying multiple DDBMSs, we found that MySQL Cluster would be the best fit, since SchalaDB needs OLTP for scheduling and OLAP for runtime queries. In addition, it is open-source, scalable, can run fully in cluster memory, and implements ACID transactions. Further details about why MySQL Cluster was chosen over other options are discussed in [12]. In Figure 2, we show the centralized architecture in Chiron. In Figure 3, we show how we implemented SchalaDB in d-Chiron using MySQL Cluster. Figures 4 and 5 show a comparison of centralized and distributed task scheduling driven by a centralized and distributed DBMS, respectively. We can see that a DDBMS-driven scheduling eases scheduling and reduces the overhead caused by message passing between master and workers.

3. Contributions and Concluding Remarks

Therefore, we are not aware of a SWMS solution that both allows for analytical queries integrating domain data, provenance, and execution at runtime and is highly scalable. We proposed a decentralized MTC task scheduler, which is the core of an HPC system, using distributed data management techniques aiming at high performance for a SWMS. This is the first work that frequently attains over 80% of parallel efficiency running on a 1,008 cores cluster while managing workflow data in a same database available for runtime queries. Observing the tendency of the curves plotted in a comprehensive set of performance tests [12], we could see that the solution could still scale if we had access to an even larger cluster with at least few thousands of CPU cores. In addition to benchmarking workflows, we successfully ran two real workflow case studies: one in oil and gas and other in bioinformatics domains. Finally, we show that our implementation runs at least two orders of magnitude faster than the implementation that uses a centralized data management and scheduling [12,15]. Besides the performance gains, by using SchalaDB's scheduling, a complex part of the SWMS engine source code can be outsourced to a specialized system, *i.e.*, the DDBMS. This dissertation was developed in the context of a set of published works:

- The core ideas of SchalaDB and some results in d-Chiron were presented in the prestigious ACM/IEEE Supercomputing conference as a poster [15].
- An approach to analyze performance data integrated with domain dataflow and provenance was presented in [14]. We could quantify the performance bottlenecks that a centralized scheduling was causing in Chiron SWMS. It motivated this work and won the second-best paper award in the workshop. It also derived in [10], presented in a workshop held in conjunction with ACM/IEEE Supercomputing.
- A strategy to publish data stored in the wf-Database on the semantic web using ontology, RDF, and triple stores was presented in [2], which followed a linked data publication strategy presented in [11]. It ran for best poster award. It was derived from an undergraduate dissertation that I co-supervised.
- A DDBMS-driven approach for fault tolerance in SWMSs [5]. It is part of an undergraduate dissertation that I co-supervised.

- The results of the dissertation have contributed to a new direction of research related to data reduction in scientific workflows, presented in a workshop in conjunction with Supercomputing [8]. Even though it has additional work, developed after the dissertation, we consider it a derived result from the scalability of d-Chiron.

References

[1] Atkinson, M., Gesing, S., Montagnat, J., Taylor, I. Scientific workflows: past, present and future. *FGCS*, 75:216–227, 2017.

[2] Castro, R., Souza, R., Sousa, V.S., Ocaña, K.A.C.S., Oliveira, D. de, Mattoso, M. Uma abordagem para publicação de dados de proveniência de workflows científicos na web semântica. *Simpósio Brasileiro de Banco de Dados*, 1–6, 2015.

[3] Deelman, E., Vahi, K., Juve, G., Rynge, M., Callaghan, S., Maechling, P.J., Mayani, R., Chen, W., Ferreira da Silva, R., et al. Pegasus, a workflow management system for science automation. *FGCS*, 46(C):17–35, 2015.

[4] GitHub. d-Chiron Repository. Available at: github.com/hpcdb/d-Chiron

[5] Miranda, P. *Um mecanismo de tolerância a falhas em execuções paralelas de workflows apoiadas por banco de dados*. BSc dissertation, UFRJ, 2015.

[6] Ogasawara, E., Dias, J., Oliveira, D., Porto, F., Valduriez, P., Mattoso, M. An algebraic approach for data-centric scientific workflows. *PVLDB*, 4(12):1328–1339, 2011.

[7] Oliveira, D., Costa, F., Silva, V., Ocaña, K., Mattoso, M. Debugging scientific workflows with provenance: achievements and lessons learned. *Simpósio Brasileiro de Banco de Dados*, 1–10, 2014.

[8] Oliveira, D., Ogasawara, E., Baião, F., Mattoso, M. SciCumulus: a lightweight cloud middleware to explore many task computing paradigm in scientific workflows. *IEEE Int. Conf. on Cloud Computing*, 378–385, 2010.

[9] Özsu, M.T., Valduriez, P. *Principles of distributed database systems*. 3 ed. New York, Springer, 2011.

[10] Silva, V., Neves, L., Souza, R., Coutinho, A.L.G.A., Oliveira, D. de, Mattoso, M. Integrating domain-data steering with code-profiling tools to debug data-intensive workflows. *WORKS*, 59–63, 2016.

[11] Souza, R., Cottrell, L., White, B., Campos, M.L., Mattoso, M. Linked open data publication strategies: Application in networking performance measurement data. *ASE International Conference on BigData/SocialCom/CyberSecurity*, 1–7, 2014.

[12] Souza, R. *Controlling the parallel execution of workflows relying on a distributed database*. MSc dissertation, COPPE/UFRJ, 2015.

[13] Souza, R., Silva, V., Coutinho, A.L.G.A., Valduriez, P., Mattoso, M. Online input data reduction in scientific workflows. *WORKS*, 44–53, 2016.

[14] Souza, R., Silva, V., Neves, L., De Oliveira, D., Mattoso, M. Monitoramento de desempenho usando dados de proveniência e de domínio durante a execução de aplicações científicas. *Workshop em Desempenho de Sistemas Computacionais e de Comunicação*, 1–14, 2015.

[15] Souza, R., Silva, V., Oliveira, Daniel, Valduriez, P., Lima, A.A.B., Mattoso, M. Parallel execution of workflows driven by a distributed database management system. *Poster in IEEE/ACM Supercomputing*, 1–3, 2015.

[16] Wozniak, J.M., Armstrong, T.G., Wilde, M., Katz, D.S., Lusk, E., Foster, I.T. Swift/T: large-scale application composition via distributed-memory dataflow processing. *IEEE/ACM Int. Symp. Cluster, Cloud and Grid Computing*, 95–102, 2013.

Uma Abordagem em Paralelo para Matching de Grandes Ontologias com Balanceamento de Carga

Tiago Brasileiro Araújo[1], **Orientador: Carlos Eduardo Santos Pires**[1]

[1] Departamento de Sistemas e Computação
Universidade Federal do Campina Grande (UFCG)
Caixa Postal 10.106 – 58.429-900 – Campina Grande, PB – Brasil

tiagobrasileiro@copin.ufcg.edu.br, cesp@dsc.ufcg.edu.br

Abstract. *The Ontology Matching (OM) process is applied in order to identify correspondences between concepts of two distinct ontologies. The major challenges of large ontology matching are the high execution time and the excessive consumption of computational resources. Thus, this work presents the Partition-Parallel-based Ontology Matching (PPOM) approach that partitions the input ontologies and performs the comparisons between concepts in parallel, applying the MapReduce framework. This work also proposes two techniques to minimize the problem of load imbalancing, common in parallel approaches. The experimental results indicate that the PPOM approach is scalable and improves the efficiency of the OM process.*

Resumo. *O processo de Matching de Ontologias (MO) é aplicado com o intuito de identificar correspondências entre os conceitos de duas ontologias distintas. Um dos maiores desafios do matching de grandes ontologias é o elevado tempo de execução e o excessivo consumo de recursos de computacionais. Assim, este trabalho apresenta uma abordagem para o Matching de Ontologias baseado em Particionamento e Paralelismo (MOPP) que particiona as ontologias de entrada e executa as comparações entre conceitos em paralelo, a partir da aplicação do framework MapReduce. O trabalho propõe ainda duas técnicas para amenizar o problema de desbalanceamento de carga, comum em abordagens em paralelo. Os resultados experimentais indicam que a abordagem MOPP é escalável e melhora a eficiência do processo de MO.*

1. Introdução

Atualmente, existe uma vasta quantidade de informações relacionadas a diferentes áreas do conhecimento, por exemplo, saúde, clima e ciência. Uma maneira de representar o conhecimento de uma determinada área é por meio da aplicação de ontologias (de domínio) [Euzenat and Shvaiko 2013]. Em uma ontologia, a informação é representada como um conjunto de conceitos, instâncias e relações entre conceitos (detalhado no *Capítulo 1*).

Uma vez que as ontologias podem apresentar sobreposição de conteúdo, torna-se necessária a identificação (semi-) automática de semelhanças entre conceitos como suporte para diferentes tarefas, por exemplo, integração de dados e ligação de dados. O processo que determina os pares de conceitos similares (correspondências) entre ontologias é chamado Matching de Ontologias (MO) [Euzenat and Shvaiko 2013]. No MO tradicional, duas ontologias O_1 e O_2 (normalmente modelando o mesmo ou domínios

semelhantes) são recebidas como entrada. Posteriormente, comparações entre pares de conceitos são realizadas, seguindo o produto cartesiano, ou seja, todos os conceitos da ontologia O_1 são comparados com todos os conceitos de O_2.

Para calcular a similaridade entre dois conceitos, múltiplos algoritmos de correspondência (*matchers*) são aplicados. Os *matchers* exploram propriedades lexicais, estruturais ou semânticas dos conceitos usando diferentes funções de similaridade [Euzenat and Shvaiko 2013]. Portanto, os *matchers* tendem a apresentar complexidades computacionais diferentes (conforme formalizado no *Capítulo 4.1*). O valor de similaridade, que é produzido por cada *matcher*, varia entre 0 (sem similaridade) e 1 (semelhança total). Para cada par de conceitos, o valor de similaridade é gerado por meio de medidas de agregação (por exemplo, média e média ponderada) dos valores de similaridade parcial produzidos pelos *matchers*. Se o valor de similaridade é maior que um limiar de similaridade, o par de conceitos é considerado como uma correspondência.

Ao lidar com grandes ontologias, o MO tradicional (produto cartesiano) é lento e requer uma elevada quantidade de recursos computacionais [Amin et al. 2015]. Portanto, para otimizar o processo de MO, podem ser aplicadas abordagens para reduzir o espaço de busca (isto é, minimizar a quantidade de comparações a serem realizadas) e/ou executar o MO em paralelo. Para minimizar o espaço de busca de correspondência, cada ontologia de entrada (O_1 e O_2) é individualmente particionada e os conceitos são divididos em sub-ontologias (partições), de maneira que não há sobreposição entre as sub-ontologias. Depois disso, as sub-ontologias de O_1 e O_2 com certo grau de similaridade são pareadas e os conceitos dentro de cada par de sub-ontologias são combinados seguindo o produto cartesiano. Por sua vez, as abordagens de MO em paralelo visam reduzir o tempo de execução distribuindo os pares de conceitos entre os vários recursos (por exemplo, computadores ou máquinas virtuais) de uma infra-estrutura computacional. Neste sentido, o presente trabalho propõe uma abordagem para o Matching de Ontologias baseado em Particionamento e Paralelismo (MOPP) com o objetivo de reduzir a quantidade de comparações entre conceitos (quando comparado com o produto cartesiano) e paralelizar a comparação dos pares de conceitos por meio da aplicação do *framework* MapReduce, buscando reduzir o tempo de execução do processo de MO como um todo.

Particularmente, um dos principais problemas enfrentados pelas abordagens de MO em paralelo (com a utilização do MapReduce) é o desbalanceamento de carga. O desbalanceamento de carga ocorre quando alguns nós executam comparações por um longo período enquanto outros nós permanecem ociosos. Esse problema influencia diretamente a eficiência do MO, uma vez que o nó mais lento domina o tempo total de execução. Uma razão para o desequilíbrio de carga é que as abordagens de MO baseadas em partições podem produzir pares de sub-ontologias que geram diferentes quantidades de pares de conceitos a serem comparados. Assim, se a distribuição de carga de trabalho entre os nós é baseada na quantidade de pares de sub-ontologias, um nó provavelmente receberá mais pares de conceitos que outros. Além disso, uma vez que a quantidade de comparações para cada par de conceitos varia, um nó pode receber pares de conceitos que resultam em mais comparações do que os outros nós (descritos nos *Capítulos 2.5 e 4.1*). Para endereçar o problema de balanceamento de carga o presente trabalho também propõe duas técnicas de balanceamento de carga para serem aplicadas à abordagem MOPP: básica e refinada (com baixo nível de granularidade).

2. Objetivos e Principais Contribuições

Tomando como base os principais desafios enfrentados pelo processo de OM, sobretudo no que diz respeito à eficiência, o principal objetivo deste trabalho é propor uma abordagem para o Matching de Ontologias baseado em Particionamento e Paralelismo (MOPP). Além do objetivo principal, este trabalho possui os seguintes objetivos secundários que fundamentaram a relevância da abordagem proposta: a)minimizar o problema de desbalanceamento de carga, relacionados à execução do processo de MO em paralelo; b)avaliar a abordagem proposta (MOPP) no que diz respeito ao desempenho do processo e à qualidade dos resultados (alinhamentos) gerados.

A partir dos objetivos elaborados, o presente estudo atingiu contribuições relacionadas à eficiência do processo de MO, além de contribuições no que diz respeito à técnicas de balanceamento de carga e na identificação de novos direcionamentos para pesquisas futuras. Relacionada à eficiência do processo de MO, a partir do particionamento das ontologias é possível reduzir a quantidade de comparações a serem realizadas e consequentemente reduzir o tempo de execução do processo. Com o objetivo de potencializar ainda mais a eficiência do MO, as comparações são executadas em paralelo pelos nós de uma infraestrutura computacional.

Em relação ao desbalanceamento de carga, este trabalho propôs a utilização de duas técnicas de balanceamento de carga: básica e refinada (ambas as técnicas são ilustradas no *Capítulo 4.3*). A primeira técnica baseia-se na quantidade de pares de conceitos contidos em cada par de subontologias para guiar a distribuição dos pares de conceitos entre os nós. A segunda técnica é mais robusta e refinada, uma vez que ela analisa a quantidade de comparações realizadas (pelos *matchers*) em cada par de conceitos. Com base na quantidade de comparações, a técnica refinada de balanceamento distribui uniformemente os pares de conceitos entre os nós, de forma que todos os nós executam uma quantidade semelhante de comparações.

A abordagem proposta neste trabalho foi submetida a análises experimentais envolvendo eficiência do processo de MO e a qualidade dos alinhamentos gerados, aplicando ontologias reais e sintéticas (conforme detalhado no *Capítulo 5.2*). A abordagem também foi submetida a experimentos comparativos com uma abordagem estado da arte. Além disso, foi analisada a aplicação das duas técnicas de balanceamento de carga propostas neste trabalho na abordagem MOPP. Foram criados diferentes cenários experimentais com o objetivo de analisar os desafios referentes ao desempenho do processo de MO (descritos no *Capítulo 5*).

3. Matching de Ontologias baseado em Particionamento e Paralelismo

A abordagem MOPP, apresentada nesta seção, utiliza o *framework* MapReduce (cujo fluxo de execução está descrito no *Capítulo 2.3*) para paralelizar a etapa de *matching* de subontologias (etapa ilustrada no *Capítulo 2.4.1*). Neste trabalho, foi aplicado o algoritmo PAP (*Partition, Anchor, Partition*) [Hamdi et al. 2010] para particionar as ontologias, uma vez que esse algoritmo obteve melhores resultados quando comparado com outros algoritmos de particionamento estado da arte. Resumidamente, a abordagem proposta funciona da seguinte forma (ilustrado na Figura 1): para cada par de subontologias a serem comparadas, os conceitos da subontologia menor (em termos de quantidade de conceitos) são replicados na fase de *map*. Os conceitos da subontologia maior são enviados juntamente

com os conceitos replicados da subontologia menor para serem comparados na mesma tarefa de *reduce* (fase de *reduce*). As etapas da abordagem MOPP, bem como o funcionamento das técnicas de balanceamento de carga, estão descritos e ilustrados no *Capítulo 4* da dissertação.

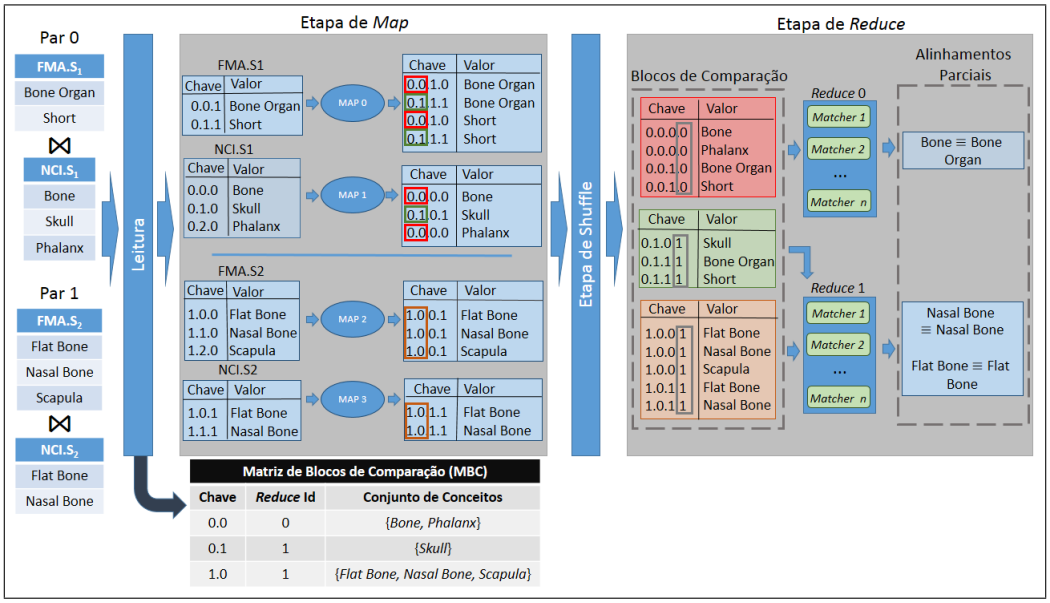


Figura 1. *Matching* dos pares de subontologias com balanceamento refinado de carga.

Etapa 1: Leitura de conceitos e Pré-processamento. Nesta etapa é realizada a leitura dos conceitos contidos nos pares de subontologias. Durante a leitura dos conceitos, é feito o cálculo da quantidade de pares de conceitos e comparações resultante de cada par de subontologias. Esta etapa é de fundamental importância para abordagem MOPP, pois tais informações são armazenadas na Matriz de Blocos de Comparação (MBC) e posteriormente utilizadas pelas duas técnicas de balanceamento de carga (básica e refinada). Inicialmente, cada par de subontologias é associado a um identificador (*par_id*). Cada conceito (tupla) recebe uma chave de identificação composta por: $\langle par\_id.conceito\_id.replica \rangle$, onde *conceito_id* é o identificador do conceito na subontologia correspondente e *replica* indica se o conceito deve ser replicado (*replica* = 1) ou não (0) na etapa de *map*.

Etapa 2: Map. Esta etapa é responsável pela definição das chaves dos conceitos, com o objetivo de guiar a formação dos blocos de comparação. As tarefas de *map* recebem os pares chave-valor $\langle chave, conceito \rangle$ originados da etapa de leitura. Os conceitos com o valor *replica* igual a 1 são replicados. Para cada conceito replicado, o *conceito_id* da chave é substituído pelo *bloco_comparação_id* do bloco comparação com o mesmo *par_id* do conceito. Além disso, as tarefas de *map* acrescentam o *reduce_id* à chave do conceito. Essa parte da chave indica para qual tarefa de *reduce* o conceito será enviado. O valor do *reduce_id* é obtido a partir da MBC (gerada na etapa anterior), com o intuito de distribuir uniformemente as comparações entre conceitos (isto é, balancear a carga de trabalho).

Etapa 3: Shuffle. Nesta etapa, os pares chave-valor $\langle chave, conceito \rangle$ são particionados, ordenados e agrupados. Os pares $\langle chave, conceito \rangle$ são agrupados e ordenados

pela chave, com o intuito de definir os blocos de comparação, tal como ilustrado na Figura 1. Os conceitos são agrupados baseados nos dois primeiros valores de suas chaves, ou seja, *par_id* e *conceito_id*. Para ordenar os conceitos em um bloco de comparação, todos os valores da chave são considerados. O último valor da chave (*reduce_id*) dos conceitos determina para qual tarefa de *reduce* os conceitos que compõem um bloco de comparação deverão ser enviados. Por fim, os pares $\langle chave, conceito \rangle$ são agrupados em cada bloco de comparação e enviados para a etapa de *reduce*.

Etapa 4: Reduce. O objetivo desta etapa é executar as comparações dos pares de conceitos contidos nos blocos de comparação. As tarefas de *reduce* recebem os blocos de comparação e, para cada bloco, realizam a comparação do subconjunto de conceitos da subontologia maior com todos os conceitos do subontologia menor. Os *matchers* executam as comparações entre os conceitos e um valor de similaridade (agregado) é retornado para cada par de conceitos. Os pares de conceitos com um valor acima do limiar de similaridade (ϕ) são considerados correspondências. Para concluir, a saída de cada tarefa de *reduce* é um conjunto de correspondências denominado alinhamento parcial.

4. Experimentos

Nesta seção são descritos os cinco experimentos que avaliam a abordagem MOPP e a aplicação das técnicas de balanceamento de carga, em uma infra-estrutura de computacional distribuída (descrito no *Capítulo 5*). Nos experimentos foram avaliadas a eficiência do processo de MO e a qualidade dos alinhamentos gerados. Os cinco experimentos abordam respectivamente as seguintes questões de pesquisa: a) Com relação ao desempenho, a abordagem MOPP é comparável às abordagens estado da arte existentes?; b) A técnica refinada de balanceamento de carga é capaz de melhorar a eficiência da abordagem MOPP?; c) Os algoritmos de particionamento reduzem o tempo de execução, sem comprometer a eficácia?; d) A técnica refinada de balanceamento de carga pode reduzir a quantidade de conceitos replicados na abordagem MOPP?; e) A abordagem proposta é escalável em um contexto onde são utilizados múltiplos matchers?

Os principais resultados obtidos em cada uma das cinco avaliações experimentais foram, respectivamente:

- Quanto ao tempo de execução, a abordagem MOPP com balanceamento refinado de carga superou as abordagens MOPP com balanceamento básico e Gross para todas as variações no número de núcleos. A abordagem MOPP com balanceamento refinado chegou a reduzir o tempo de execução em até 24% em relação à abordagem de Gross e 14% em relação à abordagem MOPP com balanceamento básico.
- A abordagem MOPP com balanceamento refinado reduziu o tempo de execução em até 87% em relação à aplicação da técnica básica. Esse resultado é reflexo do balanceamento de carga promovido pela técnica refinada, uma vez que mesmo em cenários onde alguns pares de conceitos potencialmente causariam um elevado grau de desbalanceamento de carga, a técnica consegue balancear a quantidade de comparações realizadas em cada tarefa de *reduce*.
- Com a utilização 4 nós, o tempo de execução da abordagem sem o algoritmo de particionamento é o dobro em relação ao tempo de execução da abordagem com o algoritmo de particionamento. Contudo, a aplicação do algoritmo de partici-

onamento impactou no decréscimo de 2% na eficácia dos alinhamentos gerados (mensurados a partir da métrica *F-measure*).
- No cenário onde foram utilizados 32 nós, a abordagem MOPP com a técnica refinada reduz pela metade a quantidade de conceitos replicados em relação à abordagem MOPP com a técnica básica.
- No que diz respeito a eficiência, a diferença no tempo de execução entre o *matcher Label* e a combinação dos *matcher Label* e *Annotation* tende a diminuir à medida em que o número de nós aumenta. Com apenas um nó, a diferença de tempo de execução é 3.960 segundos, com 32 nós essa diferença é reduzida para apenas 163 segundos. Quanto à eficácia dos alinhamentos, a combinação dos *matchers Label* e *Annotation* produz um aumento de 6% na *F-measure*, saindo de 65% para 71%.

5. Resultados e Conclusões

É importante destacar o caráter inovador das contribuições apresentadas neste trabalho. Primeiramente, como apresentado no *Capítulo 3* da dissertação, existem poucos trabalhos que abordam o processo de MO em paralelo. Além disso, a abordagem MOPP não apenas é executada em paralelo, como também aplica técnicas de particionamento de ontologias com o intuito de melhorar ainda mais o desempenho do processo de MO. Em segundo lugar, dentre as abordagens de MO em paralelo, a abordagem MOPP é uma das poucas que utiliza o modelo programático de computação distribuída MapReduce, o qual desponta como um recurso poderoso para potencializar o desempenho do processo de MO. Por fim, este trabalho ainda propõe duas técnicas de balanceamento de carga para abordagens em paralelo, sobretudo as que utilizam MapReduce.

As contribuições apresentadas neste trabalho resultaram em duas publicações em periódicos: Journal of Information and Data Management (Qualis: B3) [Araújo et al. 2015] e Knowledge-Based Systems (Qualis: A1) [Araújo et al. 2016]. O trabalho [Araújo et al. 2015] recebeu menção honrosa como um dos quatro melhores trabalhos apresentados no Simpósio Brasileiro de Banco de Dados, realizado em 2015.

Referências

Amin, M. B., Khan, W. A., Lee, S., and Kang, B. H. (2015). Performance-based ontology matching. *Applied Intelligence*, 43(2):356–385.

Araújo, T. B., Pires, C. E., da Nobrega, T. P., and Nascimento, D. C. (2015). A parallel approach for matching large-scale ontologies. *Journal of Information and Data Management*, 6(1):18.

Araújo, T. B., Pires, C. E. S., da Nóbrega, T. P., and Nascimento, D. C. (2016). A fine-grained load balancing technique for improving partition-parallel-based ontology matching approaches. *Knowledge-Based Systems*, 111:17–26.

Euzenat, J. and Shvaiko, P. (2013). *Ontology matching*. Springer-Verlag, Heidelberg (DE), 2nd edition.

Hamdi, F., Safar, B., Reynaud, C., and Zargayouna, H. (2010). Alignment-based partitioning of large-scale ontologies. In *Advances in knowledge discovery and management*, pages 251–269. Springer.

32th Brazilian Symposium on Databases

dsw

October 2nd to 5th, 2017
Uberlândia – MG – Brazil

DATASET SHOWCASE WORKSHOP

PROCEEDINGS

Promotion

Brazilian Computer Society – SBC
SBC Special Interest Group on Databases

Organization

Universidade Federal de Uberlândia – UFU

DSW Program Chairs

Mirella M. Moro, UFMG
Carina F. Dorneles, UFSC

Editorial

It is a pleasure to introduce the SBBD Dataset Showcase Workshop – DSW. It is the first time such an event happens with SBBD, and we are proud to see the database community participation through papers submissions and reviewers' engagement.

The Dataset Showcase Workshop purpose is to provide a forum for sharing and discussing how to build and organize datasets that serve as basis for research work developed in the database community. The contribution of papers published at DSW is the final product in the form of a dataset, usually extracted from some database or Web platform, cleaned, transformed and processed, often enhanced with external data and able to be reused for experiments reproduction as well as amplified to other scenarios. Furthermore, the DSW provides a real possibility of improving collaboration between different research groups through sharing data used in scientific endeavours.

Regarding the evaluation process, all submitted papers were evaluated by at least three members of the DSW program committee. For its first edition, DSW received 15 submissions from Brazilian and French researchers. Due to their high quality and interesting datasets, 11 submissions were accepted to be published in this proceedings and presented at the Workshop.

Finally, as SBBD DSW co-chairs, we would like to thank the authors and their collaborators for submitting their work to the workshop. Likewise, we really appreciate the reviewers work for the precious time spent in the careful evaluation of all submissions. We would also like to thank SBBD organizers for their outstanding support. We wish the community an excellent workshop and success in their future work.

Welcome to the first edition of the DSW, and we hope you enjoy the workshop.

Mirella M. Moro, UFMG
Carina F. Dorneles, UFSC
DSW Program Chairs

32nd Brazilian Symposium on Databases

October 2nd to 5th, 2017
Uberlândia – MG – Brazil

Promotion

Brazilian Computer Society – SBC
SBC Special Interest Group on Databases

Organization

Universidade Federal de Uberlândia – UFU

SBBD Steering Commitee

Agma Juci Machado Traina, USP
Bernadette Lóscio, UFPE
Caetano Traina Jr., USP
Carmem Hara, UFPR
Javam Machado, UFC
Mirella M. Moro, UFMG
Vanessa Braganholo, UFF

SBBD 2017 Commitee

Steering Committee Chair
Javam Machado, UFC

Local Organization Chairs
Maria Camila N. Barioni, UFU and Humberto L. Razente, UFU

Program Committee Chair
Carmem S. Hara, UFPR

Short papers Chairs
Bernadette Lóscio, UFPE and Damires Souza, IFPB

Demos and Applications Session Chair
Daniel de Oliveira, UFF

Short Courses Chair
Vaninha Vieira, UFBA

Workshop on Thesis and Dissertations in Databases Chair
Carina Dorneles, UFSC

Tutorials Chair
Ana Carolina Salgado, UFPE

Thesis and Dissertation Contest Chair
Vânia Vidal, UFC

Workshops Chair
Fernanda Baião (UNIRIO)

Local Organization Committee

Maria Camila N. Barioni, UFU
Humberto L. Razente, UFU
José Gustavo de Souza Paiva, UFU
Marcelo Zanchetta do Nascimento, UFU
Elaine Ribeiro de Faria Paiva, UFU
João Henrique de Souza Pereira, UFU

Dataset Showcase Workshop Program Committee

Alberto H. F. Laender (UFMG)
Bernadette Farias Lóscio (UFPE)
Daniel de Oliveira (UFF)
Daniel Kaster (UEL)
Eduardo Ogasawara (CEFET RJ)
Giseli R. Lopes (UFRJ)
Helena Grazziotin (UCS)
José Palazzo M. de Oliveira (UFRGS)
Jussara Almeida (UFMG)
Renata M. Galante (UFRGS)
Robson Cordeiro (USP São Carlos)
Ronaldo S. Mello (UFSC)
Sérgio Lifschitz (PUC Rio)
Vanessa Braganholo (UFF)
Vaninha Vieira (UFBA)

External Reviewers

Altamir Gomes Bispo Junior (USP São Carlos)
Eduardo Borges (FURG)

Leonardo Mauro Moraes (UFMS)
Raphael Martins (CEFET/RJ)
Roberto de Castro (CEFET/RJ)

Table of Contents (DSW)

A Twitter Opinion Mining Gold Standard for Brazilian Uprising in 2013

Tiago Cruz de França[1,2], José Orlando Gomes[2], Jonice Oliveira[2]

[1]Department of Mathematics – Federal Rural University of Rio de Janeiro (UFRRJ)
Seropédica – RJ – Brazil

[2]Postgraduate Program in Computer Science (PPGI) – Federal University of Rio de Janeiro (UFRJ)
Rio de Janeiro, Brazil

tcruzfranca@ufrrj.br, joseorlando@nce.ufrj.br, jonice@dcc.ufrj.br

Abstract. *Social media provide valuable sources of information, produced by real people about real world events, such as demonstrations or protests. The retrieval and extraction of information from such data presents some challenges, such as the production of good data sets, useful for support analysis using machine learning approaches, for example. In this paper, we present a sentiment-annotated Twitter gold standard for the analysis of Brazilian protests in 2013. The data set consists of 4,422 Twitter messages (tweets) annotated by three raters with information about the sentiment expressed in the messages. This is a valuable resource for social media-based sentiment analysis in the context of protest events.*

1. Introduction

Social media are a source of data and information about different subjects. França et al. (2014) presented some observations concerned with the data available in those media. The authors cited that the volume of data used in those media reached into petabytes. Social media provides a platform for shared information in a population, allowing them to organize themselves, to claim their rights also making it possible to raise public awareness and knowledge of such events [Hernandez and Spiro 2013].

Further to that, it provides the support and means necessary to extract information from the data and then get an understanding of events. For instance, during events like the Brazilian street protests, people were motivated to share their opinions [França and Oliveira]. In these settings, extracting the opinions expressed in such messages is fundamental to providing insights, enabling an analysis of the underlying political processes and dynamics [Hürlimann et al. 2016]. Machine learning approaches make it possible to analyze quantities of data which would be impossible for humans to analyze.

Automatically analyzing messages written in languages such as Brazilian Portuguese, can be harder than in other languages. For example, in the context of natural language process and sentiment analysis there are no adequate tools or a sentiment lexicon for Brazilian Portuguese, such as there are for English [Freitas 2015, Araújo et al 2016]. So, in such scenarios, where messages are not written in English, the use of machine learning approaches are often a good choice [França and Oliveira 2014]. In order to enable the sentiment analysis (or opinion mining) using machine learning it is necessary to provide annotated and reliable data sets [Saif et al. 2013, Hürlimann et

al. 2016]. The annotation of those data sets are difficult procedures and there aren't many data sets available, there are even less data sets of messages in Brazilian Portuguese [França and Oliveira 2014].

This work presents a gold standard human annotated data set. The data set consists mainly of tweets written in Brazilian Portuguese. A random sample was built upon 432,975 messages related to Brazilian protests in 2013. The data set has 4,422 tweets. Three annotators assigned just one class to each tweet in the data set. The possible classes were positive, negative or neutral sentiment about the protests. We also present some agreements and inter-rater agreement measures within the data set. The data set is available in https://labcores.github.io/p_tcruzfranca/.

The remainder of this work is organized as follows. In section two, we have presented some related works. In section three, we have described some information about the events to which the data set is related. Section four has the description of the data sets, the method and consolidation of the gold standard data set. Finally, section five has the final considerations.

2. Related Works

There are some existing tweet data sets available [Hassan et al. 2013]. For instance, those providing information to analyze data about the Brexit. Two examples are the Twitter gold standard for Brexit referendum [Hürlimann et al. 2016] and the #ImagineEurope project [ImagineEurope 2013a; ImagineEurope 2013b]. They both collected a data set of tweets using hashtags as filters. In general, those data sets were collected in the time leading up to the referendum (at least in the same year, months or weeks before the referendum day).

Hürlimann et al. (2016) selected certain hashtags and user citations to get data related to the Brexit. They then took a random sample of the English message data to be annotated. The main annotation task that they performed annotated two thousands tweets into five classes. Subsequently, they calculated certain agreements and inter-rater agreements in relation to the data. This proposal intends to work in similar way, but we have annotated a predominantly Brazilian Portuguese data set, related to protests in Brazil.

There are also other data sets that exist related to the Brexit or other real events (such as political debate, discourse or discussion, for example). In Ontotext (2016) 1.5 million tweets were utilized, while [Sheth 2016] checked messages posted to support the staying in or not (the leaving) of the Europe Union, three hours before and four hours after the referendum. Priego [Priego 2016a] and Priego[Priego 2016b] also studied tweets related to that event, in order to analyze the messages supporting, the staying in or leaving of, the Europe Union.

Go et al. (2009) introduced the Stanford Twitter Sentiment Corpus which consists of two different sets; a training and a test. On the one hand, the training data set has 1.6 million of tweets labeled as either positive or negative. The labels were assigned in a automated way using messages with emoticons (e.g. =(or =D). Despite the amount of messages, there is no guarantee that the emoticons really represent the sentiment of the message. On the other hand, the test data set is manually annotated and contains 177 negative, 182 positive, and 139 neutrals tweets.

Another data set was constructed to analyze the Obama-McCain debate during USA elections in September 2008 [Shamma, Kennedy and Churchill 2009]. This data set has 3,238 tweets annotated using Amazon Mechanical Turk, utilizing at least three annotators for each tweet. The classes assigned were positive, negative, mixed or other.

Another example is the Sentiment Strength Twitter Dataset which consists of 4,242 tweets manually labeled with positive and negative sentiment strength [Thelwal, Buckey and Paltoglou 2012]. Rieser (2014) presented a gold standard annotated data set of Arabic tweets which contains 8,868 messages.

Hernandez and Spiro (2013) also investigated tweets related to the Brazilian protests in 2013. However, the authors did not share their data set. Theodoro et al. (2015) utilized Twitter data to identify influencing factors in that social media. França and Oliveira (2014) analyzed the sentiment expressed in tweeted messages related to the Brazilian protests. This related work was a study using the preliminary annotated data of our data set. Such a preliminary data set has not been published before. We do not know of any other data set of Brazilian Portuguese related the Brazilian protests in 2013.

3. Data Set Contextualization

In 2013 Brazilians were facing huge protests and social media, such as Twitter, were used by the public as a channel to express opinion about the demonstrations. These people could have been the activists directly involved in protests or everyday citizens just expressing an opinion on events without actually participating in them. During this period certain expressions (words and hashtags) associated with the protests became very common (entering the list of trending topics in social media generally). This meant that some users utilized words employed in the protest context, although their intentions were not directly related to events (i.e. advertisers/self publicists).

We will describe the method used to collect the data and to build the data set, in relation to the protests, in section 4. However, to understand the data set, in this section we have made a description of the main events related to the protests. Brazil is a huge country with practically continental dimensions and is one of the 10 largest economies in the world, moreover, it has a population of more than 200 million people, spread about in different national regions. Moreover, more than 102 millions of it's inhabitants access the Internet, among which 89% use a smartphone to access social media [Portal Brasil, 2016], in order to to share their opinion about different subjects [Hernandez & Spiro, 2013].

The protests that occurred in Brazil in 2013 were compared to the protests that happened during the impeachment of the former President Color in 1992. But in 2013, Brazilians complained about increases in public transport ticket prices, as well as government corruption, amongst other things. Indeed, since 2012, Brazilians had marched against increases in public transport fares. Despite this, only in Jul 2013 did the major media channels report events, amidst claims of violence involving both the protesters and the police that clashed with them [Datafolha 2013c]. During these demonstrations, a number of violent events took place. For instance; damage to public and private buildings, the throwing of Molotov cocktails and many people were injured by rubber bullets, tear gas and incendiary incidents. At least six deaths were associated with the protests. The last major violent clash between security forces and the population had occurred in 1968 during protests calling for the end of military dictatorship in Brazil [Brazil, 2013].

Figure 1 summarizes events that occurred between June 1 and July 31, 2013. We have focused on that interval because during these months the protests were at their most intense. The summarization of events is presented as a timeline and the information is based on the following references [Brazil 2013]. Initially, the marchers complained about rises in public transport fares. Motivated by the large numbers of protesters and the police clashes, the demonstrations immediately gained attention in social media and major Brazilian media outlets published some information on the

events happening. It is important to highlight that from the 15th until 30th of June, Brazil hosted the FIFA Confederations Cup and the World Youth Days from 23th to the 28th of July.

After a series of demonstrations in Brazil the nature and number of the public demands needed to be clarified. Besides ticket transport prices, the people also demonstrated their disappointment regarding; government corruption, certain legislative proposals (e.g. PEC 37) and the large expenses incurred by events that were to be held in the country. There was no official social leadership, with neither political party leading the protests in Brazil during this time. Therefore, a group named Anonymous Brazil published a video promoting 5 main issues. We have highlighted two of them. Firstly, a rejection of a proposed amendment to the Brazilian constitution known as PEC 37 which intended to prohibit investigations by the federal public ministry. The PEC 37 was understood to be a way of facilitating politician's departures, without them paying for their corruption. Secondly, they called for the opening of an investigation for verifying irregularities related to money spent on infrastructure for the FIFA World Cup. An event that would take place in 2014.

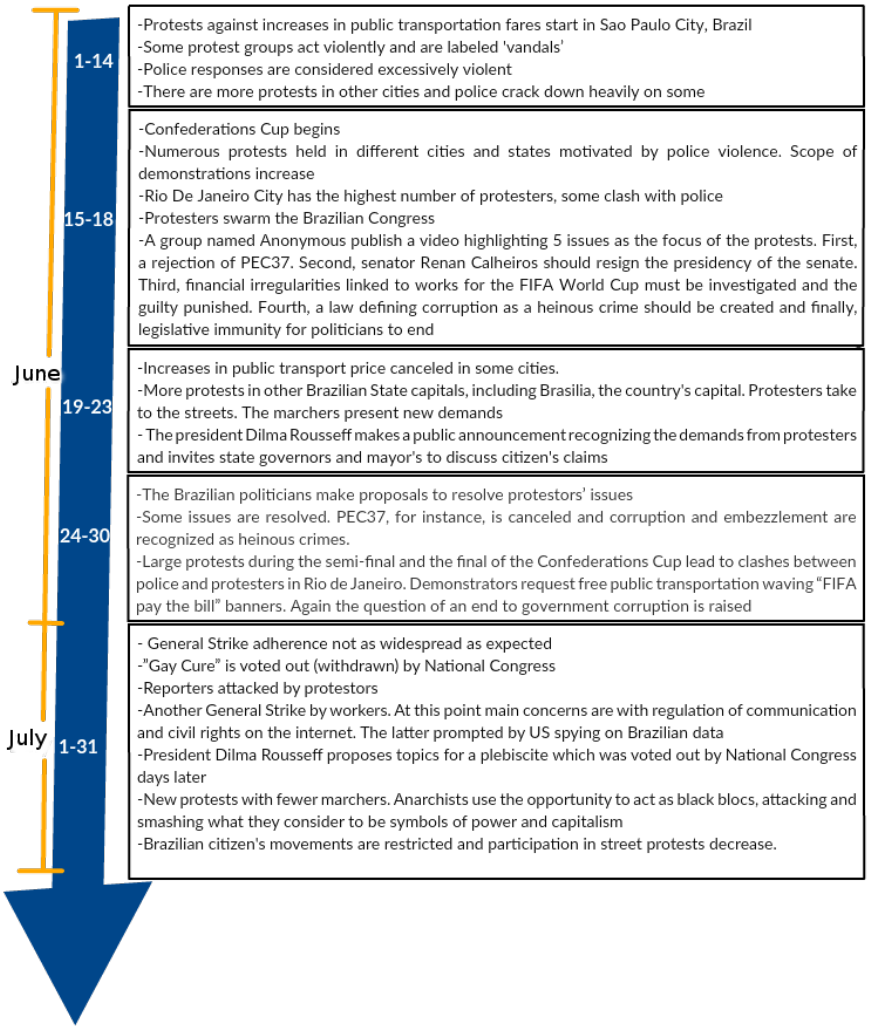


Figure 1. The Timeline of the main events during the Brazilian protests, 2013

Certain Brazilian agencies conducted surveys related to the protests. An opinion poll showed that 55% of citizens from the city of São Paulo had agreed with the

protests on June 14 [Datafolha, 2013d]. According the same survey, 41% of respondents had disagreed, while 4% had no opinion or were indifferent. In other words, 55% of citizens had a positive opinion about protests while 41% had a negative opinion. A week later, on June 21st, 66% still agreed with the protests, even though just 2 days earlier public transportation fares had been reduced [Datafolha, 2013e]. On October 28th in São Paulo city data showed that the support for the protests had decreased from 89% to 66% in October while the rate of disagreement grew from 8% to 31% [Datafolha, 2013a]. In February 2014 the interviewees considering the protests to be positive reduced to 52% while those who considered them to be negative increased to 42% [Datafolha, 2014].

A nationwide public opinion survey was made on June 24th, 2013 by the Brazilian Institute of Public Opinion and Statistics Institute (IBOPE). The survey results had shown 75% Brazilians agreed with the demonstrations and 22% had disagreed [IBOPE, 2013a]. IBOPE also asked Brazilians about political party representation and 89% of the people answered that no political party represented them [IBOPE, 2013b].

Other surveys demonstrated the importance of social media in the Brazilian protests. The results of [Datafolha, 2013b] showed that on June 19th, 93% of the activists organizing the demonstration through social media considered such media as their main source of information. According to IBOPE [IBOPE, 2013b] results on June 20th, 62% of marchers knew of the protests through Facebook and 75% of people invited other people using Facebook and Twitter.

4. Method and Data Set Consolidation

Our aim is provide a random sample to support the opinion mining (sentiment analysis) from tweets related to Brazilian protests in 2013. Many demonstrations occurred during this year in Brazil between June and August, which represents a time period relating to events as summarized in section 3.

The first step in data retrieval was to define some filters. During the protests we observed the content posted relating to the events in which we are interested. Then we defined some hashtags used to index the content published, as expressions associated with the protests. Secondly, we kept in mind that we wanted to understand the opinion of the users from Twitter. So we defined a protocol to assist in our analysis. The protocol comprises of a defining guide question and some possible answers (the classes) since we are interested in support supervised machine learning analysis to extract information from the data set.

In general, there are two basic ways to get posts from Twitter, one is using the Twitter API and the other is using Twitter stream[1]. Both approaches are provided by a REST API and require some credentials from a user to access resources. We have used the Twitter API to build our raw data set. That API imposes some restrictions, such as limits of requests and limits of time (posts newer than 7 days) to query and deliver data.

4.1 Sample and Filtering (Data Collections)

In regards to data collection, we used the Twitter's API during the Brazilian protests between June 1 and July 31, 2013 (inclusive). In order to cover posts related to the protests, we utilized 22 hashtags used by the activists which are presented in List 1. The criteria chosen for those hashtags was based on the manual identification of common keywords associated with the protests events in Brazil during that period.

1 https://dev.twitter.com/overview/api

Each tweet is returned by the API as a JSON representation. That JSON comprises all metadata provided by Twitter. Those metadata include, for example, data about the user that published the message, a timestamp, possibly a geolocation and so on. Those tweets formed a raw data set with 432,975 messages published by 180,005 different users (approximately 2.41 tweets per user). The information contained in the raw data set is summarized in Table 1. Figure 2 depicts the frequency of tweets per day between the Jun 1 and Jul 31, 2013. We should point out that most messages were written in the Brazilian Portuguese language.

#acordabrasil, #vemprarua, #ForaFifa, #ogiganteacordou, #anonymousbrazil, #MPL, #passelivre, #pec37, #mudabrasil, #ChangeBrazil, #anonymousbrazil, #protesto, #foradilma, #protestorj, #protestabrasil, #primaverabrasileira, #forafeliciano, #ocupa, #copapraquem, #protest, #pec33, #pec99

List 1. Hashtags used to retrieve Twitter's messages

Table 1. Basic raw data set information

	# Amount
Total of messages	432,975
Total of users	180,005 (average of ~2.41 messages per user)
Period of data collect	61 days (from Jun 1 up to Jul 31, 2013)

We build a random sample of 1% of messages for each day from the stream represented by the raw data set. In other words, for each day we randomly chose 1% (with a minimum of 1 message in a day) of data to be labeled. The total amount of messages was 4,422 tweets.

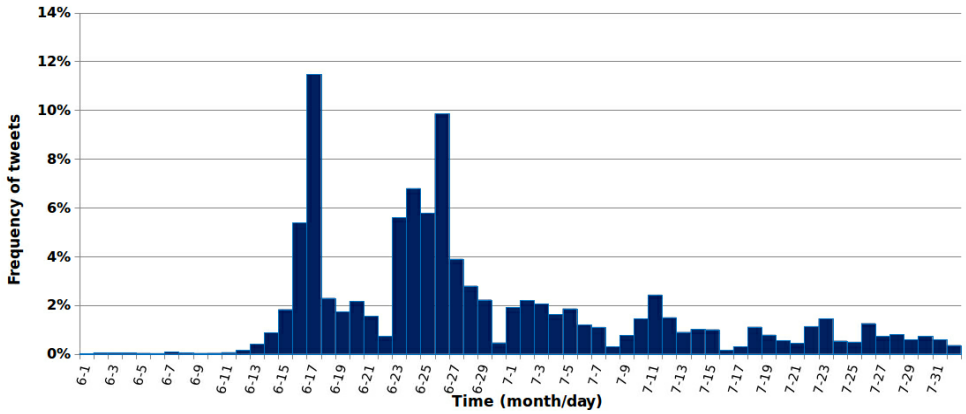


Figure 2. Frequency of Tweets per Day

4.2 Annotation

The 4,422 tweets sampled were presented to three volunteer annotators. Before starting the annotation task, each annotator received basic explanations related to both "why" and "when" the dataset was built. Then, the following two explanations were presented. The first one was concerned with the motivation for understanding how the activists and population expressed their opinion (or sentiment) in the social media (mainly on Twitter). The second, was related to the period of data retrieval that occurred during the many Brazilian protests events in 2013.

The aim was to classify whether the opinion expressed in the messages was in some way positive or negative, regarding the protests. To do that, we defined a **key question** as a guide for the classification task. The question states *"(does) the tweet text expresses an opinion of agreement (positive), disagreement (negative) or neither (neutral) to the protests?"*. Only one answer should be assigned to each message. Thus,

the annotators judged whether a message expressed an opinion in favor (positive) or against (negative) the protests. If a classifier judged that a message didn't express a positive or negative opinion for any reason, that message would be considered neutral. In other words, **we have defined three (answer) classes**: **positive, negative or neutral.**

We should note that the annotation tasks were started eight months after the protests events. At first, a pilot was built with only 200 messages [França and Oliveira, 2014]. After that, the sample was built in a random way taking 1% of messages for each day of the raw data set. We also assume a minimum of one message per day, if the 1% represents a value less than one. The entire labeled data set was then finished, one and half years after the protests. The results of the annotation task are in Table 2. This approach enables us to get some important feedback, such as that the majority of the messages were in Brazilian Portuguese. Furthermore, we could also take some inter-rater agreement measures between the annotators.

Table 2. Rating among annotators and classes

Classes	Annotators-1	Annotators-2	Annotators-3
Positive	1,786	1,723	1,945
Negative	191	277	294
Neutral	2,445	2,422	2,183
# Total of Labeled Messages		4,422	

4.3 Agreement

The approach of classifying the messages three times with different annotators enables us to verify the inter-rater agreement among such raters. The inter-rater agreement aggregates information about how reliable the annotated data set is. That means low inter-rater agreement could produce unreliable results. This could happen because of problems in annotation method, during the annotation task, or in the data set.

Table 3 shows the distribution of messages with regard to the number of annotators opinion rater agreement, providing a different view of agreement. In Table 4 we present two inter-rater agreement metrics (Fleiss' Kappa and Krippendorff's alpha) for the sentiment annotations. Fleiss' Kappa is useful when all messages are labeled by the annotators. Krippendorff's alpha allows the inter-rater agreement to be calculated even if not all the data is labeled for all volunteers.

Table 3. Agreement among the annotators (raters)

	# tweets	%
Unanimous sentiment	3,449	~ 78%
Two sentiments	936	~ 21%
Three sentiments	37	~ 1 %
Total	4,422	100%

Table 4. Inter-rater agreement among raters

Statistic inter-rater agreement	Result
Fleiss' Kappa	73%
Krippendorff's alpha	65%

If inter-rater agreement is low or negative, then someone could conclude that an automated supervised classification task is not useful because it's result is not reliable. That is, the results of an automated classification will reflect the manually annotated task. Thus, it will not represent non-valuable information to understand real case scenarios. In our case, we argue that inter-rater agreement values enable someone to know how reliable the classification is and thus how valuable the data set is.

There are some points about both Kappa value and Krippendorff's alpha that we want to highlight. Landis (1977) had proposed a scale in which Fleiss' Kappa greater than 0.6 would be good values, between 0.41 and 0.6 moderate, and less than 0.41 fair, slight or poor when less than 0. However, there is no widely agreed acceptance about the interpretation of Kappa results [Gwet 2014]. Besides that, when the number of classes are fewer, the kappa tends to be higher [Sim and Wright 2005].

Krippendorff's alpha presents some disadvantages when used to measure nominal data. It is more general than Fleiss' Kappa, but has some weaknesses such as it ignores how many times a subject was annotated. Another consideration is that just the disagreement is evaluated while the agreement is ignored. In other words, just the disagreements define the result. Finally, the disagreements are larger for nominal data leading alpha to a small result.

4.4 Data Set Consolidation and Description

The gold standard was obtained after a consolidation on the annotations presented in Table 2. That consolidation procedure follows two steps. We use 1) a majority vote for the sentiment, and 2) when the raters disagreed entirely (each one assigned three different sentiments to a tweet), then the decision was assign the tweet as neutral. As presented in Table 3, a small amount of messages were assigned in three different class (just 37, less than 1%).

Table 5. The gold standard final consolidation

	# consolidation	%
Positive	1752	~40%
Negative	198	~4%
Neutral	2472	~56%
Total of Labeled Messages (#)	4422	

Unfortunately, according to Twitter policy[2] a third party cannot provide entire tweet objects, only the tweet's IDs which in turn must be shared as a spreadsheet or PDF file. Therefore, the raw data set is available for download as a list of IDs. We also provide a python script to retrieve tweets by IDs using the Twitter's API. The resource API utilized enables the recovery of posts older than 7 days. Therefore, although the data set being composed by messages are unrecoverable using Twitter API search query, it is possible when the tweet's IDs are known.

The shared data set has the tweet IDs, the assigned classes by humans in three distinct column and the consolidation column. The classes were assigned as follows; messages classified as *positive* has received an uppercase "P" as mark. Those classified as *negative* or *neutral* received an uppercase mark as either "N" or "NN", respectively. Similarly, the raw data set is available for download as a spreadsheet, but containing just the tweet IDs.

Most of the messages were classified as being neutral. It should be noted that neutral messages are more common among social media messages than in the surveys presented in section 3. This is because the neutral class groups together all messages that did not present positive or negative ratings. This means that any result which does not represent feeling, will be classified as neutral. For example, a message reporting an event or an advertising message indexed with one of the hashtags adopted in the project could be used to gain attention, for example. In addition, IBOPE and Datafolha question users, which is different from extracting their opinion from a text that is not the answer

2 https://dev.twitter.com/overview/terms/policy

to a pre-prepared question, but rather a free expression by the user. In the Gold Standard data set, the positive and negative opinions in June were 34% and 5% respectively. In July, the positive was 50% and the negative 4%.

Table 6. Examples of tweets and annotations assigned

	Tweet text	Assigned sentiment
1	Falta de leitos nos hospitais públicos leva milhares de pacientes recorrem ao Ministério Público para se tratarem #VempraRua #SOSSUS	Positive
2	Cara, desisto de fazer vocês entenderem. Não adianta ficar nessa mobilização " #mudabrasil" sem propôr mudanças reais.	Negative
3	O pau comendo em várias cidades do Brasil e o Fantástico começa com música feliz, falando sobre a Copa das Confederações. #protestorj	Neutral
4	Convoco lutadores de Jiu-Jitsu e Muay Thai a participar dos protestos para dar surra nos baderneiros e saqueadores de lojas. #mudabrasil	Negative (it received two negative and one positive assignment)
5	A torcida continuou cantando o hino nacional, mas n foi p fazer bonito para a rede Globo e sim p mostrar sua forca! #OGiganteAcordou	Neutral (received the three possible sentiments annotations)

5. Final Considerations

In this work we have described a Gold Standard data set for the Brazilian protests in 2013. Besides the data set description, we also presented the method, consolidation and some measures of agreement and inter-rater agreement among three human annotators. The annotated data set provides a resource for observing the social and discourse dynamics associated to the protests in Brazil between June 1st and July 31st, 2013.

The messages present in the data set were classified in one of three classes: positive, negative or neutral. Those tags represent whether the messages present a agreement or disagreement about the protests. If no sentiment could be assigned for any reason, then the message was classified as neutral. After the human annotation task, we defined two basic consolidation decisions. The final result would be the sentiment with more assignment, or if a total disagreement occurred (each annotator assigned a different tag), the message was defined as neutral.

This data set is useful for anyone interested in exploring data about the protests, or to evaluate machine learning analysis specifically using messages written in the Brazilian Portuguese language, since our messages are mostly Brazilian Portuguese messages. The data set also supports the use or evaluation of, tools for natural process languages for Portuguese or tweet messages.

Besides the gold standard data set, we have also presented some overviews about the raw data set retrieved during the events. We named that data set the ‘raw data set’. We should point that the raw data set is a convenient sample returned by Twitter, since we have no way to define a sampling method. We knew previously that Twitter has restrictions or limits on the data retrieval and so we will use a sample without any statistical definition. Another point to be taken into consideration is the fact that many citizens have no access to or an account on Twitter. For instance, poor people often have no means to access the Internet. Therefore, even before the Twitter restrictions, social media is, in general selective.

References

Brazil (13 jul 2017). 2013 protests in Brazil. *Wikipedia*. https://en.wikipedia.org/w/index.php?title=2013_protests_in_Brazil&oldid=790394354.

Datafolha (2013 6AD). Paulistanos defendem continuidade de protestos e foco em saúde e educação - Opinião Pública. http://datafolha.folha.uol.com.br/opiniaopublica/2013/06/1300362-paulistanos-defendem-continuidade-de-protestos-e-foco-em-saude-e-educacao.shtml, [accessed on Jul 30].

DataFolha (14 jun 2013). Paulistanos aprovam protestos, mas rejeitam vandalismo e tarifa zero - Opinião Pública. http://datafolha.folha.uol.com.br/opiniaopublica/2013/06/1295431-paulistanos-aprovam-protestos-mas-rejeitam-vandalismo-e-tarifa-zero.shtml, [accessed on Jul 30].

Datafolha (14 jun 2013a). Paulistanos aprovam protestos, mas rejeitam vandalismo e tarifa zero - Opinião Pública. http://datafolha.folha.uol.com.br/opiniaopublica/2013/06/1295431-paulistanos-aprovam-protestos-mas-rejeitam-vandalismo-e-tarifa-zero.shtml, [accessed on Jul 30].

Datafolha (28 oct 2013b). Apoio às manifestações cai de 74% para 66% - Opinião Pública. http://datafolha.folha.uol.com.br/opiniaopublica/2013/10/1363246-apoio-as-manifestacoes-cai-de-74-para-66.shtml, [accessed on Jul 30].

FRANCA, T. C. and Oliveira, J. Proceedings of III Brazilian Workshop on Social Network Analysis and Mining (BraSNAM), Brasilia. Anais do Congresso da Sociedade Brasileira de Computação, 2014.

França, T., Faria, F., Rangel, F., Farias, C. and Oliveira, J. (2014). *Big Social Data: Princípios sobre Coleta, Tratamento e Análise de Dados Sociais*. Curitiba-PR: Bernadette Farias Lóscio.

Go, A., Bhayani, R. and Huang, L. (2009). Twitter Sentiment Classification using Distant Supervision. *Processing*, p. 1–6.

Hürlimann, M., Davis, B., Cortis, K., et al. (2016). A Twitter Sentiment Gold Standard for the Brexit Referendum. In *Proceedings of the 12th International Conference on Semantic Systems*. , SEMANTiCS 2016. ACM. http://doi.acm.org/10.1145/2993318.2993350.

IBOPE (24 jun 2013a). 75% dos brasileiros são favoráveis às manifestações públicas. http://www.ibopeinteligencia.com/noticias-e-pesquisas/75-dos-brasileiros-sao-favoraveis-as-manifestacoes-publicas/, [accessed on Jul 30].

IBOPE (25 jun 2013b). 89% dos manifestantes não se sentem representados por partidos. http://www.ibopeinteligencia.com/noticias-e-pesquisas/89-dos-manifestantes-nao-se-sentem-representados-por-partidos/, [accessed on Jul 30].

ImagineEurope (2017). Building a Twitter Dataset to Find out How People View the EU (with images, tweets) ·. https://storify.com/ImagineEurope/building-a-twitter-dataset-to-find-out-views-on-th, [accessed on Jul 30].

ImagineEurope ([S.d.]). EU Twitter Sentiment Analysis (with images, tweets) · ImagineEurope. https://storify.com/ImagineEurope/initial-sentiment-analysis, [accessed on Jul 30].

Monroy-Hernandez, A. and Spiro, E. (2013). Research Notes » Blog Archive » How is the Brazilian Uprising Using Twitter? . http://blogs.harvard.edu/andresmh/2013/07/how-is-the-brazilian-uprising-using-twitter/, [accessed on Feb 4].

Ontotex (2016). White Paper: #BRexit Twitter Analysis. *Ontotext*. https://ontotext.com/white-paper-brexit-twitter-analysis/, [accessed on Jul 30].

Portal Brasil (14 jun 2013). Maioria da população é a favor dos protestos, mostra Datafolha - 14/06/2013 - Cotidiano. http://www1.folha.uol.com.br/cotidiano/2013/06/1294919-maioria-da-populacao-e-a-favor-dos-protestos-mostra-datafolha.shtml, [accessed on Jul 30].

Portal Brasil (2016). Pesquisa revela que mais de 100 milhões de brasileiros acessam a internet — Portal Brasil. http://www.brasil.gov.br/ciencia-e-tecnologia/2016/09/pesquisa-revela-que-mais-de-100-milhoes-de-brasileiros-acessam-a-internet, [accessed on Jul 30].

Priego, E. (21 jun 2016a). “Stronger In”: Looking Into a Sample Archive of 1,005 StrongerIn Tweets. *Ernesto Priego*. https://epriego.wordpress.com/2016/06/21/stronger-in-looking-into-a-sample-archive-of-1005-strongerin-tweets/, [accessed on Jul 30].

Priego, E. (21 jun 2016b). “Vote Leave”: Looking Into a Sample Archive of 1,100 vote_leave Tweets. *Ernesto Priego*. https://epriego.wordpress.com/2016/06/21/vote-leave-looking-into-a-sample-archive-of-1100-vote_leave-tweets/, [accessed on Jul 30].

Refaee, E. and Rieser, V. (2014). An Arabic Twitter Corpus for Subjectivity and Sentiment Analysis. . European Language Resources Association. https://researchportal.hw.ac.uk/en/publications/an-arabic-twitter-corpus-for-subjectivity-and-sentiment-analysis, [accessed on Jul 30].

Saif, H., Fernández, M., He, Y. and Alani, H. (2013). Evaluation datasets for Twitter sentiment analysis: a survey and a new dataset, the STS-Gold. . http://www.di.unito.it/~patti/essem13/index.html, [accessed on Jul 30].

Shamma, D. A., Kennedy, L. and Churchill, E. F. (2009). Tweet the debates: Understanding community annotation of uncollected sources. In *In WSM ’09: Proceedings of the international workshop on Workshop on Social*.

Sheth, A. (24 jun 2016). #Brexit: “there is a big trouble for #remain” — Some Lessons from Real-time #socialmedia Analysis. https://www.linkedin.com/pulse/brexit-big-trouble-remain-some-lessons-from-real-time-amit-sheth, [accessed on Jul 30].

Thelwall, M., Buckley, K. and Paltoglou, G. (jan 2012). Sentiment Strength Detection for the Social Web. *J. Am. Soc. Inf. Sci. Technol.*, v. 63, n. 1, p. 163–173.

Theodoro, I., Stearns, B., França, T. C. and Oliveira, J. (2015). Padrões de Comportamento de Usuários nos Protestos Brasileiros via Twitter. In *XII Simpósio Brasileiro de Sistemas Colaborativos (SBSC), 2015*.

Dados de Monitoramento de Projetos de Inclusão Digital do Ministério da Ciência, Tecnologia, Inovações e Comunicações

Diego Pasqualin[1], **Edemir Maciel**[1], **Luis C. E. de Bona**[1]
Lucas Oliveira[1], **Marcos Sunye**[1]

[1]Centro de Computação Científica e Software Livre
Departamento de Informática
Universidade Federal do Paraná (UFPR)
R. Cel. Francisco H. dos Santos, 100 – Curitiba – PR – Brasil

{dpasqualin,bona,lfo14}@inf.ufpr.br

Abstract. *This paper presents open data collected through SIMMC, an award winning monitoring system used to track the digital inclusion projects by the Brazilian Ministry of Science, Technology, Inovations and Communications. Besides assisting the government in the management and expansion of the projects "Telecentro", "Gesac" and "Digital Cities", the system has the innovative feature of releasing all collected data in a publicly available Web site, to allow the development of third party applications. SIMMC stores and provides around 1,2 million new records a day, totalling more than 1,5 billion since it's release, displaying data about software and hardware inventory from computers and network usage from routers, allowing the assessment of the deployed computational park, inventory changes, in addition to anomaly detection on the network usage and discrepancies between hired and measured bandwidth.*

Resumo. *Esse artigo apresenta os dados abertos coletados pelo SIMMC, premiado sistema de monitoramento e transparência pública empregado no acompanhamento dos projetos de inclusão digital do Ministério da Ciência, Tecnologia, Inovações e Comunicações. Além de auxiliar o governo no gerenciamento e expansão dos projetos Telecentro, Gesac e Cidades Digitais, o sistema tem o caráter inovador de disponibilizar todos os dados coletados em uma página Web acessível publicamente, para permitir o desenvolvimento de aplicações de terceiros. O SIMMC armazena e disponibiliza cerca de 1,2 milhão de entradas diárias e totaliza mais de 1,5 bilhão de entradas desde seu lançamento, com dados sobre inventário de hardware e software de computadores e uso de rede de conexões de Internet, que permitem verificar o estado do parque computacional instalado, alterações de inventário, além de detectar anomalias no uso da banda de Internet e discrepâncias entre banda entregue e contratada.*

1. Introdução

O SIMMC [Pasqualin et al. 2017 no prelo] é o sistema de monitoramento e transparência desenvolvido pelo Centro de Computação Cientifica e Software Livre (C3SL)[1] para o antigo Ministério das Comunicações que é atualmente utilizado pelo Ministério da Ciência,

[1]Página oficial do C3SL: http://www.c3sl.ufpr.br.

Tecnologia, Inovações e Comunicações (MCTIC)[2], com o objetivo de acompanhar a implantação e uso dos seus projetos de inclusão digital, presentes em cidades e regiões remotas, onde o investimento privado é baixo ou inexistente. Os projetos em andamento são: i) *Gesac*[3], que prioriza regiões remotas e de baixo IDH, oferecendo conexão com a Internet a escolas, unidades de saúde, vilas indígenas, quilombos, entre outros; ii) *Cidades Digitais*[4], que implanta infraestrutura de rede de alta velocidade em pequenas cidades, conectando órgãos governamentais e disponibilizando acesso gratuito em praças públicas, assim como aplicativos de e-government buscando melhorar a eficiência da administração pública; iii) *Telecentros*[5], projeto que leva a comunidades carentes um laboratório de informática com acesso à Internet, equipado para realização de cursos e treinamentos, agindo como um espaço para integração, cultura e lazer.

A avaliação da efetividade dos projetos foi historicamente realizada de duas formas: através de sistemas de monitoramento providos pelos próprios provedores de Internet contratados para oferecer o serviço, ou através do envio de servidores públicos do governo para verificação in loco. O primeiro método carece em transparência e o segundo apresenta custo elevado com deslocamento e baixa eficiência, devido à necessidade de amostragem e longos intervalos entre as avaliações para redução de custo. Nesse contexto, o SIMMC foi criado como uma ferramenta de monitoramento automatizado e transparência pública, buscando aumentar a eficiência da gestão pública e auxiliar na expansão dos projetos de inclusão digital do MCTIC.

O SIMMC foi vencedor do 3º Concurso de Boas Práticas da CGU na categoria "Promoção da Transparência Ativa ou Passiva"[6] e monitora hoje cerca de 6300 Pontos de Presença (PdPs)[7], localizados em mais de 2400 municípios brasileiros. O sistema coleta hoje dados de disponibilidade, inventário de *hardware* e *software*, além de uso de rede, recebendo mais de 1,8 milhões de novas entradas diariamente e totalizando mais de 1,5 bilhão desde seu lançamento, em 2014. Todos os dados coletados são disponibilizados em gráficos, relatórios e mapas no portal público, em `http://simmc.c3sl.ufpr.br/`, e também em formato aberto, descrito em detalhes nesse artigo.

Atualmente, os dados são utilizados para identificar a instalação de novas conexões e telecentros, verificar se os recursos empregados estão de fato sendo utilizados - executando realocação ou atualização de equipamentos e banda conforme necessidade -, detectar situações que indicam furto de equipamentos ou falha dos provedores de Internet na entrega da banda contratada, além do planejamento da expansão dos projetos. A publicação dos dados em formato aberto fomenta ainda sua utilização em outros contextos, permitindo junção com outros repositórios, que podem trazer novas interpretações aos indicadores coletados.

O artigo se organiza da seguinte maneira. Na Seção 2, são descritos os repo-

[2]Página oficial do MCTIC: `http://www.mcti.gov.br/`.

[3]Página oficial do projeto Gesac: `http://www.mc.gov.br/gesac`.

[4]Página oficial do projeto Cidades Digitais: `http://www.mc.gov.br/cidades-digitais`.

[5]Página oficial do projeto Telecentro: `http://www.mc.gov.br/telecentros`.

[6]SIMMC vence concurso de boas práticas da CGU: `http://www2.mcti.gov.br/index.php/imprensa/todas-as-noticias/institucionais/38138-mc-recebe-premio-da-cgu-por-transparencia`.

[7]Um PdP é uma localidade física (escola, telecentro, quilombo) onde os dispositivos monitorados (computadores, roteadores) estão instalados.

sitórios disponibilizadas em formato aberto. A Seção 3 exibe casos de uso comuns desses repositórios, como exibidos no portal do SIMMC e utilizados pelo MCTIC para acompanhamento dos projetos. Em seguida, a Seção 4 descreve como os dados são obtidos, a Seção 5 descreve alguns desafios no desenvolvimento do sistema e limitações dos dados coletados e, por fim, a Seção 6 exibe a conclusão do projeto.

2. Dados Abertos

Os dados dos projetos do MCTIC são coletados pelo SIMMC desde 2014. Até o momento, já acumularam cerca de 1,5 bilhão de entradas no banco de dados, com uma frequência diária de aproximadamente 1,8 milhões de novas entradas. Ao todo são 6302 PdPs ativos no sistema, contemplando 58% das cidades brasileiras (detalhes na Tabela 1), com dados de tráfego de rede coletados em intervalos de cinco minutos e de inventário de *hardware* e *software* coletados diariamente.

Apesar do projeto Gesac possuir mais pontos monitorados, o maior volume de dados vem das Cidades Digitais, representando 66,70% da base atual, seguido pelo Gesac com 33,03% e Telecentros, com 3,98%[8]. O motivo é o fato de que cada órgão na Cidade Digital possui vários dispositivos conectados, com tráfego armazenado individualmente (sem agregação) e distinguível pelo IP de origem.

Tabela 1: Número de PdPs monitorados por projeto e sua distribuição nas cidades brasileiras (5.570 no total).

	Gesac	*Telecentros*	*Cidades Digitais*	**TOTAL**
PdPs	4837	758	898	6302
Cidades	2072 (37%)	1088 (20%)	65 (1%)	3225 (58%)

A Figura 1 exibe a evolução do sistema quanto ao tráfego de rede medido e número de PdPs monitorados. As quedas observadas no uso de rede são devido à indisponibilidades ocorridas no sistema de coleta, de cerca de 3,88% desde o lançamento.

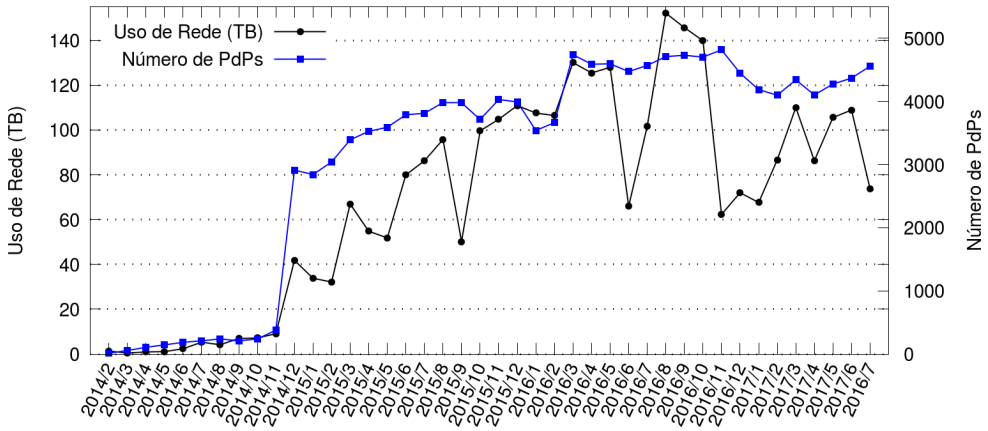


Figura 1: Uso de rede e número de pontos monitorados ao longo do tempo.

[8]A soma ultrapassa 100% porque alguns telecentros também possuem conexão Gesac, duplicando os meios de coleta.

Tabela 2: Dicionário de metadados sobre os PdPs.

Campo	*Descrição*
id_point	Identificação única do PdP
is_active	Indica se ponto está ativo (ainda é monitorado)
is_gesac	Verdadeiro se o PdP é uma conexão Gesac
is_telecenter	Verdadeiro se o PdP é um Telecentro
is_digital_city	Verdadeiro se o PdP pertence a uma Cidade Digital
street	Logradouro
neighborhood	Bairro
zipcode	CEP
city	Cidade
region	Região (ex: SUDESTE)
name	Nome do PdP
latitude	Coordenada geográfica, latitude
longitude	Coordenada geográfica, longitude

2.1. Dicionário de Dados

Todos os dados coletados são disponibilizados com frequência diária na página de dados abertos do C3SL, no endereço `http://dadosabertos.c3sl.ufpr.br/simmc`. A página exibe um arquivo único para todos os projetos chamado `pontos-de-presenca.csv`, contendo metadados sobre os PdPs, detalhadas na Tabela 2. Cada projeto possui então seu respectivo diretório (`telecentro`, `gesac` e `cidade-digital`), com arquivos que contém os indicadores e métricas coletadas, nomeadas com o indicador como prefixo (`uso-de-rede` ou `inventario`), seguido da data no formato `YYYY-MM-DD` (exemplo: `gesac/uso-de-rede_2017-01-01.csv`).

As métricas disponíveis para análise de uso de rede são descritas na Tabela 3, com a coluna "Projetos" indicando em quais projetos os campos estão disponíveis. Alguns campos são particulares a um projeto. O campo `access_point_type`, presente somente nas cidades digitais, indica se o PdP está localizado em órgãos governamentais, quando ele é chamado de PAG, ou se ele representa uma conexão de Internet pública sem fio, normalmente posicionada nas praças das cidades, chamado de PAP. Telecentros possuem o campo "macaddr" (*MAC Address*), que identifica unicamente as máquinas dentro de um mesmo telecentro. Os pontos GESAC, por sua vez, possuem a informação da banda contratada pelo MCTIC. A Tabela 4 exibe as métricas de inventário, disponíveis somente para o projeto Telecentro. Os indicadores podem ser associados com os metadados através do campo `id_point`, presente em ambos os arquivos.

3. Utilização dos Dados

O acompanhamento através do SIMMC permite ao MCTIC avaliar diversos aspectos dos projetos monitorados. Os indicadores de uso de rede e inventário possibilitam a geração de alertas informando sobre novas implantações, sobre conexões ou computadores desconectados, alterações de *hardware*, baixo uso da rede contratada, entre outros. A geração de métricas com base nesses dados pode ser utilizada inclusive para definir a logística na

Tabela 3: Dicionário de dados de uso de rede. O campo "Projetos" indica em quais projetos os campos estão presentes, onde "T" significa Telecentro, "G" conexão GESAC e "C" Cidade Digital.

Campo	*Descrição*	*Projetos*
id_point	Identificação única do PdP	T/G/C
collect_time	Hora da coleta, formato HH:MM:SS	T/G/C
collect_date	Data da coleta, formato YYYY-MM-DD	T/G/C
download_bytes	*Bytes* baixados pelo dispositivo	T/G/C
upload_bytes	*Bytes* enviados pelo dispositivo	T/G/C
download_packages	Número de pacotes de dados baixados	T/G/C
upload_packages	Número de pacotes de dados enviados	T/G/C
macaddr	MAC *Address*, id da máquina no telecentro	T
hired_download_kbps	Banda de *download* contratada	G
hired_upload_kbps	Banda de *upload* contratada	G
access_point_type	Ponto de acesso público (PAP) ou governamental (PAG)	C
ip	IP do dispositivo que gerou o tráfego	C

compra de novos equipamentos. Por exemplo, para receber novos computadores um telecentro deve manter 70% dos computadores atuais em operação, com uma disponibilidade maior do que 80%.

A expansão da banda de Internet pode ser planejada de forma similar. A Figura 2a exibe um exemplo de gráfico de rede onde a banda medida (linha sólida) se aproxima diariamente da banda contratada (linha tracejada), tanto em *download* quanto em *upload.* Esse fenômeno indica que a operadora está entregando a banda contratada e que o PdP é utilizado regularmente. O MCTIC pode então optar por aumentar a largura de banda para melhor satisfazer as necessidades dos usuários desse PdP. Em contrapartida, quando o ponto é subutilizado o MCTIC pode desligá-lo ou realocá-lo para melhor aproveitamento do recurso.

Os gráficos de rede também podem ser utilizados para verificar se o provedor de Internet está de fato entregando a banda contratada pelo MCTIC, situação evidente ao se observar plateaus abaixo da linha tracejada, como pode ser visto na Figura 2b. Atualmente não é possível gerar tráfego artificial no PdP para saturar o *link* e medir esse fenômeno de forma precisa, portanto a identificação de plateaus é interpretada como uma estimativa.

Outro exemplo de uso do SIMMC é a gestão do inventário, que consiste na identificação de alteração dos computadores dos telecentros, que pode ser interpretada como simples manutenção dos equipamentos, ou furto. A Figura 3 destaca o relatório de mudança de inventário, onde pode-se observar que um dos computadores do telecentro "Kit - Centro Inclusão digital", identificado pelo *MAC Address* da placa de rede, teve um pente de memória subtraído, detectado na coleta do dia 07/08/2014.

3.1. Oportunidades

Dentro das atividades ainda não desenvolvidas, podemos citar a incorporação no sistema de indicadores sociais e de presença de provedores privados de Internet nos municípios. Esses indicadores, em conjunto com os dados coletados pelo SIMMC, permitiriam identificar se a cobertura dos projetos de inclusão digital do governo corresponde aos locais

Tabela 4: Dicionário de dados de inventário, somente para projeto Telecentro.

Campo	*Descrição*
id_point	Identificação única do PdP
collect_date	Data da coleta, formato YYYY-MM-DD
macaddr	MAC Address, id da máquina no telecentro
disk1_size	Tamanho do disco rígido 1, em GB
disk1_used	Espaço ocupado no disco 1, em GB
disk2_size	Tamanho do disco rígido 2, em GB
disk2_used	Espaço ocupado no disco 2, em GB
machine_type	Indica se é um cliente ou servidor ("client" ou "server")
processor	Modelo e frequência do processador
memory	Tamanho da memória (em MB)
os_type	Sistema operacional (Linux, Windows, etc)
os_distro	Versão do sistema operacional (Ubuntu, XP, etc)
os_kernel	Versão do kernel do sistema operacional

de maior necessidade, guiando assim a expansão dos projetos.

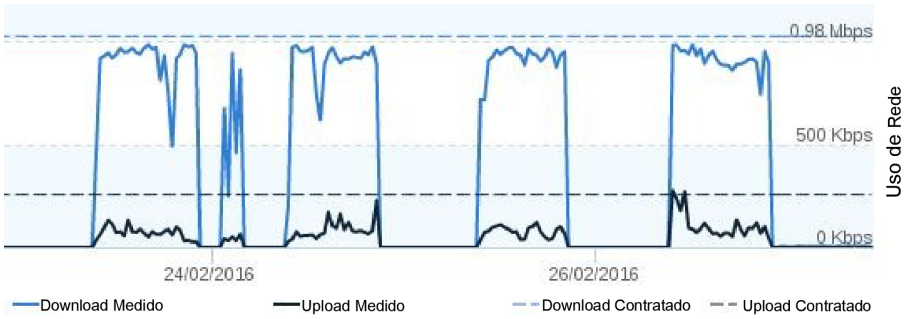
Os telecentros ainda poderiam ter seu padrão de uso analisado. O conhecimento dos dias e horários de maior ocupação auxiliariam na decisão sobre melhores datas para aplicação de cursos e treinamento, além de sugerir aos cidadãos os melhores horários para visitas.

Esses exemplos de uso do SIMMC não são uma descrição exaustiva de todas as aplicações possíveis do sistema. Esperamos que a disponibilização dos dados abertos fomente novas interpretações e criem novas funcionalidades para os dados já coletados.

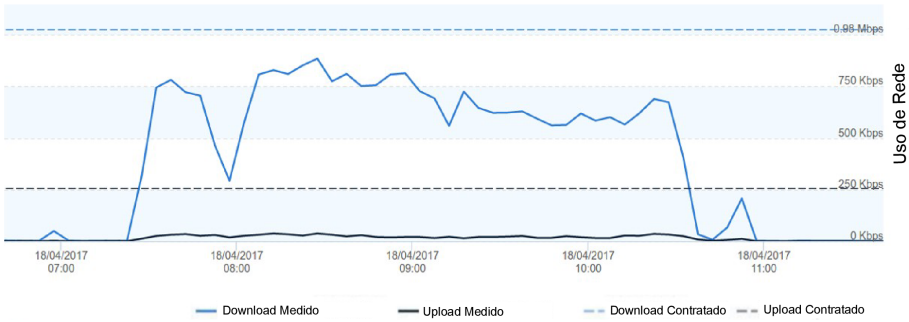
4. Arquitetura do Sistema de Monitoramento

O SIMMC é um sistema de código aberto[9], disponibilizado sob licença GPL[10] e implementado como três módulos principais ilustrados na Figura 4: coleta, armazenamento e visualização. O módulo de coleta é responsável por coletar as métricas dos dispositivos, considerando ambiente altamente heterogêneo dos projetos monitorados. Nos telecentros, o MCTIC fornece computadores e *link* de Internet e deseja obter indicadores sobre seu estado e uso, incluindo disponibilidade, inventário, mudanças de inventário e tráfego de rede. Pontos Gesac e Cidades Digitais recebem roteadores e devem ser monitorados quanto a disponibilidade e uso de rede. A gama de tecnologias envolvidas é variada e, portanto, cada projeto possui uma solução de coleta particular. Para os telecentros, um agente de monitoramento instalado nos computadores envia diariamente os indicadores coletados. Ele foi desenvolvido para funcionar em vários sistemas GNU/Linux e Windows e programado para enviar os dados uma vez ao dia, em horários aleatórios, afim de evitar carga no servidor e uso excessivo da banda local do telecentro. No projeto Gesac o monitoramento é ativo, disparado pelo servidor do SIMMC utilizando protocolo SNMP. Nas Cidades Digitais os PdPs internos são acessíveis somente pelo servidor central da cidade, única ponte com a Internet. Esse servidor coleta os dados da rede interna e os envia

[9]SIMMC código fonte: `https://gitlab.c3sl.ufpr.br/minicom/simmc`.
[10]Licença GPL: `http://www.gnu.org/licenses/gpl.html`



(a) Exemplo de gráfico exibindo uso de rede quando banda medida se aproxima da banda contratada, indicando alto uso de rede.



(b) Exemplo de um plateau abaixo da banda contratada, indício de que o provedor de Internet não está cumprindo com o acordado.

Figura 2

periodicamente ao servidor do SIMMC via protocolo *rsync*, em uma solução híbrida entre os telecentros e conexões Gesac.

O módulo de armazenamento foi implementado no modelo de *Data Warehouse* (DW) [Kimball and Ross 2011] utilizando o SGBD PostgreSQL. A Figura 5 exibe o fluxo dos dados e os principais componentes do DW. A área temporária (AT) é otimizada para inserções e se encarrega de armazenar os dados de coleta até a consolidação noturna, processo que faz a validação dos dados recebidos, adicionando os dados válidos nas ”tabelas fato”e os inválidos na tabela de rejeitados, que pode ser então verificada a posteriori para detecção de problemas nos agentes ou tentativas de ataque. As ”tabelas fato”são por fim sumarizadas nos *Data Marts* (DM), visões dos dados agrupadas para acelerar as consultas realizadas pelo módulo de visualização (portal Web).

O módulo de visualização consiste em uma interface Web acessível publicamente[11], que oferece gráficos, relatórios e mapas, três formas distintas e complementares para visualizar e analisar os dados armazenados. Independente da forma, a navegação segue a lógica de agrupamentos geográficos. O usuário inicia a navegação com a visualização de todo o território nacional e pode então navegar para as regiões, estados e cidades, alcançando no último nível os detalhes de um PdP específico. Tal padronização permite a troca intuitiva entre relatórios, gráficos e mapas, de acordo com necessidade e

[11]Portal do SIMMC: `http://simmc.c3sl.ufpr.br`.

	Sistema Operacional	Processador	Memória	Disco
Kit - Centro Inclusão Digital				
MAC Address - 00:27:13:ae:18:02				
05/02/14	Ubuntu 12.04.4 LTS	Intel(R) Core(TM) i5 CPU 650 @ 3.20GHz	3.62 GiB	465 GB
07/08/14	Ubuntu 12.04.4 LTS	Intel(R) Core(TM) i5 CPU 650 @ 3.20GHz	1.71 GiB	465 GB

Figura 3: Relatório de alteração de inventário exibindo redução na quantidade de memória de um computador do telecentro "Kit - Centro Inclusão Digital".

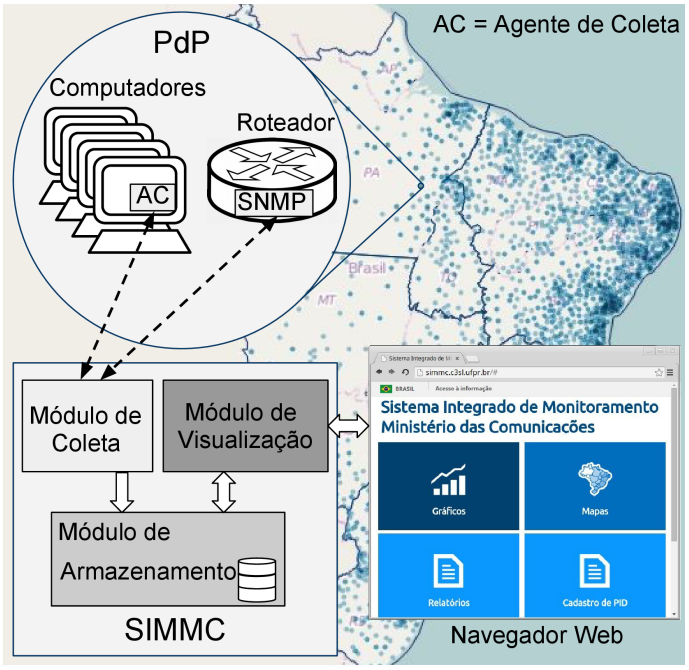


Figura 4: Arquitetura geral do SIMMC.

interesse do usuário. A Figura 4 exibe todos os PdPs monitorados, representados pelos pontos em azul, com a tonalidade variando de acordo com sua concentração (quanto mais PdPs, mais escuro o azul). A Figura 6 exibe um recorte da página de gráficos, com o histórico do número de PdPs do projeto Gesac ao longo dos últimos seis meses.

5. Desafios e limitações

Os principais desafios no desenvolvimento do SIMMC estão relacionados a heterogeneidade dos equipamentos monitorados e armazenamento do grande volume de dados coletados e analisados diariamente. A implementação inicial do mecanismo de consolidação dos dados e geração dos *Data Marts* (Seção 4) apresentou degradação contínua de performance seguindo acúmulo de dados e adição de novos PdPs. O problema de escalabilidade fica evidente na Figura 7 quando, no final de maio, o tempo de execução do processo de consolidação ultrapassou 800 minutos (13 horas). A redução drástica vista em seguida se deu através da aplicação de duas medidas, que reduziram o tempo de consolidação para estáveis 40 minutos. Primeiramente, um processo de de-normalização foi empregado para reduzir o número de junções e acelerar as consultas e agregações [Hoffer et al. 2004]. Si-

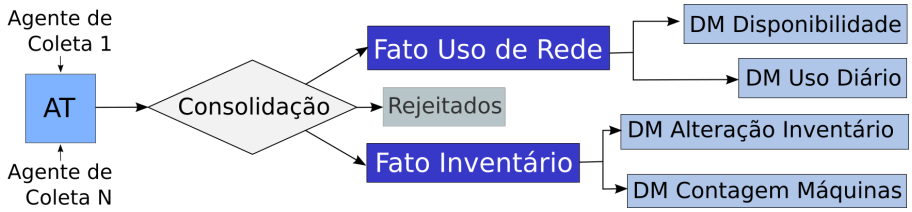


Figura 5: Amostra do fluxo dos dados no módulo de armazenamento.

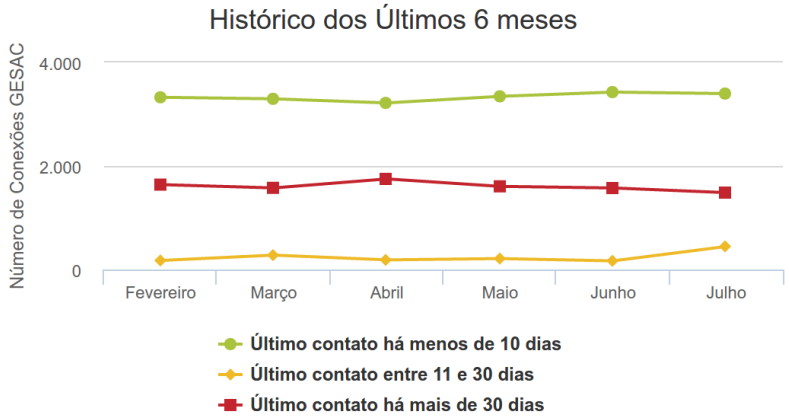


Figura 6: Fluxo dos dados no módulo de armazenamento.

multaneamente, foram feitas modificações para somente atualizar os *Data Marts* ao invés de gerá-los integralmente a cada execução do processo [Mumick et al. 1997], como a modelagem tradicional de DW sugere. Outra limitação de sistema consiste em sua arquitetura, específica para os projetos do MCTIC. Estamos estudando soluções de armazenamento e implementações que permitam maior generalização, considerando modelos *NoSQL* [Pasqualin et al. 2016], assim como *Polystores* [Duggan et al. 2015], procurando garantir simultaneamente flexibilidade e desempenho na análise de grandes volumes de dados.

Além dos desafios da ferramenta, podemos citar desafios e limitações dos dados. Já foi mencionada a dificuldade de garantir que a banda de Internet contratada para o projeto Gesac esteja de fato sendo entregue, devido à impossibilidade atual em se saturar o *link* para verificação. A medição do uso de rede possui uma limitação mais relevante no projeto Telecentro, onde as métricas são coletadas pelo agente de monitoramento instalado em cada computador. Apesar dos esforços para incentivar sua instalação[12], não há garantia de que todos os dispositivos conectados à rede do Telecentro estejam de fato sendo monitorados, o que pode causar distorções no cálculo de uso de rede total.

6. Conclusão

Nesse artigo apresentamos os repositórios de dados criados a partir do SIMMC, um sistema de monitoramento e transparência pública, utilizado pelo MCTIC para avaliar o desempenho e projetar a expansão dos seus projetos de inclusão digital. Foram detalhados os dados contidos nesses repositórios, a forma como são coletados, como lidamos

[12]O agente de monitoramento é embarcado no sistema operacional que acompanha os computadores. Além disso o sistema de gerência do Telecentro solicita instalação do agente a cada inicialização.

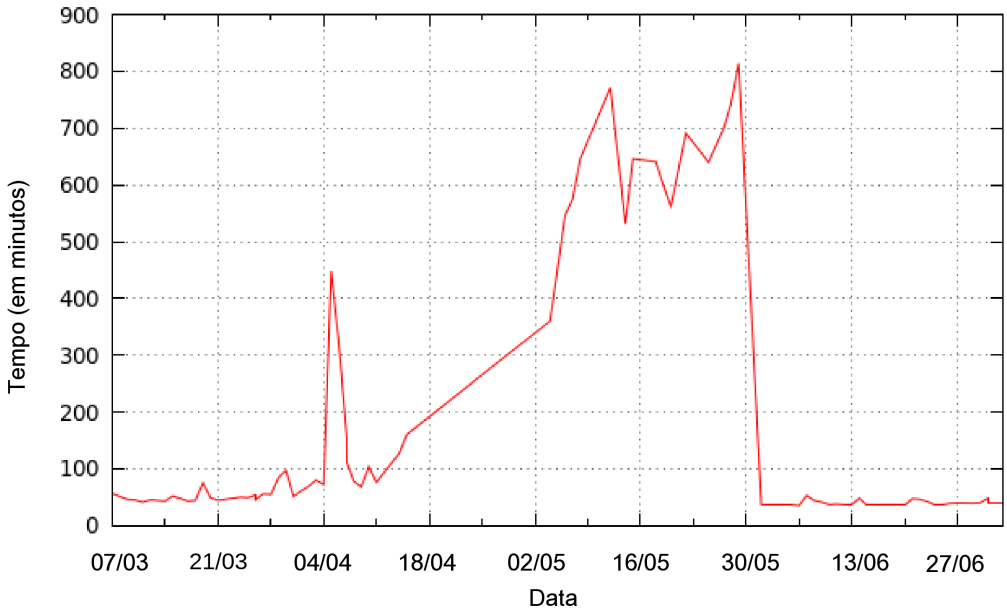


Figura 7: Tempo de execução do processo de consolidação dos dados.

com eles no SIMMC e a geração e organização dos arquivos nos repositórios.

Esperamos que a publicação dos dados em formato aberto ampliem o alcance social do sistema, fomentando o desenvolvimento de aplicações de terceiros com novas interpretações dos indicadores coletados.

Referências

Duggan, J., Elmore, A. J., Stonebraker, M., Balazinska, M., Howe, B., Kepner, J., Madden, S., Maier, D., Mattson, T., and Zdonik, S. (2015). The bigdawg polystore system. *ACM Sigmod Record*, 44(2):11–16.

Hoffer, J. A., Prescott, M., and McFadden, F. (2004). *Modern Database Management (7th Edition)*. Prentice-Hall, Inc., Upper Saddle River, NJ, USA.

Kimball, R. and Ross, M. (2011). *The data warehouse toolkit: the complete guide to dimensional modeling*. John Wiley & Sons.

Mumick, I. S., Quass, D., and Mumick, B. S. (1997). Maintenance of data cubes and summary tables in a warehouse. In *ACM Sigmod Record*, volume 26, pages 100–111. ACM.

Pasqualin, D., Bona, L., Trois, C., Didonet, M., Sunye, M., Almeida, C., Castilho, M., Weingaertner, D., Maciel, E., and Tissot, H. (2017 no prelo). Transparency meets management: a monitoring and evaluating tool for governmental projects". In *14th ACS/IEEE International Conference on Computer Systems and Applications AICCSA 2017*. IEEE.

Pasqualin, D., Souza, G., Buratti, E. L., de Almeida, E. C., Del Fabro, M. D., and Weingaertner, D. (2016). A case study of the aggregation query model in read-mostly nosql document stores. In *Proceedings of the 20th International Database Engineering & Applications Symposium*, pages 224–229. ACM.

Deduplicação de Nomes e Redes de Co-autoria na DBLP

Mariana O. Silva, Michele A. Brandão

Universidade Federal de Minas Gerais (UFMG) – Belo Horizonte, MG – Brasil

{mariana.santos,micheleabrandao}@dcc.ufmg.br

Abstract. *This article describes a dataset collected from the DBLP digital library, a repository with bibliographic data of Computer Science. This dataset includes approximately 15 million records collected in September 2016. From this dataset, two datasets were created. The first one has the original data collected from the DBLP with name deduplication treatment. The second one presents three co-authorship social networks built using the snowball sampling technique.*

Resumo. *Este artigo descreve um dataset coletado da biblioteca digital DBLP, um repositório com dados bibliográficos de Ciência da Computação. Este conjunto de dados inclui aproximadamente 15 milhões de registros coletados em Setembro de 2016. A partir deste dataset, foram criados dois conjuntos de dados. O primeiro possui os dados originais coletados da DBLP com tratamento de deduplicação de nomes. O segundo apresenta três redes sociais de coautoria construídas utilizando a técnica snowball sampling.*

1. Introdução

Bibliotecas Digitais (BDs) são repositórios organizados de uma ou mais coleções online, que provêm acesso à informações e conhecimento para seus usuários. Tais repositórios proporcionam diversos serviços, como pesquisa, visualização dos dados e armazenamento de metadados que descrevem seu conteúdo e suas interações. No contexto acadêmico, as bibliotecas digitais são consideradas importantes fontes de informação, fornecendo uma interface centralizada para o acesso a diversas publicações científicas.

Muitos estudos relevantes são realizados a partir dos dados coletados desses repositórios [Brandão and Moro 2017]. Por exemplo, para avaliar a qualidade e o impacto das publicações [Omodei et al. 2017; Weitzel 2006], identificar relevantes temas de pesquisa [Ohira and Prado 2002; Villarreal and Schaeffer 2016], revelar tendências e padrões de colaboração em redes sociais de coautoria [Brandão and Moro 2012; Brandão et al. 2013; Chen et al. 2017], dentre outros. Em particular, estudos focados na análise de interações entre pessoas ou organizações, bem como detectar padrões presentes nessas interações permitem prever o comportamento de uma rede e analisar diferentes aspectos da mesma.

Esses estudos podem ser usados por agências de fomento ou instituições de pesquisa e para tal pressupõem-se que os dados sejam de alta qualidade [Laender et al. 2008; Lee et al. 2007]. Porém, manter essa alta consistência geralmente não é uma tarefa simples. Um dos principais e mais complexos desafios enfrentados para melhorar a qualidade dos dados é a deduplicação de nomes [Laender et al. 2008]. Esse problema pode ocorrer de duas formas: quando um mesmo autor publica utilizando nomes similares, mas distintos (sinônimos); ou quando autores diferentes compartilham o mesmo nome, ou variações parecidas (homônimos).

Nesse contexto, a DBLP[1] (*Digital Bibliography & Library Project*) é um exemplo de biblioteca digital que possui informações bibliográficas sobre as principais publicações de Ciência da Computação. Essa biblioteca armazena dados de pesquisadores da área da computação (ou ciências vizinhas) de todo o mundo. Em setembro de 2016, essa coleção possuía cerca de 1,780,000 autores e 3,400,000 publicações. Dessa forma, a DBLP proporciona dados bibliográficos reais e úteis que podem auxiliar na análise de redes sociais acadêmicas. No entanto, a presença de sinônimos e homônimos é o problema principal. Por sua amplitude e representatividade, escolhemos a DBLP como fonte de coleta para construir o conjuntos de dados apresentado neste artigo.

Após descrever os trabalhos relacionados e aplicações do conjunto de dados (Seção 2), as principais contribuições deste artigo são: uma metodologia para construir dois conjuntos a partir dos dados da DBLP (Seção 3); uma descrição detalhada e quantitativa dos dois conjuntos além da disponibilização online dos mesmos (Seções 4 e 5); Finalmente, apresentamos as conclusões e trabalhos futuros (Seção 6).

2. Trabalhos Relacionados e Aplicações

As bibliotecas digitais são sistemas de informação extremamente complexos que envolvem conjuntos de objetos digitais e seus respectivos metadados [Gonçalves et al. 2004]. Esses dados podem ser provenientes de fontes variadas, mas relativos a uma mesma área de interesse e possuem o propósito de atender a uma determinada comunidade [Borgman 1999]. O vasto conteúdo presente nessas bibliotecas digitais podem conduzir a análises de dados interessantes, tais como tendências de pesquisa [Ferreira 2012], padrões de colaboração em redes sociais [Freitas et al. 2008], predição de links [Hasan et al. 2006], recomendação de colaborações [Brandão et al. 2013], pesquisas em qualidade da informação [Han et al. 2004], entre outras.

Dentre os diversos campos de pesquisa, a análise de redes sociais tem se tornado um assunto extremamente abordado e relevante. Trabalhos com as mais diversas finalidades têm sido realizados para analisar diferentes aspectos de uma rede social. Tais redes podem mostrar padrões de cooperação entre pesquisadores e o impacto de suas publicações [Börner et al. 2005], podem ser utilizadas para recomendação de colaboração [Brandão and Moro 2012] e avaliar grupos de pesquisa e programas de pós-graduação [Lopes et al. 2011]. Outro exemplo de aplicação de dados derivados de BDs é a análise da qualidade de agrupamento [Brandão and Moro 2017]. Ademais, utilizamos as redes sociais de co-autoria do conjunto de dados apresentado neste artigo para avaliar se métricas para força de relacionamentos podem ser usadas também para avaliar a qualidade de comunidades[2].

Geralmente, redes sociais acadêmicas apresentam uma estrutura grande e volumosa de dados, o que impossibilita uma análise manual detalhada. Para analisar essas redes, é necessário desenvolver métodos que possam tratar este grande volume de dados. Dessa forma, usuários e desenvolvedores enfrentam diversos desafios ao realizarem pesquisas mais precisas e detalhadas de tais redes. Diante desses desafios, realizamos um estudo que especifica e valida modelos para as redes sociais acadêmicas, incluindo a definição de uma infraestrutura de banco de dados que permite armazenar e manter os dados das redes[3]. Tal

[1]DBLP: `http://dblp.uni-trier.de/`

[2]Relatório técnico em `http://www.dcc.ufmg.br/~mirella/projs/apoena`

[3]Relatório técnico em `http://www.dcc.ufmg.br/~mirella/projs/apoena`

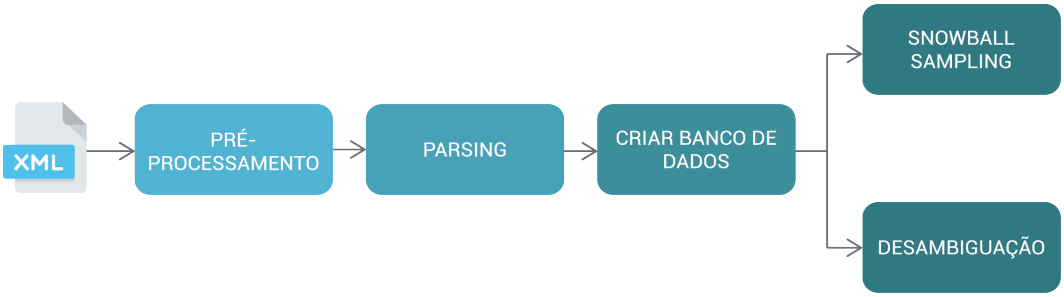


Figura 1: Etapas da metodologia.

infraestrutura também facilita a utilização de técnicas de análise de redes sociais. Esse estudo compara o desempenho de diferentes tipos de SGBDs (Sistemas de Gerenciamento de Bancos de Dados) utilizando o conjunto de dados descrito na Seção 4.

A DBLP é uma abrangente fonte de pesquisas científicas em Ciência da Computação, capaz de facilitar a análise de dados em diversas áreas. Por exemplo, Yang and Leskovec [2015] estudam um conjunto de 230 redes sociais grandes, onde uma rede científica de colaboração foi criada a partir dos dados coletados da DBLP. Os autores apresentam uma metodologia que compara e avalia quantitativamente as diferentes definições estruturais de comunidades. Verificou-se que o método proposto apresenta uma melhoria relativa de 30% em relação a métodos de agrupamento local do estado da arte. Por outro lado, Lange and Naumann [2011] utilizam o conjunto de dados da DBLP para avaliar uma abordagem proposta por eles para medir a semelhança de dois registros.

Finalmente, além de pesquisas voltadas à análise de redes sociais de co-autoria e problemas de deduplicação e desambiguação de dados, novos desafios de pesquisa também podem se beneficiar a partir dos dados coletados e tratados neste trabalho. Alguns exemplos importantes são análises de agrupamento de comunidades, formação de times na pesquisa científica e recomendação de colaborações.

3. Metodologia

A DBLP começou como uma pequena e limitada base de dados, mas tornou-se uma grande biblioteca digital contendo trabalhos de quase todos os campos de estudo em computação. Todo o conjunto de dados da DBLP está disponível online como um grande arquivo XML. O arquivo *dblp.xml* (um simples arquivo XML ASCII) contém todos os registros bibliográficos presentes na biblioteca e é acompanhado pelo arquivo *dblp.dtd*, um conjunto de regras que define quais tipos de dados e entidades fazem parte do documento XML.

A Figura 1 apresenta as etapas da metodologia, desde a obtenção do arquivo XML até a criação dos dois conjuntos de dados. Antes de realizar a leitura do *dblp.xml*, realizamos um pré-processamento para remover a codificação utilizada para caracteres especiais. Com o arquivo tratado, executamos o *parser* do mesmo. Em seguida, foram gerados dois conjuntos de dados a partir dos dados coletados: uma base de dados com nomes desambiguados e outra com três redes sociais de co-autoria.

A identificação de registros duplicados é um processo complexo e composto de várias etapas. A Figura 2 apresenta uma visão geral do processo de deduplicacão de acordo com a abordagem *Data Deduplication* [Christen 2012]. Assim, para eliminarmos

Figura 2: Processo de deduplicação dos dados.

os registros duplicados, seguimos as seguintes etapas: *(1)* pré-processamento responsável por dividir os nomes dos autores em primeiro nome, nome do meio e último nome; *(2)* indexação de todos os registros por uma chave de bloco (BK), utilizando a técnica *Soundex* [Odell and Russell 1918]; *(3)* comparação de todos os registros pertencentes a cada um dos blocos por meio da função de similaridade *Jaro Winkler* [Winkler 1990]; *(4)* classificação dos registros como duplicados, não duplicados e como possíveis duplicados de acordo com um limiar de similaridade.

Para a criação das redes, utilizamos o dataset original, sem o tratamento de nomes ambíguos. Além disso, foi utilizada a técnica de amostragem não probabilística conhecida como *snowball sampling* para filtrar e diminuir seu volume [Goodman 1961]. Essa técnica foi escolhida por ser mais direcionada do que outras técnicas de amostragem não aleatórias [Pearson 2012]. Para adquirir uma amostra, é necessária uma "semente", geralmente de indivíduos conhecidos envolvidos no comportamento sob análise [Snijders et al.]. Aqui, os nós sementes escolhidos foram os bolsistas vigentes do CNPq[4] de Ciência da Computação. Note que para resolver o problema de nomes não padronizados na DBLP, buscamos manualmente o perfil dos bolsistas. Assim, garantimos a corretude das informações de cada um deles na rede. A partir dos nós sementes, foram feitas duas coletas para aumentar a amostra e criar a três redes reais de tamanhos diferentes. Ao final, são três redes criadas a partir da DBLP: (*Rede 0*) formada apenas pelos bolsistas vigentes do CNPq, que fazem parte da DBLP; (*Rede 1*) formada pela *Rede 0* e seus vizinhos; e (*Rede 2*) formada pela *Rede 1* e seus vizinhos.

4. Descrição dos Dados Coletados

De acordo com Seção 3, os dados foram coletados do arquivo *dblp.xml* disponível na DBLP. Este arquivo é modelado a partir do formato BibTeX *.bib. Existem dois tipos de registros neste arquivo: registros de publicação e registros de pessoas. Os registros de publicação são fornecidos por um dos seis elementos: *article*, artigo em periódico ou revista; *inproceedings*, artigo publicado em conferência ou workshop; *proceedings*, volume de trabalhos de uma conferência ou workshop; *book*, autoria de monografia ou coleção editada de artigos; *incolletion*, parte ou capítulo em uma monografia; *phdthesis*, tese de doutorado; *masterthesis*, tese de mestrado; *www*, página da web.

Para formar o conjunto de dados, consideramos apenas os dados referentes às pessoas (autores e editores) e suas publicações. Para simplificar e filtrar o dataset, foram

[4]`http://cnpq.br/bolsistas-vigentes/` (Abril de 2017)

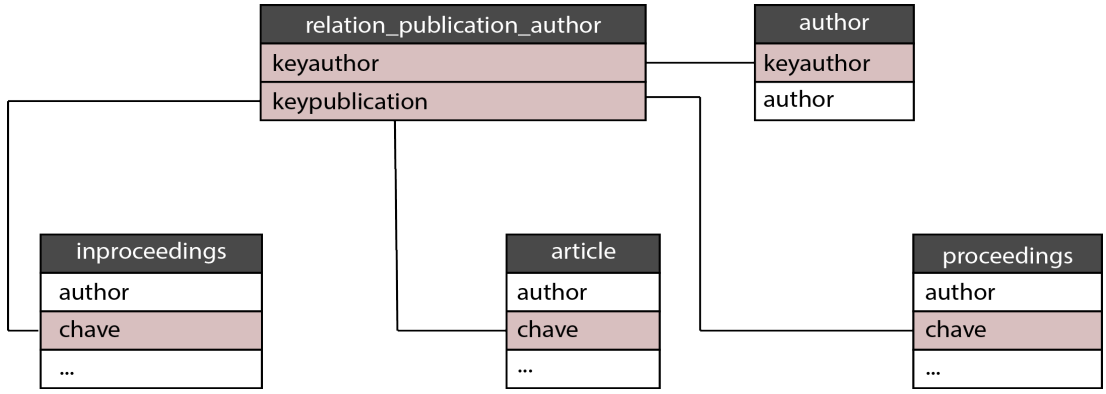


Figura 3: Esquema do banco de dados relacional. Note que exibimos apenas os principais atributos das tabelas devido a restrições de espaço.

Tabela 1: Descrição das redes sociais criadas.

Rede	# autores	# pares (# dist)	MedPubA	Modularidade	Coeficiente de clusterização médio
0	394	3898 (738)	9,89	0,686	0,377
1	68397	249352 (110357)	3,65	0,86	0,672
2	314444	1297929 (540571)	4,13	0,691	0,524

coletadas publicações originadas apenas dos elementos: *article*, *inproceedings* e *proceedings*. Além disso, criamos uma tabela que relaciona autores (ou editores) com suas publicações. O conjunto de dados coletado foi armazenado em um sistema de gerenciamento de banco de dados relacional (SGBDR), mais especificamente o MySQL. O esquema de dados relacional possui 5 tabelas conforme apresentado na Figura 3. Na tabela *author* estão armazenados o nome e o id dos pesquisadores. As tabelas *article*, *inproceedings* e *proceedings* possuem informações detalhadas de publicações. Finalmente, a tabela *relation_publication_author* representa o relacionamento entre autores e publicações, que podem estar em *article*, *proceedings* ou *inproceedings*.

Em seguida, foi realizado um processo de deduplicação de nomes e construção das três redes sociais. A Tabela 1 descreve as principais propriedades de cada rede social criada, que são o número de pesquisadores (autores de artigos), número de publicações, número médio de publicações por autor, número de pares de co-autores (e número de pares distintos de co-autores), modularidade da rede e o coeficiente de clusterização médio. A modularidade é uma medida capaz de determinar a qualidade da divisão feita na rede, enquanto o coeficiente de clusterização médio representa o valor médio do grau com que os nós de uma rede (ou grafo) tendem a se agrupar.

Para ilustrar a estrutura das redes sociais de co-autoria construídas a partir dos dados coletados da DBLP, a Figura 4 exemplifica a rede origem de coautoria com apenas pesquisadores vigentes do CNPq (*Rede 0*). Os nós do grafo representam os autores e cada aresta representa a colaboração entre dois autores. O tamanho do nó é proporcional ao seu grau, ou seja, ao número de arestas adjacentes a ele. A espessura da aresta é proporcional ao número de publicações que dois autores possuem. Já a cor dos nós identifica o componente conectado a que pertencem. Observa-se que o maior componente conectado da rede é

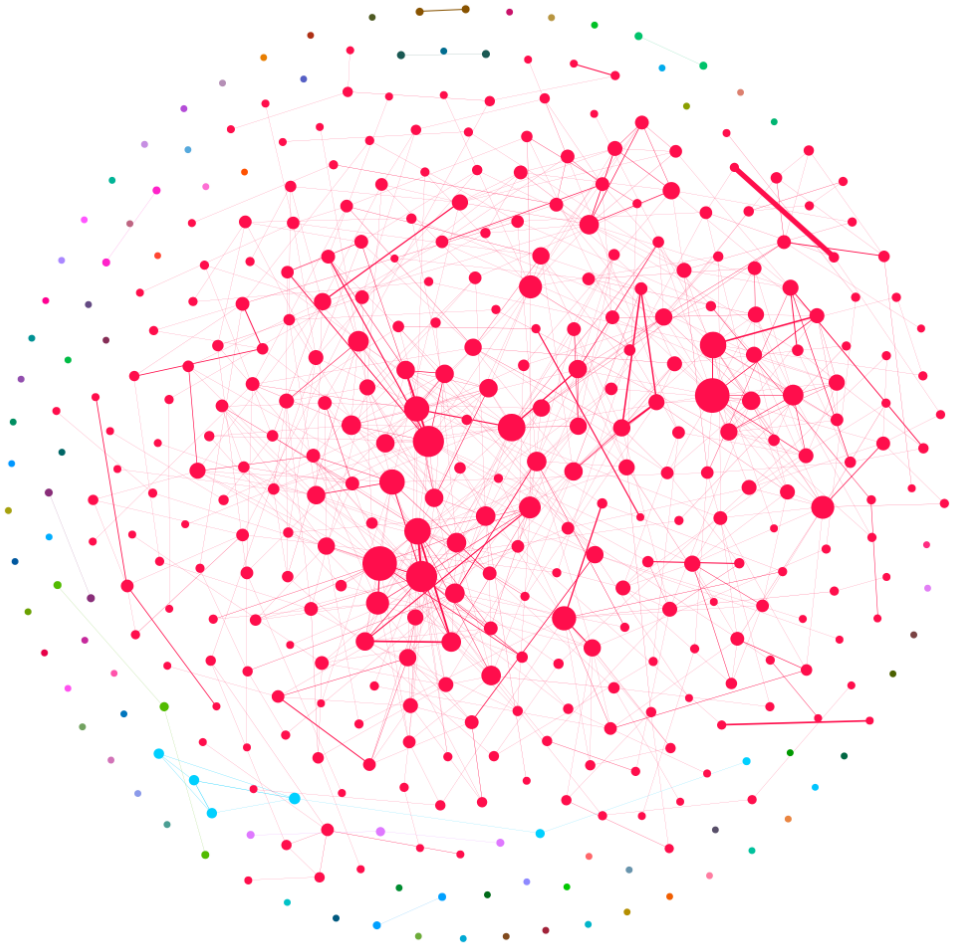


Figura 4: Representação gráfica da *Rede 0*.

grande (em vermelho), abrangendo cerca de 76% da rede, o que significa tratar-se de uma comunidade bem conectada.

O conjunto de dados completo está disponível na página do projeto Apoena[5] e é composto por três conjuntos de dados compactados:

- **Dblp.zip** - contém os dados coletados sem a deduplicação de nomes dos pesquisadores;
- **Dblp_name_desambiguation.zip** - contém os dados coletados com nomes ambíguos resolvidos;
- **Dblp_social_networks.zip** - contém os dados das três redes de coautoria.

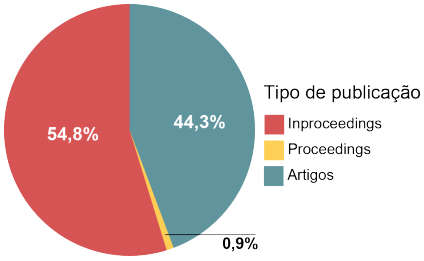
5. DBLP em Números

O conjunto de dados coletado por meio da metodologia descrita na Seção 3 consiste em aproximadamente 15 milhões de registros. A Tabela 2 apresenta uma descrição quantitativa do conjunto de dados. A Figura 5(a) mostra a distribuição dos tipos de publicações

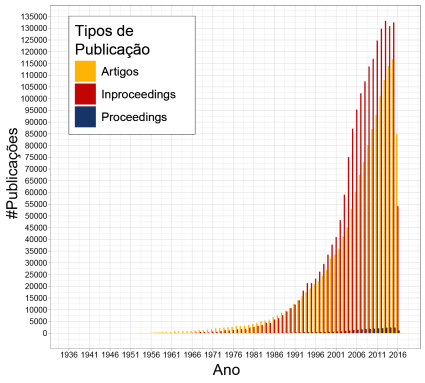
[5]`http://homepages.dcc.ufmg.br/~mirella/projs/apoena/datasets.html`

Tabela 2: Conjunto de dados coletado (16 de Setembro de 2016).

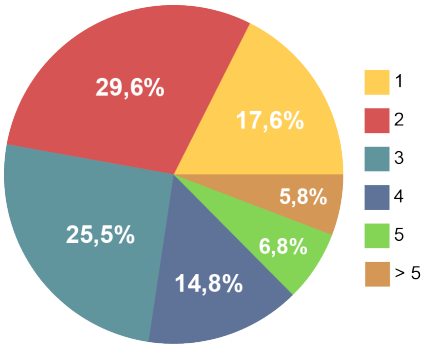
Dados	**Número de registros**
Publicações em artigos	1.505.020
Autores	1.779.971
Publicações em proceedings	31.549
Publicações em inproceedings	1.861.226
Relação entre autores e publicações	9.707.161
Total	**14.884.927**



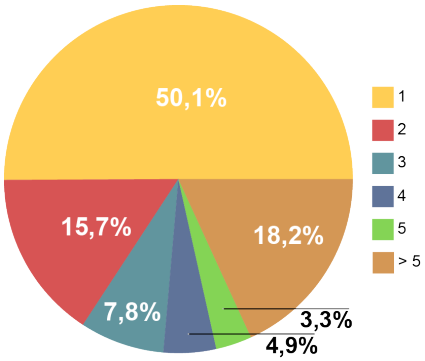
(a) Distribuição dos tipos de publicação



(b) Publicações por ano (1936−2017)



(c) Número de coautores por publicação



(d) Número de publicações por autor

Figura 5: Estatísticas sobre o conjunto de dados.

presentes no conjunto de dados. Observa-se que aproximadamente 55% das publicações são provenientes de conferências ou workshops (*inproceedings*), cerca de 44% são artigos de um periódico ou revista e apenas 0,9% representam *proceedings*. Ou seja, a maioria das publicações presentes no conjunto de dados são de artigos ou *inproceedings*.

A Figura 5(b) apresenta a evolução do número de publicações ao longo dos 81 anos da DBLP (1936 - 2017). Para este diagrama, as publicações foram agrupadas por seu tipo e ano de publicação. É possível notar que o número de publicações originadas de artigos e *inproceedings* estão correlacionados. Porém, o número de publicações em *inproceedings* cresce mais rapidamente que os outros tipos de publicação. Além disso, observa-se que a partir dos anos 90 houve um grande crescimento do número de publicações. Provavelmente,

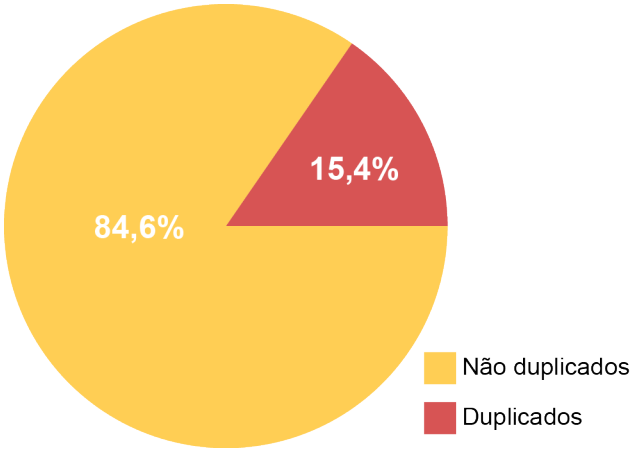


Figura 6: Distribuição dos registros duplicados no conjunto de dados.

isso pode ser explicado pelo uso da Internet que se difundiu nessa época.

A Figura 5(c) mostra a distribuição do número de co-autores por publicação. Percebe-se que cerca de 30% das publicações (seja de *articles* ou *inproceedings*) têm dois co-autores e 25,5% têm três co-autores. Além disso, apenas 5,8% das publicações apresentam mais de 5 co-autores. O maior número de pesquisadores em uma publicação é 287. Uma análise semelhante pode ser feita verificando-se a média de publicações por autor. A Figura 5(d) apresenta essa análise. Observa-se que aproximadamente metade da comunidade de autores publicam apenas uma única vez na DBLP, cerca de 16% publicam duas vezes e mais de 18% publicam mais de cinco vezes.

Segundo Lee et al. 2007, os desafios e limitações relacionados à qualidade dos dados presentes nas bibliotecas digitais têm várias origens. Por exemplo, erros na entrada de dados, ausência de padrões de execução, imperfeição de softwares de coleta, geração de metadados em larga escala, ambiguidade de nomes de autores, entre outros. Dentre esses, o problema de ambiguidade de nomes de autores vem se destacando na comunidade científica, devido à sua inerente dificuldade. Conforme Seção 3, para a criação de um dos conjuntos de dados, aplicamos um processo de deduplicação de acordo com a abordagem *Data Deduplication*. Para isso, foi utilizada uma função de deduplicação com limiar de 95% [Christen 2012], fazendo com que fosse possível alcançar resultados satisfatórios. Após o processo, foram encontrados 289.598 registros duplicados, reduzindo o número de autores de 1.779.971 para 1.593.237. Na Figura 6, podemos ver a distribuição dos registros duplicados detectados no conjunto de dados, abrangendo mais de 15% dos dados.

6. Conclusões

Neste artigo, apresentamos dois conjuntos de dados a partir dos dados da DBLP. Um conjunto possui os nomes dos autores deduplicados, enquanto que o outro é composto por três redes sociais de co-autoria. Ademais, apresentamos as principais aplicações, desafios e limitações desses conjuntos. Como trabalhos futuros, planejamos construir mais redes sociais de co-autoria considerando diferentes perfis de pesquisadores na DBLP. Além disso, pretendemos aprimorar a qualidade dos dados coletados, tanto explorando novas técnicas de deduplicação de nomes propostas na literatura, quanto incluindo metadados aos conjuntos de dados para melhorar a sua compreensão e facilitar o seu uso.

Agradecimentos. Trabalho parcialmente financiado por CAPES, CNPq e FAPEMIG.

Referências

Borgman, C. L. (1999). What are digital libraries? competing visions. *Inf. Process. Manage.*, 35(3):227–243.

Börner, K., Dall'Asta, L., Ke, W., and Vespignani, A. (2005). Studying the emerging global brain: Analyzing and visualizing the impact of co-authorship teams. *Complexity*, 10(4):57–67.

Brandão, M. A. and Moro, M. M. (2017). Social professional networks. *Computer Communications*, 100(C):20–31.

Brandão, M. A. and Moro, M. M. (2012). Recomendação de colaboração em redes sociais acadêmicas baseada na afiliação dos pesquisadores. In *Procs. of SBBD (Short Papers) - Simpósio Brasileiro de Bancos de Dados*, pages 73–80.

Brandão, M. A. and Moro, M. M. (2017). Strength of Co-authorship Ties in Clusters: a Comparative Analysis . In *Procs. of AMW - Alberto Mendelzon International Workshop on Foundations of Data Management*, Montevideo, Uruguai.

Brandão, M. A., Moro, M. M., and Almeida, J. M. (2013). Análise de fatores impactantes na recomendação de colaborações acadêmicas utilizando projeto fatorial. In *Procs. of SBBD (Short Papers) - Simpósio Brasileiro de Bancos de Dados*, pages 5:1–5:6.

Chen, Y., Ding, C., Hu, J., Chen, R., Hui, P., and Fu, X. (2017). Building and analyzing a global co-authorship network using google scholar data. In *Procs. of WWW - International Conference on World Wide Web Companion*, pages 1219–1224, Perth, Austrália.

Christen, P. (2012). A survey of indexing techniques for scalable record linkage and deduplication. *IEEE transactions on knowledge and data engineering*, 24(9):1537–1555.

Ferreira, A. A. (2012). *Contributions for solving the author name ambiguity problem in bibliographic citations*. PhD thesis, Universidade Federal de Minas Gerais.

Freitas, C., Nedel, L. P., Galante, R., Lamb, L. C., Spritzer, A. S., Fujii, S., de Oliveira, J. P. M., Araújo, R. M., and Moro, M. M. (2008). Extração de conhecimento e análise visual de redes sociais. *SEMISH - Seminário Integrado de Software e Hardware, Belém do Pará, Brasil, SBC*, pages 106–120.

Gonçalves, M. A., Fox, E. A., Watson, L. T., and Kipp, N. A. (2004). Streams, structures, spaces, scenarios, societies (5s): A formal model for digital libraries. *ACM transactions on information systems (TOIS)*, 22(2):270–312.

Goodman, L. A. (1961). Snowball sampling. *The annals of mathematical statistics*, pages 148–170.

Han, H., Giles, L., Zha, H., Li, C., and Tsioutsiouliklis, K. (2004). Two supervised learning approaches for name disambiguation in author citations. In *Procs. of JCDL - Joint ACM/IEEE conference on Digital Libraries*, pages 296–305, Tucson, USA.

Hasan, M. A., Chaoji, V., Salem, S., and Zaki, M. (2006). Link prediction using supervised learning. In *Procs. of SDM - Workshop on Link Analysis, Counterterrorism and Security*.

Laender, A. H., Gonçalves, M. A., Cota, R. G., Ferreira, A. A., Santos, R. L., and Silva, A. J. (2008). Keeping a digital library clean: new solutions to old problems. In *Procs. of DocEng - ACM symposium on Document engineering*, pages 257–262. ACM.

Lange, D. and Naumann, F. (2011). Frequency-aware similarity measures: why arnold schwarzenegger is always a duplicate. In *Proceedings of the 20th ACM international conference on Information and knowledge management*, pages 243–248. ACM.

Lee, D., Kang, J., Mitra, P., Giles, C. L., and On, B.-W. (2007). Are your citations clean? *Communications of the ACM*, 50(12):33–38.

Lopes, G. R., Moro, M. M., Da Silva, R., Barbosa, E. M., and de Oliveira, J. P. M. (2011). Ranking strategy for graduate programs evaluation. *Procs. of ICITA - International Conference on Information Technology and Applications*, pages 59–64.

Odell, M. and Russell, R. (1918). The soundex coding system. *US Patents*, 1261167.

Ohira, M. L. B. and Prado, N. S. (2002). Bibliotecas virtuais e digitais: análise de artigos de periódicos brasileiros (1995/2000). *Ciência da Informação*, 31(1):61–74.

Omodei, E., De Domenico, M., and Arenas, A. (2017). Evaluating the impact of interdisciplinary research: A multilayer network approach. *Network Science*, 5(2):235–246.

Pearson, M. (2012). Social network analysis: An overview. *Social Networks*.

Snijders, T. A., Steglich, C. E., and Schweinberger, M. Modeling the co-evolution of networks and behavior. In van Montfort, K., Oud, H., and Satorra, A., editors, *Longitudinal Models in the Behavioral and Related Sciences*. Lawrence Erlbaum.

Villarreal, S. E. G. and Schaeffer, S. E. (2016). Local bilateral clustering for identifying research topics and groups from bibliographical data. *Knowledge and Information Systems*, 48(1):179–199.

Weitzel, S. d. R. (2006). O papel dos repositórios institucionais e temáticos na estrutura da produção científica. *Em Questão*, 12(1).

Winkler, W. E. (1990). String comparator metrics and enhanced decision rules in the fellegi-sunter model of record linkage. In *Procs. of the Section on Survey Research*, pages 354–359.

Yang, J. and Leskovec, J. (2015). Defining and evaluating network communities based on ground-truth. *Knowledge and Information Systems*, 42(1):181–213.

FiSmo: A Compilation of Datasets from Emergency Situations for Fire and Smoke Analysis

Mirela T. Cazzolato, Letricia P. S. Avalhais, Daniel Y. T. Chino
Jonathan S. Ramos, Jessica A. de Souza,
Jose F. Rodrigues-Jr, Agma J. M. Traina

[1] Institute of Mathematics and Computer Science
University of Sao Paulo
Sao Carlos, Brazil

{jessicasouza, jonathan, mirelac}@usp.br
{agma, chinodyt, junio, letricia}@icmc.usp.br

Abstract. *In this work, we present FiSmo, a compilation of datasets from emergency situations, composed of images, videos, regions of interest (ROIs), annotations, and features. These datasets were employed in the context of the RESCUER Project; they were used in the experimental analysis of techniques created in a set of works carried out at the Databases and Images Group (GBDI) of the University of Sao Paulo. These works were focused on the analysis of images and videos regarding the presence of fire, smoke, and explosions in emergency situations. The available data is composed of four image and two video datasets: fire/smoke detection in images; fire segmentation in images; smoke segmentation in images; content-based image retrieval; temporal segmentation of fire segments in videos; and fire detection in videos. All datasets were preprocessed according to the involved context, including annotation steps carried out by a set of subjects, training images and ROIs. Furthermore, the extracted feature vectors are also available, providing features of color and texture. FiSmo can be employed for experimentation of computational techniques and systems designed to work with images and videos from emergency situations.*

1. Introduction

Digital images and videos have been used in many fields of study. Nowadays, a massive number of image data is available on the internet due to the explosion of mobile devices, which captures and uploads images and videos to the cloud. To take advantage of this, the RESCUER Project[1] was developed to support the analysis of information regarding crises in large scale events using crowd-sourced data: images, videos, and text captured and sent by users. The Databases and Images Group (GBdI)[2], from the Institute of Mathematics and Computer Science of the University of Sao Paulo (ICMC-USP), is responsible for the image and video analysis functionalities of the RESCUER architecture. The computational system has to work in real-time to produce accurate and reliable information. Late responses or decisions based on inaccurate information may lead to financial losses

[1]RESCUER Project: Reliable and Smart Crowdsourcing Solution for Emergency and Crisis Management – www.rescuer-project.org

[2]GBdI: Databases and Images Group – www.gbdi.icmc.usp.br

and/or injuries. Therefore, the most relevant images and videos have to be identified as soon as possible. The relevant images and videos are the ones that pose emergency situations and can effectively assist in decision making. Defining the proper dataset to evaluate fire, smoke and explosion detection algorithms is a crucial step.

In this paper, we divulge a compilation of datasets of images and videos that present emergency scenarios called FiSmo[3]. Considering the high cost of making simulations in emergency scenarios to take pictures and make videos, we assume that images and videos gathered from social media website were suitable to reflect the real case scenario of emergency situations. Our dataset is composed of images retrieved from the Flickr[4] social media under the Creative Commons license; videos obtained from YouTube; and simulations carried out during the RESCUER project. These data were labeled according to the presence or absence of fire/smoke. Therefore, the datasets provide a proper material for the validation of the algorithms developed for emergency image and video analysis.

Subsets of the FiSmo have been used in a series of works, where in each work, the subset was adapted according to its needs:

- Fast Fire Detection -FFireDT [Bedo et al. 2015, Bedo et al. 2016]: combines low-level features and evaluation functions to support instance-based learning to detect fire in images and support similarity-enabled Relational Database Management Systems (RDBMS) in disaster-relief tasks [Oliveira et al. 2016];
- BowFire [Chino et al. 2015]: detects and segments fire in still images, by combining color features with texture classification on superpixel regions;
- SmokeBlock [Cazzolato et al. 2016]: segments and detects smoke in still images using superpixel segmentation and local color and texture features from images;
- SPATFIRE [Avalhais et al. 2016]: a fire event detection method that works with videos and takes advantage of spatial color modeling and motion pattern.

Additionally to the images and videos, the datasets are also composed of a set of features extracted from the images, regions of interest and annotations, obtained in preprocessing steps and manual efforts of aforementioned works.

FiSmo is divided into two main parts: FiSmo-Images, presented in Section 2, containing images datasets and information extracted from these images; and FiSmo-Videos, presented in Section 3, containing videos and annotations. In Section 4 we discuss the applicability, challenges, and limitations of the datasets, including the public location of the files. Finally, in Section 5 we present the conclusion of this work.

2. FiSmo-Images: Still Images from Social Media

In this section we present FiSmo-Images (**Fi**re and **Smo**ke Images), which is composed of four datasets: Flickr-FireSmoke, Flickr-Fire, BoWFire, and SmokeBlock. In the following subsections, we describe the process of collecting the images, preprocessing the data and the annotation task. Then, for each specific task, we present the modifications made in the data, in order to provide enough information for experimental analysis steps.

[3]FiSmo datasets are available at `https://goo.gl/uW7LxW`

[4]`https://www.flickr.com/`

2.1. Data Collection and Preprocessing

The collection of the images was carried by using the Flickr API[5], in August of 2014. All images downloaded were available under Creative Commons license. A total of 5,962 images were retrieved, using a set of textual keywords presented in Table 1.

Table 1. List of keywords used to retrieve images using Flickr API.

Keywords used to collect images using Flickr API				
fire	smoke	emergency	flames	burning
protest	boston marathon	car fire accident	criminal fire	fire department
firefighter	urban fire	house burning	criminal fire	fire car accident

Removing duplicated images: A similarity comparison between the images was carried, in order to identify duplicate images. The resulting 406 images identified as duplicated were removed, resulting in a dataset composed of 5,556 images.

Annotation of images according to the emergency scenario: Figure 1 shows examples of images from this dataset. Despite the keywords used during the acquisition, part of the images (e and f) do not have fire and/or smoke, and others (g and h) have fire and/or smoke, but are not from emergency scenarios (e.g. lightening purposes).

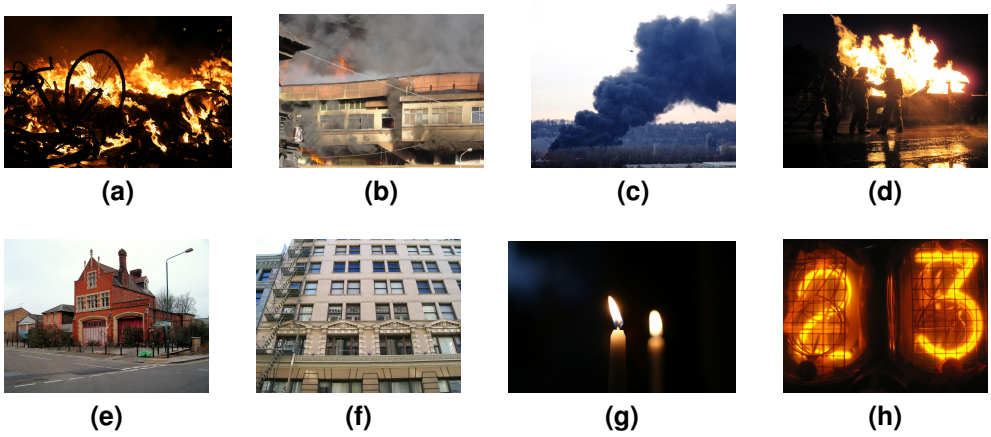


Figure 1. Sample images from the gathered dataset: (a-d) present fire and/or smoke; (e-f) do not present fire and/or smoke; and (g-h) present fire and/or smoke; but not from emergency situations.

As shown in Figure 1, not all of the retrieved images contain visual traces of fire/smoke. Thus, the next preprocessing step carried was the manual annotation of the images. For this task, we asked seven subjects to perform the annotation, all of them between 20 and 30 years old and familiar with the problem.We observed a disagreement between what is considered fire/smoke. For example: *can the image showed in Figure 1-g be considered as "fire"?* Part of the subjects considered fire for lightening purposes as "not fire" since it was not related to an emergency scenario. The same disagreement applies for the annotation as "smoke", for example when the smoke was coming out of the exhaust pipe of a car. Considering this scenario, all subjects were oriented to annotate the images according to the following questions:

[5]The Flickr API: `www.flickr.com/services/api/`

- *Does the image contain fire from an emergency situation?* (yes/no)
- *Does the image contain smoke from an emergency situation?* (yes/no)

The set of 5,556 non-duplicate images was divided into subsets constructed in a way that each image was annotated by at least two subjects. After the first round of annotation, all images with disagreement (i.e. different annotations) were submitted to a second round, performed by a third subject. After this process, we obtained the first dataset of FiSmo-Images, called **Flickr-FireSmoke**. This dataset contains 5,556 images classified as fire (y/n) and smoke (y/n), and its class distribution is described in Table 2. It was used as the basic ground truth for the experimental analysis of works regarding the detection of fire and smoke. For each specific task, the dataset was adapted in order to provide the proper information regarding the application context, resulting on specific datasets. In the next subsections, we present these variations.

Table 2. Class distribution of the 5,556 images from Flickr-FireSmoke dataset.

Flickr-FireSmoke Dataset	
class	***# images***
fire and smoke	527
only fire	1,077
only smoke	369
none	3,583

2.2. The Flickr-Fire Dataset

In emergency situations, urban and crowded scenarios may contain a vast amount of information to be processed in a short interval of time. During an explosion in a stadium, for example, many users can use send pictures, short movies and text messages to social media websites, and it is important to process and filter all this information for the rescue forces. To address this problem, Bedo *et al.* [Bedo et al. 2015, Bedo et al. 2016] proposed the FFireDt method, that classifies a given input image based on past cases (pre-annotated images), relying on a content-based image retrieval module. FFireDt is able to select similar images, helping to filter the information and build an overview of the emergency scenario for the authorities. In this work, the authors used a subset of Flickr-FireSmoke, in order to obtain a balanced dataset, called **Flickr-Fire**, composed of:

- 2,000 images, 1,000 labeled as *"fire"* and 1,000 labeled as *"not fire"*;
- Six files containing low-level features, extracted from the 2,000 images, and the corresponding classes. The Feature Extractor Methods (FEM) used were: Color Layout, Scalable Color, Color Structure, Color Temperature, Edge Histogram and Texture Browsing.

Oliveira *et al.* [2016] also employed this dataset on a system to support civilian crisis situations relying on a RDBMS.

2.3. The BoWFire Dataset

Several methods regarding the problem of fire detection on videos have been proposed in the last years [Celik and Demirel 2009, Zhang et al. 2014]. To detect fire on videos, the

methods explore color features on the video frames and refine their output using temporal features of the videos. There are also a few works that detect fire on still images based on color. These works achieved good results on forest fire or controlled environments because the fire contrasts with the background. However, when changing the emergency scenario to urban regions, the images may also contain reddish/yellowish objects, which can be mistaken as fire. This problem was studied by Chino *et al.* [2015], where the authors proposed the BoWFire, a method based on color and texture analysis.

In order to train and validate the BoWFire method, the authors constructed the **BoWFire** dataset. The BoWFire dataset is a subset of the Flickr-FireSmoke dataset, with images from emergency situations with fire in urban scenarios. The BoWFire dataset also has images with no visible fire containing reddish or yellowish objects and sunsets, which can be mistaken as fire. To validate the method, the authors constructed a ground truth composed of masks of the fire regions in the images. Figure 2 shows examples of the BoWFire images and their respective masks. The BoWFire dataset has a second dataset used as a training set. This second dataset consists of images classified as fire and non-fire. It is important to note that non-fire images also contain red or yellow objects. Figure 3 shows examples of the training set. In summary, the BoWFire dataset consists of:

- 226 images, 119 labeled as “fire” and 107 labeled as “not fire”;
- 226 masks of the regions containing fire, that were used as ground truth;
- 240 images (regions of interest – ROIs) of 50x50 pixels resolution, 80 labeled as *“fire”* and 160 *“not fire”*. These images were used for training purposes.

Figure 2. Sample images and the corresponding mask of the fire regions.

Figure 3. BoWFire ROIs with fire (1st row) and without fire (2nd row).

2.4. The SmokeBlock Dataset

Many computer vision methods in the literature address the problem of detecting smoke. However, they usually rely on the movement present on videos to differentiate smoke regions from background objects, such as trees and clouds [Cazzolato et al. 2016]. The

existing methods that work with still images propose the classification of pixels analysing only the color of pixels, mainly considering smoke regions as depicting grayscale colors. However, in real scenarios smoke regions depict different colors considering many factors, e.g. illumination, temperature and the material being burned. To overcome these problems Cazzolato *et al.* [2016] proposed the SmokeBlock, a smoke detection method using color and texture patterns.

SmokeBlock was trained with a set of ROIs depicting different colors and patterns of images, as presented in Figure 4. For each input image, the method classifies the image as smoke or not smoke, and outputs the segmented smoke regions of the image (if it was classified as smoke). The experimental analysis was carried out using a subset of the dataset Flickr-FireSmoke, named **SmokeBlock** dataset, which is composed of:

- 1,666 images, 832 labeled as *"smoke"* and 834 labeled as *"not smoke"*;
- 10 files containing the low-level features of the images and the corresponding classes. The FEM employed were: Color Layout, Color Structure, Color Temperature, Edge Histogram, Haralick, LPB, Normalized Histogram, Scalable Color, Texture Spectrum and Zernike.
- 103 images (ROIs), 43 labeled as *"smoke"* and 60 labeled as *"not smoke"*.
- Low-level features of the images and the corresponding classes. The FEM employed were: Color Layout and Haralick;

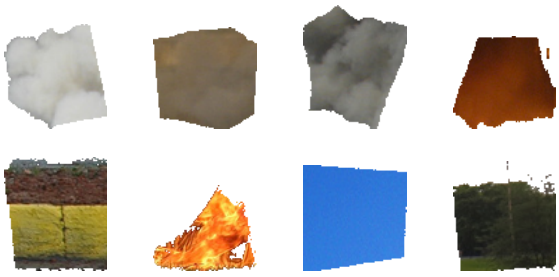


Figure 4. SmokeBlock ROIs with smoke (1st row) and without smoke (2nd row).

2.5. FiSmo-Images: Summarization

FiSmo-Images consists of a set of the image datasets Flickr-FireSmoke, Flickr-Fire, BoWFire and SmokeBlock, and it is summarized in Table 3. The images from all datasets are labeled, according to the purpose of the work, as fire/not fire and smoke/not smoke.

Table 3. Summarization of the FiSmo-Images datasets.

FiSmo-Images				
Dataset Name	Purpose	# images	Features	ROIs
FireSmoke	Fire and Smoke detection	5,556	No	No
Flickr-Fire	Global fire detection and content-based image retrieval	2,000	Yes	No
BoWFire	Fire detection and segmentation	226	No	Yes
SmokeBlock	Smoke detection and segmentation	1,666	Yes	Yes

3. The FiSmo-Videos: Unconstrained Videos for Event Segmentation

There has been an increasing interest from the computer vision community in researching about several video-related problems over the past two decades. One of the outcomes of this is the effort to make publicly available datasets with benchmarks, as an example, the multimedia event recognition (MED) dataset composed of high-level complex event categories provided by TREC Video Retrieval Evaluation (TRECVid) [6]. Notwithstanding, with regard to the fire emergencies scenario, there is a lack of standard benchmark datasets for fire detection publicly available.

The literature shows that, in general, previous studies focusing on fire detection constrain the domain of the problem through the assumption that the scenes are captured in a stationary set up [Töreyin et al. 2006, Celik and Demirel 2009, Zhang et al. 2014, Qureshi et al. 2016], or have a limited influence of camera motion [Habiboğlu et al. 2012]. The validation of these methods is usually based on datasets of short video clips acquired with static cameras. For instance, the *MIVIA*[7] [Di Lascio et al. 2014] dataset, one of the most used dataset for fire detection, is composed of 14 videos with fire and 17 which do not contain fire. More recently, in the work [Qureshi et al. 2016] the authors introduced the *QuickBlaze*[8] dataset, with a total of 30 short-duration videos in which 50% has fire.

Besides the fact that the existing datasets were built under the stationary camera constraint, there is also another limitation regarding the type of problems that can take advantage of them. Methods that focus on fire detection in videos with categoric binary output can assess their performance from such data. In fact, those are the most common task for video fire detection in the literature. However, if the problem is more complex, e.g., event segmentation where the focus is to detect the approximate time interval where the event of interest takes place, then these datasets are no longer adequate. This is because each video is usually a single sequence with fire or a single sequence without fire.

3.1. Video Data Acquisition

In order to provide support to the task of detecting events of fire from unconstrained videos, we have collected videos from two distinct sources. The first subset was extracted from YouTube, and the second was provided by the RESCUER Project partners after a fire simulation in an industrial park. More details are described in the following sections.

3.1.1. The FireVid Dataset

The FireVid dataset was collected from YouTube, using an adapted version of the crawler tool TubeKit[9]. This tool works by making periodically requests with queries that use previously selected keywords. For this purpose, the set of keywords showed in Table 4 was selected. 97 videos were downloaded, from which 27 were selected for the annotation task totalizing $83,675$ frames. The videos are unconstrained in respect to several aspects:

[6] http://trecvid.nist.gov

[7] http://mivia.unisa.it/datasets/video-analysis-datasets/fire-detection-dataset

[8] http://vgl-ait.org/cvwiki/doku.php?id=quickblaze:main

[9] http://www.tubekit.org

- Video quality: resolution, frame rate, encoding algorithm;
- Point of view: ground level shooting and aerial videos captured by drone and from a helicopter;
- Scene lightening: several lightening conditions such as sunny, dusk, night, shade;
- Scale and distance: different distances to the main scene and presence of distance variation in a single video;
- Camera motion: mostly hand-held shooting with different amount of motion.

Table 4. Keywords used to retrieve videos.

Keywords used to collect videos from YouTube		
fire	smoke	explosion
flames	burning	blaze
campfire	bonfire	combustion
ignite	wildfire	firefighters

3.1.2. The RESCUER Video Dataset

Specialists working on the RESCUER Project in a partnership with experts from several areas and companies belonging to a chemical industrial park located in Camaçari, Brazil, had performed a fire exercise simulation with victims. During this simulation, videos were shot with different handheld mobile devices. The scenes captured the firefighters extinguishing the fire from different angles of view and distances. Besides, there is reasonable variability regarding the amount of movement from the camera handler perspective, light conditions, resolution, and duration. Shooting the videos while taking into account these diversities is important in order to simulate the scenario where people with different camera devices, shooting skills, local position and motion behavior would be more likely to represent a real situation. A total of $29,895$ frames from 61 videos were annotated in this dataset. Figure 5 shows a sample of frames from the FireVid and RESCUER datasets.

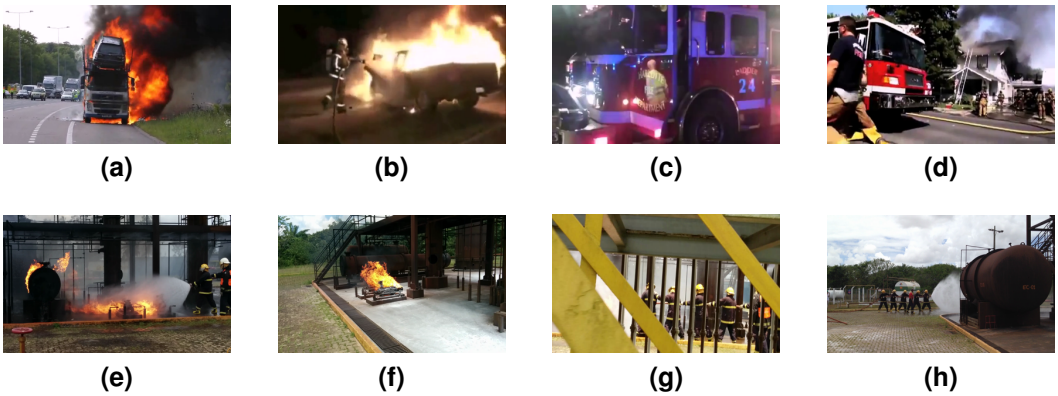


Figure 5. Sample frames from the video dataset: (a-b) fire segment, (c-d) non-fire segment on FireVid dataset; (e-f) fire segment and (g-h) non-fire segment on RESCUER videos dataset.

3.2. Annotation Protocol and Ground-truth

We provide frame-level ground-truth, instead of a single label for each video. For each frame of a video, two different annotators assigned that frame to one of the labels: *"fire"*, *"notFire"* or *"ignore"*. Similarly to how the annotation was conducted for the Flickr-FireSmoke dataset, the annotators were asked to label a frame with *"fire"* if they were able to consider that, the frame isolated, has fire. The label *"notFire"* was assigned to every frame in which the annotators were confident that there was no fire in the frame. However, if a frame shows a fire like lightening but the annotator was not able to tell whether it is fire or any other lightening source, then he/she was instructed to assign the *"ignore"* label. The *"ignore"* label was also assigned to every frame that is part of any post-edition transition effect. We adopted the use of the last label to exclude from our analysis segments that are not useful to evaluate. For each frame that the two annotators did not agree, a third annotator was cast so to define the label.

After the manual annotation task, we process the frame annotation to create the ground-truth for each transition. For each subsequent pair of frames f_i and f_{i+1}, the transition label t_i represents a transition of a fire segment if f_i = *"fire"* and f_{i+1} = *"fire"*, then t_i = *"fire"*. If one of the frames is *"fire"* and the other is *"notFire"*, or both are *"notFire"*, then t_i = *"notFire"*. For the case in which at least one of the frames is *"ignore"*, then we assign t_i = *"ignore"*. This annotation scheme allows one to use both datasets to validate not only simple fire detections methods, but also temporal segmentation of fire methods, such as the work presented by [Avalhais et al. 2016].

4. Discussion

The effort to build FiSmo was made due to the lack of datasets of images and videos depicting real situations. In the literature is possible to find small sets of images and videos, generally of a restricted scenario (for example, forest fire), and without any additional information to be used as ground truth. FiSmo provides a feasible set of additional information that can be used as a basis for experimental analysis.

It is worth to clarify that the datasets Flickr-Fire, BoWFire, and SmokeBlock are subsets of the Flickr-FireSmoke dataset, and their images were randomly selected. Each of these subsets was build in order to aid with the validation of the corresponded methods comprising different tasks, and provides the features and ground truth that meet the purpose of the task.

4.1. Applicability

FiSmo can be applied in the analysis of emergency scenarios, regarding the detection of fire and smoke in still images and videos. By providing a set of images retrieved from social media, the analysis can be carried using real data, with realistic characteristics such as different resolutions, illuminations, angles, and situations. The images and videos also contain labels, manually created, making it feasible for classification and clustering tasks. Additionally, the low-level features extracted from the images enable the use of this dataset for content-based image retrieval analysis.

4.2. Challenges and Limitations

One of the main challenges regarding the detection of fire and smoke is the subjectivity of the problem. During the annotation process, one may consider, for instance, a candle

inside a cup as fire (Figure 6-a). However, one can label the image as not fire, since a candle is not related to an emergency situation. Also, a third opinion may lead a person to label the image as not fire, because the flame is not visible. Here we have three different points of view, and whether they are right or wrong is subjective, and it depends on the application scenario. Fire regions may be mistaken by objects like lights, flashlights, and sunset (Figure 6-b). Also, fire flames may present different colors, depending on the temperature, the material being burned, and the illumination.

The same issue can be applied to smoke. In some situations, even for the human eye, it is difficult to differentiate regions depicting smoke. In Figure 6-c is possible to observe smoke and water presenting similar texture and color patterns. Particularly, smoke detection has many challenges, for instance: regions may present transparency, leading to more difficulties during the detection; depending on the illumination, the temperature, and the material being burned, smoke can present different colors (Figure 6-d); and smoke is visually similar to other objects, such as mist, water, trees, and clouds.

(a) (b) (c) (d)

Figure 6. Challenges regarding fire and smoke detection in images: (a) fire can be used for lightening and (b) may depict different colors; (c) smoke can be mistaken by objects such as water, making it difficult to determine the region depicting only smoke and (d) can also present different colors.

4.3. Download and Citation Request

FiSmo is public for research, under the Creative Commons license, and available at `https://goo.gl/uW7LxW`. In case this dataset (or part of it) is used for scientific, industrial, or academic purposes, or in case it is publicly mentioned for whatever purpose, please include the citation to this work.

5. Conclusion

We presented the FiSmo compilation of images and videos datasets. FiSmo provides annotations, regions of interest and low-level features obtained from the data. This information can be used as a basis for experimental analysis of several works since it is built from real images and videos. The datasets can be used in several applications regarding the analysis of fire, explosion and smoke in emergency scenarios. The dataset is public for research, available online, and can be used under the Creative Commons license.

Acknowledgments

This research is supported, in part, by CNPq, FAPESP, the RESCUER project, funded by the European Commission (Grant: 614154) and by the CNPq/MCTI (Grant: 490084/2013-3), CAPES and CAPES-PDSE (Process: 88881.134068/2016-01). The authors would like to thank Alceu Ferraz Costa, for his contribution in the data collection and analysis process.

References

Avalhais, L. P. S., Rodrigues, J., and Traina, A. J. M. (2016). Fire detection on unconstrained videos using color-aware spatial modeling and motion flow. In *2016 IEEE 28th International Conference on Tools with Artificial Intelligence (ICTAI)*, pages 913–920.

Bedo, M. V. N., Blanco, G., Oliveira, W. D., Cazzolato, M. T., Costa, A. F., Jr., J. F. R., Traina, A. J. M., and Jr., C. T. (2015). Techniques for effective and efficient fire detection from social media images. In *ICEIS 2015 - Proceedings of the 17th International Conference on Enterprise Information Systems, Volume 1, Barcelona, Spain, 27-30 April, 2015*, pages 34–45.

Bedo, M. V. N., de Oliveira, W. D., Cazzolato, M. T., Costa, A., Blanco, G., Rodrigues-Jr, J. F., Traina, A., and Traina Jr, C. (2016). Fire detection from social media images by means of instance-based learning. In Springer, editor, *Lecture Notes in Computer Science*, pages 1–22 (to appear). Springer International Publishing.

Cazzolato, M. T., Bedo, M. V., Costa, A., Souza, J. A., Traina Jr, C., Rodrigues Jr., J. F., and Traina, A. J. M. (2016). Unveiling smoke in social images with the smokeblock approach. In *Proceedings of the 31st ACM/SIGAPP Symposium on Applied Computing*, pages 1–6 (to appear). ACM Press.

Celik, T. and Demirel, H. (2009). Fire detection in video sequences using a generic color model. *Fire Safety Journal*, 44(2):147–158.

Chino, D. Y. T., Avalhais, L. P. S., Rodrigues-Jr, J. F., and Traina, A. J. M. (2015). Bowfire: Detection of fire in still images by integrating pixel color and texture analysis. In *Proceedings of the 2015 28th SIBGRAPI Conference on Graphics, Patterns and Images*, SIBGRAPI '15, pages 95–102, Washington, DC, USA. IEEE Computer Society.

Di Lascio, R., Greco, A., Saggese, A., and Vento, M. (2014). Improving fire detection reliability by a combination of videoanalytics. In *International Conference Image Analysis and Recognition*, pages 477–484. Springer.

Habiboğlu, Y. H., Günay, O., and Çetin, A. E. (2012). Covariance matrix-based fire and flame detection method in video. *Machine Vision and Applications*, 23(6):1103–1113.

Oliveira, P. H., Fraideinberze, A. C., Laverde, N. A., Gualdron, H., Gonzaga, A. S., Ferreira, L. D., Oliveira, W. D., Jr., J. F. R., Cordeiro, R. L. F., Jr., C. T., Traina, A. J. M., and de Sousa, E. P. M. (2016). On the support of a similarity-enabled relational database management system in civilian crisis situations. In *ICEIS 2016 - Proceedings of the 18th International Conference on Enterprise Information Systems, Volume 1, Rome, Italy, April 25-28, 2016*, pages 119–126.

Qureshi, W. S., Ekpanyapong, M., Dailey, M. N., Rinsurongkawong, S., Malenichev, A., and Krasotkina, O. (2016). Quickblaze: early fire detection using a combined video processing approach. *Fire Technology*, 52(5):1293–1317.

Töreyin, B. U., Dedeoğlu, Y., Güdükbay, U., and Çetin, A. E. (2006). Computer vision based method for real-time fire and flame detection. *Pattern Recognition Letters*, 27(1):49 – 58.

Zhang, Z., Shen, T., and Zou, J. (2014). An improved probabilistic approach for fire detection in videos. *Fire Technology*, 50(3):745–752.

GitSED: Um Conjunto de Dados com Informações Sociais baseado no GitHub

Natércia A. Batista, Gabriela B. Alves, André L. Gonzaga, Michele A. Brandão

Universidade Federal de Minas Gerais (UFMG) – Belo Horizonte, MG – Brasil

{natercia,gabrielabrant,andregonzaga,micheleabrandao}@dcc.ufmg.br

Abstract. *Software repositories analysis can reveal, for example, patterns associated to interactions between developers and factors related to their productivity. This paper presents GitSED (Socially Enriched Dataset based on GitHub): a dataset based on GitHub that is curated (cleaned and reduced), augmented with external data, and enriched with social information on developers' interaction. Finally, GitSED is easy to use and replicate in order to perform different social analyses, such as community detection and developer ranking.*

Resumo. *A análise de repositórios de software pode revelar, por exemplo, padrões associados às interações entre desenvolvedores e fatores relacionados à produtividade dos mesmos. Nesse contexto, este artigo apresenta GitSED (Socially Enriched Dataset based on GitHub): um conjunto de dados curado (limpo e reduzido), expandido com dados externos e enriquecido com informações sociais sobre interações entre desenvolvedores no GitHub. Finalmente, o GitSED é fácil de ser utilizado e replicado em diferentes análises sociais, tais como detecção de comunidades e ranqueamento de desenvolvedores.*

1. Introdução

Analisar repositórios de software permite responder perguntas importantes e interessantes como "Quais padrões governam as interações entre desenvolvedores?", "Como os sentimentos dos desenvolvedores influenciam seu desempenho?" ou "Quem são os melhores desenvolvedores que uma empresa pode contratar para um trabalho específico?". Pesquisas recentes exploram esses tópicos e muitos outros relacionados [Barnett et al. 2016; Bartusiak et al. 2016; Brandão and Moro 2017; Casalnuovo et al. 2015; Padhye et al. 2014; Rocha et al. 2016; Sinha et al. 2016].

Existem vários repositórios de software diferentes, incluindo SourceForge, GitHub, Google Code e BitBucket. Neste artigo, focamos na mineração das interações entre desenvolvedores do GitHub[1], um website que fornece serviço de hospedagem online e gerenciamento de código fonte, bem como controle de versão distribuído aos usuários. Atualmente, o GitHub possui mais de 53 milhões de repositórios (Fevereiro 2017) e 14 milhões de usuários (Abril 2016)[2]. Muitos estudos utilizam os dados do GitHub para analisar sentimento[Sinha et al. 2016], estudar a correlação entre as mensagens dos *commits* e a tendência a defeitos nos projetos de Java [Barnett et al. 2016], mostrar práticas típicas que programadores usam para lidar com exceções [Kery et al. 2016], entre outros.

[1]GitHub: `http://github.com`

[2]O maior repositório de códigos fontes do planeta: `github.com/features` e `github.com/about/press`

Outros estudos focam na análise dos aspectos sociais dos repositórios de software [Bartusiak et al. 2016; Bettenburg and Hassan 2010; Casalnuovo et al. 2015; Vasilescu et al. 2015]. O estudo de tais aspectos sociais pode ajudar a entender o que influencia a produtividade dos desenvolvedores [Casalnuovo et al. 2015; Vasilescu et al. 2015], e como as interações sociais entre desenvolvedores e usuários influenciam na qualidade do software [Bettenburg and Hassan 2010].

Neste artigo, contribuímos para preencher tais deficiências através da criação de uma base de dados do GitHub com as seguintes características notáveis: **curada** por ter sido limpa e focada em apenas três linguagens de programação, **aumentada** por adicionar dados sobre repositórios e desenvolvedores não disponíveis no GHTorrent[3] e **enriquecida** com informações de redes sociais que permitem medir a força dos relacionamentos. Tais informações de redes sociais facilitam o estudo e computação de diferentes métricas para estudo das interações entre desenvolvedores em uma rede de codificação social. Consequentemente, permitem determinar métricas mais adequadas para medir tal força [Alves et al. 2016; Batista et al. 2017]. Além disso, disponibilizamos online[4] o conjunto de dados completo e todos os códigos fonte desenvolvidos para coleta e modelagem.

Para validar e mostrar de forma prática a utilidade de nossa base de dados, nós a avaliamos experimentalmente em uma aplicação real: construindo e analisando uma rede de desenvolvimento colaborativa. Ao analisar a rede social, vimos que: a maioria dos desenvolvedores são ativos em poucos repositórios, o número de conexões entre desenvolvedores varia entre diferentes linguagens de programação e poucos pares de desenvolvedores possuem interação em mais de um repositório. Esses resultados são relevantes para irmos além, mais profundamente e até mesmo desenvolver estudos novos sobre a força de colaboração em redes de desenvolvimento social [Bartusiak et al. 2016; Casalnuovo et al. 2015; Tsay et al. 2014].

Após discutirmos os trabalhos relacionados (Seção 2), as contribuições deste trabalho são sumarizadas a seguir:

- Apresentação de um conjunto de dados do GitHub curado, aumentado e enriquecido chamada GitSED (*Github Socially Enriched Dataset*) para JavaScript, Ruby e Java, a metodologia usada para coleta e o mecanismo de armazenamento por meio de uma modelagem não-relacional (Seção 3). Note que tal modelagem diferencia-se do GHTorrent, o qual utiliza uma modelagem relacional mesmo quando os dados são armazenados no MongoDB[5];
- Exemplos de aplicação real do conjunto de dados (Seção 4); e
- Discussão sobre as limitações e desafios relacionados à criação e uso do conjunto de dados (Seção 5).

2. Trabalhos Relacionados

Análises do GitHub oferecem noções que podem ajudar a melhorar a qualidade do processo de desenvolvimento de software e sua qualidade. Por exemplo, Dabbish et al.

[3]GHTorrent: `http://ghtorrent.org/`

[4]GitSED e códigos fontes em *`http://homepages.dcc.ufmg.br/~mirella/projs/apoena/datasets.html`*

[5]MongoDB: `https://www.mongodb.com/`

[2012] mostram como inferir os objetivos técnicos e visões dos desenvolvedores a partir de sua rede de atividades, prevendo quais quais projetos tem chance de ficar ativos no longo prazo. Ademais, Jiang et. al [2013] estudam a disseminação de projetos para entender como isso ocorre e então, melhorar a popularidade de tal projeto.

Além disso, estudos do GitHub sobre linguagens de programação específicas mostram diferentes padrões de desenvolvimento e colaboração que são peculiares a cada linguagem. Por exemplo, Kery et. al [2016] mostram práticas típicas que programadores usam para tratar exceções em projetos Java. Adicionalmente, Proksch et al. [2016] elaboraram um conjunto de dados que inclui códigos fontes de 360 repositórios C#. Em uma perspectiva mais social, Padhye et. al [2014] consideram 89 projetos populares no GitHub (e seus 108.00+ forks) para estudar os níveis de participação de diferentes comunidades em projetos e apresentar resultados agrupados por linguagem de programação (as onze mais populares foram consideradas).

Nesse contexto, construímos o GitSED para JavaScript, Ruby e Java, uma base de dados que facilita a construção de redes sociais de desenvolvedores para essas linguagens de programação. Essas redes de colaboração permitem a investigação de diferentes padrões de colaboração em repositórios nas três linguagens de programação. Especificamente, nosso conjunto de dados facilita a computação de métricas sociais, tanto topológicas quanto semânticas. Existem diferentes métricas para calcular tal aspecto [Bartusiak et al. 2016; Casalnuovo et al. 2015; Tsay et al. 2014]. Note que GitSED difere dos conjuntos de dados propostos por Gousios [2013] , Kery et. al [2016] e Proksch et al. [2016] por facilitar tais análises sociais.

3. Metodologia

Para construir o GitSED, inicialmente extraímos dados de uma grande base de dados, o GHTorrent. Em seguida, construímos a rede social de colaboração conectando desenvolvedores que contribuem para um mesmo repositório. Isso permite o uso do GitSED para análise de colaborações.

Descrição do conjunto de dados. Neste trabalho, extraímos dados do GHTorrent [Gousios 2013], um projeto aberto que fornece uma base de dados do GitHub. Segundo Cosentino et al. [2016], o GHTorrent é o meio mais popular de coleta e monitoramento de dados do GitHub. Diferente da maioria dos estudos cobertos na pesquisa desses autores (cerca de 70%), disponibilizamos nossa base de dados publicamente em `http://homepages.dcc.ufmg.br/~mirella/projs/apoena/datasets.html`. O GitSED originou de uma base de dados completa do GHTorrent, disponibilizada em 15 de setembro de 2015. Inicialmente, a base de dados possuía um total de 1.987.760 projetos (32 GB de dados). Desses projetos, 1.204.212 eram forks[6]. Nós então removemos todos os repositórios forks, pois as mudanças feitas em um repositório fork precisam ser aprovadas no repositório base por meio de *pull requests*[7], resultando em 529.405 projetos não-fork. Mesmo com tais filtragens, a base de dados resultante ainda era grande para processar e necessitaria de grande poder computacional para lidar com todos os dados.

[6]Cópia de um repositório que permite aos usuários modificar e experimentar o código sem afetar o projeto original.

[7]Mudanças realizadas por meio de *commits* em um repositório externo devem ser aprovadas no repositório base.

Tabela 1: Estatísticas do GitSED por linguagem de programação.

Linguagem de Programação	# Repositórios	# Nós	# Arestas
JavaScript	90.363	88.586	3.196.846
Ruby	59.225	51.475	4.620.128
Java	52.601	69.119	2.122.017

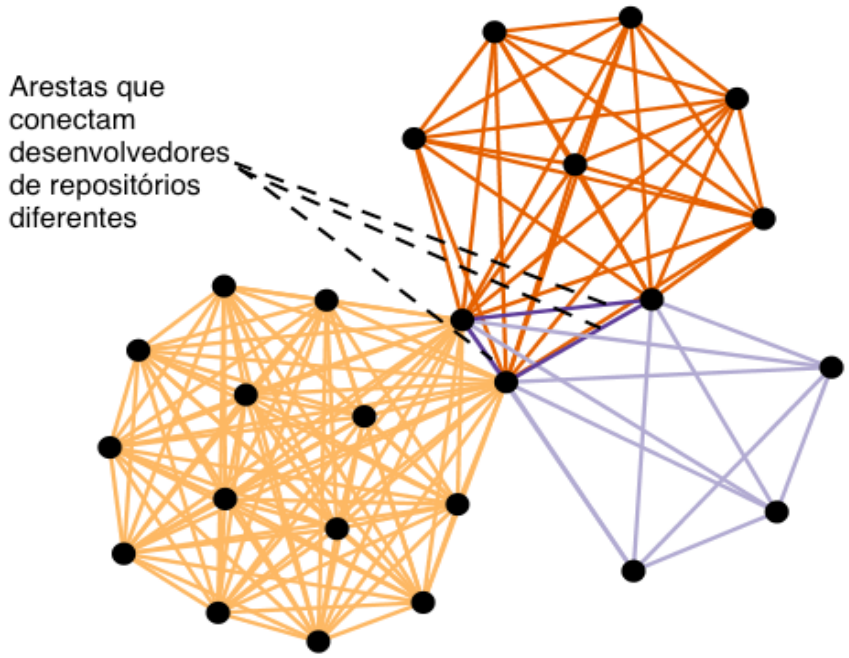


Figura 1: Exemplo de conexões entre desenvolvedores que contribuíram em um ou mais repositórios. Note a clique conectando todos os desenvolvedores em um repositório.

Filtrando linguagens de programação específicas. Com o objetivo de reduzir a base de dados e tornar mais fácil de usá-la, focamos em três linguagens de programação, sem perda de generalidade e seguindo trabalhos anteriores (Section 2). Inicialmente, consideramos JavaScript com 90.363 repositórios, que representa 17% dos repositórios não-fork. Tal linguagem de programação é também a mais utilizada no GitHub. Em seguida, incluímos Ruby com 59.225 repositórios e Java com 52.601 repositórios por serem a segunda e terceira linguagens de programação, respectivamente, com maior número de repositórios no GHTorrent. Finalmente, construímos três redes sociais de colaboração para cada uma dessas linguagens de programação.

Rede social de colaboração entre desenvolvedores e suas análises. Para melhorar ainda mais o potencial de aplicações do GitSED, o enriquecemos para possibilitar análises de redes de sociais de desenvolvimento. Tais redes podem ser modeladas como um grafo ponderado $\mathcal{G} = (\mathcal{V}, \mathcal{E})$ onde $\mathcal{V}$ é o conjunto de nós que representam desenvolvedores, e $\mathcal{E}$ é o conjunto de arestas não direcionadas que conectam aqueles que contribuíram em um mesmo repositório. A Tabela 1 apresenta as estatísticas das redes de colaboração entre desenvolvedores no GitSED para JavaScript, Ruby e Java. Note que as arestas possuem pesos associados que medem a conexão social, por meio de diferentes métricas que são apresentadas na Tabela 2. Como esperado, a rede social final é formada por várias

Tabela 2: Descrição das siglas das métricas sociais especificadas na Figura 2.

Sigla	Descrição
NO	Sobreposição de vizinhos (neighborhood overlap)
AA	Adamic-Adar
PA	Preferential Attachment
SR	Quantidade de repositórios compartilhados
JCSR	Quantidade de contribuição em repositórios compartilhados
JCOSR	Quantidade de *commits* em repositórios compartilhados em relação ao total do repositório
JWCOSR	Quantidade do número de linhas por *commit* em repositórios compartilhados em relação ao total do repositório
PC	Quantidade de repositórios compartilhados normalizada pela quantidade de colaboradores por repositório no tempo de ingresso do par de desenvolvedores
GPC	Potencial de colaboração entre desenvolvedores baseado no tempo de contribuição simultânea

cliques, pois cada repositório forma um, conforme mostra a Figura 1. Nesse exemplo foram selecionados três repositórios da rede de JavaScript (diferenciados pela coloração) com baixo número de desenvolvedores, visto que a visualização da rede completa é de difícil compreensão devido a densidade da mesma. Tais cliques estão conectadas por desenvolvedores que tiveram contribuições em comum em mais de um repositório.

Ademais, notamos algumas diferenças entre as redes de diferentes linguagens. Por exemplo, apesar da rede de JavaScript ter mais repositórios e nós, a rede de Ruby é mais densa. Isso pode indicar que os desenvolvedores de Ruby tendem a contribuir para repositórios distintos mais do que desenvolvedores de JavaScript. Para a rede da linguagem Java, há um número de nós (desenvolvedores) superior à rede de Ruby, porém estão presentes menos da metade da quantidade de contribuições. Essas intuições só foram possíveis devido ao nosso conjunto de dados curado e aumentado.

Outras propriedades do GitSED. Após trabalhar na base de dados original, apresentamos como aumentar consideravelmente a utilidade e aplicação do nosso novo conjunto de dados. Inicialmente, acrescentamos o número de linhas em que cada desenvolvedor contribuiu para o repositório. A razão para tal acréscimo é descrita a seguir. Considere dois desenvolvedores distintos: o primeiro trabalha por quatro horas e faz um *commit* no fim e o segundo trabalha o mesmo número de horas mas faz *commit* a cada 10 minutos. Claramente, a importância ou relevância de suas contribuições não podem ser medidas unicamente pelo número de *commit*. Apesar disso, o GHTorrent não compartilha o número de linhas impactadas a cada *commit*. Assim, desenvolvemos um *crawler* utilizando *BeautifulSoup* e *Request* (módulo do *Python*). O *crawler* coleta os dados corretamente, exceto para aqueles repositórios que não estão mais disponíveis para coleta. Por isso, sincronizamos o GitSED para incluir a interação entre desenvolvedores apenas dos repositórios encontrados pelo *crawler*. Consequentemente, o GitSED considera interações de 65.799 repositórios: 28.584 de JavaScript, 17.342 de Ruby e 19.873 de Java. É importante ressaltar que a redução na quantidade de repositórios na rede também é explicada pela existência de vários repositórios particulares. Analisando os dados, observamos que cerca de 50% dos repositórios de cada uma das redes apresentadas na Tabela 1 possuem um único desenvolvedor. Esses repositórios particulares não geram relacionamentos. Portanto, não estão presentes na modelagem final das redes.

A segunda propriedade é o tempo. Especificamente para os repositórios, adicionamos atributos para datas de início e fim (*create_date* e *end_date*, respectivamente).

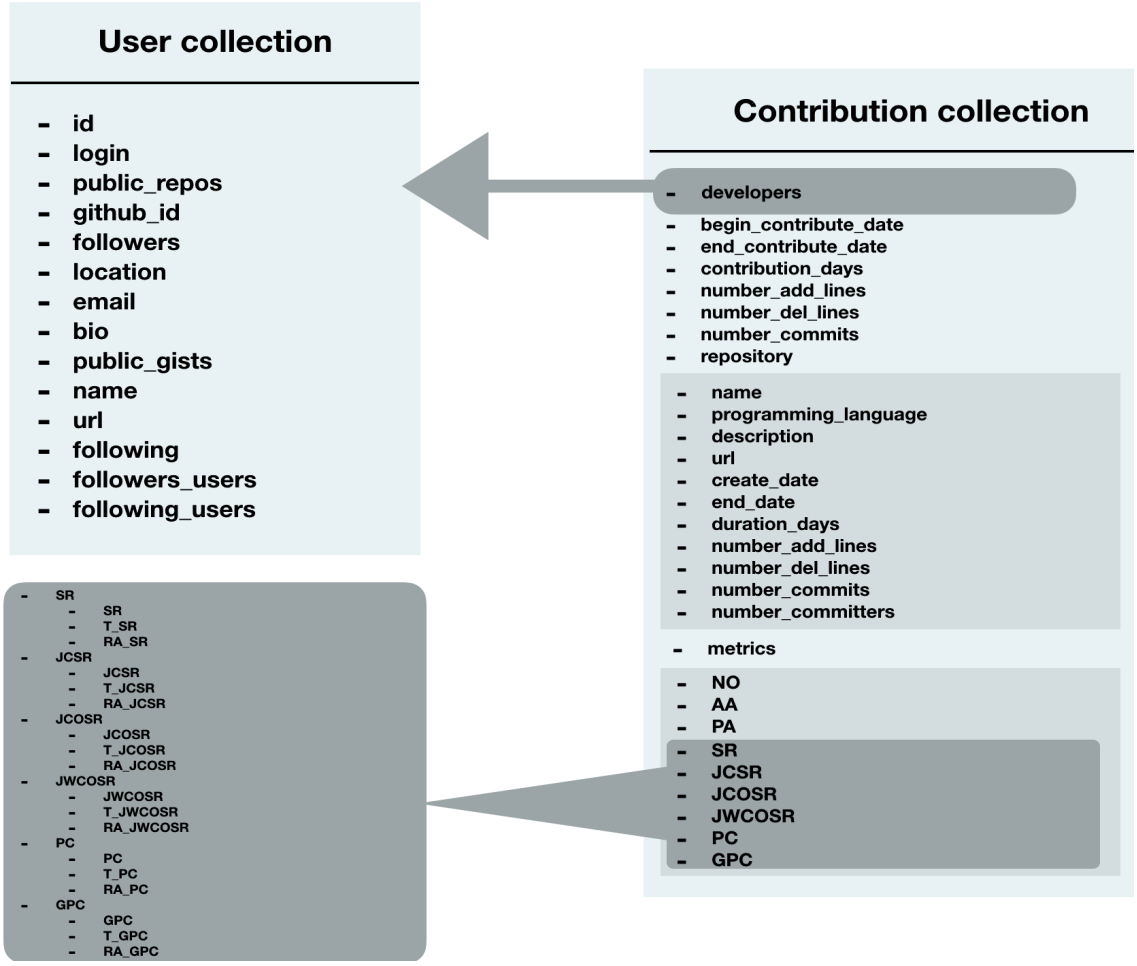


Figura 2: Modelagem do GitSED no MongoDB: coleções de documentos (*collection*) e pares chave-valor (*key-value*). A Tabela 2 descreve as métricas sociais.

Analisando as datas de criação dos repositórios presente na base extraída do GHTorrent e seus respectivos *commits*, identificamos que em alguns casos as datas de alguns *commits* antecedem as próprias datas iniciais de seus repositórios. Dessa forma, *create_date* é definida como a data mínima entre *created_at* do GHTorrent e a data do primeiro *commit*, identificando o início de dado repositório. Além disso, a data de término dos mesmos foi definida como a última data de *commit* no repositório. Finalmente, tais datas permitem a definição da duração do repositório por meio da adição do campo *duration_days*.

A terceira propriedade é a estatística dos *commits*: a quantidade de *commits* (*number_commits*), soma das linhas adicionadas e deletadas (*number_add_lines* e *number_del_lines*) para cada *commit*, e o número de "committers" no repositório (*number_committers*). O último representa os desenvolvedores que efetivamente contribuíram para cada repositório, ou seja, não necessariamente o número original de membros naquele repositório. Finalmente, outra propriedade relacionada aos aspectos sociais do repositório são explicadas no decorrer da modelagem não-relacional.

Modelagem não-relacional do GitSED. A seguir, descrevemos uma modelagem não-relacional para armazenar e trabalhar facilmente com o novo conjunto de dados, conforme

ilustrado na Figura 2. Note que a modelagem refere-se a um banco de dados NoSQL (Not SQL) orientado a documentos chamado MongoDB. Escolhemos esse banco de dados por permitir a inserção de dados em formato JSON (JavaScript Object Notation), o mesmo que o GitHub usa para suas APIs[8], além de fornecer uma estrutura simples e robusta para armazenamento dos dados. Como o MongoDB é baseado em coleções de documentos, foram criadas duas coleções principais que são de usuários e contribuições. Para a criação dessas coleções, processamos e tratamos os dados. A coleção de usuários assemelha-se à forma como o GitHub disponibiliza seus dados, em que para cada documento é armazenado suas informações básicas, como nome, login, url, localização, entre outros. É importante ressaltar que nem todo usuário no conjunto de dados tem uma relação de contribuição na outra coleção, visto que ao selecionar as linguagens JavaScript, Java e Ruby para o GitSED, as contribuições referem-se a usuários que pertencem a repositórios dessas linguagens. Para a coleção de usuários, foram considerados todos os disponíveis e seus respectivos dados, pois no GitHub eles podem colaborar com projetos e repositórios de diferentes linguagens de programação. Assim, o usuário no GitSED pode contribuir em outras linguagens além do JavaScript, Ruby e Java. As informações sobre cada usuário na base de dados também descrevem os repositórios públicos de um usuário (*public_repos*), sua lista de seguidores (*followers_users*) e a lista de usuários a quem tal usuário segue (*following_users*). Além disso, as informações de usuários podem ser facilmente expandidas para a inclusão de outras características como gênero, permitindo análises relevantes sobre diversidade [Vasilescu et al. 2015].

O acesso as informações de colaboração podem ser obtidas através da coleção *contributions*, na qual cada documento representa uma contribuição por um par de desenvolvedores[9]. Para cada contribuição, são armazenados os dados do repositório e totalizadas as contribuições conjuntas de cada par de desenvolvedores. A rede social de colaboração representa todas as interações entre desenvolvedores para os repositórios compartilhados em cada uma das linguagens de programação. Quando desenvolvedores contribuem em diferentes repositórios, um valor é incrementado para representar a interação desse par. Quantificamos tais valores para cada par de desenvolvedores, considerando seu número de *commits* e as quantidades de linhas adicionadas e deletadas nos repositórios. Também adicionamos a data de início e fim de tal interação. Ambas as datas permitem computar a interação de tempo para cada par de desenvolvedores, e podem ser úteis para análises complexas da rede baseada em tempo.

Em relação às métricas de redes sociais, adicionamos valor social a base de dados ao incorporar tais métricas, descritas brevemente na Tabela 2. As três primeiras métricas apresentadas abordam propriedades topológicas da rede (NO, AA e PA), enquanto as demais utilizam propriedades semânticas para realização dos cálculos. Para cada uma das métricas semânticas, foram realizadas combinações com a *tieness* [Brandão and Moro 2017] e *resource allocation*, chamadas, respectivamente, de T e RA, conforme apresentado na Figura 2. Mais detalhes sobre as formas de cálculo das métricas podem ser encontrados em [Batista et al. 2017]. Aspectos distintos representados por métricas diferentes permitem medir a força de interação entre desenvolvedores. Até o momento, a coleção *contributions* tem os campos que representam as métricas de colaboração social propos-

[8]APIs do GitHub: `https://developer.github.com/`

[9]O modelo permite a inserção de colaborações entre mais de dois desenvolvedores, caso seja necessário.

tas por Alves et. al [2016] e Batista et. al [2017] , além de métricas topológicas, como *Acadamic-Adar* [Adamic and Adar 2003], sobreposição de vizinhos (*neighborhood overlap*) [Easley and Kleinberg 2010] e *preferential attachment* [Liben-Nowell and Kleinberg 2003]. Ademais, o GitSED permite o cálculo de outras métricas que representam a força da colaboração entre desenvolvedores, especialmente métricas topológicas.

4. Aplicações

Essencialmente, qualquer estudo que precise de uma amostra do GitHub pode se beneficiar do GitSED. Especialmente, aqueles relacionados à análise de redes sociais [Batista et al. 2017]. O principal diferencial é o acesso direto e rápido às informações de desenvolvedores, dados de sua produtividade (por exemplo, número de linhas por *commit*) e a rede social sem ter que processar uma grande quantidade de dados.

Por exemplo, estudos sobre correlação são importantes para identificar a dependência entre variáveis e melhor representar sua realidade armazenada nos dados. De fato, Sinha et al. 2016 investigam o sentimento dos desenvolvedores ao analisar os logs dos *commits*. Eles encontraram uma forte correlação entre o número de arquivos modificados e os sentimentos expressos. Além disso, Barnett et al. 2016 estudam a correlação entre volume e conteúdo das mensagens dos *commits*, e a tendência a defeitos em projetos Java. Similarmente, Alves et al. 2016 e Batista et al. 2017 analisaram a correlação entre diferentes propriedades topológicas e semânticas que podem ser usadas para medir a força de colaboração social entre desenvolvedores. Todas esses estudos podem utilizar facilmente o GitSED.

Além das análises de correlação, GitSED pode ser usado para identificar desenvolvedores principais (*hubs*) na rede social, descobrir repositórios com mais interação ativa entre desenvolvedores, e até mesmo predizer futuras interações entre desenvolvedores. Todas essas análises podem revelar intuições para melhorar o processo de desenvolvimento de software, como também identificar os melhores desenvolvedores para resolver um problema e os repositórios mais populares.

Em geral, o conjunto de dados apresentado neste trabalho beneficia tais aplicações ao permitir a computação fácil e direta de diferentes métricas de redes sociais. Essas métricas podem ser topológicas, incluindo coeficiente de clusterização, sobreposição de vizinhos, coeficiente *adamic-adar*, entre outras [Easley and Kleinberg 2010]. Mais importante, elas também poderiam ser semânticas, incluindo a frequência absoluta de interação entre desenvolvedores, o número de repositórios compartilhados, a força de colaboração e outras [Alves et al. 2016; Batista et al. 2017; Casalnuovo et al. 2015].

5. Desafios e Limitações

Existem três limitações na nossa base de dados, conforme descritas a seguir.

Número limitado de linguagens de programação. Essa versão inicial da base de dados considera apenas repositórios desenvolvidos em três linguagens de programação, JavaScript, Ruby e Java, as quais são muito comuns. Entretanto, estamos trabalhando para incluir e possibilitar o processamento de todas as linguagens de programação.

Perda de *commits*. Ao usar o *crawler* para coletar o número de linhas, alguns *commits* não são coletados devido às razões descritas na Seção 3. Essa é uma limitação pois

alguns *commits* de determinados pares de desenvolvedores podem não estar disponíveis para coleta e por isso, estes pares ficam ausentes na rede. Entretanto, isso também é uma vantagem, porque todos os usuários falsos (*fake*) criados pelo GHTorrent não são coletados pelo *crawler*, sendo excluídos do GitSED.

Ambiguidade de nomes para usuários do GitHub. Segundo Vasilescu et al. 2015, a ambiguidade de nomes ocorre quando o nome e email de *committers* são definidos localmente ou quando GHTorrent insere contas de usuários falsos para contribuições feitas por desconhecidos. Nesse ponto, apenas o segundo caso é resolvido, pois o *crawler* ignora os usuários desconhecidos. Ademais, solucionar a ambiguidade de nomes em uma base de dados é um desafio [Han et al. 2017; Lee et al. 2007].

6. Conclusão

Nesse artigo, apresentamos uma base de dados do GitHub curada, aumentada, enriquecida e modelada de forma não-relacional, chamada GitSED. Tal conjunto de dados armazena a rede social de colaboração entre desenvolvedores de três linguagens de programação (JavaScript, Ruby e Java). Além disso, GitSED permite a análise de diferentes aspectos sociais, incluindo links de colaboração entre desenvolvedores. O GitSED também oferece diferentes vantagens, como sua facilidade de usar e replicar, bem como a capacidade de computar diferentes métricas de redes sociais.

Como trabalhos futuros, planejamos incluir repositórios de diferentes linguagens de programação e transformar em uma rede social heterogênea incluindo diferentes tipos de interação entre desenvolvedores, como os *pull requests*, comentários, *issues*, etc. O objetivo principal é lançar novas versões do GitSED cobrindo mais linguagens de programação e métricas sociais pré-computadas. Além disso, planejamos incluir métricas relacionadas à produtividade dos desenvolvedores.

Agradecimentos. Trabalho parcialmente financiado por CAPES, CNPq e FAPEMIG.

Referências

Adamic, L. A. and Adar, E. (2003). Friends and neighbors on the web. *Social Networks*, 25(3):211 – 230.

Alves et al., G. B. (2016). The strength of social coding collaboration on github. In *Procs. of SBBD*, pages 1–6, Salvador, Brasil.

Barnett et al., J. G. (2016). The relationship between commit message detail and defect proneness in java projects on github. In *Procs. of MSR*, pages 496–499, Austin, USA.

Bartusiak, R. et al. (2016). Cooperation prediction in github developers network with restricted boltzmann machine. In *Procs. of ACIIDS*, pages 96–107, Da Nang, Vietnã.

Batista et al., N. A. (2017). Collaboration Strength Metrics and Analyses on GitHub. In *Procs. of WI - Aceito para publicação*, Leipzig, Alemanhã.

Bettenburg, N. and Hassan, A. E. (2010). Studying the impact of social structures on software quality. In *Procs. of ICPC*, pages 124–133, Braga, Portugal.

Brandão, M. A. and Moro, M. M. (2017). Social professional networks. *Computer Communications*, 100(C):20–31.

Brandão, M. A. and Moro, M. M. (2017). The strength of co-authorship ties through different topological properties. *Journal of the Brazilian Computer Society*, 23(1):5.

Casalnuovo, C., Vasilescu, B., Devanbu, P., and Filkov, V. (2015). Developer onboarding in github: The role of prior social links and language experience. In *Procs. of ESEC/FSE*, pages 817–828, Bergamo, Itália.

Dabbish, L., Stuart, C., Tsay, J., and Herbsleb, J. (2012). Social coding in GitHub: transparency and collaboration in an open software repository. In *Procs. of CSCW*, pages 1277–1286, Seattle, USA.

Easley, D. and Kleinberg, J. (2010). *Networks, crowds, and markets: Reasoning about a highly connected world*. Cambridge University Press.

Gousios, G. (2013). The GHTorrent dataset and tool suite. In *Procs. of MSR*, pages 233–236.

Han et al., H. (2017). Semantic fingerprints-based author name disambiguation in Chinese documents. *Scientometrics*, 111(3):1879–1896.

Jiang, J. et al. (2013). Understanding project dissemination on a social coding site. In *Procs. of WCRE*, pages 132–141, Koblenz, Alemanhã.

Kery, M. B., Le Goues, C., and Myers, B. A. (2016). Examining programmer practices for locally handling exceptions. In *Procs. of MSR*, pages 484–487, New York, USA.

Lee, D., Kang, J., Mitra, P., Giles, C. L., and On, B.-W. (2007). Are your citations clean? *Communications of the ACM*, 50(12):33–38.

Liben-Nowell, D. and Kleinberg, J. (2003). The link prediction problem for social networks. In *Procs. of CIKM*, pages 556–559, New Orleans, USA.

Padhye, R., Mani, S., and Sinha, V. S. (2014). A study of external community contribution to open-source projects on GitHub. In *Procs. of MSR*, pages 332–335, Hyderabad, India.

Proksch et al., S. (2016). A Dataset of Simplified Syntax Trees for C#. In *Procs. of MSR*, pages 476–479, New York, USA.

Rocha, L. M. A., Silva, T. H. P., and Moro, M. M. (2016). Análise da contribuição para código entre repositórios do github. In *Procs. of SBBD*, pages 103–108, Salvador, Brasil.

Sinha, V., Lazar, A., and Sharif, B. (2016). Analyzing developer sentiment in commit logs. In *Procs. of MSR*, pages 520–523, New York, USA.

Tsay, J., Dabbish, L., and Herbsleb, J. (2014). Influence of social and technical factors for evaluating contribution in github. In *ICSE*, pages 356–366.

Vasilescu, B., Serebrenik, A., and Filkov, V. (2015). A Data Set for Social Diversity Studies of GitHub Teams. In *Procs. of MSR*, pages 514–517, Florence, Itália.

Vasilescu et al., B. (2015). Quality and productivity outcomes relating to continuous integration in github. In *Procs. of ESEC/FSE*, pages 805–816, Bergamo, Itália.

IntergenicDB: Banco de dados de regiões intergênicas de Bactérias Gram-Negativas

Daniel L. Notari[1], Jovani Dalzochio[1], Camila R. T. Andrade[1], Jórdan R. Rosa[1], Hugo A. Klauck[2], Scheila de Ávila e Silva[2]

[1] Área do Conhecimento de Ciências Exatas e Engenharias – Universidade de Caxias do Sul, Rua Francisco Getúlio Vargas, 1130, CEP 95070-560 - Caxias do Sul – RS – Brazil

[2] Instituto de Biotecnologia – Universidade de Caxias do Sul, Rua Francisco Getúlio Vargas, 1130, CEP 95070-560 - Caxias do Sul – RS – Brazil

{dlnotari,sasilva6}@ucs.br, {jovanidalzochio, camila.rachel, jordan.rs2006, hugoklauck}@gmail.com

Abstract. *There are several biological databases devoted to make available regulatory sequences for their analysis. However, there are not databases specialized on intergenic sequences of bacteria with information from each associated gene. Thus, the IntergenicDB portal is a public repository that is available on the web. IntergenicDB allows researchers to query information of intergenic regions and associated biological functions through a user-friendly interface. This article aims to describe the data retrieve, clean, association and load of the IntergenicDB Portal dataset, and present applications for this data.*

Resumo. *Existem alguns bancos de dados que disponibilizam sequências regulatórias, mas não um banco de dados de sequências intergênicas de bactérias com as informações de cada gene associado. Assim, o portal IntergenicDB, um repositório público, vem ao encontro dessa necessidade. Ele permite aos pesquisadores consultar as informações no banco de dados integrados de regiões intergênicas e funções biológicas associadas por meio de uma interface amigável. Este artigo tem como objetivo descrever a busca, manipulação e carga do conjunto de dados das regiões intergênicas do Portal IntergenicDB e apresentar aplicações para esses dados.*

1. Introdução

Os fenômenos biológicos são muito complexos e requerem a integração de diversas áreas do conhecimento para a comprovação ou refutação de hipóteses. A interface interdisciplinar mais antiga (e talvez a mais conhecida) entre a Biologia e as Ciências Exatas é a Bioestatística. Gradualmente, nos últimos anos, a Biologia tem utilizado as ferramentas proporcionadas pela Informática e pela Matemática para a resolução de problemas nos mais diversos campos: desde a Genética até a Ecologia [Barrera et al. 2004; Gannon e Reed 2009; Lesk 2013; Marx 2013].

Um dos maiores desafios da era pós-genômica é a determinação de quando, onde e como os genes são "ligados" e "desligados". A diferença entre duas espécies está muito mais relacionada com a transcrição de seus genes do que com a estrutura destes em si [Kanhere e Bansal 2005; Lehninger et al. 2013]. Assim, o estudo da regulação gênica contribui para a construção do conhecimento a respeito da funcionalidade dos genes em diferentes espécies, na questão da diferenciação celular em organismos multicelulares, na resposta celular frente às mudanças ambientais, entre outras questões [Howard e Benson 2003; Cotik et al. 2005].

Neste contexto, elementos regulatórios da transcrição gênica encontram-se em sequências denominadas intergênicas, as quais consistem em um elemento não transcrito que contém as sequências responsáveis pelo processo de regulação de início e término da expressão dos genes. Em organismos procariotos, como bactérias e outros organismos unicelulares, as sequências intergênicas estão relacionadas a um ou mais genes [Zaha et al. 2014].

Ao estabelecer uma analogia, os elementos *downstream* (como os genes) representam a memória de um computador e os elementos *upstream* (como os promotores) os programas que atuam nessa memória. Assim, o estudo dos elementos *upstream* pode prover modelos sobre a constituição do "programa" e de como este opera [Howard e Benson 2003]. Assim, as informações biológicas relacionadas a uma determinada sequência intergênica, como gene associado a ela, papel biológico do gene e outras informações, contribuem para a ampliação do conhecimento biológico relacionado aos elementos regulatórios.

Contudo, encontrar um banco de dados biológico que possua um gene associado a uma região intergênica específica não é trivial. Aliado a isso, o conhecimento computacional não é homogêneo entre pesquisadores de Bioinformática que precisam realizar análise de diferentes fontes de dados. O IntergenicDB é um banco de dados público desenvolvido para o estudo de sequências intergênicas [Notari et al. 2014] e, foi modelado para armazenar informações relevantes sobre a estrutura das sequências intergênicas de bactérias Gram-negativas.

Para tanto, este artigo descreve a metodologia utilizada para construir o banco de dados IntergenicDB, descreve o *dataset* disponibilizado e as aplicações que fazem uso dos seus dados. O artigo está organizado com as seções de Materiais e Métodos,

Resultados, Aplicações com Trabalhos Relacionados e Conclusões.

2. Material e Métodos

Esta seção apresenta as fontes de dados utilizadas, o modelo conceitual e lógico do *dataset* e a descrição do software para a geração do *dataset*.

2.1 Fonte de Dados

Foram utilizados os dados dos bancos de dados biológico GenBank [Clark et al. 2016] e Kegg [Kanehisa et al. 2017]. O National Center for Biotechnology Information (NCBI) provê uma gama de recursos *online* de dados e informações biológicas, incluindo alguns bancos de dados, como o banco de dados de sequências de ácido nucleico (GenBank) e um banco de dados de citações e resumos para revistas científicas (PubMed) [NCBI 2016].

O Genbank é um abrangente banco de dados que contém sequências de DNA com mais de 340 000 organismos nomeados [Clark et al. 2016]. Os dados desse banco de dados são obtidos, principalmente, através de submissões individuais feitas por laboratórios ou por lotes de submissões feitas por projetos de sequenciamento de larga escala. Os dados do Genbank são disponibilizados através de ferramenta de busca Entrez [Maglott et al. 2011] e podem ser obtidos através de chamadas URL ou *download* de arquivos disponibilizado pelo NCBI. O segundo banco de dados utilizado foi o KEEG (Kyoto Encyclopedia of Genes and Genomes). Este banco de dados é uma enciclopédia de genes e genomas, e fornece significado a genes e genomas, tanto em nível molecular quanto a níveis mais elevados [Kanehisa et al. 2017]. Hoje, o KEEG é composto por quinze bancos de dados com curadoria manual e um banco de dados gerado computacionalmente.

Para a coleta das informações de organismos, genes, família, reino e região intergênica foi realizado o *download* dos arquivos dos organismos usando o FTP do GenBank. A única informação que não tem origem no GenBank é o papel biológico. Para tal, foi realizada uma busca no banco de dados Keeg Brite utilizando a própria ferramenta disponível no seu *website*.

Os arquivos texto são usados com muita frequência para submeter ou adquirir informações de bancos de dados biológicos. O formato de arquivo texto chamado de FASTA é um tipo de arquivo comum utilizado para armazenar sequências de ácidos nucleicos, aminoácidos e RNA [Pearson e Lipman 1988].

2.2 Modelo de Dados

A Figura 1 apresenta o diagrama Entidade-Relacionamento do banco de dados IntergenicDB utilizando a notação de Heuser (2009). Os dados armazenados neste banco de dados são as seguintes [Notari 2012]: i) cada região promotora está ligada a um organismo, que possui um nome, um reino, uma família, um tipo de molécula e um papel biológico; ii) cada gene possui um nome, um símbolo, um número de identificação de início e fim, uma função e uma quantidade percentual de CG; iii) cada

região intergênica possui um número para a posição inicial e final na sequência, um tamanho, sua sequência de nucleotídeos e o tipo de fita a que pertence.

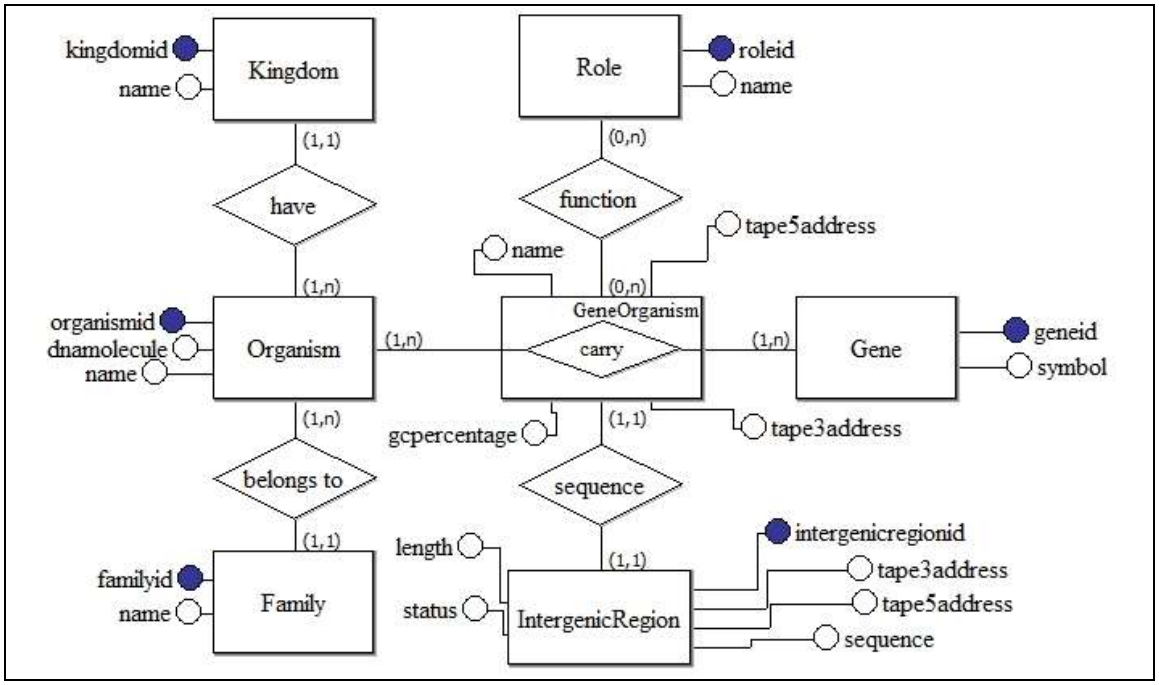


Figura 1: Modelo Conceitual do IntergenicDB

Family (familyid, name), *Kingdom* (kingdomid, name), *Gene* (geneid, symbol), *Role* (roleid, name)
Organism (organismid, name, dnamolecule, #familyid, #kingdomid)
GeneOrganism (geneorganismid, name, gcpercentage, tape5address, tape3address, #geneid, #organismid)
GeneOrganismRole (geneorganismroleid, #geneorganismid, #geneid)
IntergenicRegion (intergenicregionid, tape5address, tape3address, sequence, length, status, #geneorgnismid)

Figura 2: Modelo Lógico do IntergenicDB

A Figura 2 apresenta as tabelas geradas a partir da aplicação e adaptação das regras de tradução de Heuser[1] (2009) no modelo conceitual da Figura 1. A tabela *Family* possui os dados de todas as famílias de organismos. A tabela *Kingdom* possui os dados de reino dos organismos. A tabela *Organism* possui os dados dos organismos biológicos. A tabela *Gene* contém os dados dos genes para cada organismo biológico. A tabela *Role* possui os dados sobre que papel o gene desempenha. A tabela *IntergenicRegion* armazena as regiões intergênicas de cada gene. A tabela *GeneOrganism* contém os dados de cada gene de um organismo. E, por fim, a tabela *GeneOrganismRole* contém os dados dos papéis biológicos de cada gene de um organismo.

[1] Coluna sublinhada indica chave primária, enquanto que *hashtag* (#) indica chave estrangeira

2.3 Importador de Dados

A Figura 3 apresenta a sequência de atividades realizadas para a geração dos dados para o IntergenicDB. Esta ferramenta foi denominada MMDBImportTool. Nela é possível configurar o diretório FTP do GenBank para a busca dos dados, bem como, buscar as informações banco de dados KEGG BRITE [Dalzochio 2014].

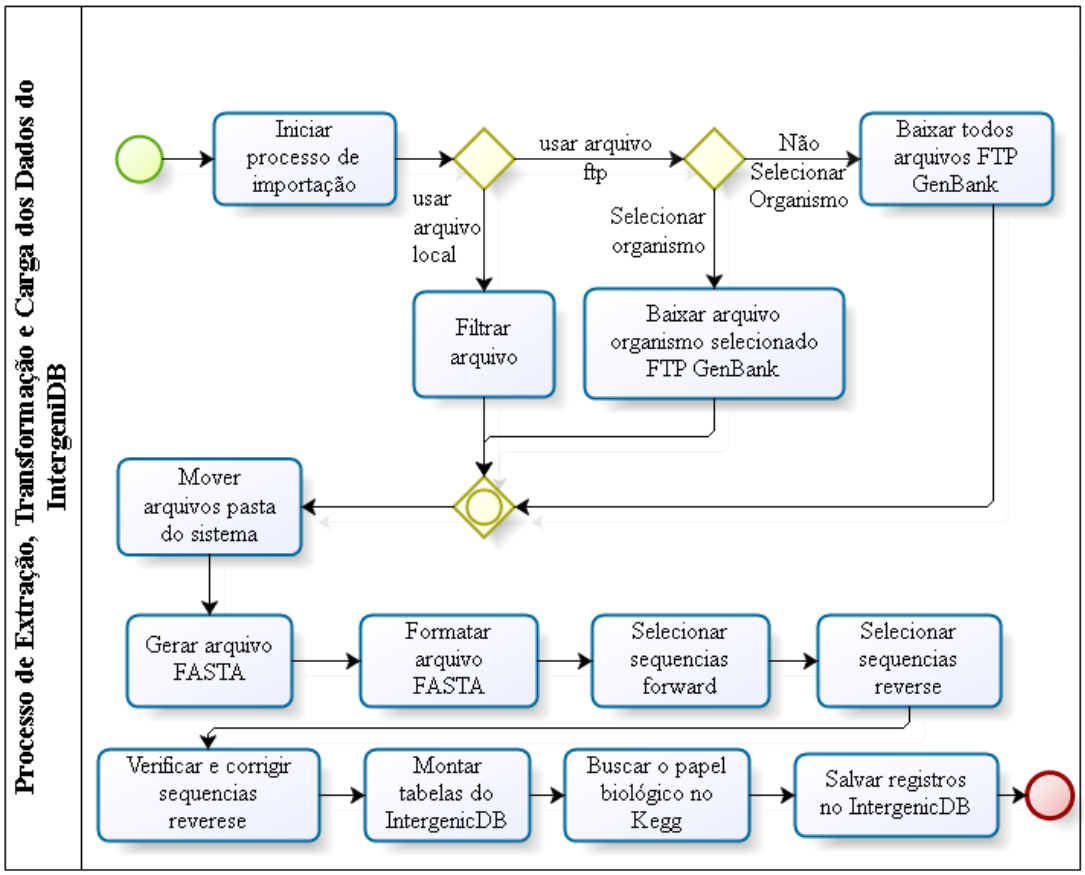


Figura 3: Atividades executadas pelo Importador de Dados

O processo inicia com a configuração do acesso ao FTP do GenBank com a seleção dos arquivos dos organismos a serem processados. Estes arquivos são baixados e salvos em uma pasta local. Sempre que um novo organismo for necessário, faz-se a busca desse arquivo. Após isso, usam-se os arquivos locais para processamento.

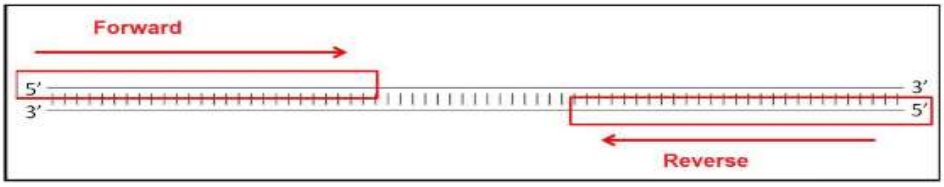


Figura 4: Leitura de fita foward X fita reverse

Na sequência, o *script* ExecutePytonFileMuchosArquivos é responsável por gerar um arquivo do tipo FASTA que será utilizado pelos outros *scripts*. Este arquivo é utilizado para separar os dados da fita dupla de DNA, denominada *forward* e *reverse* (*scripts* ExecutePytonFileSelecionaFoward e ExecutePytonFileSelecionaReverse).

Os arquivos *foward* e *reverse* devem ser tratados de forma diferente, pois como pode ser visto na Figura 4, a forma como deve ser recuperada a informação da posição de início da sequência gênica muda conforme o tipo da sequência [Dalzochio 2014].

Tabela 1: Mapeamento entre os dados extraídos do GeneBank com as tabelas do IntergenicDB

Origem	**Destino**	**Descrição**
Genbank (tag organism)	Family.Name	Nome da família do organismo
Genbank (tag organismo	Kingdom.Name	Reino do organismo
Genbank (tag organism)	Organism.Name	Nome do organismo
Genbank (tag mol_type)	Organism.DnaMolecule	Molécula de DNA
Genbank (tag gene)	Gene.Symbol	Símbolo do gene
Keeg	Role.Name	Papel biológico do gene
Genbank (tag product)	GeneOrganism.Name	Nome do organismo
Genbank (gerado scripts)	GeneOrganism.GCPercentage	% de GC da sequência de DNA
Genbank (tag complement)	GeneOrganism.Tape3Address	Fita de DNA sentido *forward*
Genbank (complement)	GeneOrganism.Tape5Address	Fita de DNA sentido *reverse*
Genbank (usa arquivo FASTA)	IntergenicRegion.Sequence	Sequência região intergênica
Genbank (usa arquivo FASTA)	IntergenicRegion.Tape3Address	Posição sequência *forward*
Genbank (usa arquivo FASTA)	IntergenicRegion.Tape5Address	Posição sequência *reverse*
Genbank (usa arquivo FASTA)	Length	Tamanho da região intergênica
Genbank (Genbank)	Status	F – Foward, R – Reverse

Por fim, o *script* ExecutaPythonFileCorrigeSequenciaReversa é responsável por corrigir as sequências reversas, invertendo-as e depois substituindo os nucleotídeos A por T, T por A, C por G e G por C [De Robertis, 2017; Alberts 2017]. O processamento dos *scripts* na linguagem *python* é realizado para obter os dados de interesse para a população do banco de dados IntergenicDB. O penúltimo passo envolve buscar as informações do papel biológico no Kegg. Por fim, os dados são salvos na tabela do

banco de dados.

3. Resultados

A execução dos passos do Importador MMDBImportTool descritos na Figura 3 geraram os dados salvos nas tabelas do IntergenicDB descritas na Figura 2. A tabela 1 apresenta a relação entre os dados importados do GenBank e os dados armazenados no IntergenicDB. Os dados da Tabela 1 compreendem: i) Origem - refere-se ao dado extraído do arquivo de dados; ii) Destino - refere-se à tabela e coluna onde o dado foi colocado; e, iii) Descrição - refere-se a dizer ao que o dado se refere.

O conjunto de dados gerado no formato CSV[2] para pesquisadores utilizarem envolveu a execução de uma consulta SQL no banco de dados IntergenicDB mostrada na Figura 5.

```
Select o.name as organism, g.symbol as gene, k.name as kingdom, f.name as family, r.name as role,
ir.tape5address, ir.tape3address, ir.sequence
from organism o, gene g, geneorganism go, kingdom k, family f, role r,geneorganismrole gor,
intergenicregion ir
where o.familyid = f.familyid and o.kingdomid = k.kingdomid and o.organismid = go.organismid and
g.geneid = go.geneid and  go.geneorganismid = gor.geneorganismid and r.roleid = gor.roleid and
go.geneorganismid = ir.geneorganismid
```

Figura 5: Consulta SQL para gerar o DataSet

A estrutura do *dataset* gerado pela execução da consulta da Figura 4 possui a seguinte estrutura:

- *organism*: nome do organismo
- *gene*: símbolo do gene
- *kingdom*: reino do organismo
- *family*: nome da família do organismo
- *role*: papel biológico do gene
- *tape5address*: posição da sequência no sentido *forward*
- *tape3address*: posição da sequência no sentido *reverse*
- *sequence*: sequência de dados da região intergênica

O conjunto de dados possui cadastrados 75252 linhas com os dados combinados. Além disso, possui dados de 88 organismos, 15095 genes, 12565 funções biológicas e 55635 sequências de DNA de regiões intergênicas diferentes. Os dados importados limitam-se às bactérias *gram negativas.*

4. Aplicações e Trabalhos Relacionados

O objetivo do IntergenicDB é ser um portal de consultas para regiões intergênicas sobre os dados armazenados de organismos procariontes [Avila e Silva 2011; Notari 2012]. O portal possui uma área administrativa de acesso privado e uma

[2] Disponível para download em http://www.bioinfoucs.com/intergenicdb_dataset.zip

área de consulta pública.

A área administrativa do portal pode ser acessada mediante usuário e senha. O acesso a essa área possibilita o gerenciamento dos cadastros de usuários e grupos do portal, sendo que o grupo ao qual o usuário está associado lhe proverá mais ou menos acessos. É possível também consultar individualmente os dados biológicos inseridos no banco de dados, não sendo permitida a inserção, nem a manutenção dos dados. A área administrativa ainda provê aos administradores acesso a dados de geoposição dos usuários que acessam o portal e a administração dos textos das páginas de Início e Ajuda do portal [Andrade 2017].

Para otimizar o desempenho da consulta à base dados, devido aos diversos relacionamentos e volume resultante, foi implementado um *data warehouse,* utilizando o 6modelo estrela [Rosa 2017]. A tabela fato (Figura 6) é formada somente pelos identificadores das tabelas dimensões. A tabela fato é estática, ou seja, os novos dados não são adicionados a tabela no momento da sua inserção. Para isso, é necessário rodar uma rotina que destrói a tabela fato e recria.

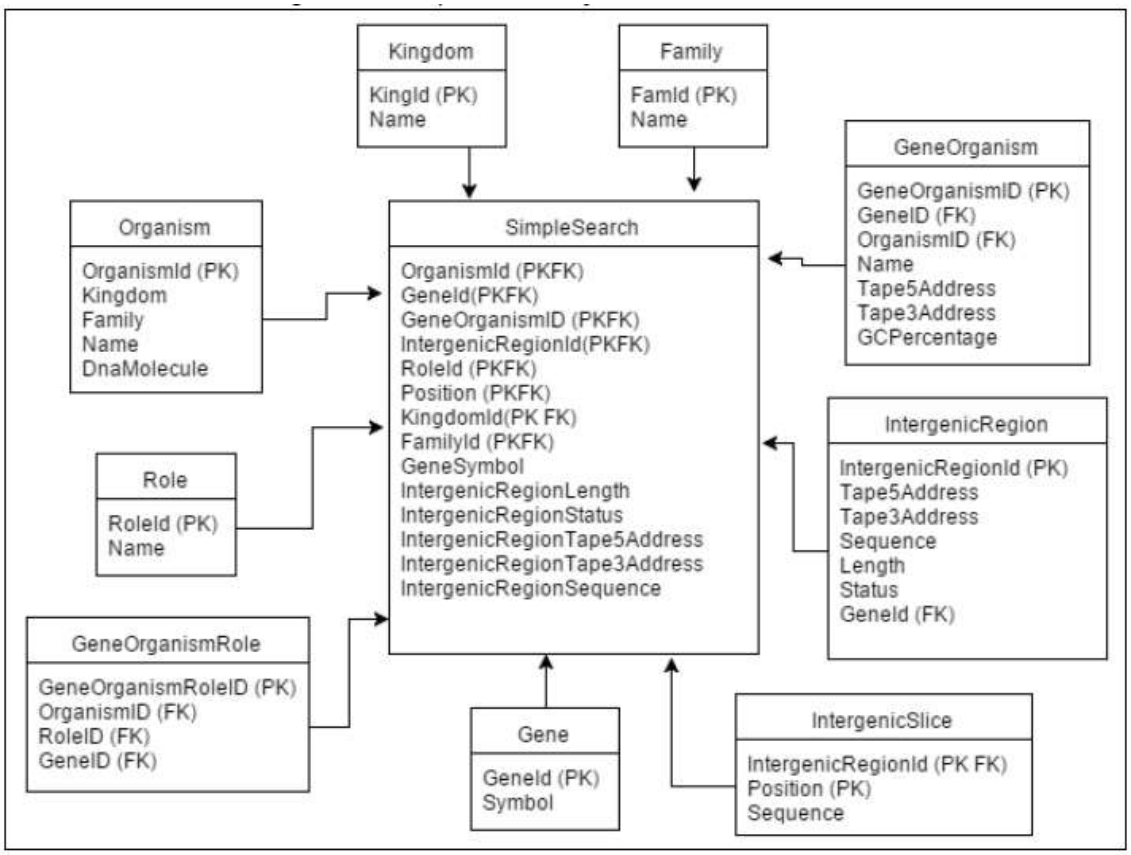


Figura 6: Tabela fato e suas tabelas dimensões

A consulta, disponibilizada no portal do IntergenicDB, possui interface baseada em *query builders* dos portais como, por exemplo, *"PubMed Advanced Search Builder"*

do NCBI[3], "*Free text search"* do EBI[4] "*ARSA"* do DDBJ[5] e outros. A partir dos parâmetros selecionados pelo usuário, o mecanismo de consulta monta uma consulta SQL. O usuário necessita possuir conhecimento em operadores lógicos para montar a pesquisa que deseja [Rosa 2017]. Uma vez com os parâmetros informados, o mecanismo gera a consulta buscando os identificadores dos parâmetros em nas tabelas dimensões e assim, após obtê-las, a consulta segue na tabela fato. Com os identificadores filtrados a partir da tabela fato, a consulta busca seus nomes nas tabelas dimensões. A consulta somente ocorre no dentro do intervalo de linhas da paginação determinada pelo usuário, ou seja, se o usuário parametrizar na consulta que deseja visualizar 50 linhas por página, a consulta somente ocorrerá dentro desse intervalo para minimizar o volume de dados a ser manipulado.

4.1 Trabalhos Relacionados

O banco de dados denominado lncRNA [Scott et al. 2017] é uma coleção de informações de animais equinos. Foi criado a partir da aplicação de um *pipeline* para análise de sequências de RNA candidatas. A montagem do banco de dados usou os dados de 59 cavalos armazenados no NCBI de sequências de RNA com anotação genômica de regiões intergênicas.

Tong et al. (2017) analisaram o perfil de transcrições de RNA de glândulas mamárias bovinas, utilizando-se o banco de dados SRA (Sequence Read Archive) do NCBI. Vários programas de bioinformática foram analisados para a montagem dos transcritos de glândulas mamárias bovinas visando à identificação e categorização regiões intergênicas não codificantes. Neste trabalho, não foi criado um banco de dados persistente.

O banco de dados RiboGap [Naghdi et al. 2017] é uma ferramenta para encontrar regiões não codificantes de RNAs, bem como para descobrir funções biológicas de sequências simples com funções reguladoras.

Os trabalhos apresentados destacam a análise de sequências de RNA para diferentes organismos. O Integenicdb foi criado para armazenar sequências de regiões intergênicas de genes baseados no DNA de bactérias gram-negativas permitindo futuras análises *in silico* usando por ferramentas de bioinformáticas.

5. Conclusão

Existe uma gama de ferramentas destinadas à análise de elementos regulatórios da expressão gênica. Ao mesmo tempo em que isso amplia o espaço de hipóteses geradas, a diversidade de padrões (ou a falta de) de implementação torna-se um fator limitante na obtenção e comparação de resultados.

O desenvolvimento de um banco de dados para integrar os dados das regiões

[3] https://www.ncbi.nlm.nih.gov/pubmed/advanced
[4] http://www.ebi.ac.uk/ena/data/warehouse/search
[5] http://ddbj.nig.ac.jp/arsa/

intergênicas em um repositório único mostra-se importante no auxílio ao pesquisador na correlação dessas informações com outras ferramentas *on-line,* destinadas à análise de elementos regulatórios da transcrição gênica.

Adicionalmente, está em desenvolvimento a conexão entre as sequências depositadas neste banco de dados com outras ferramentas on-line de análise de regiões regulatórias. Considerando esta questão, este projeto visa integrar as regiões intergênicas armazenadas no IntergenicDB com outras ferramentas disponíveis na internet para análise de dados e/ou predição de sequências como: promotores (ex: BacPP, NNPP, Scope, PromPredict), motivos consensuais (ex: ClustalO, WebLogo), terminadores (ex: WebGeSTer DB, Arnold), fatores de transcrição e seus sítios de ligação (Tfsitescan), dentre outras ferramentas.

Referências

Alberts, B. (2017) Fundamentos da Biologia Molecular. Porto Alegre: Artmed.

Andrade, C. R. T. D. (2017) "IntergenicDB 2.0". Trabalho de Conclusão do Curso de Sistemas de Informação da Universidade de Caxias do Sul.

Avila e Silva, S. (2011) "Redes neurais artificiais aplicadas no reconhecimento de regiões promotoras em bactérias Gram-negativas". Tese de Doutorado, Programa de Pós-Graduação em Biotecnologia da Universidade de Caxias do Sul.

Barrera, J.; Cesar-Jr, R. M.; Ferreira, J. E., Gubitoso, M. E. (2004) "An Environment for knowledge discovery in biology". Computers in Biology and Medicine 34: 427-447.

Clark, K., Karsch-Mizrachi, I., Lipman, D. J., Ostell, J., & Sayers, E. W. (2016) "GenBank". Nucleic Acids Research, 44(Database issue), D67–D72.

Cotik, V.; Zaliz, R. R.; Zwir, I. (2005) "A hybrid promoter analysis methodology for prokaryotic genomes". Fuzzy Sets and Systems152: 83-102.

Dalzochio, J. (2014) "Implementação de novas funcionalidades para o portal IntergenicDB e povoamento do seu banco de dados". Trabalho de Conclusão do Curso de Sistemas de Informação da Universidade de Caxias do Sul.

De Robertis, E. D. P. (2017) Bases da biologia celular e molecular. 16. ed. Rio de Janeiro: Grupo Gen.

Gannon, D. & Reed, D. (2009) "Parallelism and the cloud, The fourth paradigm: Data intensive scientific discovery". T. Hey et al., eds. Washingtong: Microsoft Research, 131-135.

Heuser, C. (2009) Projeto de Banco de Dados. Porto Alegre: Bookman, v. 4.

Howard, D.; Benson, K. (2003) "Evolutionary computation method for pattern recognition of cis-acting sites". BioSystems, 72: 19-27.

Kanhere, A.; Bansal, M. (2005) "Structural properties of promoters: similarities and differences between prokaryotes and eukaryotes". Nucleic Acids Research 33:3165-

3175.

Kanehisa, M., Furumichi, M., Tanabe, M., Sato, Y., & Morishima, K. (2017) "KEGG: new perspectives on genomes, pathways, diseases and drugs". Nucleic Acids Research, 45(Database issue), D353–D361.

Lehninger, A. L.; Cox, M. M.; Nelson, D. L. (2013) Principles of biochemistry. 6. ed. New York: Worth.

Lesk, A. (2013) Introduction to bioinformatics. Oxford University Press.

Marx, V. (2013) "Biology: The big challenges of big data". Nature 498, 255–260.

Maglott, D., Ostell, J., Pruitt, K. D., & Tatusova, T. (2011) "Entrez Gene: gene-centered information at NCBI". Nucleic Acids Research, 39(Database issue), D52–D57.

Naghdi, M. R., Smail, K., Wang, J. X., Wade, F, Breaker, R. R, Perreault, J. (2017) "Search for 5′-leader regulatory RNA structures based on gene annotation aided by the RiboGap database", Methods, Vol. 117, Pages 3-13.

NCBI Resource Coordinators. (2016) "Database resources of the National Center for Biotechnology Information". Nucleic Acids Research, 44(Database issue), D7–D19.

Notari, D. L. (2012) "Desenvolvimento de workflows científicos para a geração e análise de diferentes redes de interatomas". Tese de Doutorado, Programa de Pós-graduação em Biotecnologia, Universidade de Caxias do Sul.

Notari, D. L., Molin, A., Davanzo, V., Picolotto, D., Ribeiro, H. G., & Silva, S. de A. e. (2014) "IntergenicDB: a database for intergenic sequences". Bioinformation, 10(6), 381–383.

Pearson, W. and Lipman, D. (1988) "Improved tools for biological sequence comparison (amino acid/nucleic acid/data base searches/local similarity)". Proceedings of the National Academy of Sciences, 85, 2444-2448.

Rosa, J. (2017) "Criação e implementação de um data warehouse para reformatar o mecanismo de pesquisa do portal IntergenicDB 2.0". Trabalho de Conclusão do Curso de Ciência da Computação da Universidade de Caxias do Sul.

Scott, E. Y., Mansour, T., Bellone, R. R., Brown, C. T., Mienaltowski, M. J., Penedo, M. C., … Finno, C. J. (2017). "Identification of long non-coding RNA in the horse transcriptome". BMC Genomics, 18, 511.

Tong, C., Chen, Q., Zhao, L., Ma, J., Ibeagha-Awemu, E. M., & Zhao, X. (2017). "Identification and characterization of long intergenic noncoding RNAs in bovine mammary glands". BMC Genomics, 18, 468.

Zaha, A; Ferreira, H. B.; Passaglia, L. M. P. (2014) Biologia Molecular Básica. Porto Alegre: Artmed.

LattesDoctoralDataset: Uma Coleção de Dados Estratificados sobre o Conjunto de Doutores Cadastrados na Plataforma Lattes

Thiago M. R. Dias[1], Alberto H. F. Laender[2], Gray F. Moita[1]

[1]Centro Federal de Educação Tecnológica de Minas Gerais
[2]Departamento de Ciência da Computação - Universidade Federal de Minas Gerais

thiago@div.cefetmg.br, laender@dcc.ufmg.br, gray@dppg.cefetmg.br

Abstract. *Studies on scientific data have attracted the interest of researchers from several areas of knowledge, in view of their potential to better understand how research in a given area has been carried out, or how groups of researchers have collaborated when developing their work. Thus, this work describes the process of extracting, treating and characterizing a stratified dataset containing information about individuals with curricula registered in the Lattes Platform and holding a PhD degree. It also presents a quantitative description of the collected data, as well as an overall description of the datasets made available.*

Resumo. *Os estudos sobre dados científicos têm atraído o interesse de pesquisadores de diversas áreas do conhecimento, tendo em vista seu potencial para melhor compreender como as pesquisas em uma determinada área têm sido realizadas, ou como grupos de pesquisadores têm colaborado no desenvolvimento de seus trabalhos. Assim, este trabalho descreve o processo de extração, tratamento e caracterização de uma coleção de dados estratificados contendo informação sobre os indivíduos com currículos cadastrados na Plataforma Lattes e que possuam doutorado concluído. O trabalho também apresenta uma descrição quantitativa sobre os dados coletados, bem como uma descrição geral dos conjuntos de dados disponibilizados.*

1. Introdução

Uma nova geração de serviços disponíveis principalmente na Web está mudando a forma de divulgar e disponibilizar a produção científica e tecnológica. Existe, atualmente, uma tendência que reforça a troca de informações e a colaboração entre as pessoas. A forte relação entre os domínios científico e socioeconômico tem gerado um interesse crescente pela compreensão dos mecanismos que norteiam as atividades científicas, sendo possível apontar diversos trabalhos que analisam aspectos específicos como as características da linguagem e dos discursos empregados na comunicação científica (Hoffnagel, 2009), bem como a relação de colaboração entre pesquisadores e grupos de pesquisa (Ding, 2011; Revorendo et al., 2012; Stroele, Zimbrão e Souza, 2012).

Para Mugnaini et al. (2014), o levantamento da produção científica de um país permite estudar diversos aspectos que podem ser qualificados como resultados mensuráveis de seu respectivo sistema de ciência, tecnologia e inovação. Acompanhar o

fluxo de comunicação científica das diversas áreas facilita o processo de avaliação dos resultados de pesquisa, cujas características são tão diversificadas quanto a própria ciência. No entanto, o grande volume de dados sobre produção científica disponível em diferentes formatos e em diferentes repositórios dificulta a realização de estudos, bem como a consulta por parte de usuários que necessitam de uma visão unificada desses dados para, por exemplo, possibilitar a identificação de grupos de indivíduos que estejam trabalhando com determinado tema em diferentes instituições ou regiões.

Estudos bibliométricos, principalmente em grandes repositórios bibliográficos, não são tarefas triviais tendo em vista a quantidade de dados a serem analisados e as características dos repositórios que, em sua maioria, não possuem um padrão definido. Atualmente, grande parte desses estudos tem utilizado como principais fontes de dados resultados de consultas a repositórios internacionais que apresentam dados sobre trabalhos científicos, geralmente publicados em periódicos indexados. Entretanto, muitos desses repositórios negligenciam trabalhos publicados em periódicos nacionais que geralmente não são indexados e grande parte dos artigos publicados em anais de congressos, que constituem importante meio de publicação de algumas áreas do conhecimento como, por exemplo, a Ciência da Computação (Laender et al., 2008).

Assim, é evidente a dificuldade existente para se realizar estudos abrangentes que possam apresentar, de forma ampla, análises sobre a produção científica de um grande conjunto de indivíduos que estejam vinculados a diferentes instituições ou que atuem em áreas distintas, como, por exemplo, o conjunto de todos os pesquisadores com um determinado nível de formação ou de uma determinada área de atuação. Diante disso, este trabalho apresenta uma coleção de dados estratificados extraídos dos currículos de todos os indivíduos com doutorado concluído cadastrados na Plataforma Lattes. Essa coleção, denominada LattesDoctoralDataset, inclui dados sobre a formação, orientações concluídas e em andamento, produção científica e colaborações desses indivíduos, possibilitando, desta forma, a realização de diversos estudos sobre esse segmento de pesquisadores brasileiros.

2. Fonte de Dados

Para a geração da coleção de dados apresentada neste trabalho, foram coletados da Plataforma Lattes os currículos de todos os doutores ali registrados. Segundo Lane, em artigo publicado na revista Nature [Lane 2010], medir e avaliar o desempenho acadêmico de seus pares é um fator crucial para qualquer comunidade científica. A autora descreve esforços empregados para a construção de repositórios confiáveis de dados científicos que poderiam permitir análises com o objetivo de explorar e compreender como a ciência tem evoluído. Embora tais esforços sejam importantes, alguns apresentam problemas que comprometem o sucesso dessas iniciativas. Neste cenário, a Plataforma Lattes é citada como exemplo de boas práticas para o fornecimento de dados de alta qualidade sobre a produção científica de um país e de como a sua utilização tem sido incentivada por órgãos federais, instituições acadêmicas e agências de fomento a pesquisa. Por fim, a autora destaca que a Plataforma Lattes é uma das fontes de dados sobre pesquisadores mais confiáveis existentes atualmente.

Para Ferraz, Quoniam e Maccari [2014], até o presente momento, não existe no mundo um repositório curricular nacional semelhante à Plataforma Lattes, sendo que somente repositórios de dados referenciais, de onde se podem extrair referências

bibliográficas, e fontes de informação secundárias estão disponíveis para livre acesso. Dessa forma, a Plataforma Lattes é um instrumento da maior importância para o estudo da produção científica brasileira.

Mugnaini, Leite e Leta [2011] destacam que muito embora não se apresente como uma base de indexação e catalogação de publicações científicas, a Plataforma Lattes é uma fonte inesgotável de informação sobre a ciência brasileira, sob diversos aspectos e abordagens. Os autores ressaltam que, apesar de todo o volume de informação disponível, o que se observa ainda é uma baixa frequência de estudos cientométricos realizados por especialistas brasileiros que utilizam a Plataforma Lattes, e que isso é reflexo das limitações de seus mecanismos de recuperação e extração de informação. Os autores ainda destacam o fato de o repositório reunir toda a produção científica brasileira, o que viabilizaria estudos que só seriam possíveis se conduzidas em diversas fontes internacionais, mas a um custo considerável para seus autores.

É importante observar ainda que a análise dos dados contidos nos currículos da Plataforma Lattes pode fornecer informações importantes para a compreensão do conhecimento científico brasileiro e como ele tem evoluído, tendo em vista a quantidade de trabalhos recentes que têm considerado esses currículos como principal fonte de dados [Oliveira et al., 2012, Perez-Cervantes, Mena-Chalco e Cesar-Junior, 2012, Digiampietri, Mugnaini e Alves, 2013, Mena-Chalco et al., 2014, Roos et al., 2014, Furtado et al., 2015, Silva et al., 2016, Brito, Quoniam e Mena-Chalco, 2016, Sidone, Haddad e Mena-Chalco, 2017]. No entanto, as restrições de acesso impostas pelo CNPq, como, por exemplo, a necessidade de se validar *captchas* para acesso a cada um dos currículos, tem limitado bastante o potencial de estudos sobre os dados curriculares da Plataforma Lattes. Além disso, particularidades do repositório, como ambiguidade entre nomes de indivíduos e a falta de vínculos explícitos entre os coautores dos trabalhos cadastrados, dificultam bastante a análise dos dados, já que a identificação das colaborações passa a ser uma tarefa não trivial, contribuindo para o fato de que a maioria dos atuais trabalhos tem analisado apenas grupos específicos de indivíduos ou pequenos períodos de tempo.

Neste contexto, este trabalho apresenta um importante recurso para disseminação do conteúdo da Plataforma Lattes, abrangendo uma parcela significativa de seus dados já pré-processados, contendo informações relevantes sobre os doutores com currículos ali cadastrados. Logo, tendo em vista a abrangência do conjunto de indivíduos considerado e a diversidade dos dados disponibilizados, inúmeros estudos poderão ser viabilizados, possibilitando assim ampliar o conhecimento sobre a nossa comunidade científica.

3. Coleta dos Dados

Para a criação dos estratos de dados que compõem o LattesDoctoralDataset, utilizou-se um arcabouço denominado LattesDataXplorer (Figura 1) desenvolvido especificamente para a coleta, extração e tratamento de dados da Plataforma Lattes [Dias, 2016]. Esse arcabouço adota técnicas usualmente empregadas na coleta e extração de dados de documentos disponíveis na Web para realizar essas tarefas sobre os currículos da Plataforma Lattes.

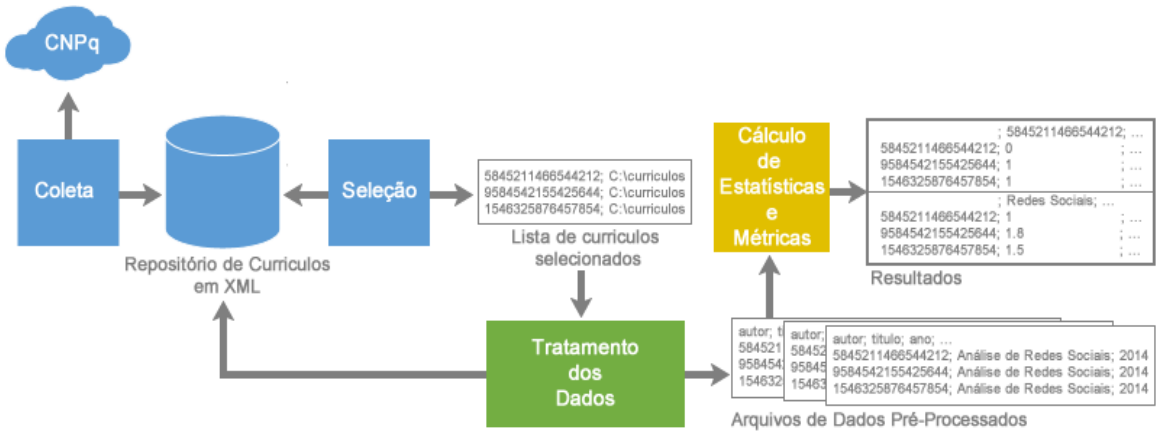


Figura 1: Visão geral do LattesDataXplorer.

A Figura 2 apresenta os componentes do módulo de coleta e extração de dados do LattesDataXplorer. O processo de coleta e extração dos dados da Plataforma Lattes envolve três etapas que são realizadas por meio de três componentes específicos que, para minimizar o custo computacional envolvido, executam respectivamente as seguintes funções: 1) extração de URLs, que visa obter os códigos de identificação de todos os currículos cadastrados na plataforma, possibilitando assim acessar individualmente cada um deles; 2) extração de Ids e Datas de Atualização, que visa extrair de cada currículo o seu identificador individual e a data de sua última atualização; e 3) coleta dos currículos, que visa coletar e armazenar em um repositório local os currículos cuja data de atualização na Plataforma Lattes seja divergente da data de atualização armazenada localmente ou que ainda não tenham sido coletados.

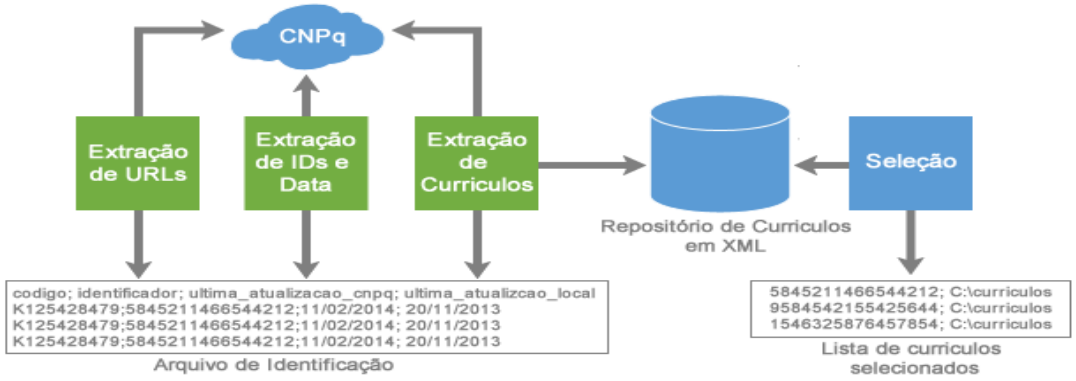


Figura 2: Processo de extração e seleção de dados.

Essas etapas são necessárias para manter o repositório local sempre atualizado sem que se faça necessário coletar novamente todos os currículos a cada nova extração, possibilitando assim a realização de análises com dados sempre atualizados. Além disso, é importante ressaltar que os currículos atualizados podem não só conter novos dados, como também ter dados já registrados alterados ou excluídos, o que torna o processo de atualização de campos específicos uma tarefa complexa. Com a estratégia adotada, todo currículo atualizado é substituído pela sua versão mais recente, o que ameniza consideravelmente o processo de atualização do repositório local de currículos.

O arquivo de identificação é a base para a extração de dados dos currículos. Toda a vez que seja necessário atualizar o repositório local de currículos, a primeira etapa do processo de extração é executada, resultando na extração de todos os códigos cadastrados na Plataforma Lattes. Códigos já registrados no arquivo de identificação são ignorados por corresponderem a currículos já incluídos no repositório local, enquanto que novos códigos são inseridos ao final do arquivo, já que representam novos currículos que ainda não foram coletados.

Posteriormente, com o uso dos códigos de identificação, são acessados os cabeçalhos de cada um dos currículos e extraídos os respectivos códigos identificadores e as datas de atualização junto à Plataforma Lattes, tanto para currículos já extraídos como para os novos currículos, atualizando o arquivo de identificação a cada nova extração. O acesso ao cabeçalho possibilita maior rapidez a todo o processo, agilizando de forma significativa a extração de dados, já que não se faz necessário esperar que todo o currículo seja gerado.

Finalmente, ocorre a extração de dados dos currículos. Inicialmente, o extrator verifica no arquivo de identificação se existem currículos cuja data de atualização na Plataforma Lattes é divergente da data de atualização local. Em caso afirmativo, esses currículos são coletados, substituindo-se as atuais versões armazenadas no repositório local e tendo as suas respectivas datas de atualização alteradas no arquivo local de identificação. Em seguida, são coletados os novos currículos cadastrados na Plataforma Lattes, cujos códigos de identificação foram inseridos ao final do arquivo local de identificação. Diante disso, os dados desses currículos são extraídos pela primeira vez e sua data de atualização local é registrada no arquivo de identificação. Todo esse processo possibilita manter um repositório atualizado com baixo custo computacional, já que apenas um pequeno percentual dos currículos é atualizado com frequência.

Com todos os currículos armazenados localmente em formato XML, é possível manipular os dados com mais flexibilidade, permitindo explorar todo o potencial que os dados curriculares da Plataforma Lattes oferecem. Para isso é utilizado o módulo de seleção de currículos, que permite a criação de subgrupos de currículos por meio de consultas expressas em XPath.

Para a geração dos conjuntos de dados que compõem a coleção disponibilizada neste trabalho, foram selecionados todos os indivíduos com doutorado concluído informado em seus currículos. Nesse processo, foi possível identificar alguns indivíduos que, mesmo não tendo inserido qualquer informação sobre os seus cursos de doutorado, foram incluídos na coleção a partir da informação de seus programas de pós-doutorado.

4. Tratamento dos Dados

Na etapa de Tratamento dos Dados, o componente correspondente é responsável por tratar os currículos dos grupos selecionados e gerar um conjunto de dados para análises posteriores. Esse componente utiliza a lista de currículos selecionados, gerada para um determinado grupo, identificando os currículos a serem analisados. Essa lista inclui o identificador e o local de armazenamento de cada currículo selecionado, o que possibilita que apenas os currículos presentes na lista sejam considerados nessa etapa. Esse componente é responsável por tratar cada um dos currículos selecionados, produzindo arquivos de dados pré-processados que possibilitam a realização de análises

da produção científica, do processo de formação e orientação de alunos, e das redes de colaboração científica dos indivíduos que compõem os grupos selecionados (Figura 3).

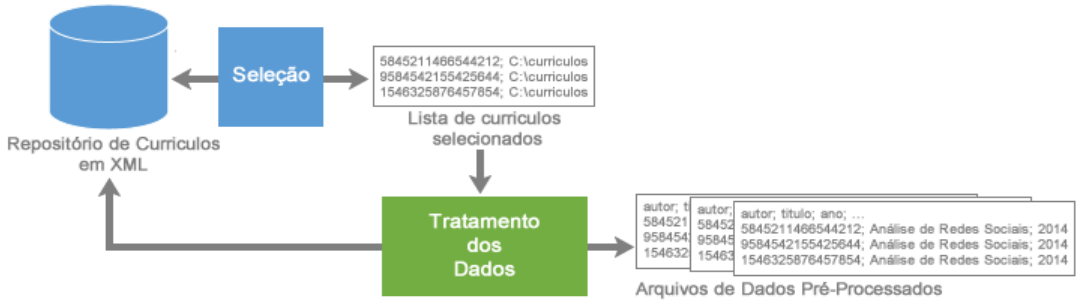


Figura 3: Processo de tratamento dos dados.

O maior desafio no tratamento dos dados coletados da Plataforma Lattes está relacionado com a maneira como cada indivíduo preenche os dados em seus currículos. Como as entradas são realizadas manualmente, não é uma situação incomum dois indivíduos cadastrarem o mesmo trabalho com informações divergentes, como o título do trabalho diferente ou, até mesmo, a lista de coautores incompleta [Boaventura et al., 2014].

É importante destacar que este trabalho não possui ênfase na desambiguação de autores, sendo esta tarefa foco de diversos outros trabalhos que tentam tratar a melhor forma de desambiguar nomes de autores [Ferreira, Gonçalves e Laender, 2012]. Neste trabalho, são considerados os identificadores de cada autor e não o nome dos autores para o processo de identificação de colaborações. Ou seja, cada indivíduo é referenciado pelo seu identificador único na Plataforma Lattes. O método de identificação utilizado neste trabalho para a caracterização das colaborações científicas dos doutores é descrito por Dias e Moita [2015].

Além da identificação das colaborações, como resultado da etapa de tratamento dos dados são produzidos arquivos de dados pré-processados que contêm todas as informações necessárias para o cálculo de diversas métricas bibliométricas e baseadas em análise de redes sociais. O cálculo das estatísticas e métricas a partir desses arquivos é tremendamente facilitado, já que eles sumarizam os dados contidos nos currículos.

5. Descrição da Coleção de Dados

Os dados utilizados para criar a coleção de dados apresentada neste trabalho foram coletados em junho de 2017, quando a Plataforma Lattes incluía o total de 5.251.540 currículos. Esses dados foram coletados exatamente conforme o processo descrito na seção anterior. Para a coleta, foram selecionados todos os indivíduos com doutorado concluído ou que tivessem realizado algum estágio de pós-doutorado, resultando em um conjunto contendo o total de 265.187 currículos. De modo geral, esse conjunto de currículos possui data de atualização recente e inclui, em quase sua totalidade, algum tipo de publicação registrada. Esse grupo de indivíduos que, em sua maioria, atua em pesquisa, seja em instituições de ensino superior ou em institutos de ciência e tecnologia, é também responsável pela formação de alunos de mestrado e doutorado nos principais programas de pós-graduação *stricto sensu* do país, sendo vários deles reconhecidos por sua elevada produção científica. Com isso, ressalta-se que o conjunto de indivíduos considerado para a geração da coleção de dados descrita neste trabalho

compreende grande parte dos docentes dos programas de pós-graduação reconhecidos pela CAPES e dos bolsistas de produtividade em pesquisa do CNPq.

Conforme ressaltado por Dias [2016], apesar de o conjunto de indivíduos com doutorado concluído representar apenas 5,38% de todos os currículos cadastrados na Plataforma Lattes, esses indivíduos são detentores de 64,67% dos artigos em anais de congressos e 74,51% dos artigos em periódicos registrados em todo o conjunto de currículos contidos na Plataforma Lattes, corroborando, assim, a importância da coleção apresentada neste trabalho.

É importante destacar, ainda, a diversidade dos dados registrados no conjunto de currículos considerado, que referem-se a artigos publicados em anais de congresso e em periódicos, apresentação de trabalhos científicos, participação em eventos, nível de formação acadêmica, orientações realizadas, dentre outros. É importante ressaltar, ainda, que um determinado trabalho pode estar registrado em currículos distintos, já que pode ter sido realizado em colaboração envolvendo mais de um indivíduo. Logo, no repositório da Plataforma Lattes, um trabalho pode aparecer várias vezes, tendo em vista que ele pode ter sido registrado por cada um de seus autores. A Tabela 1 apresenta o quantitativo geral de todos os trabalhos registrados nos currículos dos doutores coletados para a geração da coleção LattesDoctoralDataset.

Tabela 1: Quantitativo dos dados dos currículos dos doutores em junho de 2017.

Tipo de Trabalho	Geral
Artigos em Anais de Congresso	9.051.680
Artigos em Periódico	4.660.430
Capítulos de Livro	1.054.844
Demais Trabalhos	445.894
Livros	397.934
Textos em Jornais e Revistas	858.789
Trabalhos Técnicos	1.342.416
Outras Produções Bibliográficas	625.621

A quantidade de dados registrada corrobora a importância da Plataforma Lattes, confirmando a sua condição de um dos principais repositórios de dados científicos atualmente existentes em todo o mundo [Lane, 2010] e caracterizando-se como uma fonte extremamente rica para análise da produção científica brasileira. A partir da Tabela 1, é possível observar a tendência de publicação de artigos em anais de congresso, seguida em menor número pela publicação de artigos em periódicos e de capítulos de livro.

Os estratos de dados estão disponíveis em formato *.csv*, em oito arquivos *.rar*, nos quais as colaborações científicas, correspondentes a pares de colaboradores, estão divididas em dois arquivos. Além disso, um descritor, detalhando os dados contidos em cada um dos estratos, pode ser encontrado no arquivo *descritor.pdf*. Informações adicionais como, por exemplo, datas de atualização dos estratos, podem ser encontradas no arquivo *info.txt*.

A Tabela 2 apresenta a descrição dos estratos de dados disponibilizados. Como pode ser observado, os dados dos doutores estão organizados em conjuntos de dados que agrupam informações extraídas dos currículos e que possibilitam a realização de análises específicas. É importante ressaltar que os dados contidos nesses estratos podem

ser combinados por meio dos identificadores dos respectivos currículos, propiciando um amplo escopo de análises que podem ser realizadas com os dados ora disponibilizados.

Tabela 2: Descrição dos estratos de dados disponibilizados.

Estrato	Descrição
Formação Acadêmica	Dados sobre a formação acadêmica de cada doutor da graduação até o pós-doutorado.
Proficiência	Dados sobre nível de conhecimento (compreensão, fala, leitura e escrita) de cada doutor nos idiomas que domina.
Orientações	Dados sobre as orientações em andamento e concluídas de cada doutor nos diversos níveis de capacitação.
Produção Científica	Dados sobre a produção científica de cada doutor nos diversos tipos de veículo de publicação (periódicos, anais de conferências, livros, etc.).
Atuação Profissional	Dados sumarizados sobre a atuação profissional de cada doutor.
Colaborações	Dados sobre as colaborações científicas de cada doutor com os demais indivíduos do mesmo grupo, tendo como base as publicações produzidas no quinquênio 2012 a 2016.

O primeiro estrato de dados (*Formação Acadêmica*) sumariza a formação acadêmica de cada um dos doutores. Nesse estrato são apresentados os dados de cada doutor referentes aos cursos de graduação, especialização, mestrado e doutorado concluídos, bem como dos programas de pós-doutorado realizados, incluindo o ano de início, ano de conclusão e local de realização. Caso um doutor tenha realizado mais de um curso em um determinado nível de formação, o mais recente é considerado. As análises deste conjunto de dados possibilitam compreender o processo de formação nos diversos níveis de capacitação dos doutores com currículos cadastrados na Plataforma Lattes. Estudos que visam compreender a duração média de cada nível de capacitação ou que levem em consideração as instituições em que os doutores se capacitaram podem propiciar informações inéditas sobre o processo de formação desses indivíduos.

Já o estrato *Proficiência* apresenta para cada doutor o seu nível de proficiência em cada idioma informado. Esse nível de proficiência é indicado para cada habilidade específica (compreensão, fala, leitura e escrita), sendo que para cada uma dessas habilidades podem ser registrados os seguintes níveis de conhecimento: *pouco*, *razoável* e *bom*. Vele ressaltar que, caso não tenha sido feito qualquer registro de conhecimento sobre um determinado idioma, nenhuma informação sobre esse idioma é apresentada. Este conjunto de dados pode ser de grande relevância para estudos que visam compreender o grau de conhecimento de um idioma estrangeiro pelos doutores de uma determinada área e realizar a correlação desse conhecimento com a respectiva produção científica internacional, uma vez que quanto maior o conhecimento de idiomas estrangeiros, maior o potencial de inserção internacional desses indivíduos.

O estrato *Orientações* apresenta para cada um dos doutores o total de orientações em andamento e concluídas nos seguintes níveis de capacitação: *Iniciação Científica*, *Graduação*, *Especialização*, *Mestrado*, *Doutorado, Pós-Doutorado* e *Outra Natureza*. Este estrato de dados considera apenas o quantitativo de orientações em cada um desses níveis, não incluindo os cursos ou instituições onde tais orientações foram realizadas, sendo basicamente o somatório de orientações em cada um dos níveis de capacitação. Logo, com esses dados, é possível realizar estudos que visem analisar o volume de orientações de cada doutor ao longo de sua carreira. Além disso, esses dados

também permitem realizar análises comparativas sobre o processo de orientação nas diversas áreas do conhecimento.

O estrato *Produção Científica* caracteriza-se por ser o conjunto de dados que permite realizar diversas análises sobre a produção científica dos pesquisadores brasileiros. Este conjunto de dados apresenta o quantitativo dos principais tipos de publicação produzidos por esses pesquisadores (artigos em anais de congressos e periódicos, livros, capítulos de livros, textos em jornais e revistas, e trabalhos técnicos), bem como das apresentações de trabalho e demais tipos de produção técnica (ex., material didático, relatórios de projeto, comunicações, etc.). Como o conjunto de doutores é responsável pela maior parte da produção científica registrada nos currículos cadastrados na Plataforma Lattes, este estrato representa de forma consistente a produção geral registrada nos currículos de todos os indivíduos. Além disso, análises que consideram a correlação entre produção científica e tempo de carreira podem ainda utilizar o conjunto de dados de *Formação Acadêmica*, da mesma forma que análises que consideram a correlação entre produção científica e número de orientações podem ser viabilizadas utilizando também dados do estrato *Orientações*, possibilitando assim apresentar informações importantes sobre a comunidade científica brasileira.

O estrato *Informações Profissionais* apresenta um conjunto de dados específicos sobre cada um dos doutores. Além do identificador do doutor, ele inclui também a sua grande área e a área específica de atuação, bem como dados sobre o seu vínculo profissional e a sua localização geográfica, tomando como base o seu endereço profissional. Para indivíduos que tenham mais de uma grande área e área de atuação registradas em seus currículos, apenas a primeira delas foi considerada neste conjunto de dados. Assim, a partir dos dados de cunho profissional dos doutores, diversas análises que consideram as instituições em que eles estão vinculados podem ser realizadas, bem como análises que levam em consideração as regiões geográficas em que estão localizados. Além disso, os dados deste estrato possibilitam o agrupamento dos doutores por áreas e grandes áreas de atuação, potencializando ainda mais as análises a serem realizadas.

Por fim, o último estrato de dados, *Colaborações*, descreve o conjunto de colaborações em trabalhos publicados pelos doutores no quinquênio de 2012 a 2016. Ele associa cada doutor e aos seus colaboradores, indicando, ainda, a quantidade de trabalhos, que podem ser de natureza distinta, realizados em colaboração. A partir desses dados é possível caracterizar e analisar a rede de colaboração científica dos doutores brasileiros, possibilitando a realização de diversos estudos baseados em análise de redes. A combinação dos dados deste e de outros estratos possibilita a elaboração de diversos estudos sobre a produção científica brasileira.

Os estratos de dados descritos acima podem ser obtidos a partir do seguinte endereço eletrônico: *https://github.com/thiagomagela/LattesDoctoralDataset*.

6. Considerações Finais

Considerando o grande interesse de diversos trabalhos recentes que visam analisar dados de publicações científicas, os conjuntos de dados disponibilizados neste trabalho caracterizam-se como importante fonte de informação para diversos novos estudos em diferentes áreas. Por sumarizar dados específicos, como produção científica, formação acadêmica, orientações em andamento e concluídas, informações profissionais e

trabalhos em colaboração, os conjuntos de dados descritos possibilitam diversos novos estudos com grande facilidade, tendo em vista a forma como estão formatados.

Por fim, espera-se que com a disponibilização desses conjuntos de dados, pesquisadores de áreas distintas do conhecimento possam realizar estudos bibliométricos que visem apresentar informações relevantes para toda a comunidade científica nacional, corroborando assim a importância dos dados disponibilizados e a relevância dos dados curriculares da Plataforma Lattes para estudos bibliométricos.

Referências

Brito, A. G. C., Quoniam, L., Mena-Chalco, J. P. (2016) Exploração da Plataforma Lattes por assunto: proposta de metodologia. *TransInformação*, 28(1): 77-86.

Dias, T. M. R. (2016) *Um Estudo da Produção Científica Brasileira a partir de Dados da Plataforma Lattes*. Tese de Doutorado, Programa Pós-Graduação em Modelagem Matemática e Computacional, CEFET-MG.

Dias, T. M. R., Moita, G. F. (2015) A method for the identification of collaboration in large scientific databases. Em Questão, 21(2): 140-161.

Digiampietri, L. A., Mugnaini, R., Alves, C. (2013) Analysis of Participation in Supervised Production of Advisors: A Case Study in Computer Science. In *Proceedings of the Brazilian Workshop on Social Network Analysis and Mining*, Maceió, Brasil.

Ding, Y. (2011) Scientific collaboration and endorsement: Network analysis of coauthorship and citation networks. *Informetrics*, 5(1): 187-203.

Ferraz, R. R. N., Quoniam, L., Maccari, E. A. (2014) The Use of Scriptlattes tool for extraction and on line availability of academic production from a departament of stricto sensu in management. In *Proceedings of the International Conference on Information Systems and Technology Management*, São Paulo, Brasil, pp. 663-679.

Ferreira, A. A., Gonçalves, M. A., Laender, A. H. F. (2012) A Brief Survey of Automatic Methods for Author Name Disambiguation. *SIGMOD Record*, 41(2): 15-26.

Furtado, C. A. et al. (2015) A Spatiotemporal Analysis of Brazilian Science from the Perspective of Researchers' Career Trajectories. *PLOS ONE*, 10(10): e0141528.

Laender, A. H. F. et al. (2008) Assessing the research and education quality of the top Brazilian Computer Science graduate programs. *SIGCSE Bulletin*, 40(2): 135-145.

Lane, J. (2010) Let's make science metrics more scientific. *Nature*, 464(7288): 488-489.

Mena-Chalco, J. P. et al. (2014) Brazilian bibliometric coauthorship networks. *JASIST*, 65(7): 1424-1445.

Mugnaini, R., Leite, P., Leta, J. (2011) Fontes de informação para análise de internacionalização da produção científica brasileira. *PontodeAcesso*, 5(3): 87-102.

Mugnaini, R. et al. (2014) Comunicação científica no Brasil (1998-2012): indexação, crescimento, fluxo e dispersão. *Transinformação*, 26(3): 239-252.

Oliveira, E. A. et al. (2012). Comparison of Brazilian researchers in clinical medicine: are criteria for ranking well-adjusted? *Scientometrics*, 90(2): 429–443.

Perez-Cervantes, E., Mena-Chalco, J. P., Cesar-Junior, R. M. (2012) Towards a Quantitative Academic Internationalization Assessment of Brazilian Research Groups. In *Proceedings of the IEEE 8th International Conference on e-Science*, Chicago, IL, USA.

Revoredo, K. et al. (2012) Mining scientific literature for analysis of collaboration in research communities. In: *Proceedings of the Brazilian Workshop on Social Network Analysis and Mining*, Curitiba, Brasil.

Roos, D. H. et al. (2014) Brazilian scientific production in areas of biological sciences: a comparative study on the modalities of full doctorate in Brazil or abroad. *Scientometrics*, 98(1): 415-427.

Sidone, O. J. G., Haddad, E. A., Mena-Chalco, J. P. (2017) Scholarly publication and collaboration in Brazil: The role of geography. *JASIST*, 68(1): 243-258.

Silva, T. H. P. et al. (2016) The Impact of Academic Mobility on the Quality of Graduate Programs. *D-Lib Magazine*, 22(9/10).

Ströele, V., Zimbrão, G., Souza, J. M. (2012) Análise de redes sociais científicas: modelagem multi-relacional. In: *Proceedings of the Brazilian Workshop on Social Network Analysis and Mining*, Curitiba, Brasil.

MAMMOSET: An Enhanced Dataset of Mammograms

Paulo H. Oliveira[1], Lucas C. Scabora[1], Mirela T. Cazzolato[1], Marcos V. N. Bedo[1,2], Agma J. M. Traina[1], Caetano Traina-Jr.[1]

[1]Institute of Mathematics and Computer Sciences – USP – São Carlos, Brazil

[2]Fluminense Northwest Institute – UFF – S. A. Pádua, Brazil

{pholiveira,lucascsb,mirelac}@usp.br, {bedo,agma,caetano}@icmc.usp.br

Abstract. *In this paper, we present* MAMMOSET*, a compilation of datasets consisting of regions of interest (ROIs) of mammograms.* MAMMOSET *is composed of data collected from three diversified sources, namely* DDSM*,* MINI-MIAS*, and* VIENNA*. Accordingly, the images of* MAMMOSET *were obtained from distinct medical scanners and annotated in different manners. Our contribution refers to the standardization of the collected images, as well as the organization of the metadata. Additionally, we generate a high-dimensional data representation, which is a composition of features extracted by nine image processing extractors to represent color, texture, and shape. Finally, we provide a detailed description of the organized metadata and extracted features, as well as a discussion on the Principal Component Analysis (PCA).* MAMMOSET *can be employed in several supervised and non-supervised tasks, such as classification, clustering, data visualization and Content-Based Image Retrieval (CBIR).*

1. Introduction

Picture Archiving and Communication Systems (PACS) enable the management of huge quantities of medical images, patient data, as well as expert-driven annotations. The annotated medical images stored within PACS are useful for several applications. For instance, they can be used for computer-aided diagnosis, in which they are employed to automatize tasks involving segmentation and classification [Tang et al. 2009]. Furthermore, it is possible to use those annotated images to enable Content-Based Image Retrieval (CBIR), in which the expert can visualize the most similar cases from the past as a second opinion [Kinoshita et al. 2007]. Although millions of medical images are generated every day at hospitals and radiological centers, many annotated medical image datasets have rather few instances [Suckling et al. 1994, Heath et al. 2001]. Such contradiction is due to two reasons. First, medical data are usually protected by national and international laws and the patients have the right not to have their information exposed to the public. Therefore, to be made available, medical images are either anonymized or consist of data from patients who have consented to provide them. Second, as medical images are mainly taken to avoid more intrusive evaluations, e.g. biopsy, the acquisition and annotation protocol for such data is burdensome for experts [Gillies et al. 2016]. In fact, annotation frequently depends on the consensus between experts or on biochemical evaluations. In the case of mammograms, datasets in the literature usually have no more than one thousand instances.

It is worth mentioning data analysis on mammograms is a challenging task due to their nature [Tang et al. 2009]. For instance, they can be troublesome to preprocess, as

distinct medical scanners can generate images of varied resolutions. Second, there are important points in medical breast images, the *regions of interest* (ROIs), which potentially include abnormalities that must be identified and annotated. Although such ROIs can be cropped by either radiologists or automatic segmentation methods, the annotation process is subject to the protocol of each radiologist/hospital so that metadata typically include different levels of details. Therefore, in several situations, such data can be viewed as semistructured. Notice, even if the annotation follows a certain pattern within the same dataset, data from distinct medical datasets are very likely to be quite heterogeneous, which demands an effort to integrate them when data from multiple repositories are taken into account. Under such considerations, we have worked on finding annotated medical breast images. Our contribution refers to the curation of a mammogram dataset, named MAMMOSET, as follows. First, we integrated data from multiple datasets in order to standardize the medical image files and properties. Additionally to image standardization, we worked on the integration of different metadata and labels. Lastly, we applied nine image processing techniques to extract engineered features for the medical images, which we call *feature vectors* of the original instances. After an extensive study of the literature, we selected datasets DDSM, MINI-MIAS, and VIENNA to be integrated into MAMMOSET. The datasets we included have already been in use in several studies, as shown in Table 1.

Table 1. Summarized year-sorted list of studies on the integrated datasets.

Study	Description	Dataset(s)
[Heath and Bowyer 2001]	Segmentation/Classification	DDSM
[Watanabe et al. 2011]	Classification	DDSM
[Silva et al. 2009]	Feature selection	VIENNA and DDSM
[Traina et al. 2011]	Feature selection	VIENNA
[Casti et al. 2013]	Classification	MINI-MIAS and DDSM
[Oliveira et al. 2017]	Data indexing	VIENNA, DDSM and MINI-MIAS

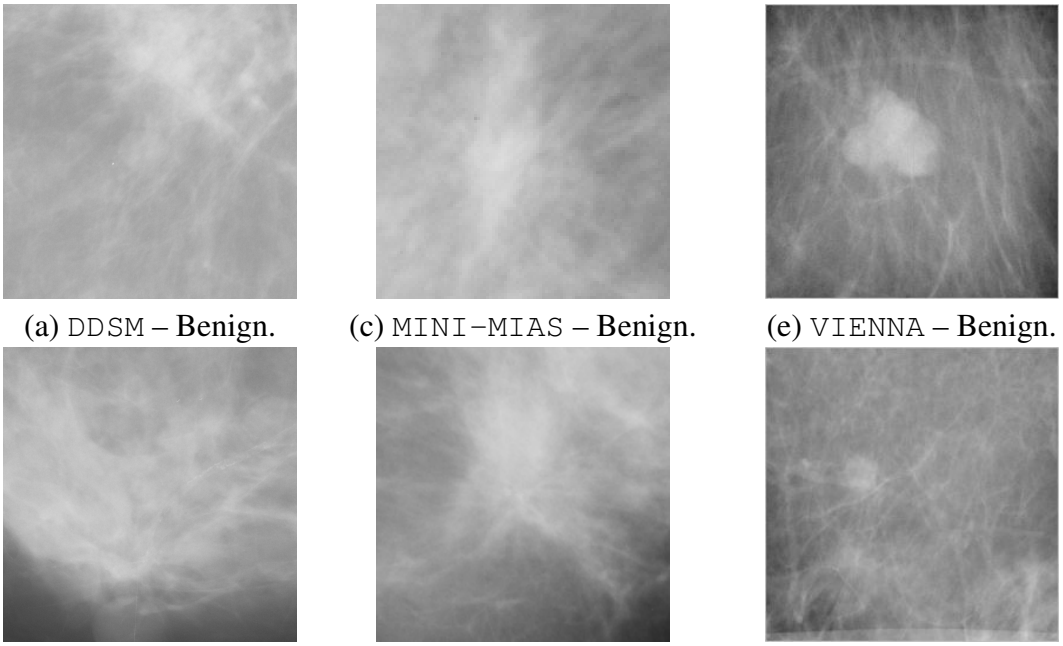
The remainder of this paper is as follows. Section 2 presents a detailed discussion on the curation of MAMMOSET and statistical data description. Section 3 covers potential challenges and limitations on the use of our dataset. Section 4 provides information on how to obtain our dataset, as well as on how to properly acknowledge it in further studies. Finally, Section 5 concludes the work.

2. MAMMOSET: An Enhanced Dataset of Mammograms

This section describes data gathering, standardization, and preprocessing. Furthermore, it details a *schema* for storing MAMMOSET, as well as provides a straightforward statistical analysis regarding the data distribution according the metadata attributes.

2.1. Data Collection and Preprocessing

We integrated data from three datasets, namely DDSM, VIENNA and MINI-MIAS to generate MAMMOSET. The DDSM (Digital Dataset for Screening Mammography) repository [Heath et al. 2001] is composed of more than three thousand medical breast images with 12 bits per pixel, organized into four categories based on the view of the breast image,



(a) DDSM – Benign. (c) MINI−MIAS – Benign. (e) VIENNA – Benign.

(b) DDSM – Malignant. (d) MINI−MIAS – Malignant. (f) VIENNA – Malignant.

Figure 1. Excerpt from the integrated datasets regarding two classes.

which are *(i)* LCC: Left CranioCaudal, *(ii)* RCC: Right CranioCaudal, *(iii)* LMLO: Left MedioLateral Oblique, and *(iv)* RMLO: Right MedioLateral Oblique. DDSM was originally created in a collaborative effort involving the Massachusetts General Hospital, the University of South Africa and the Sandia National Laboratories. Moreover, additional cases were provided by the Washington University School of Medicine. The current DDSM repository is maintained at University of South Florida[1]. DDSM is divided into cases with annotated labels, which are "normal", "cancer", or "benign". The maximum of four images is assigned to one case, two for the left breast and two for the right breast. We took $2,892$ cropped and annotated ROIs with 256×256 pixels from the DDSM images.

We named VIENNA the dataset created by the Department of Radiology at University of Vienna[2]. VIENNA dataset is composed of mammograms collected from the Breast Imaging Reporting and Data System (BI-RADS) Tutorium, which was held at the same university. The repository includes 447 images of ROIs, with 1024×1024 pixels of resolution, representing tumoral tissues and comprising mixed mammogram views. To merge this dataset into MAMMOSET, we resized the images to 256×256 in order to standardize the resolutions of the images, keeping them as thumbnails, since the focus of MAMMOSET is on the extracted features. VIENNA metadata includes the lesion type (calcification or mass), their respective subtype (e.g. amorphous calcification or circumscribed mass), the presence of architectural distortion and asymmetric density, and the lesion severity.

Finally, MINI−MIAS (Mammographic Image Analysis Society) repository consists of 118 valid ROIs of mixed mammogram views [Suckling et al. 1994], with various resolutions. MINI−MIAS is a reduced version of the MIAS dataset, in which the original images were cropped to present a resolution of 1024×1024 pixels. The dataset is avail-

[1]The DDSM repository: http://marathon.csee.usf.edu/Mammography/Database.html

[2]The VIENNA repository: http://www.medaustria.at/medaustria/index.html

able online[3]. In addition to the images, MINI-MIAS provides metadata corresponding to the background tissue, class and severity of the abnormality, as well as coordinates to the center of the abnormality and the approximate radius of a circle enclosing it. Such coordinates and radius allowed us to extract the ROIs from the images, which were embedded in MAMMOSET instead of the whole images themselves.

Figure 1 presents two images from each integrated dataset. We randomly selected those examples according to a common annotated metadata among all three datasets, the labels "Benign" and "Malignant". Figures 1(a–b) show two ROIs from two distinct DDSM cases with benign and malignant findings respectively. Analogously, Figures 1(c–d) show examples from VIENNA and Figures 1(e–f) from MINI-MIAS. Furthermore, all three repositories were acquired in different time spans. Originally, we collected DDSM ROIs in February of 2017, VIENNA in August of 2009 and MINI-MIAS in February of 2017. Last but not least, we converted the MINI-MIAS images from the .pgm file format to .png, since the .pgm images were not supported by the implementation of the feature extractor methods we used[4].

2.2. Feature Extraction

We selected nine FEMs to generate a multidimensional representation of the dataset images as *feature vectors*. Such extractors were selected to represent three characteristics, namely color, texture, and shape. Table 2 presents the employed extractors, their category and a reference for their description.

Table 2. Feature extractor methods employed in MAMMOSET.

Extractor	Category	Reference
Normalized Histogram	Color	[Nixon and Aguado 2012]
BIC Histogram	Color	[Stehling et al. 2002]
Edge Histogram	Texture	[Won et al. 2002]
Haralick	Texture	[Haralick et al. 1973]
Rotation Invariant LBP	Texture	[Guo et al. 2010]
Texture Spectrum	Texture	[He and Wang 1990]
Zernike	Shape	[Khotanzad and Hong 1990]
Haar Extractor	Shape	[Wang et al. 1998]
Daubechies	Shape	[Wang et al. 1998]

The extractors Normalized Histogram and BIC Histogram enable, respectively, the gathering of global (256 features) and local (512 features) distributions of the grayscale levels within images. The texture extractors represent the correlation and co-occurrence of pixels. Particularly, Edge Histogram provides a summary of the number of edge types within each image (150 features), whereas the Haralick extractor calculates dimensions as variances and moments based on co-occurrence matrices (24 features). Likewise, Rotation Invariant LPB (108 features) and Texture Spectrum (8 features) enable the gathering of local correlation among grayscale values within each image. Finally, shape extractors

[3]The MINI-MIAS repository: http://peipa.essex.ac.uk/info/mias.html

[4]Arboretum: https://bitbucket.org/gbdi/arboretum/downloads/

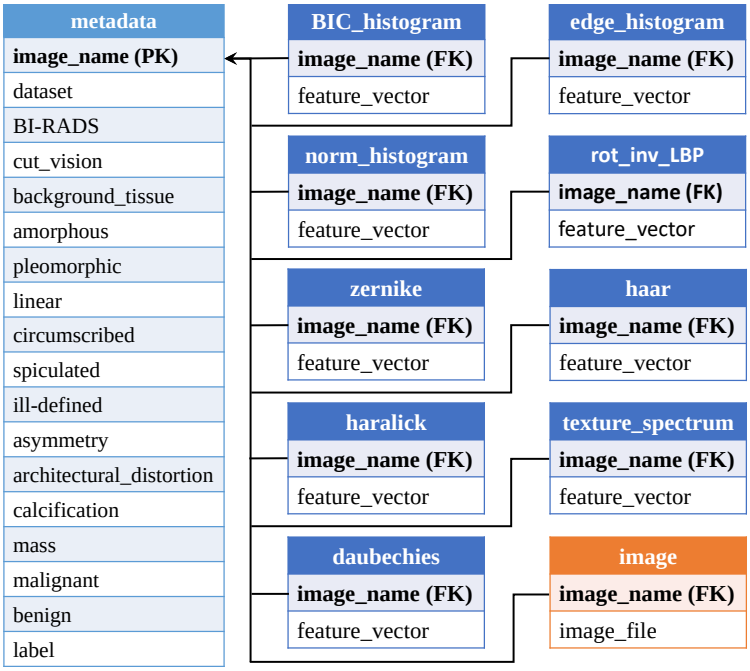


Figure 2. Schema for MAMMOSET.

enable the gathering of edges and moments within the ROIs of mammograms. For instance, the Zernike extractor represents a certain type of grayscale moments (36 features), whereas the Haar (16 features) and Daubechies (16 features) wavelets transforms summarize the two-dimensional set of gray pixels into a vector representation. The resulting feature vector has 1126 dimensions, which are already normalized to the $[0, 1]$ real interval according to the criteria of each extractor. Notice, although the generated features are low-level, they provide a very generic description of grayscale ROIs. The addition of any new ROI into MAMMOSET dataset also requires this feature extraction process and the concatenation of the features into a single *feature vector*.

2.3. Data Description

We propose a *schema* with normalized tables to store all MAMMOSET data into a relational DBMS. Figure 2 presents our solution. Basically, it is composed of eleven tables that include the images, the metadata and the low-level extracted features. Accordingly, the table "metadata" consists of textual and numerical information about each image. The table "image" is employed to store the images themselves. Finally, the remaining tables store the low-level feature vectors, one table per feature extractor. The tables have $3,457$ tuples, each tuple corresponding to one image from MAMMOSET. Tables for feature vectors share the same structure, which consists of the image name followed by the feature vector itself. The image name is both the primary key of its respective table and a foreign key (i.e. "FK" in the figure) to the metadata table, whose structure consists of the image name acting as a primary key (i.e. "PK" in the figure) followed by seventeen attributes.

The metadata are detailed in Table 3. The last attribute, called *label*, is a synthetic attribute, created to easily allow grouping or filtering elements according to the characteristics they share. It consists of an integer formed as the sum over the binary attributes *calcification*, *mass*, *malignant* and *benign*. This sum weighs each binary attribute as fol-

Table 3. Description of the metadata.

Attribute	Description	Domain	Observation
image_name	Identifier	String	
dataset	Dataset of origin	`DDSM`, `VIENNA`, `MINI-MIAS`	
BI-RADS	Severity of lesion	Integers from 1 to 5	Only `DDSM` and `VIENNA`
cut_vision	Mammographic view	LCC, RCC, LMLO, RMLO	Only `DDSM`
background_tissue	Type of tissue	FattyGlandular, Fatty, DenseGlandular	Only `MINI-MIAS`
amorphous pleomorphic linear	Subcategories of calcification	True, False	Only `VIENNA` and `MINI-MIAS`
circumscribed spiculated ill-defined	Subcategories of mass	True, False	Only `VIENNA` and `MINI-MIAS`
asymmetry architectural_distortion	Other subcategories	True, False	Only `VIENNA` and `MINI-MIAS`
calcification mass	Has calcification/mass	True, False	
malignant benign	Has malignant/benign lesions	True, False, ? ("?" means "Missing")	"?" only appears in pairs, i.e. for both malignant/benign
label	Categorization of *calcification*, *mass*, *malignant* and *benign*	Integers from 4 to 19	

lows: *(i) calcification* by 2^0, *(ii) mass* by 2^1, *(iii) malignant* by 2^2, and *(iv) benign* by 2^3. For instance, if the image is annotated with *calcification* and *malignant* lesion, the calculated label is $2^0 + 2^2 = 5$. There are no missing values for attributes *calcification* and *mass*. However, they do exist for attributes *malignant* and *benign*. Therefore, we add the value 2^4 to the sum of *label* whenever the value of either of them is missing. For instance, if the image is annotated with "?" for attributes *malignant* and *benign*, but *mass* is annotated with "true", the calculated label is $2^4 + 2^1 = 18$. The maximum value for *label* is 19, which happens only when all binary attributes are true. Likewise, the minimum value is 4, as all images are annotated either with *mass* or with *calcification*.

Figure 3 shows the proportion of elements from each original dataset regarding

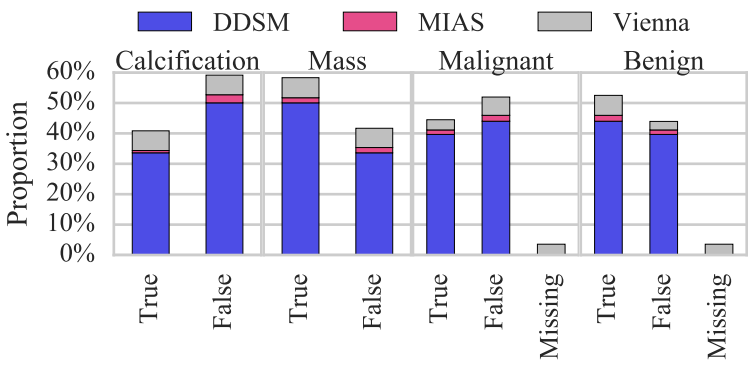


Figure 3. Proportion of datasets according to images with *calcification*, *mass*, *malignant* lesions and *benign* lesions.

binary attributes (including missing values) *calcification*, *mass*, *malignant* and *benign*. Notice nearly 41% of all images contain *calcification* and nearly 58% include *mass*. Moreover, almost 45% of all images contain *malignant* lesions. The remaining 55% of entries for *malignant* attribute were divided into 51% of ROIs without annotations of malignancy and 4% of ROIs with missing values ("Missing" in Figure 3). Regarding *benign* attribute, 53% of entries are annotated with "True". The remaining 47% were divided into 43% annotated with "False" and 4% of missing values. We highlight the 4% of missing values correspond solely to the `VIENNA` dataset.

Finally, Figure 4 presents the ratio of instances by their label and by dataset. The highest ratios correspond to four labels, namely (5) – images annotated with *calcification* and *malignant* lesions, (6) – images annotated with *mass* and *malignant* lesions, (9) – images annotated with *calcification* and *benign* lesions, and (10) – images annotated with *mass* and *benign* lesions. Values of label from 11 to 19 are instances with missing binary attribute values. Each of them accounts for no more than 2% of all instances and contributes to `MAMMOSET` being heavily unbalanced with regard to attribute *label*.

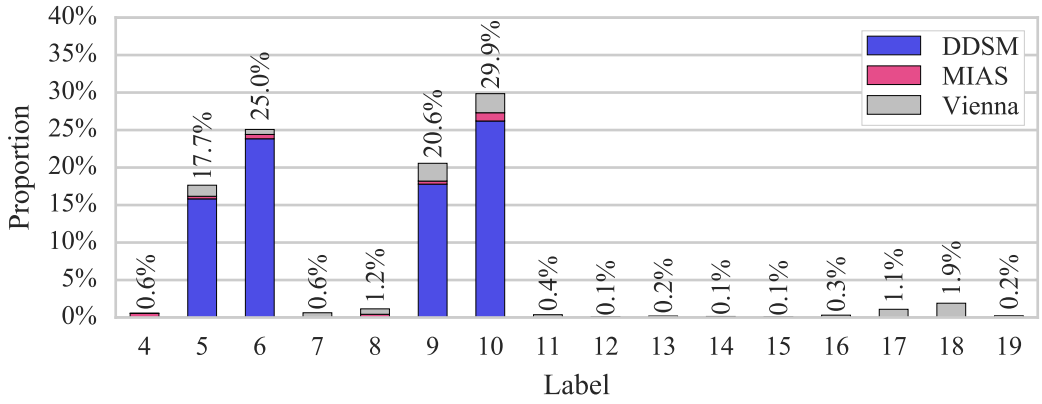


Figure 4. Proportion of datasets with respect to each label.

2.4. Principal Component Analysis

After the curation of the high-dimensional dataset, we verified some of its properties. For instance, we performed a dimensionality reduction by relying on the Principal Component Analysis (PCA) to determine whether or not the dataset is intrinsically high-dimensional. First, we eliminated the attributes whose variance is not null, which generated a dataset with 1111 rather than the original 1126 dimensions. Next, we calculated the eigenvalues by subtracting the mean of each attribute and by rescaling the data. Each eigenvalue was used to determine the percentages of variances represented by each principal component (the "influence" of each dimension on the analysis). Figure 5 presents the scree-plot [Peres-Neto et al. 2005] of the percentage of the variance regarding each dimension.

The percentages of variances show that the intrinsic dimensionality of `MAMMOSET` seems to be from medium to high, as the first two dimensions obtained by PCA represent only 22.9% of the overall variance indicated in Figure 5. The first three components represent only 28.6% of the overall variance, so scatter plots in two or three dimensions may not be enough to draw patterns from the reduced data. It is worth mentioning that it takes, at least, ten dimensions to capture more than 50% of the scree-plot area (the overall

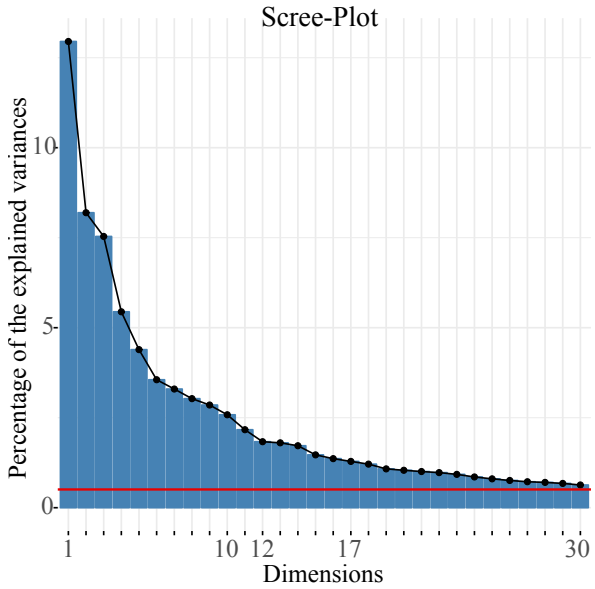


Figure 5. Variance represented by each dimension with regard to PCA.

variance for ten dimensions is 53.83%) and, at least, fourteen dimensions to surpass 60% of the overall variance. In fact, a common heuristic based on scree-plots is to consider at least the number of dimensions closer to the "elbow" of the curve, as in the 12 to 17 dimensions of Figure 5.

A second criterion for selecting the number of primary components is also displayed in Figure 5, represented by the red line in the graph. It represents the widely employed Kaiser-Guttman criterion, which selects all components whose respective eigenvalue is greater than 1.0 (single variable information). In this case, the criterion indicates that at least 99 dimensions should be kept to properly characterize the original high-dimensional dataset. Notice, although many other criteria can be used to select the number of dimensions [Peres-Neto et al. 2005], the results obtained from both scree-plot and the Kaiser-Guttman criterion indicate that the intrinsic dimensionality is indeed medium/high.

We report a data analysis on both data distribution and density of labels "Malignant" and "Benign" in `MAMMOSET` images. Such labels can assume three values, namely "True (T)" for those ROIs whose findings are the same of the label, "False (F)" for those ROIs that are not the same of the label and "?" for the images whose findings were labeled as neither "True (T)" nor "False (F)". Accordingly, we generated two scatter plots to spatially represent the labeled data distribution (points) and density (ellipses), as presented in Figure 6. The first step we took was to perform a random sampling of `MAMMOSET` to retrieve 346 instances (nearly 10% of the original data). Next, we performed the PCA reduction with regard to the two principal components, so that each instance can be seen as a two-dimensional point and depicted according to its label. The data density is showed as ellipsoidal regions that include, at least, 95% of the instances for each label.

Figure 6(a) shows the data distribution regarding label "Malignant". Even though the ellipsoidal groups "T" and "F" are not quite separated (as already expected according to the results in Figure 5), there can be a better separation if the "?" elements become "F". Likewise, Figure 6(b) presents the data distribution with regard to label "Benign".

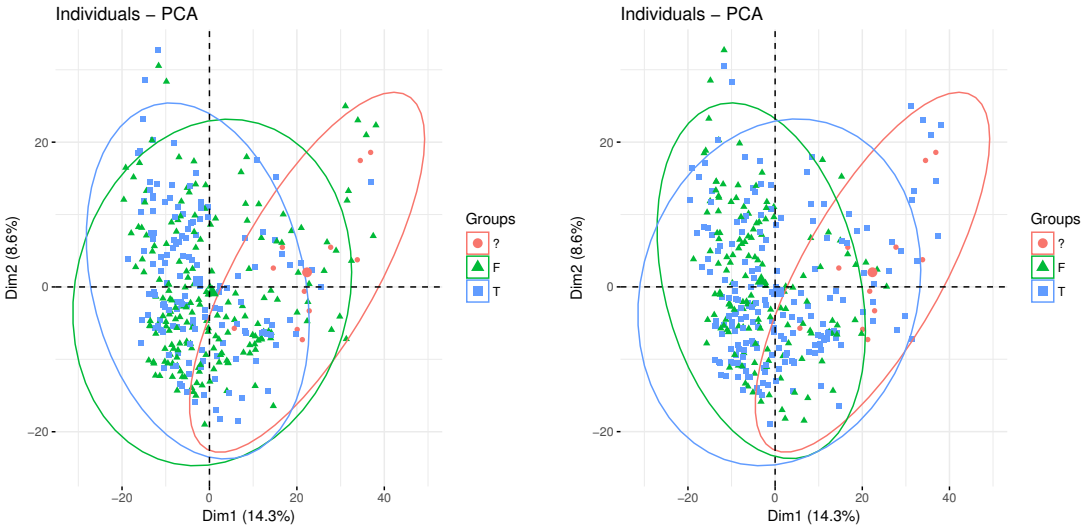


Figure 6. Data distribution and density of MAMMOSET with regard to a random sampling of 10% of the original elements. The elements are represented by their two first principal components and associated with labels (a) *Malignant* and (b) *Benign*.

Accordingly, the groups for "Benign" findings have also a separation trend if "?" elements become "T". Notice the data distribution in Figures 6(a–b) are similar, but labels "Benign" and "Malignant" are not mutually exclusive as there are ROIs with both findings.

3. Applicability and Challenges for MAMMOSET

MAMMOSET can be used in several applications of mammogram analysis. We highlight some scenarios of applicability and challenges regarding the curated data.

Applicability. MAMMOSET has a great potential to be employed in applications related to radiology and medicine. Most interestingly, MAMMOSET can be employed in numerous data analysis problems, such as Content-Based Image Retrieval and classification. The use of MAMMOSET in CBIR tasks enables the retrieval of similar MAMMOSET entries, so that a second opinion is automatically provided to the expert. Classification tasks can also benefit from MAMMOSET for labeling undiagnosed ROIs. Additionally, *hubs* (frequently returned elements in k-nearest neighbor queries in high-dimensional spaces) on MAMMOSET are yet to be identified. Subspace clustering of MAMMOSET is also intriguing, as the dataset is intrinsically medium/high-dimensional. Another applicability of MAMMOSET is dimensionality reduction, in which data analysts can evaluate their reduction approaches to improve data visualization. Finally, MAMMOSET can be used to benchmark the performance of multidimensional and metric indexes.

Challenges. Two challenges stand out in the processing of MAMMOSET, namely the existence of missing and unbalanced data. The existence of missing data in the metadata table (i.e. not all labels could be found for all three integrated repositories) is due to the different mammogram annotation protocols with specific information to be filled, so that no single standards were employed in the annotations. For this reason, one must handle missing data whenever certain attributes are the focus of the analysis. Unbalanced data occurrence (as highlighted in Figure 4) is due to the fact that some labels concentrate the

majority of instances while others cover just a small portion of the dataset. In this scenario, strategies as sampling or data augmentation can be of good use. These challenges should not go unnoticed on MAMMOSET data analyses.

4. Download and Citation Request

We made MAMMOSET available for download by creating a Bitbucket repository at https://bitbucket.org/gbdi/mammoset/src. It is available for researchers and data scientists under the Creative Commons BY license[5] and is structured as follows. File mammoset.tar.gz contains the comma-separated files (.csv) related to the tables of Figure 2. Each metadata file consists of $3,457$ tuples, as well as 1 additional header line. The feature vectors of each extractor are stored in separate .csv files, according to the table names in Figure 2. File images_raw.tar.gz includes the original breast images of MAMMOSET. File mammoclear.tar.gz includes the concatenation of all feature vectors, in which attributes of null variance were removed. In case of publication and/or public use of MAMMOSET, please acknowledge us by citing this paper.

5. Conclusion

In this work, we curated the MAMMOSET dataset, which includes a compilation of mammogram ROIs. MAMMOSET is composed of data collected from three different datasets with annotations from different medical scanners. Our contribution is related to the standardization of the collected images, as well as the organization of the metadata. Additionally, we generated a high-dimensional data representation, which is a composition of the features from nine extractors to represent color, texture, and shape. MAMMOSET is organized as a relational schema and made available in a public repository under the Creative Commons license. We also provided a statistical analysis about the extracted values, as well as an initial discussion on the Principal Component Analysis of MAMMOSET. Finally, we highlighted the applicability and challenges in which the study of MAMMOSET could be further undertaken. In the future, the organized dataset along with the detailed information in this paper can be used for many data mining applications, such as CBIR, classification, clustering and dimensionality reduction.

Acknowledgments. We thank CAPES, CAPES-PDSE (process 88881.134068/2016-01), CNPq and FAPESP for the financial support. We also thank Willian Dener de Oliveira for his contribution in data preprocessing.

References

Casti, P., Mencattini, A., Salmeri, M., Ancona, A., Mangieri, F. F., Pepe, M. L., and Rangayyan, R. M. (2013). Automatic detection of the nipple in screen-film and full-field digital mammograms using a novel Hessian-based method. *JDI*, 26(5):948–957.

Gillies, R. J., Kinahan, P. E., and Hricak, H. (2016). Radiomics: Images are more than pictures, they are data. *Radiology*, 278(2):563–577.

Guo, Z., Zhang, L., and Zhang, D. (2010). Rotation invariant texture classification using LBP variance (LBPV) with global matching. *Patt. Recog.*, 43(3):706–719.

[5]Creative Commons BY 4.0 license: https://creativecommons.org/licenses/by/4.0/

Haralick, R. M., Shanmugam, K., and Dinstein, I. (1973). Textural features for image classification. *TSMC*, (6):610–621.

He, D.-C. and Wang, L. (1990). Texture unit, texture spectrum, and texture analysis. *TGRS*, 28(4):509–512.

Heath, M. and Bowyer, K. (2001). Mass detection by relative image intensity. In *IWDM*, pages 219–225. Medical Physics.

Heath, M., Bowyer, K., Kopans, D., Moore, R., and Kegelmeyer, W. P. (2001). The digital database for screening mammography. In *IWDM*, pages 212–218. Medical Physics.

Khotanzad, A. and Hong, Y. H. (1990). Invariant image recognition by Zernike moments. *TPAMI*, 12(5):489–497.

Kinoshita, S. K., Azevedo-Marques, P. M., Pereira-Jr., R. R., Rodrigues, J. A. H., and Rangayyan, R. M. (2007). Content-based retrieval of mammograms using visual features related to breast density patterns. *JDI*, 20(2):172–190.

Nixon, M. S. and Aguado, A. S. (2012). *Feature Extraction & Image Processing for Computer Vision*. Academic Press.

Oliveira, P. H., Scabora, L. C., Cazzolato, M. T., Oliveira, W. D., Traina, A. J. M., and Traina-Jr., C. (2017). Efficiently indexing multiple repositories of medical image databases. In *CBMS*, page to appear. IEEE.

Peres-Neto, P. R., Jackson, D. A., and Somers, K. M. (2005). How many principal components? Stopping rules for determining the number of non-trivial axes revisited. *Comp. Stat. & Data Analysis*, 49(4):974–997.

Silva, S. F., Traina, A. J. M., Ribeiro, M. X., Batista-Neto, J. E. S., and Traina-Jr., C. (2009). Ranking evaluation functions to improve genetic feature selection in content-based image retrieval of mammograms. In *CBMS*, pages 1–8. IEEE.

Stehling, R. O., Nascimento, M. A., and Falcão, A. X. (2002). A compact and efficient CBIR based on border/interior pixel classification. In *CIKM*, pages 102–109. ACM.

Suckling, J., Parker, P., Dance, D. R., Astley, S., Hutt, I., Boggis, C., and Ricketts, I. (1994). The mammographic image analysis society digital mammogram database. *Excerpta Medica*, 1069:375–378.

Tang, J., Rangayyan, R. M., Xu, J., El Naqa, I., and Yang, Y. (2009). Computer-aided detection and diagnosis of breast cancer with mammography. *TITB*, 13(2):236–251.

Traina, A. J. M., Traina-Jr., C., Balan, A. G. R., Ribeiro, M. X., Bugatti, P. H., Watanabe, C. Y. V., and Azevedo-Marques, P. M. (2011). Feature extraction and selection for decision making. In *Biomedical Image Processing*, pages 197–223. Springer.

Wang, J. Z., Wiederhold, G., Firschein, O., and Wei, S. X. (1998). Content-based image indexing and searching using Daubechies' wavelets. *JDL*, 1(4):311–328.

Watanabe, C. Y. V., Ribeiro, M. X., Traina-Jr., C., and Traina, A. J. M. (2011). SACMiner: A new classification method based on statistical association rules to mine medical images. In *ICEIS*, pages 249–263. Springer.

Won, C. S., Park, D. K., and Park, S.-J. (2002). Efficient use of MPEG-7 edge histogram descriptor. *ETRI*, 24(1):23–30.

Publicando e Consumindo um Conjunto de Dados Abertos Conectados da UAI

André Alencar[1], Douglas Xavier[1], Luiz Carlos Chaves[1], Damires Souza[1]

[1]Unidade Acadêmica de Informática – Instituto Federal de Educação, Ciência e Tecnologia da Paraíba (IFPB) – João Pessoa.

andrealencar@outlook.com.br,{douglasf.xavier,lucachaves}@gmail.com, damires@ifpb.edu.br

Resumo. *Com o intuito de prover visibilidade para a comunidade acadêmica, assim como se aproximar mais dos cidadãos e empresas, instituições educacionais estão desenvolvendo Portais de Dados Abertos. A ideia é publicar dados em formato aberto de forma que estes possam ser consumidos tanto por humanos quanto por softwares. Considerando inicialmente o escopo de dados da UAI do IFPB/Campus João Pessoa, um conjunto de dados abertos e conectados foi desenvolvido e publicado para consumo. Ele inclui informações sobre professores, projetos, cursos e áreas de atuação da UAI. Este trabalho apresenta o conjunto de dados, sua distribuição, a ontologia utilizada e um exemplo de seu uso, incluindo aspectos e desafios relacionados.*

Abstract. *In order to become more transparent to students and associated employees, as well as to work closer with citizens and companies, educational institutions are developing Open Data Portals. The idea is to publish information as open data in such a way that these data may be consumed both by humans and by softwares. Considering, at first, the data scope of the UAI at IFPB/Campus João Pessoa, a set of open and connected data has been developed and published for consumption. It includes data regarding professors, projects, courses and knowledge areas. This work presents the generated dataset, its distribution, the used ontology and an example of its usage, including some related aspects and challenges.*

1. Introdução

A importância dos dados para a sociedade e todos os seus consumidores, principalmente para os que fazem uso de técnicas e ferramentas de estatística, análise e visualização de dados, vem crescendo cada vez mais. Em ambientes empresariais, por exemplo, é comum a utilização do processamento e análise dos dados (buscados conjuntamente) para as tomadas de decisões estratégicas. No panorama governamental, essa tendência de publicação, uso e análise dos dados evolui também progressivamente.

Em tempos onde a sociedade clama pela transparência de seus governos, e leis são criadas de modo a garantir o acesso dos cidadãos à informação [Isotani e Bittencourt, 2015], torna-se ainda mais relevante, e, em muitos casos, obrigatória, a publicação dos dados governamentais de forma aberta. Segundo a *Open Knowledge Foundation*[1], os dados estão "abertos" quando qualquer pessoa pode livremente usá-los, reutilizá-los e redistribuí-los, estando estes sujeitos, no máximo, à exigência de creditar a sua autoria e a compartilhá-los pela mesma licença. Dados Abertos devem ser

[1] http://br.okfn.org/2016/04/13/uma-revolucao-de-dados-para-quem/

disseminados publicamente em formatos abertos (e.g., CSV[2], RDF[3]) na Web, de acordo com alguns critérios e aspectos que possibilitem sua reutilização, como, por exemplo, a disponibilização de metadados [Lóscio et al., 2017]. Assim, é possível o desenvolvimento de aplicativos que consumam esses dados. Os aplicativos ajudam a sociedade a entendê-los mais facilmente.

Na realidade de uma instituição de ensino também há vários dados e recursos como, por exemplo, currículos de pesquisadores, ementas de disciplinas, projetos desenvolvidos que podem ser abertos. Como resultado, esses dados se tornam úteis e reutilizáveis pela própria instituição ou por outras, além de favorecer a criação de aplicativos que ajudam a sociedade a conhecer melhor a instituição, seus perfis, expertises e dados em geral.

Em relação às instituições de ensino públicas, como ilustração, recentemente, a Universidade Federal do Rio Grande do Norte (UFRN) abriu seus dados por meio de um Portal de Dados Abertos[4]. No caso do Instituto Federal da Paraíba (IFPB), ainda não existe um portal com esse objetivo. Nesse panorama, este trabalho aborda o problema de publicação e consumo de um conjunto de dados abertos conectados na Web considerando como escopo inicial os dados da Unidade Acadêmica de Informática (UAI) do IFPB/Campus João Pessoa.

Em meio à necessidade de divulgar de forma criteriosa e relevante dados sobre docentes, projetos, cursos, áreas de atuação, dentre outros, busca-se fazê-lo considerando todos os preceitos de Dados Abertos Conectados [Isotani e Bittencourt, 2015; Heath e Bizer, 2011]. Para isso, foi criada uma base de dados integrada, usando o modelo RDF (*Resource Description Framework*), com alguns dados importantes da UAI à publicação em formato aberto. Os dados publicados em RDF são estruturados e permitem seu uso e reuso por aplicações, além de prover meios à ligação semântica com outros dados. Para referenciar semanticamente os dados, foi desenvolvida uma ontologia de domínio – a OUAI (*Ontology for Universities and Academic Information*) - que reusa termos de vocabulários recomendados e acrescenta outros específicos. A partir da publicação do conjunto de dados da UAI, uma aplicação web está sendo construída com fins de visualização e análise dos dados em questão. O conjunto de dados possui um SPARQL endpoint que permite seu consumo e estará também disponível em um Portal de Dados Abertos que se encontra em construção, não somente em RDF, mas também em distribuição CSV. Neste momento, o conjunto de dados em pauta pode ser acessado em: openuai.ifpb.edu.br/dataset/.

Este artigo está organizado como segue: a Seção 2 descreve o processo de construção do conjunto de dados; a Seção 3 apresenta o conjunto de dados; a Seção 4 explica como o conjunto de dados pode ser consumido e apresenta um exemplo de uso; a Seção 5 mostra desafios e limitações, e a Seção 6 tece considerações sobre o trabalho.

2. Processo de Construção

Dados considerados "abertos" dos diversos domínios do conhecimento deveriam poder ser publicados e consumidos em formato aberto na Web [Lóscio et al., 2017]. Entretanto, alguns problemas ainda precisam ser tratados para facilitar essas atividades

[2] https://www.w3.org/TR/tabular-data-primer/
[3] https://www.w3.org/RDF/
[4] http://dados.ufrn.br/

[Heath e Bizer, 2011]: (i) nem todos os dados podem ser encontrados por meio dos mecanismos de busca; (ii) não é possível especificar consultas complexas sobre os dados de maneira fácil e integrada, e (iii) os dados na Web ainda vivem, em sua maioria, isolados uns dos outros. Todas essas questões têm a ver com a necessidade de uso da semântica associada aos dados para prover seu significado, facilitar a interpretação automatizada e enriquecer as aplicações que irão consultar os dados de forma conjunta. Uma solução possível é usar o modelo RDF e os princípios de *linked data* ou dados conectados [Heath e Bizer, 2011], a saber: usar URIs (*Uniform Resource Identifiers*) para nomear objetos; usar URIs HTTP (*HyperText Transfer Protocol*) para que outras pessoas possam pesquisá-los; fornecer informações úteis usando RDF e protocolo SPARQL; e criar links entre os recursos. Por meio do modelo RDF é possível representar dados em diferentes níveis de estruturação, usar termos de vocabulários para representar esses dados e fazer uso de links nomeados.

Para este trabalho, foi especificado um processo de construção do conjunto de dados abertos conectados, como mostra a Figura 1. Na abordagem proposta, os dados são extraídos e selecionados dos currículos lattes dos professores, de dados institucionais do sistema acadêmico interno e de referências como a tabela de áreas de conhecimento fornecida pela CAPES. Para prover a conversão dos dados coletados para RDF, é necessário escolher ou criar vocabulários que possam viabilizar seu referenciamento semântico. Após a triplificação dos dados, estes são tornados acessíveis por meio de um serviço a partir do RDF *Store*.

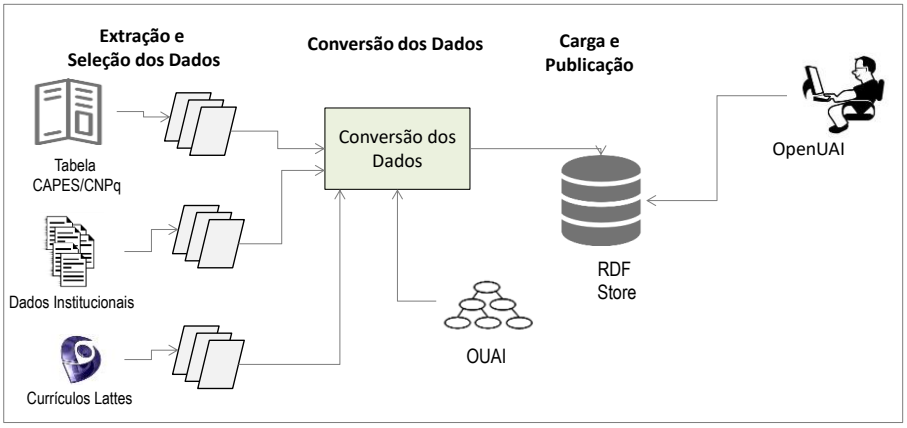


Figura 1 - Processo de Construção do Conjunto de Dados

O processo de construção foi instanciado, nesta fase inicial, de modo a contemplar um escopo de dados associado à UAI. Para esse escopo, foram definidos requisitos associados aos dados e às necessidades de análises da UAI. A ideia é ampliar o escopo inicial e, a partir da instanciação e execução do processo de forma evolutiva e incremental, estender a construção do conjunto de dados de modo a contemplar os dados das demais unidades do Instituto. Dessa forma, quatro requisitos de dados foram especificados, conforme mostra a Tabela 1.

Para viabilizar o processo, foi necessário construir uma ontologia para ser usada como referência semântica de termos e uma ferramenta de conversão dos dados. Ambas serão explicadas a seguir. Os dados convertidos em RDF foram persistidos por meio de um RDF *store*, neste caso, o Virtuoso[5]. Atualmente, existe uma aplicação que já

[5] https://virtuoso.openlinksw.com/rdf/

consome os dados do conjunto gerado, denominada de OpenUAI. A aplicação OpenUAI será mostrada na Seção 4

Tabela 1: Lista de Requisitos de Dados

Requisito de Dados	Descrição	Propriedades	Observações	Formato de Coleta
Docentes	Relação de docentes da UAI	Nome, áreas de atuação, link para currículo lattes, projetos, disciplinas.	Os dados deveriam ser extraídos, em sua maioria, da Plataforma Lattes.	XML dos currículos lattes dos professores.
Projetos	Relação de projetos de pesquisa, extensão e inovação desenvolvidos na UAI.	Título, período (início e término), descrição, categoria, responsável.	Os projetos deveriam ser extraídos da Plataforma Lattes para cada professor.	XML dos currículos lattes dos professores.
Áreas de Atuação	Relação das áreas do conhecimento estabelecidas pela CAPES/CNPq contempladas nos projetos e cursos da UAI.	Nome, professor que atua, curso que contempla, associação com projetos.	As áreas deveriam ser extraídas a partir dos currículos Lattes, de acordo com as tabelas da CAPES e CNPq.	XML dos currículos lattes dos professores e tabela com áreas em PDF.
Cursos e Matrizes Curriculares	Relação de cursos ofertados pela UAI e as disciplinas associadas.	Nome do curso, nome da disciplina, ementa e bibliografia da disciplina, professor que ministra a disciplina, período da disciplina.	As informações de cursos e suas matrizes curriculares e professores deveriam ser obtidas das Coordenações, via sistema de controle acadêmico.	Planilhas extraídas do Sistema em CSV.

2.1 A Ontologia OUAI

A publicação de dados abertos conectados em conjunto com ontologias exige menos esforço tanto na definição do modelo de dados quanto na integração e reuso de outras informações. Além disso, facilita também o consumo destes dados por aplicações.

Para prover o referencial semântico aos dados a serem convertidos, foi desenvolvida a ontologia OUAI (*Ontology for Universities and Academic Information*). A ontologia considera campos dos currículos Lattes[6], de redes acadêmicas e profissionais como LinkedIn, Academia e Research Gate [Alencar et al., 2017] e também dos dados institucionais, como disciplinas, cursos, projetos, grupos e bancas. Sua documentação pode ser consultada em: http://openuai.ifpb.edu.br/doc/ouai/.

Para o levantamento de vocabulários existentes para reuso foram considerados os repositórios ou mecanismos de busca: LOV[7], swoogle[8] e o falcons[9]. Foram pesquisados os termos compatíveis semanticamente com o escopo dos dados. Foi

[6] http://lattes.cnpq.br/
[7] http://lov.okfn.org/dataset/lov/
[8] http://swoogle.umbc.edu/
[9] http://ws.nju.edu.cn/falcons/ontologysearch/

também verificado se aquele vocabulário candidato a reuso está disponibilizado de forma estável, com as URIs dos termos ativas. Assim, como exemplo, foram considerados os vocabulários FOAF, DCTerms, DBpedia, DOAP, SWPO, entre outros. Foram levantados conceitos, propriedades simples e propriedades de objeto candidatos e avaliados conforme os requisitos de dados. Como ilustração do levantamento, reuso e definição de termos da OUAI, a Tabela 2 apresenta um fragmento.

Tabela 2: Lista de Termos Reusados ou Definidos

Vocabulário	Termos Reusados ou Definidos
FOAF	Person, name
SWPO	Researcher, Topic, Publication, Article, TechnicalReport, Inproceedings, PublicationContainer, Journal, Proceedings, PublishingCompany, hasAuthors
DBpedia	EducationalInstitution, Occupation, Project, ResearchProject
VIVO	Department, AcademicDepartment
DCTerms	Language
OUAI	User, EducationalLevel, GeneralKnowledgeArea, KnowledgeArea, Keyword, Language, DevelopmentProject, ExtensionProject, InnovationProject, lattesURL

Em alguns casos, os conceitos dos requisitos de dados não tiveram um correspondente na pesquisa realizada nos repositórios de vocabulários. Como exemplos, os conceitos de Area (de conhecimento e de atuação) e lattesURL, que são de um escopo muito específico do contexto brasileiro, não tiveram equivalentes semânticos. Como estes, outros termos foram definidos. Um fragmento da ontologia pode ser visualizado na Figura 2.

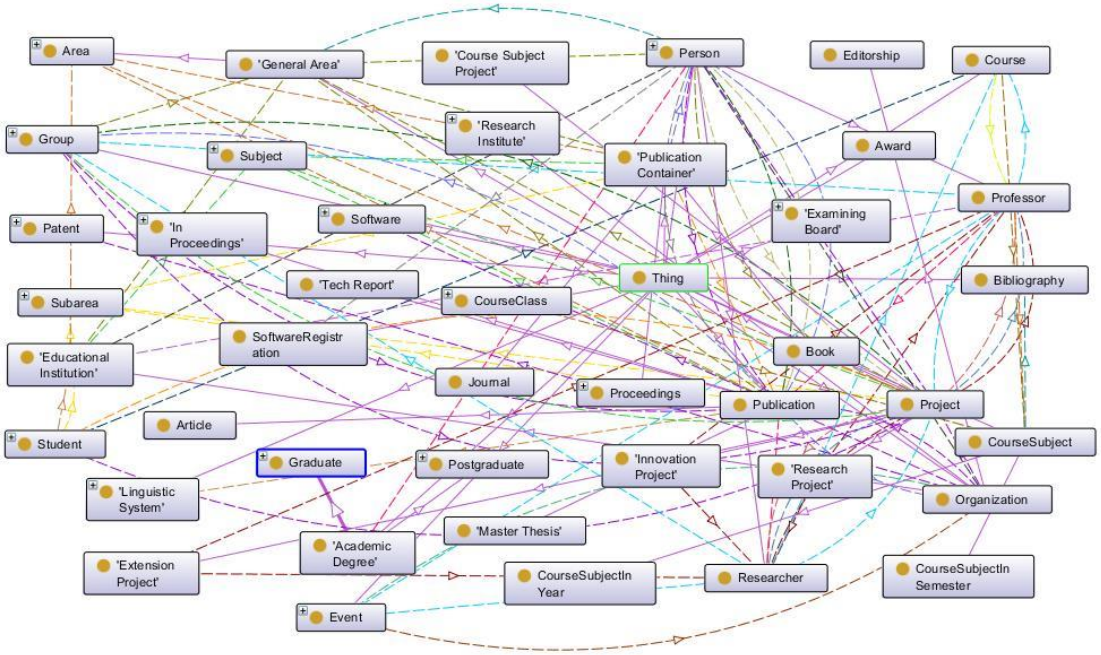


Figura 2: Fragmento da Ontologia OUAI

2.2 A Ferramenta de Conversão

Para a extração dos dados referentes a professores, todas as informações foram coletadas a partir dos perfis lattes de cada um. Sua extração foi feita usando um script, que faz a requisição dos downloads das informações dos professores em um arquivo XML. As áreas do conhecimento foram obtidas por meio das informações da Tabela de Áreas da CAPES. Entretanto, a identificação da área, de acordo com cada professor e projeto, foi realizada por meio do XML do currículo do professor. As informações dos cursos e matrizes curriculares, além de turmas de alguns anos foram obtidas por meio do sistema de controle acadêmico que gera uma planilha CSV.

A conversão dos dados para RDF é feita com a utilização dos termos da ontologia OUAI. Os dados são extraídos do CSV e dos XML e seus campos são mapeados para classes e propriedades da ontologia em questão. Cada linha do CSV é reconhecida como a instância de um recurso de uma determinada classe correspondente. No caso da lista de disciplinas, cada linha é convertida em uma instância da classe *ouai:courseSubject*, que terá suas propriedades geradas a partir dos campos do CSV. Na Tabela 3, exemplificam-se as correspondências entre alguns campos identificados no CSV e os termos correlatos encontrados na OUAI. Para os XMLs dos currículos, o mesmo princípio de mapeamento foi realizado.

Tabela 3: Mapeamento entre campos e termos da OUAI

Campo do CSV	Termo da OUAI	Tipo do Termo	Tipo de Dado
Disciplina	dc:title	Propriedade de dados	String
Período	ouai:courseSemester	Propriedade de dados	Inteiro
Carga Horária	time:hours	Propriedade de dados	Inteiro
Ementa	ouai:courseContent	Propriedade de dados	String
Bibliografia básica	ouai:hasBibliography	Propriedade de objetos	Recurso: ouai:bibliography

Assim, tanto para as linhas do CSV quanto para os objetos XML, são instanciados objetos em memória para cada recurso, onde os seus atributos são campos da linha ou campos entre as marcações XML. Assim, são montadas as triplas identificando o sujeito (recurso em questão), suas propriedades e objetos. O conjunto de triplas forma o grafo RDF que é persistido no RDF *Store*.

3. O Conjunto de Dados

Como comentado, o conjunto de dados é composto de informações relativas a docentes, projetos, áreas de atuação, cursos e disciplinas da UAI. Nessa versão, os dados do Curso de Sistemas para Internet estão incluídos. Um pequeno fragmento do conjunto de dados é mostrado na Figura 3. Neste é apresentada uma descrição de área de conhecimento ("Banco de Dados") cujo sujeito é http://openuai.ifpb.edu.br/ouai#subarea12 e de uma professora, com dados como a url do lattes e sua associação com um determinado projeto (http://openuai.ifpb.edu.br/ouai#researchproject34). O conjunto de dados completo pode ser obtido em http://openuai.ifpb.edu.br/dataset/.

```
<http://openuai.ifpb.edu.br/ouai#subarea12>
    a           <http://openuai.ifpb.edu.br/ouai#Subarea> ;
    rdfs:subclassOf  <http://openuai.ifpb.edu.br/ouai#area1> ;
    <http://purl.org/dc/terms/identifier> 10303030 ;
    <http://purl.org/dc/terms/title> "banco de dados" .
<http://openuai.ifpb.edu.br/ouai#professor10>
    a       <http://dbpedia.org/ontology/Professor> ;
    <http://openuai.ifpb.edu.br/ouai#citationName> "CAVALCANTI, V. M. B." ;
    <http://openuai.ifpb.edu.br/ouai#hasSubarea>
                                <http://openuai.ifpb.edu.br/ouai#subarea9> ;
    <http://openuai.ifpb.edu.br/ouai#isCoordinatorIn>
        <http://openuai.ifpb.edu.br/ouai#researchproject34>  ;
    <http://openuai.ifpb.edu.br/ouai#lattesURL>
        "http://lattes.cnpq.br/2868420260808800"^^xsd:anyURI ;
    <http://purl.org/dc/terms/identifier> "K4770822J4" ;
    foaf:name  "Valéria Maria Bezerra Cavalcanti" .
```

Figura 3: Fragmento exemplo do conjunto de dados da UAI

O conjunto de dados está publicado em distribuição RDF (serializado em turtle) sob licença aberta de acordo com a especificação "Licença Aberta para Bases de Dados (ODbL) do Open Data Commons"[10]. O conjunto é composto de 3.325 triplas e possui um tamanho de 333 KB. Seu acesso pode ser realizado também por meio de um SPARQL endpoint, em: http://openuai.ifpb.edu.br/sparql. Para facilitar o processamento do conjunto de dados, está em construção a disponibilização dos metadados estruturais existentes nele, assim como a publicação de mais metadados descritivos a seu respeito.

4. Consumo e Exemplo de Uso

Como primeira aplicação de consumo dos dados, foi especificada (e está sendo desenvolvida) a OpenUAI. O objetivo da OpenUAI é prover a visualização de dados da Unidade Acadêmica em questão como dados de currículos de professores, dados de pesquisa, de projetos, cursos e outros. Assim, os requisitos funcionais iniciais da aplicação foram definidos, a saber:

- Apresentar perfil dos professores da UAI, com dados do Lattes integrados a dados da UAI;
- Mostrar cursos (e disciplinas) da UAI;
- Mostrar áreas de atuação da UAI, conforme padronização do CNPq e CAPES;
- Apresentar projetos de pesquisa, extensão e de inovação e seus responsáveis.

Quanto à estrutura, a OpenUAI é uma aplicação Web dinâmica desenvolvida em Node.js[11], que consome dados por meio do SPARQL endpoint para gerar as telas da aplicação com a visualização dos dados. A Figura 3 é um exemplo dessa geração dinâmica, no qual resgata algumas triplas sobre informações de professores da Unidade, como nome, link do Lattes e área de atuação. Vale destacar que tal visualização resolveu uma antiga demanda dos alunos e professores da UAI, que sempre solicitava uma listagem das áreas de atuação de todos os professores de modo fácil e acessível.

[10] http://opendefinition.org/licenses/odc-odbl/
[11] https://nodejs.org/en/

Figura 3: Painel de Professores da UAI

Outro detalhe relevante da aplicação é que, devido ao seu perfil Web, é possível existir uma demanda variada de requisições para os mesmos recursos, e, por consequência, isso pode gerar processamentos repetitivos e até um aumento no tempo de resposta das páginas. Então, por questões de desempenho e latência, foi decidido que alguns conteúdos seriam guardados em cache. Portanto, periodicamente, é estabelecida a sincronização dos dados, principalmente, de informações dependentes de terceiros, como os dados do Lattes, para obtenção de melhores desempenhos, e também para garantir uma maior disponibilidade dos dados, especialmente nas ocasiões em que os servidores de terceiros estejam indisponíveis.

Outra importante visualização da OpenUAI, que também era bastante solicitada, se trata do painel das áreas de atuação dos professores da UAI, ilustrada na Figura 4. Esta difere da lista de professores (Figura 3), pois ela tenta exibir as áreas ajustando os seus tamanhos baseados na proporção de atuação dos professores da Unidade naquelas áreas. Com isso, fica mais fácil de identificar as áreas que possuem mais destaque na UAI, como as áreas de Metodologia e Técnicas da Computação e Sistema de Computação, além disso, a informação desse gráfico ajuda e cativa os alunos e os professores a encontrarem interesses em comum.

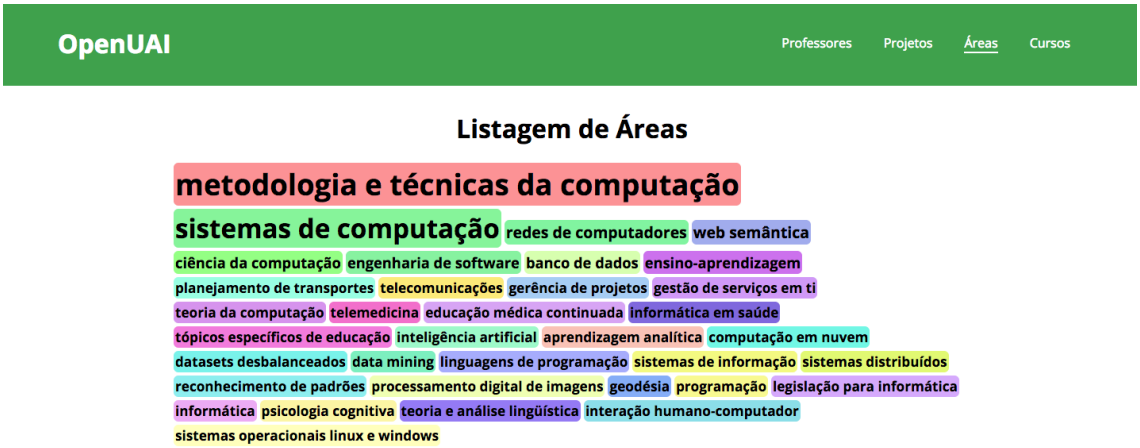


Figura 4: Painel das Áreas e Atuação Professores da UAI

Pensando no cenário fora da academia, as informações da OpenUAI auxiliarão também a comunidade externa, pois, muitas empresas que fazem parcerias com a instituição possuem dificuldades em identificar as principais expertises de cada professor. Assim, ao exibir as atuações da Unidade, por meio de seus artigos, projetos e

até disciplinas, essa carência será diminuída. O resultado é que, além de estimular um maior envolvimento das atuais parcerias, o portifólio gerado pela OpenUAI potencializará uma maior visibilidade para almejar novos parceiros e projetos.

Esses dados também podem ser uma ótima promoção para os alunos, pois cada vez mais os recrutadores vêm usando as atuações de candidatos em projetos públicos nos processos seletivos. Contudo, em muitos locais, por questões de restrições, nem sempre viabiliza-se registrar as atuações que ajudariam a construir um bom currículo profissional. Portanto, a OpenUAI irá, num próximo ciclo de desenvolvimento, apresentar os projetos de pesquisa, extensão, inovação e de disciplinas com respeito à participação dos alunos. Isso possibilitará à Unidade expor de forma mais compatível a atuação de seus futuros egressos. A ideia é também exibir resultados do ponto de vista de sua formação tecnológica por meio dos projetos desenvolvidos nas disciplinas e nos demais tipos de projetos. Também relacionado aos discentes, está sendo especificada uma aplicação que possa publicar dados dos egressos da UAI.

Os dados da UAI podem ser usados conjuntamente com dados de outras instituições. Algumas instituições já têm dados publicados de forma aberta que podem ser usados e gerar visualizações e análises mais completas em termos de projetos, perfis de pesquisadores e egressos, expertises desenvolvidas dentro do cenário da Ciência da Computação. Está em construção também o Portal de Dados Abertos da UAI onde o dataset em pauta estará disponível juntamente com outros que estão sendo construídos.

5. Desafios e Limitações

Na vida institucional dos docentes, é comum a solicitação de declarações de atuações acadêmicas para diversos fins. Logo, atualização do currículo, emissão de relatórios finais de pesquisa, publicação de trabalhos, entre tantas outras informações, são exemplos dessas solicitações. Entretanto, a compilação dessas declarações de atuações geralmente é considerada um trabalho bastante árduo, pois, muitas vezes, existem limitações quanto aos vários formatos e fontes de dados a serem manipulados para gerar informações acadêmicas e institucionais. Para agravar a situação, frequentemente, essa situação ocorre com prazos extremamente desconfortáveis para os gestores.

Em algumas situações a limitação no tratamento dos dados é tão evidente, que é possível ver redundância de trabalho, por exemplo, além de manter o currículo Lattes, geralmente, os docentes acabam sendo solicitados a enviar as mesmas informações do Lattes para outras plataformas ou relatórios, essencialmente, por causa das limitações em prover informações entre as plataformas. Já algumas plataformas tentam prover algum ponto de acesso e compartilhamento dos dados, mas ainda trazem consigo limitações derivadas pela falta de padronização e consistência de seus dados, contudo isso ainda chega a ser mais automatizado do que os cenários de dados que são mantidos e controlados de forma manual e oculta.

Então mesmo com a obtenção de uma alternativa mais automatizada para o acesso e processamento dos dados acadêmicos em pauta, algumas limitações ou políticas no tocante à coleta dos dados estão sendo gerenciadas ou definidas, a saber:

- A atualização dos dados do conjunto em relação aos professores e parte dos projetos é dependente da atualização dos currículos lattes. Então está em definição uma política de atualização dos dados, que contemplará a atualização desses dados a cada dois meses. Uma limitação ainda é que, atualmente, se

obtém os currículos de forma completa. Não existe um serviço de requisição do Lattes que permita selecionar os dados que realmente são necessários. Isso gera um esforço na etapa de seleção e atualização que poderia ser minimizado.

- Para os dados institucionais, pretende-se, para as turmas, realizar a atualização em cada início de período letivo (semestralmente). Uma limitação ainda existente nesse aspecto é que, nesse momento, o sistema de controle acadêmico do IFPB está sendo atualizado. Com a nova versão, será possível extrair informações diversas de forma mais fácil por meio da geração de planilhas CSV. Entretanto, seria mais simples e otimizado se houvesse uma API de acesso a esses dados onde fosse possível selecioná-los.

Em relação ao processo de conversão dos dados, algumas limitações e desafios também existem, como:

- A conversão de dados para RDF se constitui em uma etapa árdua, pois necessita de um vocabulário de referência de termos que nem sempre está pronto e completo. Por meio da ontologia OUAI, espera-se facilitar esse processo. Ela pode ser usada pelas diversas Unidades ou Instituições Acadêmicas de maneira geral, visto que seu escopo foi planejado e implementado para atender aos dados e termos associados à pesquisa, ensino, extensão e perfis de pessoas (e.g., pesquisadores) nesse meio.
- Ao mesmo tempo em que converter para RDF simplifica o uso e consumo dos dados, por outro lado, torna o processo mais demorado. Dessa forma, a disponibilização dos dados em distribuições como CSV e JSON pode facilitar seu uso e consumo. Portanto, é necessário prover meios para a disponibilização e consumo (por meio de uma API de acesso) às diferentes distribuições, facilitando seu processamento.
- Ainda em relação ao RDF, algumas dificuldades ocorrem quando se usa a linguagem SPARQL para criar as consultas de consumo. Pretende-se buscar simplificar isso usando meios que abstraiam a sintaxe da linguagem como, por exemplo, a partir do trabalho de Simões et al. (2015).

6. Considerações e Trabalhos Futuros

Este trabalho apresentou um conjunto de dados abertos conectados do domínio acadêmico, inicialmente construído com dados da Unidade Acadêmica de Informática do IFPB/Campus João Pessoa. Essa versão do conjunto apresenta informações sobre professores, áreas, projetos e cursos. É a primeira iniciativa de disponibilização de dados abertos do IFPB. Ao tornar esses dados disponíveis, aplicações podem consumi-los de modo a gerar visualizações que facilitem seu entendimento. Como exemplo inicial de aplicação de consumo, a OpenUAI está sendo desenvolvida. Mas, a ideia é que os alunos possam desenvolver outras aplicações usando esses dados, assim como aplicações que os utilizem conjuntamente com dados de outras instituições, gerando análises mais completas. Existem já iniciativas de trabalhos conjuntos com a UFCG e com a UFPE.

A ferramenta de conversão está sendo finalizada de modo que qualquer conjunto de dados em CSV ou XML possa ser convertida para RDF. Dessa maneira, o processo definido poderá ser utilizado em cenários acadêmicos diversos pertencentes a outras Unidades ou Instituições.

7. Referências

Alencar, A., Rocha, E., Souza, D. (2017) "An Approach for Data Integration on Academic and Professional User Profiles", submetido à Revista iSys - Revista Brasileira de Sistemas de Informação, ISSN: 1984-2902.

Heath, T. e Bizer, Christian. (2011) "Linked Data: Evolving the Web into a Global Data Space", Synthesis Lectures on the Semantic Web: Theory and Technology, 1:1, 1-136. Morgan & Claypool, 1th edition.

Isotani, S., Bittencourt, I. (2015), "Dados Abertos Conectados: Em busca da Web do Conhecimento", Editora Novatec (2015). ISBN: 978-85-7522-449-6.

Loscio, B. F., Burle, C., e Calegari, N. (2017) "Data on the Web Best Practices", disponível em: https://www.w3.org/TR/dwbp/. Acesso em 27 de julho de 2017.

Simões, N., Lopes, A., Souza, D. (2015) "Making SPARQL Query Formulation more Intuitive", In: International Conference on Information Integration and Web-based Applications & Services (IIWAS), 2015, Buxelas. Proceedings of the 17th International Conference on Information Integration and Web-based Applications & Services, 2015.

Soccer2014DS: a dataset containing player events from the 2014 World Cup

Marcos Roberto Ribeiro[1,2], Maria Camila N. Barioni[2], Sandra de Amo[2], Claudia Roncancio[3], Cyril Labbé[3]

[1] Instituto Federal de Minas Gerais (IFMG), Bambuí, Brazil

[2]Universidade Federal de Uberlândia (UFU), Uberlândia, Brazil

[3]Univ. Grenoble Alpes, CNRS, Grenoble INP, LIG, F-38000, Grenoble, France

marcos.ribeiro@ifmg.edu.br, {camila.barioni, deamo}@ufu.br,
{claudia.roncancio, cyril.labbe}@imag.fr

Abstract. *Player monitoring has become a common task in many sports. However, there is no public datasets of detailed soccer player events. Thus the creation of such datasets can be useful for diverse research fields such data mining, sports analytics, and continuous preference queries. In this paper, we describe the construction of the dataset Soccer2014DS containing player events of the 2014 Soccer World Cup. This dataset is composed of the raw extracted data collected by a web crawler and by derived streams with new calculated attributes. We also explain how we are using this dataset in experiments related to the development of a new query language.*

1. Introduction

The monitoring of sports players during matches started in the end of the XX century [Ali and Farrally 1991]. In this period, the researchers were more concerned to collect just health data from players. Since 2000, the monitoring tasks became more sophisticated due to the development of new technologies for GPS devices and softwares for video processing [Baca et al. 2009, Barris and Button 2008]. These new technologies allow collecting complex data to be used in detailed analysis of player events.

Despite the players monitoring be very common in official competitions, there are few public datasets with this information available. If we consider the soccer sport, to the best of our knowledge, there exist no public datasets with detailed player events. Thus, our main goal herein is to present the public dataset *Soccer2014DS* which contains player events of the 2014 Soccer World Cup. This dataset can be useful for diverse research fields such as data mining [Bialkowski et al. 2014, Gyarmati and Hefeeda 2015], sports analytics [Lucey et al. 2013, Perin et al. 2013], and continuous queries [Arasu et al. 2016].

This paper is organized as follows. Section 2 describes the original extracted data. Next, Section 3 presents the data streams derived from the original data. Section 4 discusses the research opportunities and Section 5 explains the dataset limitations. Finally, Section 6 concludes the paper.

2. Data Sources

The creation of the *Soccer2014DS* dataset started with the extraction of the original data available on the Huffpost Data web site[1] [Boice et al. 2014]. To the best of our knowledge, the Huffpost Data web site is the only public source containing match events of the

[1]http://data.huffingtonpost.com/2014/world-cup

2014 Soccer World Cup. This web site contains information about the 2014 Soccer World Cup provided by the company Opta Sports[2]. These data are used to display statistics and graphics about player events. Every match has an individual page where a user can delimit a time-line and see the details of this selection. Our first task was to study the source code of the Huffpost Data web site. Based on this study, we developed a web crawler to extract the data. The crawler starts the extraction in the page of the final match and follows the links to the remaining matches to complete the data collecting.

The extraction of the original raw data was performed in 2015. After this task, we organized the extracted data into the relations `Matches`, `Teams` and `Players` and the stream `Events`. The relations have just one instance and the duplicated data from all matches were eliminated. On the other hand, the stream `Events` has 64 instances (one instance per match) preserving all extracted event data. Appendix B presents the logical schema of the dataset (the derived data is addressed in Section 3).

The attributes of the relation `Matches` are `id` (match identifier), `date`, `time`, `venue` and `attendance`. Table 1(a) displays the attributes of the stream `Events`. The player coordinates (`x` and `y`) and final coordinates of the ball (`to_x`, `to_y`, and `to_z`) are expressed as a percentage of the field dimensions. The attributes `type` and `outcome` are used to identify the move performed by players. Appendix A presents the events associated with every combination of values of these attributes.

The attribute `field_pass` represents the continuous ball possession, t for true and f for false. The attribute `side` is the field side of the team, H for left side and A for right side. When the move is a pass to another player, the attribute `to` assumes the identifier of this player. For streams, we also must associate a `timestamp` for every tuple [Arasu et al. 2016]. In the stream `Events`, the `timestamp` is calculated using the attributes `min` and `sec` ($\texttt{timestamp} = \texttt{min} \times 60 + \texttt{sec}$).

Table 1. Relation attributes: (a) `Events` (b) `Players`

(a)

Attribute	Description
`id`	Event identifier
`period`	Period of math
`min, sec`	Minute and second of the event
`displaymin`	Displayed minute
`team`	Team identifier
`player_id`	Player identifier
`x, y`	Player coordinates
`type`	Type of event (move performed)
`outcome`	Result of the event
`field_pass`	Continuous ball possession
`side`	Field side of team
`to_x, to_y, to_z`	Final coordinates of ball
`to`	Player identifier of next move

(b)

Attribute	Description
`id`	Player identifier
`name`	Player name
`real_position`	Detailed position
`real_position_side`	Position side
`known_name`	Known name
`short_name`	Short name
`last_name`	Last name
`first_name`	First name
`middle_name`	Middle name
`team_id`	Team identifier
`preferred_foot`	Preferred foot
`club`	Club
`caps`	Matches played by player team
`goals`	Goals
`jersey_num`	Jersey number
`country`	Birth country
`birth_date`	Birth date
`position`	Position

The relation `Teams` is composed of the attributes `id` (team identifier),

[2] `http://www.optasports.com/`

name and iso (ISO acronym). Table 1(b) presents the attributes of the relation Players. The values for the attributes real_position, real_position_side, preferred_foot and position are shown in Table 2. Please see [Bakker 2015] for more details about the data gathered by the company Opta Sports.

Table 2. Values for Players attributes

Attribute	Values
real_position	Attacking Midfielder, Central Defender, Central Midfielder, Defensive Midfielder, Full Back, Goalkeeper, Second Striker, Striker, Wing Back, Winger
real_position_side	Centre, Centre/Right, Left, Left/Centre, Left/Centre/Right, Left/Right, Right, Unknown
preferred_foot	Both, Left, Mostly Left, Mostly Right, Right, (empty)
position	Defender, Forward, Goalkeeper, Midfielder

3. Derived Streams

After the data extraction described in the previous section, we created derived streams by applying cleaning and conversions over the original data. As described in the previous section, in order to know the exact player move, we must check the attributes type and outcome of the relation Events. In addition, the coordinates of the player and the ball, expressed by float values, could not be suitable for some applications where the user has to indicate a region of the soccer field.

To deal with practical situations, we decided to create the new derived streams: Moves(player_id, place, move) for the performed moves; and Places(player_id, place, ball, direc) for the player positioning. Using the derived streams the user can express temporal conditional preference easily without concern about type numbers of events or field coordinates. The Query 1 presents an example of query with temporal conditional preference over stream Moves. Please see [Ribeiro et al. 2017a, Ribeiro et al. 2017b] for more details about temporal conditional preferences. The logical schema presented in Appendix B shows the relationship of the derived streams and the original relations.

```
SELECT SEQUENCE FROM moves [RANGE 30 SECOND]
ACCORDING TO TEMPORAL PREFERENCES
IF PREVIOUS move = 'rec' THEN move = 'drib' BETTER move = 'pass' AND
move = 'pass' BETTER move = 'bpas' AND
IF ALL PREVIOUS place = 'mf' THEN place = 'mf' BETTER place = 'di'
```

Command Query 1: Temporal Conditional Preferences over Moves

The stream Moves contains just the moves performed by the players. So, we do not consider events without ball like cards, substitutions, etc. More precisely, the events types 17, 18, 19, 34, 43, 58, 60, 102 and the events with (type, outcome) equal to (5, 1), (6, 1), (53, 1) and (57, 1) are ignored. For the remaining move types, we use the mapping from (Events.type, Events.outcome) to Moves.move described in Table 3.

The possible values for the attribute place are defensive area (da), defensive intermediary (di), middle-field (mf), offensive intermediary (oi) and offensive area (oa). Figure 1 displays how we compute these values according to the attribute Events.x represented by dashed lines. The attribute ball (ball possession) is a mapping from the attribute field_pass. If field_pass $= t$ then ball $= 1$, and if field_pass $= f$

Table 3. Move mapping

move	(type, outcome)
pass	$(1,1)$ or $(59,1)$
bpas (bad pass)	$(1,0)$ or $(2,1)$
lbal (lost ball)	$(3,0)$, $(7,0)$, $(44,0)$, $(50,1)$, $(51,1)$, $(57,0)$, $(59,0)$ or $(61,0)$
drib (dribble)	$(3,1)$ or $(42,1)$
foul	$(4,0)$
fsuf (foul suffered)	$(4,1)$ or $(55,1)$
dled (dribbled)	$(45,0)$
bout (ball out)	$(5,0)$ or $(6,0)$
brec (ball recovery)	$(7,1)$, $(44,1)$, $(49,1)$, $(56,1)$ or $(61,1)$
int (interception)	$(8,1)$ or $(74,1)$
gsav (goalkeeper save)	$(10,1)$, $(11,1)$, $(41,1)$, $(52,1)$ or $(54,1)$
clea (clearance)	$(12,1)$
wsho (wrong shot)	$(13,1)$, $(14,1)$ or $(15,1)$
goal	$(16,1)$
rec (reception)	$(100,1)$
cond (conduction)	$(101,1)$

then $\texttt{ball} = 0$. In order to compute the attribute `direc` (move direction of a player), we consider the previous and the current place of each player. Next, we calculate the horizontal distance (*xdist*) and vertical distance (*ydist*) between these places. When $ydist = 0$ and $xdist = 0$, the direction is *none* since the player did not move. If $ydist > xdist$ then the direction is *lateral*. Otherwise, the direction is *backward* (for $xdist < 0$) or *forward* (for $xdist > 0$).

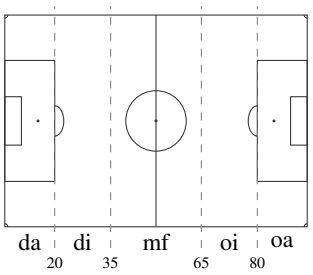


Figure 1. Soccer field division

Dataset Information. The full dataset and the importing tool are available for download in a Github repository[3]. This repository also provides additional tools for benchmarking continuous queries with temporal conditional preferences. All tools were developed using the Python language. The dataset is stored in the directory *data*. This directory contains the relations `Matches`, `Teams`, and `Players` stored in CSV (comma separated values) format. Such relations are the union of the data extracted from all matches. The *data* directory also contains the subdirectories *raw*, *events*, *moves*, and *places* to store respectively the extracted raw data, the stream `Events`, the stream `Moves` and the stream `Places`.

The subdirectory *raw* has, for every match, the JSON files *match_id.json*, *match_id-players.json* and *match_id-teams.json* where *match_id* is the match identifier. The subdirectories *events*, *moves* and *places* contain the CSV files *match_id.csv* with the stream data of the matches. Table 4 presents the total number of tuples, the number of instances, and the number of tuples per instance for all relations and streams of the dataset. The streams `Moves` and `Places` have fewer tuples than the stream `Events` due to the

[3] `https://streampref.github.io/wcimport/`

data cleaning and the computation of the new attributes.

Table 4. Data statistics

Relation/Stream	Tuples	Instances	Tuples/Instance
`Matches`	64	1	64
`Teams`	32	1	32
`Players`	736	1	736
`Events`	167,801	64	2621
`Moves`	130,607	64	2040
`Places`	137,621	64	2150

4. Research Opportunities

As we mentioned in the first section, our dataset is useful for a multitude of studies. This section outlines a non-exhaustive list of research fields that can benefit from using our dataset.

Data Mining. In the data mining field, our dataset can be explored for the validation of new techniques aiming the detection of temporal patterns and key events [Bialkowski et al. 2014, Gyarmati and Hefeeda 2015]. As the dataset has the player coordinates, we can also discover the special relations between these patterns and some specific field regions [Yue et al. 2014]. Using additional information about the localization of the matches, it could be possible to find player patterns correlated to environmental variables [Kaluarachchi and Aparna 2010].

Sports Analytics. Our dataset can be used to make various sports analysis over individual players or teams. These sports analysis can use specific data mining techniques or other approaches like data visualization. On using our dataset, specialists are able to analyze the moves performed, the player positioning, pass distances and many other variables related to a specific player [Perin et al. 2013]. In addition, the information of all team players can be combined to perform analysis over the team strategy and positioning [Lucey et al. 2013, Bush et al. 2015].

Preference Continuous Queries. An interesting research topic in the field of data streams processing is to incorporate temporal conditional preferences into continuous queries [Ribeiro et al. 2017b]. This new kind of query makes use of the implicit temporal information of data streams to select sequences of elements that best fit user preferences. As the tuples of data streams have a *timestamp*, *we can know the order of the tuples*. So, by using temporal preferences, the user can express wishes like "if there exists a value X in the past then I prefer a value Y to a value X in the current moment". Query 1 shows a practical example using the attributes of stream `Moves`.

Unlike traditional databases, data stream applications do not store all data due to limitations of time and space. The existence of datasets for data streams scenarios is important to allow the validation of new techniques for the evaluation of continuous queries. There are research works that proposed synthetic data generators [Bifet et al. 2011], but the real datasets are still useful for many specific situations.

In the work of [Ribeiro et al. 2017a] we proposed a preference model for reasoning with temporal conditional preferences on data stream scenarios. The *Soccer2014DS* dataset was used in the experiments to demonstrate the effectiveness of our proposed approach. We also used the *Soccer2014DS* dataset in the work of [Ribeiro et al. 2017b]. In

this latter work, the dataset was used to conduct an extensive set of experiments to compare the performance and the memory usage of algorithms to process continuous queries with temporal conditional preferences.

5. Limitations

This section describes the limitations of our *Soccer2014DS* dataset. The real soccer datasets collected by Opta Sports have additional information provided by a special attribute (`qualifiers`). However, these datasets are not public. The extracted data has this attribute, but it has no meaningful values. So, we drop this attribute from our dataset.

Our dataset does not have the coordinates of all players at every second. Only events related to moves, cards, fouls, and ball outs were gathered. So, positioning analysis must take this information into consideration.

We calculated just the attributes `move`, `place`, `ball` and `direc` for the derived streams `Moves` and `Places`. However, new attributes can be computed using the available information in the stream `Events`, for example, the distance of passes and shots. In addition, we decided to keep each match into an individual stream, but it is possible to join these data into a single stream if it is a requirement of the data analysis task.

6. Conclusion

In this paper, we described the creation of the public dataset *Soccer2014DS* containing player events of the 2014 Soccer World Cup. The construction of the dataset started with the extraction of data from the *Internet* using a web crawler. This extracted data was used to create the derived data by applying cleaning and conversions techniques.

The dataset and all the developed tools are available for download in a public repository. So, the dataset can be used to the development of new research works related to data mining, sports analytics, and continuous queries. We already use the *Soccer2014DS* dataset on our previous works [Ribeiro et al. 2017a, Ribeiro et al. 2017b] and we are still using this dataset on new research about continuous queries with temporal conditional preferences.

Acknowledgments. The authors thanks the Research Agencies CNPq, CAPES and FAPEMIG for supporting this work.

References

Ali, A. and Farrally, M. (1991). Recording soccer players' heart rates during matches. *Journal of Sports Sciences*, 9(2):183–189.

Arasu, A., Babcock, B., Babu, S., Cieslewicz, J., Datar, M., Ito, K., Motwani, R., Srivastava, U., and Widom, J. (2016). *STREAM: The Stanford Data Stream Management System*, pages 317–336. Springer, Berlin, Germany.

Baca, A., Dabnichki, P., Heller, M., and Kornfeind, P. (2009). Ubiquitous computing in sports: A review and analysis. *Journal of Sports Sciences*, 27(12):1335–1346.

Bakker, L. F. B. C. (2015). Visualizing football team strategies and player performance. Master's thesis, Eindhoven University of Technology, Eindhove, Netherlands.

Barris, S. and Button, C. (2008). A review of vision-based motion analysis in sport. *Sports Medicine*, 38(12):1025–1043.

Bialkowski, A., Lucey, P., Carr, P., Yue, Y., Sridharan, S., and Matthews, I. (2014). Large-scale analysis of soccer matches using spatiotemporal tracking data. In *International Conference on Data Mining (ICDM)*, Shenzhen, China.

Bifet, A., Holmes, G., Pfahringer, B., Read, J., Kranen, P., Kremer, H., Jansen, T., and Seidl, T. (2011). MOA: A real-time analytics open source framework. In *European Conference on Machine Learning and Principles and Practice of Knowledge Discovery in Databases (ECML PKDD)*, pages 617–620, Athens, Greece.

Boice, J., Fung, H., and Bycoffe, A. (2014). URL: `http://data.huffingtonpost.com/2014/world-cup` (visited on 23/04/2015).

Bush, M., Barnes, C., Archer, D. T., Hogg, B., and Bradley, P. S. (2015). Evolution of match performance parameters for various playing positions in the english premier league. *Human Movement Science*, 39:1–11.

Gyarmati, L. and Hefeeda, M. (2015). Estimating the maximal speed of soccer players on scale. In *Machine Learning and Data Mining for Sports Analytics Workshop*, Porto, Portugal.

Kaluarachchi, A. and Aparna, S. V. (2010). CricAI: A classification based tool to predict the outcome in odi cricket. In *International Conference on Information and Automation for Sustainability (ICIAFS)*, pages 250–255, Colombo, Sri Lanka.

Lucey, P., Oliver, D., Carr, P., Roth, J., and Matthews, I. (2013). Assessing team strategy using spatiotemporal data. In *ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, pages 1366–1374, Chicago, IL, USA.

Perin, C., Vuillemot, R., and Fekete, J.-D. (2013). Soccerstories: A kick-off for visual soccer analysis. *IEEE Transactions on Visualization and Computer Graphics*, 19(12):2506–2515.

Ribeiro, M. R., Barioni, M. C. N., de Amo, S., Roncancio, C., and Labbé, C. (2017a). Reasoning with temporal preferences over data streams. In *International Florida Artificial Intelligence Research Society Conference (FLAIRS)*, Marco Island, Florida.

Ribeiro, M. R., Barioni, M. C. N., de Amo, S., Roncancio, C., and Labbé, C. (2017b). Temporal conditional preference queries on streams. In *International Conference on Database and Expert Systems Applications (DEXA)*, Lyon, France.

Yue, Y., Lucey, P., Carr, P., Bialkowski, A., and Matthews, I. (2014). Learning fine-grained spatial models for dynamic sports play prediction. In *International Conference on Data Mining (ICDM)*, pages 670–679, Shenzhen, China.

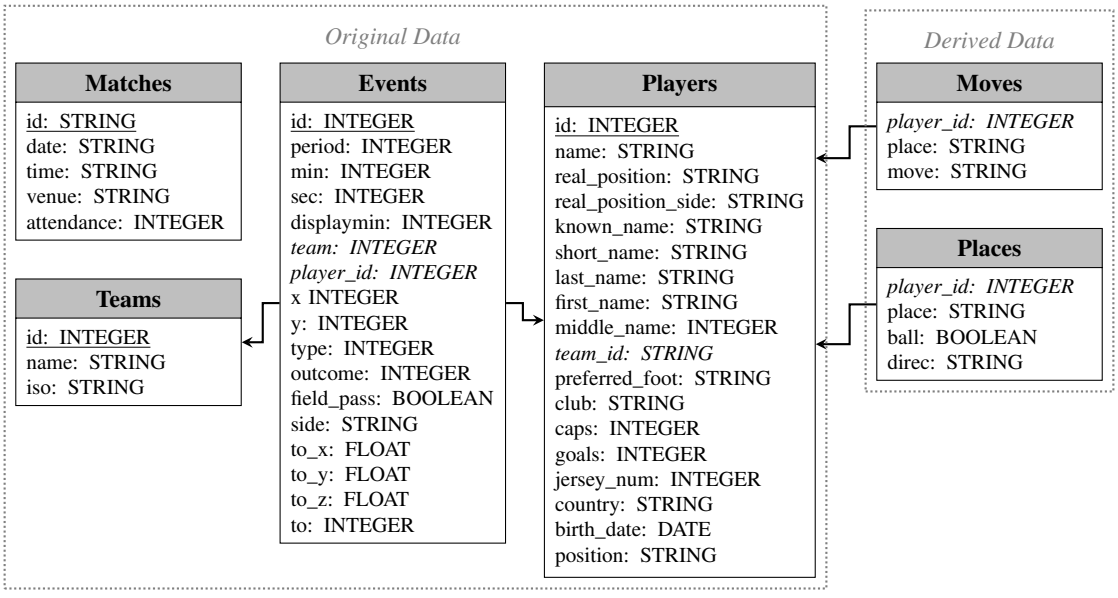
Appendix

A. Events Associated to Values of Attributes Type and Outcome

type	outcome	Event description
1	0	Non completed pass
	1	Completed pass
2	1	Pass to offside player
3	0	Dribble losing ball
	1	Successful dribble
4	0	Foul committed
	1	Foul suffered
5	0	Ball out
	1	Wined throw-in or goal kick
6	0	Ball out on goal line
	1	Wined corner kick
7	0	Dispossessed opponent without possession
	1	Dispossessed opponent with possession
8	1	Interception
10	1	Goalkeeper save shot
11	1	Goalkeeper catches crossed ball
12	1	Clearance (shot out defensive zone)
13	1	Miss (shot out goal)
14	1	Post (shot on goal frame)
15	1	Attempt saved by other player
16	1	Goal
17	1	Card
18	1	Player substituted
19	1	Player comes on (as substitute)
34	1	Player line up and formation
41	1	Goalkeeper punches ball
42	1	Skill on the ball
43	1	Deleted event

type	outcome	Event description
44	0	Lost aerial duel
	1	Wined aerial duel
45	0	Player dribbled
49	1	Ball recovery
50	1	Player dispossessed
51	1	Error (causing ball dispossession)
52	1	Goalkeeper picks up the ball
53	1	Cross not claimed by goalkeeper
54	1	Goalkeeper advances and wins possession
55	1	Offside provoked
56	1	Shield ball from opponent
57	0	Player causes throw-in reversion
	1	Player wins throw-in reversion
58	1	Goalkeeper faced to penalty kick
59	0	Goalkeeper clears ball without possession
	1	Goalkeeper clears ball with possession
60	0	Goal chance missed
61	0	Player touches ball and without possession
	1	Player touches ball and with possession
74	1	Accidental blocking
100	1	Ball reception
101	1	Ball conduction
102	1	Start playing

B. Soccer2014DS Logical Schema



Spatial Datasets for Conducting Experimental Evaluations of Spatial Indices

Anderson Chaves Carniel[1], **Ricardo Rodrigues Ciferri**[2], **Cristina Dutra de Aguiar Ciferri**[1]

[1]Department of Computer Science – University of São Paulo
13.566-590 – São Carlos – SP – Brazil

accarniel@gmail.com, cdac@icmc.usp.br

[2]Department of Computer Science – Federal University of São Carlos
13.565-905 – São Carlos – SP – Brazil

ricardo@dc.ufscar.br

Abstract. *Spatial database systems widely employ spatial indices to accelerate the processing time of spatial queries. To measure the performance of spatial indices, researchers conduct extensive experimental evaluations by varying, e.g., characteristics of the spatial datasets to be indexed and types of spatial queries to be issued. Thus, the public sharing of spatial datasets potentially benefits the research community that aims to reproduce experiments or to conduct new experiments. In this paper, we provide spatial datasets that we have been used in our experiments. We provide two types of datasets: (i) spatial datasets that represent real-world phenomena to be indexed by the spatial indices, and (ii) spatial datasets that aid in the construction of spatial queries to be issued on an indexed dataset. As a result, our spatial datasets can be used to create and perform experiments that aim to evaluate the performance of spatial indices.*

1. Introduction

Spatial database systems and Geographic Information Systems (GIS) provide the needed foundation for applications that require the management of geometric and geographic phenomena [Rigaux et al. 2001]. To this end, applications characterize geographic phenomena by using spatial data types like points, lines, and regions [Güting 1994]. For instance, points representing hydrants, lines representing streets, and regions representing engineering buildings. In order to efficiently retrieve spatial objects, spatial database systems and GIS widely employ spatial indices [Gaede and Günther 1998, Oosterom 2005]. The use of spatial indices accelerates the processing time of several types of spatial queries, such as range queries and point queries.

Several spatial indices have been proposed in the literature (e.g., [Gaede and Günther 1998] surveys more than 40 spatial indices). Examples of spatial indices are hierarchical structures like the R-tree [Guttman 1984] and the R*-tree [Beckmann et al. 1990]. They have different characteristics and can employ different parameter values. For instance, the R-tree can employ different split algorithms, while the R*-tree includes reinsertion policies. With the evolution of the storage devices, there are also specific spatial indices for newer storage devices. For instance, flash-aware spatial indices to exploit the advantages of flash-based solid state drives (SSDs). Examples

are the RFTL [Wu et al. 2003] and the FOR-tree [Jin et al. 2015], as well as generic frameworks for creating flash-aware spatial indices like FAST [Sarwat et al. 2013] and eFIND [Carniel et al. 2017a].

Frequently, extensive experimental evaluations measure the performance of spatial indices proposed in the literature. In these experiments, the parameter values of the spatial indices are varied in order to evaluate different configurations of these indices. In addition, the running environment of the experiments is also varied by indexing spatial datasets with different characteristics. For instance, the number of spatial objects, and the spatial data types and geometric configurations (e.g., number of points) of the spatial objects. In addition, the experiments may vary the types of spatial queries that will be issued to an indexed spatial dataset. For instance, the execution of points queries, and the execution of range queries with different sizes of search objects.

The public access and availability of spatial datasets employed in the experiments provide a valuable data sharing for the research community that wants to reproduce experiments or to conduct new experiments. Unfortunately, in the most of the cases, the spatial datasets of the experiments are not publicly available since they are synthetically generated by tools that do not have open access. Examples of these situations are found in [Greenel 1989, Wu et al. 2003, Emrich et al. 2010, Sarwat et al. 2013]. Although other experiments (e.g., in [Lv et al. 2011, Luo et al. 2012, Jin et al. 2015]) often use datasets based on the spatial dataset of the R-tree portal[1], the employed dataset limits the variation of other characteristics of the experiment. In addition, the experiments (e.g., in [Sarwat et al. 2013, Jin et al. 2015]) commonly do not provide details regarding the search objects used in the execution of spatial queries.

In this paper, we provide different sets of spatial datasets. We distinguish them in two types: (i) spatial datasets that can be indexed by the spatial indices, and (ii) spatial datasets to form spatial queries. Together, these types of datasets can be used in experiments to evaluate the performance of spatial indices. For the first type of spatial datasets, we provide real spatial datasets extracted from the OpenStreetMap[2] that vary (i) volume of data, (ii) spatial data types, and (iii) geometric configurations. For the second type of spatial datasets, we provide other spatial datasets that vary (i) the size of the search objects, (ii) the type of spatial query, and (ii) the correlation of the search objects with the indexed spatial dataset. We have been used these two types of spatial datasets in our experiments to measure the performance of spatial indices in SSDs [Carniel et al. 2016b, Carniel et al. 2017a, Carniel et al. 2017b]. Despite this fact, a complete description of these spatial datasets is only discussed in this paper. We also provide the open and public access to these spatial datasets at `http://gbd.dc.ufscar.br/festival/datasets.html`.

This paper is organized as follows. Section 2 details our spatial datasets, including the methodology of their creation and useful descriptions of them. Section 3 discusses how to use our spatial datasets in experiments. Finally, Section 4 finishes the paper.

[1] `http://www.chorochronos.org/?q=node/59`
[2] `http://wiki.openstreetmap.org/wiki/Main_Page`

2. Generated Spatial Datasets

In this section, we provide the complete description regarding the collection, generation, and description of our spatial datasets, following their types introduced in Section 1. Section 2.1 details the methodology that we employed to creating our spatial datasets. Our spatial datasets are stored in relational tables of the PostgreSQL with the spatial extension PostGIS. Section 2.2 provides the description of these relational tables. Finally, Section 2.3 presents the statistical descriptions of our spatial datasets.

2.1. Employed Methodology for Collecting and Generating the Spatial Datasets

We employed two different methodologies to collect and generate our spatial datasets, one methodology for each type of spatial dataset. They are detailed as follows.

Spatial datasets that can be indexed. We extract them from the OpenStreetMap (OSM). OSM is a project that provides an environment to anyone map any geographic event in the world. Thus, it crowdsources spatial data from different people. It also offers mechanisms to make available and public access of these collected spatial data[3].

Our spatial datasets consist only of geographic events of Brazil and were extracted and treated as follows. We firstly downloaded OSM data by using GeoFabrik[4]. It is a web site that permits to download OSM data, as .osm files, of specific regions and countries of the world. Since OSM data is constantly updated by people of the world, GeoFabrik provides data that are almost daily updated.

Since the format .osm uses OSM data structures to represent geographic phenomena, tools for extracting spatial objects from these files are needed. We have used the tool osm2pgsql[5], which transforms the OSM data into spatial objects stored in relational tables of a database in the PostgreSQL with the spatial extension PostGIS. osm2pgsql creates three important relational tables: (i) *planet_osm_point*, which stores only points, (ii) *planet_osm_line*, which stores only lines, and (iii) *planet_osm_polygon*, which stores only regions. Each table has two important columns: (i) *osm_id*, which stores unique identifiers of OSM; and (ii) *way*, which stores spatial objects. The other columns of these tables represent the types of events that OSM is able to map[6]. Two important notes of osm2pgsql are: (i) different versions of this tool can import a different number of spatial objects from a same .osm file, and (ii) a specific parameter (-G) should be provided for storing spatial objects with multiple components.

After importing spatial objects into relational tables, we then created specific spatial datasets by issuing SQL queries on the relational tables created by osm2pgsql. That is, we created relational tables that store spatial datasets representing specific contexts. In addition, these spatial datasets have different volumes, spatial data types, and geometric characteristics that are important to be varied in experiments evaluating spatial indices. These details are given in Sections 2.2 and 2.3.

Spatial datasets to form spatial queries. We created spatial datasets to provide search objects for point queries and range queries [Gaede and Günther 1998]. We focus on these

[3]`http://wiki.openstreetmap.org/wiki/Downloading_data`
[4]`http://download.geofabrik.de/`
[5]`http://wiki.openstreetmap.org/wiki/Osm2pgsql`
[6]`https://wiki.openstreetmap.org/wiki/Map_Features`

kind of queries because of their common usage in spatial applications. Formally, they are queries that return all the objects o from a set D where the predicate P returns *true* for a search object s, i.e., $SpatialQuery(D, s, P) = \{o | o \in D \wedge P(s, o) = true\}$. A point query specifies that s is a point and that P is the predicate *intersects*, i.e., $PointQuery(D, s \in point) = SpatialQuery(D, s, intersects)$. A range query specifies that s is a rectangular-shaped object and that P can assume any predicate, i.e., $RangeQuery(D, s$ is a rectangle, $P) = SpatialQuery(D, s, P)$.

Based on that, our spatial datasets to form spatial queries consist of a dataset containing only points and another dataset containing rectangles. To create them, we follow two different approaches. The first approach generates random objects (i.e., points and rectangles) that intersect the region (i.e., polygon) representing Brazil. Thus, there is no guarantee that a spatial query to be issued to an indexed spatial dataset will return spatial objects. On the other hand, the second approach generates search objects that are correlated with at least one object of a spatial dataset that can be indexed. For generating rectangles, there are two types of correlations: containment and intersection. This means that spatial queries that employ these search objects potentially will return spatial objects from an indexed spatial dataset. To guarantee the correlations, we generated one spatial dataset to form spatial queries for each spatial dataset that can be indexed (previously extracted). We also guarantee that the rectangles have proportional sizes in relation to the total extent of Brazil. As a result, we can construct spatial queries that return a different number of spatial objects, as detailed in Section 2.3. Section 3 discusses how spatial queries are constructed by using the search objects from these spatial datasets.

We provide the public access to the algorithms, implemented as PL/pgSQL functions, responsible for generating points and rectangles at `http://gbd.dc.ufscar.br/festival/`. Thus, researchers can use these functions to create points and rectangles that are not stored in our original spatial datasets.

2.2. Data Schema for Storing the Spatial Datasets

Here, we describe the relational tables that store our two types of spatial datasets.

Spatial datasets that can be indexed. We have extracted the OSM data related to Brazil in two different dates, 3 May 2016 and 16 January 2017. We call these files as *brazil2016.osm* and *brazil2017.osm*, respectively. Then, by following the methodology in Section 2.1, we created the relational tables detailed in Table 1. Thus, these relational tables store our spatial datasets. The columns of Table 1 are detailed as follows. The first column provides the name of the relational tables. The second column gives the .osm file of origin. The third column specifies the version of the employed osm2pgsql. Finally, the last column characterizes the context of the relational table, that is, the real-world phenomena that the spatial objects represent. Note that *brazil2017* is a relational table derived from the union of tables generated by osm2pgsql, forming, therefore, a bigger relational table. Other relational tables can be also combined. All these relational tables have the same columns. They are:

- *id*: has sequential values and is the primary key of a table.
- *osm_id*: provides the unique identifier used by OSM. By using this value, we can get the full description of a spatial object by using tools like Nominatim[7].

[7] `http://wiki.openstreetmap.org/wiki/Nominatim`

- *way*: stores spatial objects.

Spatial datasets to form spatial queries. According to the methodology in Section 2.1, we created two types of spatial datasets. We store them in two respective relational tables, *generated_point* and *generated_rectangles*. Their structures are detailed in Tables 2 and 3, respectively. These relational tables store randomly created or correlated spatial objects. In addition, the generated rectangles have proportional sizes in relation to a given spatial dataset that can be indexed. The number of generated rectangles for each proportional size is given in the third column in Table 3.

Table 1. Description of the relational tables that store our spatial datasets that can be indexed.

Name	.osm File of Origin	osm2pgsql Version	Spatial Data Type of *way*	Context of the Spatial Objects
brazil_buildings2016	*brazil2016.osm*	0.91.0-dev (64 bit id space)	region	buildings[8] of Brazil, e.g., universities, hotels, schools, hospitals, warehouses, stadiums, houses, churches, etc.
brazil_buildings2017	*brazil2017.osm*			
brazil_highways2017	*brazil2017.osm*	0.93.0-dev (64 bit id space)	line	highways[9] of Brazil, e.g., roads, footpaths, streets, cycleways, raceways, etc.
brazil_points2017	*brazil2017.osm*		point	locations mapped as points, e.g., toilets, telephones, banks, hydrants, etc.
brazil2017	*brazil2017.osm*		region, line, point	union among all the regions, lines, and points of the *brazil2017.osm*

Table 2. Description of the relational table (*generated_point*) that stores points to form point queries.

Column	Data Type	Description
id	integer	sequential numerical values representing the primary key
is_correlated	Boolean	Boolean values are used to indicate if the point is correlated to a spatial dataset or not
dataset	text	possible textual values are the names of the spatial datasets in Table 1. If *is_correlated* is *true*, it indicates the dataset in which *geom* is correlated; otherwise, it may indicate the target dataset to process the spatial query containing a search object from *geom*
geom	point	there are 100 randomly created points and 100 points correlated for each spatial dataset in Table 1

[8] http://wiki.openstreetmap.org/wiki/Map_Features#Building
[9] http://wiki.openstreetmap.org/wiki/Map_Features#Highway

Table 3. Description of the relational table (*generated_rectangle*) that stores rectangles to form range queries.

Column	Data Type	Description
id	integer	sequential numerical values representing the primary key
is_correlated	Boolean	Boolean values are used to indicate if the point is correlated to a spatial dataset or not
type	text	stores NULL, if *is_correlated* is *false*. Otherwise, it stores the type of correlation that was considered to create the rectangle: *intersects* or *contains*. This means that a rectangle *intersects* (or *contains*) at least one spatial object from the *dataset*
dataset	text	possible textual values are the names of the spatial datasets in Table 1. If *is_correlated* is *true*, it indicates the dataset in which *geom* is correlated; otherwise, it may indicate the target dataset to process the spatial query containing a search object from *geom*
percentage	double	percentage area of the total extent of Brazil. Examples are: 0.001%, 0.01%, 0.1%, and 1%
geom	region	there are 100 randomly created rectangles and 100 rectangles correlated for each spatial dataset in Table 1 considering values of *type*

2.3. Statistical Description

Table 4 describes the statistical description of our relational tables that store the spatial datasets that can be indexed. The first feature is with respect to the data volume, that is, the number of spatial objects of each dataset (Table 4a). In addition, we show the minimum, maximum, and average value of the following features: (i) number of points (Table 4a); (iv) length (Table 4b), only for datasets storing lines, and (ii) area and perimeter (Table 4c), only for datasets storing regions;. We can conclude from Table 4 that our spatial datasets have a great diversity of geometry characteristics, which can aid in the evaluation of spatial indices.

Table 4. Statistical description of the relational tables that store our spatial datasets that can be indexed.

Relational Table	# of Spatial Objects	# of Points		
		Min	Max	Avg
brazil_buildings2016	534,926	4	506	7
brazil_buildings2017	1,486,557	4	744	8.29
brazil_highways2017	2,644,432	2	1,550	9.30
brazil_points2017	770,842	1	1	1
brazil2017	5,577,373	1	94,047	14.86

(a)

Relational Table	Length		
	Min	Max	Avg
brazil_highways2017	0.02	100,000	674.38
brazil2017	0	100,000	1,054.92

(b)

Relational Table	Area			Perimeter		
	Min	Max	Avg	Min	Max	Avg
brazil_buildings2016	0.0005	4,756,911	672.81	0.23	22,730.19	19.09
brazil_buildings2017	∼3.66e-05	4,756,911	368.56	0.03	22,730.19	63.4
brazil2017	0	∼9.30e12	13,421,569.76	0	24,572,026.11	670.32

(c)

The statistical description of *generated_point* and *generated_rectangle* are showed in Tables 5 and 6, respectively. These tables show the minimum, maximum, and average number of spatial objects that are returned when we execute point queries and range queries based on these spatial datasets. The number of spatial queries followed the number of objects reported in Tables 2 and 3. More specifically, the number of point queries

was of 200 for each indexed spatial dataset: 100 with random points, and 100 with correlated points. The number of range queries was of 300 for each indexed spatial dataset and for each percentage value: 100 with random rectangles generated for the indexed spatial dataset, 100 correlated rectangles with the type of intersection, and 100 correlated rectangles with the type of containment. This means that we consider only the correlated objects that were generated taking as a basis the indexed spatial dataset (Table 1). For instance, the first row of Table 5 shows that the maximum number of objects from *brazil_buildings2016* returned by the 100 point queries using correlated points based on *brazil_buildings2016* is equal to 3. Note that in Table 5, in most of cases, the point queries using random points did not return objects from the indexed datasets since the search objects were randomly generated considering the total extent of Brazil (Section 2.1).

In Table 6, we use (I) to indicate correlated rectangles based on the intersection and (C) to indicate rectangles based on the containment. Further, these symbols also indicate the predicate used in our range queries, where (I) means *intersects*, and (C) means *contains*. Note that the percentage area of the rectangles is shown in the second column of Table 6.

Table 5. Statistical description of *generated_point*.

Indexed Dataset	Random Points			Correlated Points		
	# of Spatial Objects			# of Spatial Objects		
	Min	Max	Avg	Min	Max	Avg
brazil_buildings2016	0	0	0	1	3	1.03
brazil_buildings2017	0	0	0	1	2	1.01
brazil_highways2017	0	0	0	1	5	1.7
brazil_points2017	0	0	0	1	1	1
brazil2017	1	10	7.5	4	15	9.04

Table 6. Statistical description of *generated_rectangle*.

Indexed Dataset	Perc.	Random Rectangles			Correlated Rectangles (I)			Correlated Rectangles (C)		
		# of Spatial Objects			# of Spatial Objects			# of Spatial Objects		
		Min	Max	Avg	Min	Max	Avg	Min	Max	Avg
brazil_buildings2016	0.001%	0	39	0.55	1	74,206	21,679.51	1	74,200	22,279.39
	0.01%	0	11,719	204.35	3	95,023	24,144.46	17	95,053	30,326.52
	0.1%	0	5,261	216.31	20	99,479	37,079.97	199	99,501	38,193
	1%	0	85,304	7,822.47	211	113,261	55,214.83	728	102,912	53,780.87
brazil_buildings2017	0.001%	0	82	1.2	6	278,021	88,266.88	9	277,630	101,127.05
	0.01%	0	11,771	134.96	6	515,477	206,265.15	17	515,475	223,487.47
	0.1%	0	515,117	13,443.87	190	520,496	281,023.15	760	519,977	256,348.36
	1%	0	144,873	14,166.93	641	529,540	242,520.75	780	657,213	267,309.71
brazil_highways2017	0.001%	0	1,027	47.12	8	16,582	2,633.49	8	15,859	2,405.85
	0.01%	0	4,994	284.75	44	108,791	12,349.97	139	83,037	12,778.12
	0.1%	0	215,533	10,843.76	843	217,628	36,309.8	826	232,674	37,941.99
	1%	37	344,322	39,497.31	3,392	470,558	159,296.06	2,037	473,662	153,861.3
brazil_points2017	0.001%	0	132	6.34	1	31,102	3,889.45	1	31,778	2,727.45
	0.01%	0	3,139	116.19	23	47,212	8,818.45	4	47,527	9,169.48
	0.1%	9	73,036	3,741.52	72	72,344	18,426.5	38	72,417	14,098.62
	1%	37	90,343	11,723.36	389	141,498	51,208.44	429	140,891	48,481.03
brazil2017	0.001%	2	1,350	78.05	22	292,907	52,215.29	3	294,792	39,986.56
	0.01%	12	26,358	684.35	69	573,095	70,012.62	117	572,782	82,713.27
	0.1%	80	569,545	30,808.87	795	624,866	196,012.61	409	617,289	182,953.84
	1%	506	593,943	76,815.42	1,052	979,319	394,746	4,016	952,634	404,238.33

3. Utilization of the Spatial Datasets in Performance Evaluations

Our spatial datasets can be used according to the purpose in which they are created. The spatial datasets described in Table 1 have the purpose to be handled by spatial indices, such as:

- constructing a spatial index on a spatial dataset. For instance, create an R-tree on *brazil_buildings2016*. Future operations like updates, queries, and deletions are possible after the construction.
- constructing a spatial index on a part of a spatial dataset. For instance, create an R-tree on a percentage of the elements contained in *brazil_buildings2016*. Thus, the remaining elements can be used in future operations like insertions after the processing of spatial queries.

The spatial datasets described in Tables 2 and 3 have the purpose to form spatial queries. That is, these spatial datasets are dedicated to creating point queries and range queries. This creation can be made as follows:

- for point queries: we select a list of points P from *generated_point*. For each point p in P, we can form point queries $PointQuery(D, p)$, where D is a given indexed spatial dataset. For instance, we can form 100 point queries to be processed by an R-tree created on *brazil_buildings2016*, where the points are correlated to the indexed spatial dataset.
- for range queries: we select a list of rectangles R from *generated_rectangle*. For each rectangle r in R and given a predicate P, we can form range queries $RangeQuery(D, r, P)$. For instance, we can form 100 range queries with the predicate *contains* to be processed by an R-tree created on *brazil_buildings2016*, where the rectangles have a correlation of containment with the indexed spatial dataset and percentage size of 0.01%.

The main benefits to using our spatial datasets to be handled by spatial indices are that (i) they are based on real geographic data and thus do not have fixed distributions of objects, (ii) they provide different geometric complexities and data volume, and (iii) they can be combined to create other specific datasets. These factors contribute to a complete empirical evaluation of spatial indices that can be done by creating different workloads based on our spatial datasets.

With respect to the benefits of our spatial datasets to form spatial queries, we provide several types of search objects that directly stress a spatial index. The main reason is that we include two types of search objects: randomly created search objects, which aid to evaluating if the spatial index shows a good performance in discarding of data that do not belong to the answer, and correlated search objects, which may force the traversal of many pages of the spatial index. In addition, the rectangles of range queries permit the variation of the number of objects to be returned since they have different sizes.

We have been used the spatial datasets of this paper in experiments to evaluate the performance of spatial indices in different scenarios. We first analyzed and correlated the performance of spatial indices managed in hard disks and SSDs by using the spatial dataset *brazil_buildings2016* and by executing range queries by using rectangles randomly generated and stored in *generated_rectangle* [Carniel et al. 2016b]. The use of

these datasets in this experiment led us to identify important design goals that flash-aware spatial indices should address to deliver good performance on SSDs. As a result, we propose eFIND as a solution for spatial indexing on SSDs. Its performance was measured by indexing the spatial dataset *brazil_buildings2017* and by processing range queries based on randomly created rectangles stored in *generated_rectangle* [Carniel et al. 2017a]. Another experiment using the same spatial datasets were conducted in order to measure the performance of spatial indices in flash simulators [Carniel et al. 2017b]. Future work includes the extension of these experiments by indexing the spatial dataset called *brazil_buildings2017_v2*, which was created by using the osm2pgsql version 0.93. This spatial dataset contains 1,485,866 regions and has similar characteristics to the *brazil_buildings2017*. In addition, we will index the other spatial datasets storing lines and points. In addition, we plan to execute range and point queries with correlated search objects. Finally, all the experiments employ FESTIval [Carniel et al. 2016a], an open-source PostgreSQL extension to benchmark spatial indices on different storage devices. FESTIval has several default spatial datasets that can be indexed, which include the ones provided in this paper. A complete documentation of FESTIval is available at `http://gbd.dc.ufscar.br/festival/`.

4. Conclusions and Future Work

This paper provides several spatial datasets that can be employed in experiments to evaluate the performance of spatial indices. Two types of spatial datasets are provided: spatial datasets to be handled by spatial indices, and spatial datasets to form spatial queries. They have particular features that aid in the evaluation of spatial indices, including empirical experiments on different storage devices [Carniel et al. 2016b, Carniel et al. 2017a, Carniel et al. 2017b]. Our spatial datasets can be downloaded at `http://gbd.dc.ufscar.br/festival/`, which also provides a framework to conduct experimental evaluations of spatial indices. As a result, researchers can reproduce old experiments and create new experiments to better understand the performance of spatial indices under different scenarios.

We plan to extract other spatial datasets from the OpenStreetMap. In addition, future work includes the generation of more rectangles and points to act as search objects of spatial queries. In fact, this task can be made by using our generator algorithm.

Acknowledgements

This work has been supported by the following Brazilian research agencies: CAPES, CNPq, and FAPESP. The first author has been supported by the grant #2015/26687-8, FAPESP. The second author has been supported by the grant #311868/2015-0, CNPq.

References

Beckmann, N., Kriegel, H.-P., Schneider, R., and Seeger, B. (1990). The R*-tree: An efficient and robust access method for points and rectangles. In *ACM SIGMOD International Conference on Management of Data*, pages 322–331.

Carniel, A. C., Ciferri, R. R., and Ciferri, C. D. A. (2016a). Experimental evaluation of spatial indices with FESTIval. In *Brazilian Symposium on Databases - Demonstration Track*, pages 123–128.

Carniel, A. C., Ciferri, R. R., and Ciferri, C. D. A. (2016b). The performance relation of spatial indexing on hard disk drives and solid state drives. In *Brazilian Symposium on GeoInformatics*, pages 263–274.

Carniel, A. C., Ciferri, R. R., and Ciferri, C. D. A. (2017a). A generic and efficient framework for spatial indexing on flash-based solid state drives. In *European Conf. on Advances in Databases and Information Systems*.

Carniel, A. C., Silva, T. B., Bonicenha, K. L. S., Ciferri, R. R., and Ciferri, C. D. A. (2017b). Analyzing the performance of spatial indices on flash memories using a flash simulator. In *Brazilian Symposium on Databases*.

Emrich, T., Graf, F., Kriegel, H.-P., Schubert, M., and Thoma, M. (2010). On the impact of flash SSDs on spatial indexing. In *Int. Workshop on Data Management on New Hardware*, pages 3–8.

Gaede, V. and Günther, O. (1998). Multidimensional access methods. *ACM Computing Surveys*, 30(2):170–231.

Greenel, D. (1989). An Implementation and Performance Analysis of Spatial Data Access Methods. In *Int. Conf. on Data Engineering*, pages 606–615.

Güting, R. H. (1994). An introduction to spatial database systems. *The VLDB Journal*, 3(4):357–399.

Guttman, A. (1984). R-trees: A dynamic index structure for spatial searching. In *ACM SIGMOD International Conference on Management of Data*, pages 47–57.

Jin, P., Xie, X., Wang, N., and Yue, L. (2015). Optimizing R-tree for flash memory. *Expert Systems with Applications*, 42(10):4676–4686.

Luo, L., Wong, M. D. F., and Leong, L. (2012). Parallel implementation of r-trees on the gpu. In *Asia and South Pacific Design Automation Conf.*, pages 353–358.

Lv, Y., Li, J., Cui, B., and Chen, X. (2011). Log-Compact R-tree: An efficient spatial index for SSD. In *International Conference on Database Systems for Advanced Applications*, pages 202–213.

Oosterom, P. V. a. N. (2005). Spatial Access Methods. In Longley, P. A., Goodchild, M. F., Maguire, D. J., and Rhind, D. W., editors, *Geographical Information Systems: Principles, Techniques, Management and Applications*, pages 385–400. 2nd edition edition.

Rigaux, P., Scholl, M., and Voisard, A. (2001). *Spatial databases: with application to GIS*. Morgan Kaufmann, 1st edition.

Sarwat, M., Mokbel, M. F., Zhou, X., and Nath, S. (2013). Generic and efficient framework for search trees on flash memory storage systems. *GeoInformatica*, 17(3):417–448.

Wu, C.-H., Chang, L.-P., and Kuo, T.-W. (2003). An efficient R-tree implementation over flash-memory storage systems. In *ACM SIGSPATIAL International Conference on Advances in Geographic Information Systems*, pages 17–24.

32th Brazilian Symposium on Databases

dbbio

October 2nd to 5th, 2017
Uberlândia – MG – Brazil

DATABASES MEET BIOINFORMATICS WORKSHOP

PROCEEDINGS

Promotion

Brazilian Computer Society – SBC
SBC Special Interest Group on Databases

Organization

Universidade Federal de Uberlândia – UFU

DBBIO Program Chair

Kary Ocaña, LABINFO/LNCC

Editorial

In the last years, there has been a revolution in the way biology research is driven by the intensive use of computing resources from supercomputers to high-performance databases. This new form of biology research is called bioinformatics which applies systematic and quantitative methods to the analysis of biological systems. Since one of the big questions in biology is to decipher the genomic information from a real live human (or any organisms), biologists can work with several data types that are collected systematically from the entire biological research community; for constructing models of how genes work together in every genome or inferring the thousands of biochemical functions they carried out.

Data sharing, integration, and annotation are essential to ensure the reproducibility of the analysis and interpretation of the experimental findings. Then, bioinformaticians and computer scientists have to sum experiences and also interact with experimental biologists for enabling the biological data integration. This is a key aspect that contributes to the success and future directions in multidisciplinary research.

The DBBio workshop aims to collaborate with scientific community efforts by proposing a broad discussion forum on the issues involved in developing database and software infrastructure to support bioinformatics and computational biology, as a new platform for scientific research and experimentation.

The DBBio workshop serves as a forum to present and discuss new ideas, as well as to establish new partnerships between research and development scientists in the areas of Database and Bioinformatics. In this way, the integration is expected in the event, giving opportunities for students and researchers from different areas to contribute to the workshop and new researchers have contact to those areas.

TOPICS OF INTEREST

- Foundation and practical application for integrating Bioinformatics and Databases sciences
- Design, implementation, and integration of biological databases
- Data-driven computational support of the foundations of molecular biology, from signal detection, sequence analysis, genomic assembly, and processing information to statistical analysis
- Data models for bioinformatics
- Data integration in bioinformatics
- Distributed databases in bioinformatics
- Semantic web techniques applied in bioinformatics

Kary Ocaña, LABINFO/LNCC
DBBIO Program Chair

32nd Brazilian Symposium on Databases

October 2nd to 5th, 2017
Uberlândia – MG – Brazil

Promotion

Brazilian Computer Society – SBC
SBC Special Interest Group on Databases

Organization

Universidade Federal de Uberlândia – UFU

SBBD Steering Commitee

Agma Juci Machado Traina, USP
Bernadette Lóscio, UFPE
Caetano Traina Jr., USP
Carmem Hara, UFPR
Javam Machado, UFC
Mirella M. Moro, UFMG
Vanessa Braganholo, UFF

SBBD 2017 Commitee

Steering Committee Chair
Javam Machado, UFC

Local Organization Chairs
Maria Camila N. Barioni, UFU and Humberto L. Razente, UFU

Program Committee Chair
Carmem S. Hara, UFPR

Short papers Chairs
Bernadette Lóscio, UFPE and Damires Souza, IFPB

Demos and Applications Session Chair
Daniel de Oliveira, UFF

Short Courses Chair
Vaninha Vieira, UFBA

Workshop on Thesis and Dissertations in Databases Chair
Carina Dorneles, UFSC

Tutorials Chair
Ana Carolina Salgado, UFPE

Thesis and Dissertation Contest Chair
Vânia Vidal, UFC

Workshops Chair
Fernanda Baião (UNIRIO)

Local Organization Committee

Maria Camila N. Barioni, UFU
Humberto L. Razente, UFU
José Gustavo de Souza Paiva, UFU
Marcelo Zanchetta do Nascimento, UFU
Elaine Ribeiro de Faria Paiva, UFU
João Henrique de Souza Pereira, UFU

Databases meet Bioinformatics Program Committee

Alexandre Lima (COPPE/UFRJ)
Ary Henrique Oliveira (UFT)
Daniel de Oliveira (IC/UFF)
Fabio Andre Porto (DEXL/LNCC)
Fernanda Baião (CCET/UNIRIO)
Fernanda Campos (DCC/UFJF)
Glauber Wagner (MIP/CCB/UFSC)
Guilherme Loss (LABINFO/LNCC)
Helena Cristina Gama Leitão (IC/UFF)
Jonas Dias (EMC Corporation)
Joseane Biso de Carvalho (LABINFO/LNCC)
Kary Ocaña (LABINFO/LNCC), chair
Luis Pacheco Arge (LABINFO/LNCC)
Luiz M. R. Gadelha Jr. (CENAPAD/LNCC)
Marcos Catanho (IOC/FIOCRUZ)
Marta Mattoso (COPPE/UFRJ)
Raquel Lopes (INCA)
Regina Braga (DCC/UFJF)
Sergio Lifschitz (PUC-Rio)
Sergio Manuel Serra da Cruz (UFRRJ)

Table of Contents (DBBIO)

Bio-SGBD: precisamos?

Sergio Lifschitz (PUC-Rio)

A grande área de pesquisa da biologia computacional tem apresentado desafios envolvendo o armazenamento, gestão e o acesso aos dados oriundos de pesquisas na área de ciências biológicas e afins. Há uma alta diversidade nos tipos de dados, como por exemplo, conjuntos de imagens tridimensionais, sequências de nucleotídeos de tamanhos variados, e resultados de montagens de fragmentos. Podemos citar também um aumento considerável do volume de dados biológicos, que são disponibilizados em alguns gigabytes para genomas de seres humanos, até vários petabytes nas investigações de micro-arrays. Também é um fato que a frequência de atualizações nos bancos de dados, por conta da utilização massiva das tecnologias NGS – *Next-generation Sequencing*, cresce exponencialmente. Ou seja, um contexto de pesquisas similar ao que se costuma definir por Big Data. Após mais de 25 anos de pesquisas científicas na área de bioinformática, poderia-se esperar o surgimento de algum tipo de gerenciador de bancos de dados biológicos que viesse a atender as demandas na área. Na prática, encontram-se apenas propostas ou protótipos de SGBDs que atendem parcialmente os requisitos de eficácia e eficiência existentes. Nesta palestra pretende-se discutir alguns dos principais desafios da bioinformática, específicos para a comunidade de bancos de dados, que permitam reflexão sobre as soluções especializadas, as adaptações de SGBDs (relacionais e não-relacionais) existentes e as expectativas de visão do futuro.

Integrated Visualization of Disease-Ancestry Relationships with DANCE

Gilderlanio S. Araújo[1], Paula Jennifer dos Santos[2], Eduardo M. Tarazona Santos[1], Maíra R. Rodrigues[3]

[1] Laboratório de Diversidade Genética (LDGH), Instituto de Ciências Biológicas – PPGI em Bioinformática. [2] Departamento de Ciências da Computação - Universidade Federal de Minas Gerais, Belo Horizonte - MG, Brasil

[3]Laboratório de Biologia Computacional e Bioestatística, Faculdade de Ciências Médicas e Instituto de Matemática, Estatística e Ciência da Computação UNICAMP, Campinas - SP, Brasil

[1]gilderlanio@gmail.com

Abstract. *DANCE is a network-based approach and web-tool that integrates, summarizes and allows visualization of data from two major sources of genetic data, which stores genetic associations with complex phenotypes (traits and diseases) and population genetic diversity data. It presents the genetic architecture of complex phenotypes in a cross-ethnic view for highlighting possible influences of genetic variation among populations in disease development and progression. In this study, we present network properties of DANCE networks and new features for exploring the genetic architecture of complex phenotypes considering the genetic variability between broad continental populations. To the best of our knowledge, DANCE is the first approach to provide integration of genetic-disease associations with allele frequency differentiation and linkage disequilibrium data. DANCE is available online at www.ldgh.com.br/dance.*

Resumo. *DANCE é uma abordagem baseada em redes e uma ferramenta web que integra, sumariza e permite a visualização de dados de duas principais fontes de dados genéticos, que armazena associações genéticas com fenótipos complexos (características e doenças) e dados de diversidade genética populacional. DANCE apresenta a arquitetura genética de fenótipos complexos em uma visão étnica para destacar possíveis influências de variações genéticas entre as populações no desenvolvimento e progressão de doenças. Neste estudo, apresentamos propriedades das redes disponíveis na ferramenta DANCE e novos recursos de visualização para explorar a arquitetura genética de fenótipos complexos considerando a variabilidade genética entre populações continentais. Ao nosso conhecimento, DANCE é a primeira abordagem que proporciona integração entre associações genéticas e doenças complexas com a diferenciação da frequência de alelos e dados de desequilíbrio de ligação. DANCE está disponível online em www.ldgh.com.br/dance.*

1. Introduction

A challenge for modern genetics is identifying functional genetic variants underlying molecular processes that confer risk to complex diseases or traits and developing pharmaceutical therapies for clinical treatments that are effective across populations. This is because it can happen that disease-causing genetic variants have different frequencies across populations and increases the prevalence of such diseases in specific populations, but not in others. Also, these variants can influence an

individual's response to treatment such that it can be more or less effective than expected.

Thus, it is important to know not only the genetic variants associated with diseases, but also if there are influences of population-specific variation in these associations. Although cross-ethnic Genome-Wide Association Studies (GWAS) have identified causative genes associated with complex phenotypes, such as diabetes and obesity, considering the genetic data of several populations, the European population is still over-represented in these studies. Specifically, only 20% of participants in GWAS are Asians and Africans descents [Popejoy, 2016] and admixed populations, such as Brazilian population that is derived from Native Americans, Africans and Europeans [Kehdy, 2015] are sub-represented.

Several types of human disease networks emerged to identify genetic variants that act under the progression of diseases. Network approaches have been playing an essential role in assisting the analysis of genetic association data from different points of view. However, current implementations of the human disease networks disregard genetic variability between populations and represent a single view of genetic associations with phenotypes [Goh, 2007; Darabos, 2013].

To address this issue, we developed DANCE (Disease-ANCEstry Networks) as a network-based approach to organize information on human diseases and genetic variants, in particular Single Nucleotide Polymorphisms (SNP), and its frequencies among populations [Araújo, 2015]. It is implemented as an interactive web tool (www.ldgh.com.br/dance) and its current version allows the visualization of joint information of human phenotypes and their associated SNPs based on GWAS data. DANCE includes the distribution of SNP frequencies in different continental populations (African, European and Asian) and LD between SNPs.

Biological systems are complex and difficult to understand when initially analyzed as a whole. A more suitable approach is first to understand the role of each element in the system and then to observe their connections. Thus, when a biological system is modeled as a network, the characteristics of its elements and their links can be extracted by applying a plenty of statistical network properties. These properties, in turn, can be used to explain the origins and dynamics of biological processes in specific domains of study. The network properties of nodes and edges, as well as centrality metrics may help in understanding the network as a whole, and they have been used to identify the most influential elements within a graph. Considering this perspective, we present properties of three networks in DANCE, an update of DANCE data, and how the genetic architecture of complex diseases can be queried and visualized with the assistance of DANCE.

2. Related Work

Although human phenotype networks have been proposed elsewhere in the literature, they have a number of limitations in comparison to DANCE. Particularly, the first human phenotype network [Goh, 2007] is biased, since genetic associations between genes and phenotypes were based on Mendel's principles, focusing on monogenic disorders cataloged in OMIM [Hamosh, 2005]. Other research team [Darabos, 2013] published in 2013 an integrative approach to infer human phenotype networks based on genetic associations between SNPs and phenotypes for a set of approximately six thousands SNPs and limits the genetic data for the initial phase of the HapMap data [Gibbs, 2003], which presents few individuals (n = 270) for a diverse set of population genetic data. An intrinsic feature to all the related approaches above is that

none considers the genetic variability for a large set of populations. Then, accounting for genetic population data when designing human disease networks have received no attention. On the other hand, DANCE presents data from 2504 individuals and based on genotype genome-wide data for these individuals we compute frequencies and genetic diversity for three broad populations and a large set of risk variants.

3. Data Extraction

DANCE's dataset combines information from two existing public databases: (i) SNPs associated with complex phenotypes, reported in the NHGRI GWAS Catalog [Welter, 2014] and (ii) SNP genotypes from Europeans, Africans and Asians populations stored in the 1000 Genomes Project Phase 3 [1KGP Consortium, 2015]. Currently, the GWAS Catalog contains 2088 studies with PubMed numbers regarding genetic associations for 721 complex phenotypes and it is mapped to Genome Assembly GRCh38.p5 and dbSNP Build 146.

The process of data merging follows two main steps: first, the GWAS Catalog was manually curated to remove semantic redundancies on phenotype labels. As a result, we produced a map of label changes as an input to a function for relabeling the original phenotypes aiming to standardize data; and second, we merged the available SNPs for African, European and Asian populations to find the overlap between them. In the latter step we found around 77 million SNPs with overlap in the three populations, and these are called 1KGP SNPs hereafter. At last, we performed a second merge to find the overlap between the GWAS Catalog phenotype-associated SNPs and the 1KGP SNPs. The complete process resulted in 1521 studies reporting 11.898 SNPs with clinical impact significance associated to 721 phenotypes.

After merging data, we quantified three elements that compose the genetic architecture of complex phenotypes, which are: a) the allele frequency for each SNP based on genotypes for African, Europeans and Asians b) the allele frequency differentiation (F_{ST}) [Holsinger, 2009] between African and Europeans, African and Asians, and European and Asian populations, and c) the linkage disequilibrium (LD) that represents a non-random association of alleles at different loci for pairs of SNPs. The LD is property that has been used to understand evolutionary process, allele differentiation between populations, and mainly to map genes with complex diseases by GWAS [Slatkin, 2008].

4. Modelling Networks

The curated DANCE dataset was used to create the SNP-Disease Network (SDN). The SDN allows exploring the influences of SNPs, that are risk-allele variants and their genetic population landscape associated with phenotypes. In the first version, DANCE proposed the SDN as a bipartite network G=(V, E) where the node set V can be partitioned into two sets: (A) phenotypes and (B) risk-alleles. Phenotypes are connected to SNPs if there is a known association reported by a GWAS, composing the edge set E. As such, different phenotypes interconnect through shared SNPs. The ancestry information is represented in a basic level as a property of the SNP risk-allele node and comprises its frequency in a determined population, as given by the SNP's molecular profile. Also, the measures of genetic variability were computed by F_{ST} between pairs of populations are represented as properties of SNP nodes. As result, the SDN with SNPs and phenotypes has 12.619 nodes, which 11.898 are SNPs and 721 phenotypes.

Currently, with the publication of genotype data from 2504 individuals, the

SDN network was remodelled to accommodate correlations measured for pairs of SNPs for Africans, Europeans and Asians from 1KGP. In order to add LD edges in the SDN we calculate the pairwise linkage disequilibrium based on the r^2 equation implemented in Plink v1.9 [Purcell, 2007]. Thus, including LD edges, the SDN becomes a multi-graph bipartite network G=(V, E), where the relations between phenotypes and SNPs remains and different phenotypes interconnect through shared SNPs or are indirectly connected by SNPs that are connected by LD. Considering the genetic variability we modelled three population-specific SDN networks: a) The SDN-African, which includes the LD edges from Africans, and b) the SDN-European, which includes the LD edges from Europeans, and c) the SDN-Asian, which includes the LD from Asians. Our networks are represented graphically as in Figure 1.

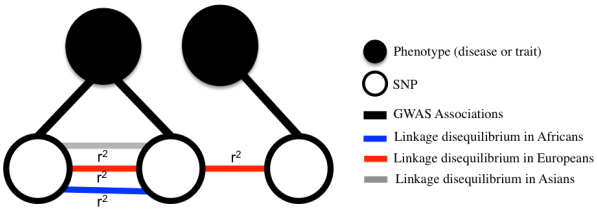


Figure 1. Graphical representation of the SNP-Disease Network with LD edges in Africans and European populations.

5. Connectivity in SDN

In order to investigate global relations between SNPs and complex phenotypes we computed properties of network connectivity. Essentially, the connectivity of a network can be explored by calculating the number of components, diameter, the average path length, average number of neighbours, and partners of multi-edges. For instance, we discussed the networks related to Europeans and Africans, due to our particular interest in the admixture process of Brazilian population [Kehdy, 2015] and limited writing space. The results of network properties analysis are summarized in the Table 1 for the SDN, SDN-African and SDN-European networks.

Table 1 - Network properties of SNP-Disease Networks.

Network Properties	Networks		
	SDN	SDN-African	SDN-European
#Nodes (SNPs)	11898	11898	11.898
#Nodes (Phenotypes)	721	721	721
#Edges (GWAS)	13.514	13.514	13.514
#Edges (Linkage Disequilibrium)	-	1.846	4.468
#Total of edges	13.514	15.360	17.982
#Components	360	296	261
Diameter	18	19	19
Avg. Path Length	6.7	6.7	6.5
Avg. Number of Neighbours	2.1	2.4	2.8

The network properties of these networks and biological implications is discussed, as follows:

- **Components.** The number of components helps us to verify if each human phenotype presents a tendency to have a distinct genetic architecture. Also, high numbers of components suggests lower connectivity. The SDN presents the most disconnected network with 360 components. The SDN-African presents the

second most disconnected network with 296 components and third the SDN-European presents 261 components. This high number of components reflects that some phenotypes have their particular and unique genetic architecture, that is specific phenotypes do not share risk-alleles with others phenotypes. The LD edges implicate in changes in structure of the networks, and it is viable to quantify these changes and impact on the connectivity of nodes. The presence of LD edges, in the European population, results in decreases of the number of components. In this case, smaller components of the networks were aggregated to the giant network component, thus reducing the number of components.

- **Diameter.** The diameter is calculated based on the distribution of the shortest paths between each node pair, and the maximum value of this distribution is determined as the diameter of the network. All networks of DANCE present high network diameter (>=18), representing low connectivity in the network.
- **Average Path Length.** The characteristic path length of the SDN network (around 6.5) also indicates tight relationships between risk-alleles and phenotypes.
- **Average number of neighbours.** This measure is the mean of the degree distribution for all nodes. Likewise, the low average number of neighbours, high values of the characteristic path length between the nodes, and the high diameter number of the network represent low connectivity. The average number of neighbours for the SDN-African and the SDN-European is equal to 2.4 and 2.8 respectively. Adding LD edges results in a slight increase in the number of neighbours for the population-specific networks.
- **Partners of multi-edge.** After an overlap procedure of the SND-African with the SDN-European we found a network with all LD edges and we observed that 1.673 SNPs are partners of multi-edges. SNPs partners of multi-edges mean that LD links them in both African and European populations. This is an important consideration for functional variant analysis that aims to identify additional causal SNPs in a cross-ethnic view.

6. Centrality and genetic overlap.

The network properties of variants may shed light on biological mechanisms related to complex phenotypes and are helpful for uncovering causal variants [Barabasi, 2011]. From this principle, we consider that SNPs that show many observed associations with complex phenotypes may have as strong functional biological impact. The degree distribution for SNPs indicates that few of them relate to more than two phenotypes (see Figure 2A), around 93% of SNPs relate to only one phenotype. Most SNPs are related to F_{ST} values less or equal to 0.25 and low degree ($d < 5$) in the network as show in Figure 2B. We found four SNPs highly differentiated ($F_{ST} > 0.8$) with lower degree: a) the SNP rs2814778 is associated with neutrophil count in HIV-infection, b) rs16891982 and rs1834640 is related to pigmentation traits (eye, hair, and skin) and rs1426654 with body mass index. The SNPs with highest degree (hubs in the network), that is, the variants with high numbers of associations are: rs1260326 (in gene GCKR – Glucokinase Regulator), rs180056 (in gene HFE - Hereditary Hemochromatosis), and rs3184504 (in gene SH2B3 - SH2B Adaptor Protein 3). The three SNPs are associated with 21, 15, 12 phenotypes respectively and are in exonic gene regions. These phenotypes are classified in metabolic, hematological and cardiovascular medical classes.

In view that some presented SNPs are shared between phenotype, estimation of genetic comorbidity is important to define genetic risk profiles with common

factors that intermediate the biological processes influencing the progression of diseases and traits. We explored the genetic overlap between pairs of phenotypes taking advantages of the DANCE approach. We identified the set of SNPs for each phenotype and computed the extent of shared SNPs between pairs of phenotypes with the set operations of intersection. The results of this analysis are showed in the Figure 2C, that presents the intersection of SNPs between phenotypes. We found that complex psychiatric phenotypes including autism, schizophrenia and bipolar disorder share between them more than 50 SNPs and also this fact repeats for triglycerides and cholesterol, which are metabolic traits.

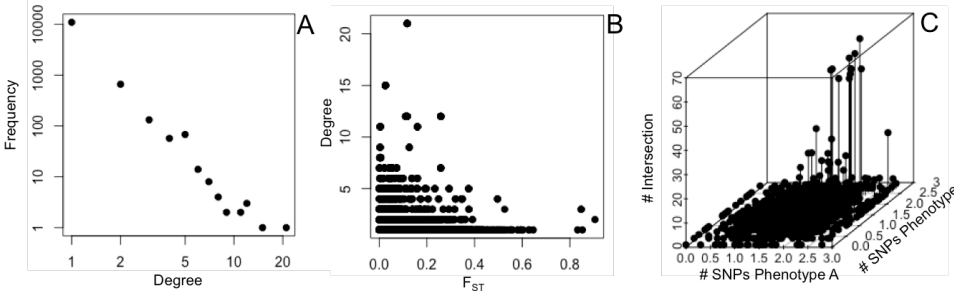


Figure 2 - (A) Distribution of degree for SNPs considering the SNP-Disease Network (SDN). (B) Distribution of degree and F_{ST} (genetic diversity index). (C) Genetic overlap between phenotypes.

7. Data visualization: an example.

DANCE is composed of a network visualization component and data filters that is implemented as a web tool to query association data and population genetic data. All the data were represented in networks structures, and also the data to construct were available in the form of tables. The web interface of DANCE is distributed in four web pages: a) Home, b) Networks, c) Documents and Tutorial, and d) Data. DANCE is available online at www.ldgh.com.br/dance.

DANCE allows identifying and interactively visualizing the sets of SNPs associated with a list of phenotypes or SNPs in specific genes. The DANCE view component is totally interactive and it shows a landscape of SNP frequencies in populations or their pairwise population differentiation (measured by F_{ST}), also currently DANCE allows querying linkage disequilibrium data for African, European and Asian populations. The web architecture of DANCE is composed of a query controller module, which provides the communication between the web interface and the server datasets, and SNP-Disease profile manager module, which updates periodically the server datasets with data from GWAS Catalog and 1000 Genomes Project.

To illustrate the use of web interface we present on the Figure 3 the query result for consulting the genetic architecture of skin pigmentation and related pigmentation traits. The network view component of DANCE shows that SNPs associated with skin pigmentation are shared with three others phenotypes related to pigmentation, such as hair, eye and freckles. On the network, the phenotypes are represented as a blue node and the genetic diversity between African and Europeans (F_{ST}) is represented in the red gradient for each SNP. In the data filters panel we choose the SNP-Disease Network, after we choose to query the SNPs and their frequency in Africans, ranging from 0 to 1. For this query the effect size ranges from 0 to 30, moreover the effect size filter and representation in networks is under construction. We choose to view the LD in Africans. At last, set the phenotype

parameter as skin pigmentation and the neighborhood as 1.

8. Final discussions and conclusions.

The implementation of the DANCE as a web tool is a proposal to fill the gap in tools that allows to explore and to construct knowledge concerning the genetic architecture of complex diseases. DANCE is a tool that automates the process of integration of genome-wide genetic association data and population genetic variability data for a broad set of populations, that comprise African, European and Asian continents. The DANCE approach allows geneticists and related professionals to build knowledge about complex phenotypes and the implications of clinical variants, and it is relevant for geneticists and epidemiologists working with admixture mapping studies and GWAS replications.

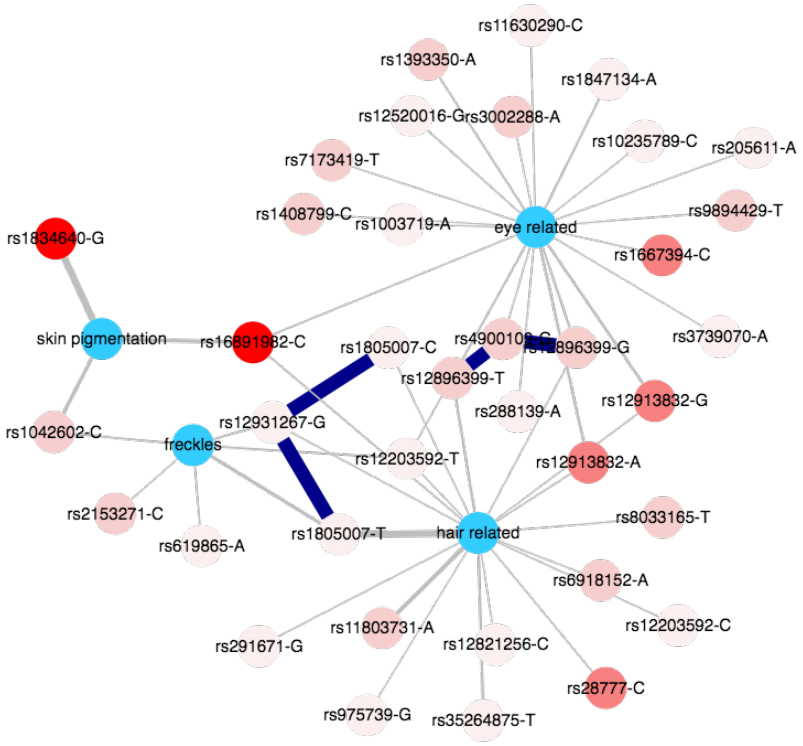


Figure 3. DANCE Analysis for Skin Pigmentation and related pigmentation phenotypes. The diversity is indicated by tones of red color. Higher tones indicate more diversity between African and European populations.

The network-based visualization methods implemented on DANCE give a valuable landscape view of genetic architecture, which includes LD, allele frequencies and frequency differentiation indexes. An ongoing work is modeling the effect size value of each SNP (odds ratio or beta values), which allows identifying protective or causative SNPs in the genetic architecture of complex diseases.

In the context of analysis of genetic variability, the current version of DANCE comprises genetic variability data for African, European and Asian populations. For further analysis we aim to include data from mixed populations, such as the admixed American and Native American populations. Integration of mixed population data will help to analyze the landscape of differential genomic diversity for under represented populations in genome-wide studies.

Centrality metrics as bipartite clustering and bipartite closeness do not fit our DANCE networks [Araújo, 2017]. The number of smallest components in the

network create a bias in the closeness and clustering distribution that tends to 1. Other metrics or creation of new that considers the distance between nodes will be used to investigate the real role of risk SNPs in further studies.

9. References.

[Araújo, 2015] Araújo, G.S., et al. (2015) Integrating, summarizing and visualizing GWAS-hits and human diversity with DANCE (Disease-ANCEstry networks). Bioinformatics, v. 32, n. 8, p. 1247-1249.

[Araújo, 2017] Araújo, G.S., (2017) Network-based Methods for Analyzing the Genetics of Human Complex Diseases. Thesis. Universidade Federal de Minas Gerais. Belo Horizonte. Brasil.

[Barabasi, 2011] Barabasi, A.-L., et al. (2011) Network medicine: a network-based approach to human disease. Nature Review Genetics 12, 56–68.

[Darabos, 2013] Darabos, C., et al. (2013) Inferring human phenotype networks from genome-wide genetic associations. In European Conference on Evolutionary Computation, Machine Learning and Data Mining in Bioinformatics, 23–34, Springer, 2013.

[Gibbs, 2003] Gibbs, R.A., et al. (2003) The International HapMap Project. Nature 426, 789–796.

[Goh, 2007]Goh, K.-I. et al. (2007) The human disease network. Proceedings of the National Academy of Sciences of the United States of America 104, 8685–8690.

[Hamosh, 2005] Hamosh, A., et al (2005). Online mendelian inheritance in man (OMIM), a knowledgebase of human genes and genetic disorders. Nucleic acids research 33, D514–D517.

[Holsinger, 2009] Holsinger, K. E., et al. (2009) Genetics in geographically structured populations: defining, estimating and interpreting F_{ST}. Nature reviews. Genetics 10, 639–650.

[Kehdy, 2015] Kehdy, F.S.G., et al. (2015) Origin and dynamics of admixture in Brazilians and its effect on the pattern of deleterious mutations. Proceedings of the National Academy of Sciences 112, 8696–8701 (2015).

[Popejoy, 2016] Popejoy, A.B. (2016). Genomics is failing on diversity. Nature 538, 161.

[Purcell, 2007] Purcell, Shaun, et al. "PLINK: a tool set for whole-genome association and population-based linkage analyses." *The American Journal of Human Genetics* 81.3 (2007): 559-575.

[Slatkin, 2008] Slatkin, M. (2008) Linkage disequilibrium–understanding the evolutionary past and mapping the medical future. Nature Reviews Genetics. 9, 477–485.

[Welter, 2013] Welter, D., et al. (2013) "The NHGRI GWAS Catalog, a curated resource of SNP-trait associations." *Nucleic acids research* 42.D1: D1001-D1006.

[1KGP Consortium, 2015] 1000 Genomes Project Consortium. (2015) "A global reference for human genetic variation." *Nature* 526.7571: 68.

Uso de bancos de dados NoSQL para gerenciamento de dados em *workflow* de Bioinformática

Fernanda Hondo[1], Polyane Wercelens[1], Waldeyr da Silva[1,2], Iasmini Lima[2], Klayton Castro[1], Ingrid Santana[1], Gabriel de Araujo[1], Aleteia Araujo[1], Maria Emília Walter[1], Maristela Holanda[1]

[1] Universidade de Brasília (UnB)

[2]Instituto Federal de Goiás (IFG)

Abstract. *Bioinformatics workflows generate massive amounts of data and an efficient management of the data and its provenance is a challenge. The use of the provenance of data has brought remarkable benefits to the research in Bioinformatics, allowing a greater accuracy in the analyses due to the reproducibility and the refinement that it provides to the experiments. Bioinformatics workflows demand scalability and performance. In such context, the use of non-relational database system (NoSQL) is becoming increasingly common. This paper presents a comparative study between the NoSQL databases Cassandra, MongoDB and OrientDB DBMS regarding the management of data and provenance, both collected in the execution of a DNA assembly workflow.*

Resumo. *Workflows em Bioinformática geram um grande volume de dados e o gerenciamento de tais dados e de sua proveniência é um desafio. A proveniência de dados tem trazido grandes benefícios às pesquisas em Bioinformática, permitindo maior acurácia nas análises devido à reprodutibilidade e ao refinamento que proporciona aos experimentos. Workflows de Bioinformática demandam escalabilidade e desempenho, contexto no qual o uso de sistemas gerenciadores de banco de dados não relacionais (NoSQL) vem se tornando cada vez mais comum. Este artigo apresenta um estudo comparativo entre os banco de dados NoSQL Cassandra, MongoDB e OrientDB em relação ao gerenciamento dos dados e da proveniência de execuções de um workflow de montagem de DNA.*

1. Introdução

A Bioinformática é uma área interdisciplinar que busca resolver problemas da Biologia Molecular utilizando ferramentas e métodos de Computação, Matemática e Estatística. Muitas dessas soluções estão associadas à execução de *workflows*. Um *workflow* é um conjunto de atividades que envolvem a execução coordenada de tarefas múltiplas realizadas por diferentes entidades de processamento [Georgakopoulos et al. 1995]. *Workflows* permitem modelar, gerenciar e coordenar a execução de experimentos científicos que envolvem diversas fases, cada uma com características, propósitos e ordem de execução particulares [Mattoso et al. 2008] [Rosa et al. 2016].

Diversas áreas da Biologia Molecular utilizam *workflows* em seus experimentos científicos [Boekel et al. 2015], nos quais frequentemente são processados dados oriundos de projetos genoma, transcritoma, metaboloma, entre outros [Wolstencroft et al. 2013] [Kohl et al. 2014]. Cada execução de um workflow ci-

entífico de Bioinformática pode gerar um grande volume de dados, os quais devem ser armazenados para novas execuções, análises ou confirmações de resultados.

A literatura apresenta diversos sistemas de armazenamento de dados e recentemente novos modelos de bancos de dados, como os Not Only SQL (NoSQL) têm sido definidos. Neste contexto de novas tecnologias de banco de dados e demanda por persistência da proveniência de dados, o objetivo deste trabalho é verificar o comportamento de diferentes bancos de dados NoSQL para gerenciar os dados de um típico *workflow* de Bioinformática e sua proveniência.

Este artigo está dividido em Seções. Na Seção 2 são apresentados conceitos de *workflows* de Bioinfomática, proveniência e NoSQL, bem como trabalhos relacionados. O método e o ambiente computacional utilizados são apresentados na Seção 3. Os resultados obtidos estão na Seção 4, seguida da Seção 5, onde são apresentadas as conclusões.

2. Dados em *workflows* de Bioinformática

2.1. Montagem de fragmentos

Um dos problemas ao qual a Bioinformática se dedica é a montagem de fragmentos de DNA oriundos do sequenciamento de alto desempenho. Esses fragmentos chamados *reads*, são *strings* de um alfabeto que representa o DNA ou o RNA. A partir de alinhamentos das *reads*, a montagem obtem sequências contíguas (*contigs*) que representam o DNA original da amostra [Zerbino and Birney 2008]. Na atualidade, basicamente três estratégias de montagem estão em uso [Bleidorn 2017]: *greedy algorithms* [Zhang et al. 2000], *overlap-layout-consensus* [Li et al. 2012] e grafos construídos com *k-mer* [Li et al. 2012]. A montagem de fragmentos pode utilizar um genoma de referência, neste caso as *reads* são alinhadas contra um genoma de organismo filogeneticamente próximo ao organismo do qual provêm as *reads*. A montagem sem um genoma de referência é chamada de montagem *de novo* [Bleidorn 2017].

Um típico *workflow* para montagem de fragmentos é normalmente composto por 3 fases sequenciais: Filtragem, Montagem e Análise [Haas et al. 2013]. A característica sequencial dessas fases demanda que o resultado de saída de cada fase seja utilizado como entrada para a fase seguinte. Na fase de Filtragem as *reads* são filtradas utilizando de parâmetros de qualidade definidos na pesquisa [Guo et al. 2013]. A fase de Montagem utiliza as *reads* filtradas para alinhá-las contra um genoma de referência a fim de montar uma sequência original comumente chamada de sequência consenso [Bleidorn 2017]. Na ausência de uma referência ou por decisão de projeto, é realizada uma montagem *de novo*. Na fase de Análise são realizadas verificações do resultado obtido na execução do *workflow* a fim de validar a hipótese inicial do experimento [Guo et al. 2013].

2.2. Modelo de Proveniência PROV-DM

O termo proveniência de dados diz respeito a origem ou procedência de um determinado dado. Em Sistemas Gerenciadores de Workflows Científicos (SGWC), a proveniência de dados tem sido aplacad para auxiliar no entendimento do ciclo de vida de *workflows* dessa natureza [Mattoso et al. 2008].

Provenance Data Model (PROV-DM) é um modelo genérico de representação de proveniência de dados. O PROV-DM descreve agente (atores), entidades e atividades en-

volvidas na geração de um dado [Moreau and Missier 2013] através de um grafo direcionado. O PROV-DM dispõe de 2 tipos para descrever a proveniência: Atividade (*Activity*) - processo executado para gerar um objeto; e Entidade (*Entity*) - utilizada para modelar qualquer objeto de proveniência. O tipo *Entity* é dividido em 4 subtipos: Agente (*Agent*), entidade que influencia, direta ou indiretamente, a execução das atividades; Coleção (Collection), representa um conjunto de Entidades; Plano (Plan), representa um conjunto de ações que um Agente deve seguir para chegar a um objetivo; Relato (Account), representa um conjunto de informações que compõem um grafo de proveniência.

O grafo de proveniência é acíclico e direcionado. Seus nós podem representar objetos, como arquivos, programas e pessoas e suas arestas representam a dependência entre os objetos. A Figura 1 mostra algumas das representações de nós utilizadas no grafo e suas possíveis relações.

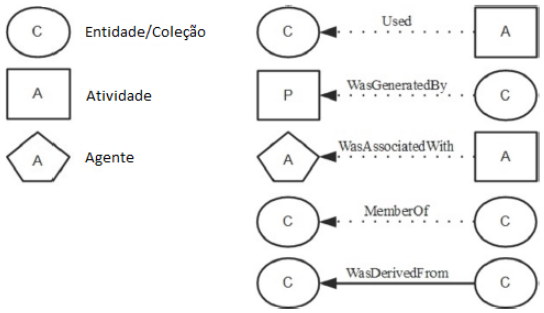


Figura 1. PROV-DM: tipos Atividade, Entidade e Agente e seus relacionamentos.

2.3. NoSQL

Os bancos de dados NoSQL (*Not only SQL*) emergiram como uma alternativa aos tradicionais Sistemas Gerenciadores de Bancos de Dados Relacionais (SGBDR). Eles são propostos como soluções escaláveis, contam com processamento distribuído, proporcionam alta disponibilidade e escalabilidade, são flexíveis e têm capacidade para armazenar dados estruturados e não estruturados. Existem quatro principais famílias de NoSQL [Corbellini et al. 2017]:

- Chave–valor: os dados são armazenados indexados por chaves, divididos em duas partes, onde cada valor está associado a uma chave única.
- Baseado em colunas: a estrutura de valores é definida como um conjunto de colunas predefinido.
- Baseado em documentos: os documentos armazenados são coleções de atributos e valores, podendo conter atributos multivalorados. Eles utilizam o conceito de chaves e valores, onde cada documento contém uma chave que o identifica.
- Grafos: os esquemas são representados por grafos direcionados ou não, em que os dados são armazenados nos vértices. Os relacionamentos são representados pelas arestas que também podem armazenar dados dependendo do banco de dados [Silva et al. 2016].

Alguns NoSQL são híbridos e implementam mais de uma família. O OrientDB, por exemplo, implementa as famílias Grafo, Chave–valor e Documento.

2.4. Trabalhos Relacionados

Em [de Paula et al. 2013] o modelo PROV-DM foi proposto para gerenciar a proveniência de dados em *workflow* de Bioinformática. O modelo PROV-DM permitiu armazenar as propriedades de cada execução de um *workflow* de Bioinformática, representando graficamente os grandes volumes de dados gerados nesses experimentos.

Em [Ferreira et al. 2014], um banco de dados relacional e um banco de dados NoSQL são comparados através da migração de dados de proveniência do PostgreSQL para o NoSQL Cassandra em um *workflow* de Bioinformática. Em [Aniceto et al. 2015] uma a análise do desempenho do NoSQL Cassandra, do PostgreSQL e do NoSQL MongoDB executando sobre dados biológicos é realizada. Em [Li et al. 2014] o *framework Provenance Lens* que fornece gerenciamento de proveniência em ambientes de nuvens e compara seu desempenho ao usar o MySQL, MongoDB e Neo4J foi definido. Em [Cheah et al. 2013] o Milieu, um *framework* focado na proveniência para experiências científicas com armazenamento em MongoDB é apresentado. [Chacko et al. 2015] especifica o *Data Foreign Wrappers* em um sistema de gerenciamento de proveniência chamado PERM para implementar a proveniência usando armazenamento com MongoDB.

Em [Fiannaca et al. 2016] é apresentado o *BioGraphDB*, um banco de dados de bioinformática capaz de integrar diferentes tipos de fontes de dados utilizando o OrientDB. Em [Bonnici et al. 2014] tem-se um banco de dados NoSQL denominado ncRNA-DB construído sobre o OrientDB capaz de integrar dados de RNA, DNA, proteínas e doenças. Em [Costa et al. 2017] o GeNNET é definido, uma plataforma integrada de análise que visa unificar *workflows* científicos com bancos de dados em grafo com o uso do Neo4J para armazenar resultados após a análise de transcritomas.

Com o propósito de expandir o conhecimento produzido na área, nosso artigo apresenta um estudo comparativo entre três diferentes tipos de NoSQL, um orientado a colunas (Cassandra), um orientado a documentos (MongoDB) e um híbrido (OrientDB usado como grafo) para gerenciamento de proveniência de dados de um *workflow* de Bioinformática. Adicionalmente, analisamos o desempenho na inserção e extração dos dados biológicos (dados brutos).

3. Método

O *workflow* em Bioinformática escolhido para análise foi o RNA-Seq do fungo *Aspergillus fumigatus* [Latgé 1999]. A escolha deste *workflow* deu-se porque o mesmo possui as fases típicas de filtragem, montagem e análise. Para os bancos de dados NoSQL, a análise foi aplicada no Cassandra (Colunas), no MongoDB (Documento) e no OrienteDB (Híbrido, Grafo e documento). A escolha do Cassandra e do MongoDB deve-se à sua expressividade na literatura relacionada. O OrientDB foi incluído com o objetivo de analisar as diferentes famílias como solução para o gerenciamento da proveniência e armazenamento dos dados de entrada (brutos) do *workflow*.

Para medir o desempenho no tempo de inserção e extração de arquivos, a primeira fase do *workflow*, a filtragem, foi repetida quatro vezes utilizando cada um dos bancos de dados NoSQL. Ao longo das quatro repetições da fase de filtragem, coletamos os tempos de inserção e extração dos arquivos em cada banco de dados. Os dados brutos são seis arquivos no formato FASTQ com tamanho médio de 1.33 GB cada, somando

aproximadamente 8 GB. Para garantir a isonomia dos testes de desempenho, as mesmas condições de execução foram estabelecidas para os três bancos de dados em um ambiente com seis máquinas virtuais (16GB de RAM, 200GB de armazenamento e 4 núcleos de CPU) provisionadas em uma nuvem privada.

Uma vez coletados os tempos, executamos o *workflow* completo para capturar e armazenar a proveniência dos dados. Todas as informações pertinentes a execução, como o agente do experimento, atividades por ele executadas em cada fase, arquivos utilizados e gerados, informações do provedor, ambiente e máquinas utilizadas por cada atividade, foram modeladas e inseridas nos diferentes NoSQL (Cassandra, MongoDB e OrientDB) de acordo com suas respectivas abordagens (orientada à colunas, a documentos e grafo) seguindo o modelo génerico ilustrado na Figura 2 baseado no PROV-DM.

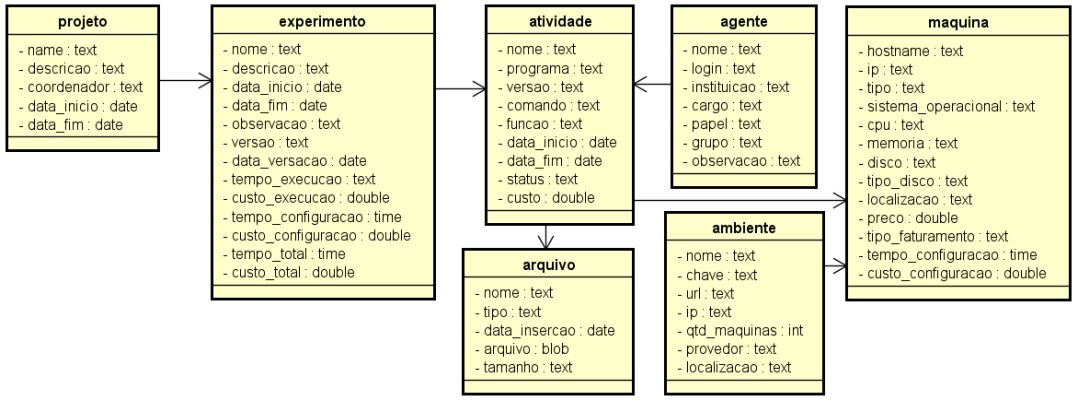


Figura 2. Modelo de dados genérico para proveniência.

4. Resultados

Os dados de proveniência em cada um dos bancos de dados NoSQL foram armazenados e capturados usando como base a modelagem genérica apresentada na Figura 2. Para validação do grafo de proveniência, realizamos diferentes consultas em cada banco de dados. É possível gerar grafos da execução total do *workflow* e também para cada fase executada. A Figura 3 apresenta a parte do grafo de proveniência da fase de filtragem do *workflow*. Na figura é possível verificar as entidades (arquivos FASTQ e FASTA), uma atividade (filtragem), um agente e seus relacionamentos. A figura foi gerada utilizando a API gráfica Prefuse [Heer et al. 2005] a partir de uma mesma consulta genérica, aplicada de acordo com a linguaguem de cada bancos de dados NoSQL.

Os tempos de inserção sofreram bastantes variações em relação aos tempos de extração. De maneira geral, todos os bancos de dados NoSQL demonstraram melhor desempenho na extração, onde houve um equilíbrio de desempenho entre o OrientDB e o MongoDB, ficando o Cassandra com o desempenho menos satisfatório. O MongoDB destacou-se no desempenho de inserção apresentando sempre o melhor tempo para todos os arquivos com média de 120s com margem de erro de 10s para mais ou para menos. O MongoDB foi também o NoSQL que manteve o maior equilíbrio nos tempos de inserção e extração, com variação de 20% entre esses dois tempos desconsiderando a margem de erro. Os resultados de desempenho nos tempos de inserção e extração podem ser observados no gráfico Figura 4.

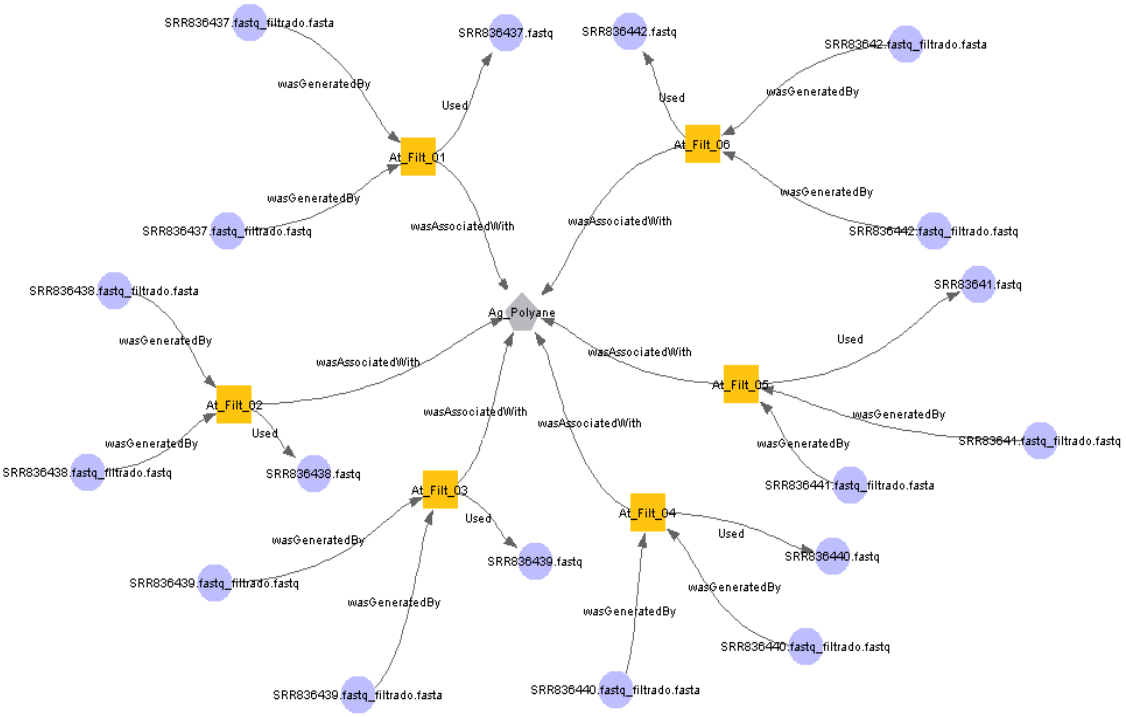


Figura 3. Grafo de proveniência da fase de filtragem do *workflow*.

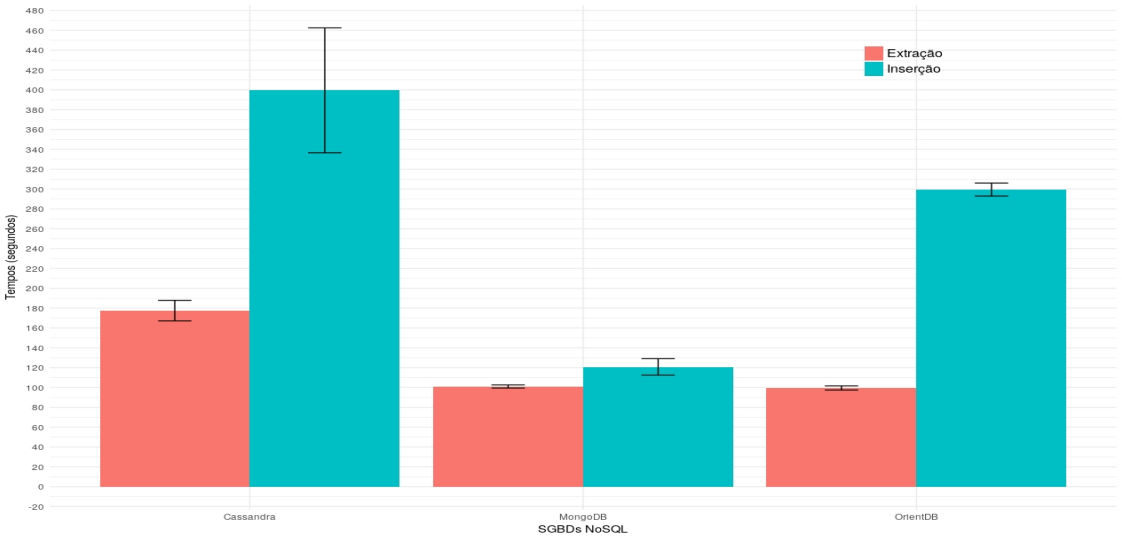


Figura 4. Média dos tempos de inserção e extração da execução do *workflow*.

5. Conclusão

Este artigo apresenta um estudo inicial sobre o uso do NoSQL para a gerência de dados de proveniência e armazenamento dos dados na execução de *workflows* científicos, especialmente em Bioinformática. Os NoSQL escolhidos foram o Cassandra, o MongoDB e o OrientDB, e analisamos o impacto dessas diferentes famílias de bancos de dados NoSQL no gerenciamento dos dados de proveniência e no armazenamento de dados biológicos.

Os resultados obtidos até aqui são encorajadores, pois o armazenamento de dados em NoSQL para gerenciar o armazenamento e a proveniência dos dados apresentaram

bons resultados. Desta forma, este trabalho aponta para melhorias da reprodutibilidade dos experimentos baseados em *workflows* de Bioinfomática. Entretanto, novos experimentos precisam ser realizados expandindo a diversidade de *workflows* e a quantidade de repetições por *workflow* para uma amostra mais abrangente de tempos de execução. Outro ponto a ser explorado é o uso de replicação dos bancos de dados para verificar seu comportamento num ambiente de processamento distribuído.

6. Agradecimentos

Os autores agradecem à UnB pelo apoio à realização deste trabalho, através da concessão de recursos. Fernanda Hondo agradece à CAPES pela concessão de bolsa de estudo.

Referências

Aniceto, R., Xavier, R., Guimarães, V., Hondo, F., Holanda, M., Walter, M. E., and Lifschitz, S. (2015). Evaluating the cassandra nosql database approach for genomic data persistency. *International journal of genomics*, 2015.

Bleidorn, C. (2017). Assembly and data quality. In *Phylogenomics*, pages 81–103. Springer.

Boekel, J., Chilton, J. M., Cooke, I. R., Horvatovich, P. L., Jagtap, P. D., Käll, L., Lehtiö, J., Lukasse, P., Moerland, P. D., and Griffin, T. J. (2015). Multi-omic data analysis using galaxy. *Nature biotechnology*, 33(2):137–139.

Bonnici, V., Russo, F., Bombieri, N., Pulvirenti, A., and Giugno, R. (2014). Comprehensive reconstruction and visualization of non-coding regulatory networks in human.

Chacko, A. M., Basheer, A. M., and Kumar, S. M. (2015). Capturing provenance for big data analytics done using sql interface. In *Electrical Computer and Electronics (UPCON), 2015 IEEE UP Section Conference on*, pages 1–6. IEEE.

Cheah, Y.-W., Canon, R., Plale, B., and Ramakrishnan, L. (2013). Milieu: Lightweight and configurable big data provenance for science. In *Big Data (BigData Congress), 2013 IEEE International Congress on*, pages 46–53. IEEE.

Corbellini, A., Mateos, C., Zunino, A., Godoy, D., and Schiaffino, S. (2017). Persisting big-data: The nosql landscape. *Information Systems*, 63:1–23.

Costa, R. L., Gadelha, L., Ribeiro-Alves, M., and Porto, F. (2017). Gennet: an integrated platform for unifying scientific workflows and graph databases for transcriptome data analysis. *PeerJ*, 5:e3509.

de Paula, R., Holanda, M., Gomes, L. S., Lifschitz, S., and Walter, M. E. M. (2013). Provenance in bioinformatics workflows. *BMC bioinformatics*, 14(11):S6.

Ferreira, G. R., Filipe Jr, C., and de Oliveira, D. (2014). Uso de sgbds nosql na gerência da proveniência distribuída em workflows científicos.

Fiannaca, A., La Rosa, M., La Paglia, L., Messina, A., and Urso, A. (2016). Biographdb: a new graphdb collecting heterogeneous data for bioinformatics analysis. *Proceedings of BIOTECHNO*.

Georgakopoulos, D., Hornick, M., and Sheth, A. (1995). An overview of workflow management: From process modeling to workflow automation infrastructure. *Distributed and parallel Databases*, 3(2):119–153.

Guo, Y., Ye, F., Sheng, Q., Clark, T., and Samuels, D. C. (2013). Three-stage quality control strategies for dna re-sequencing data. *Briefings in bioinformatics*, page bbt069.

Haas, B. J., Papanicolaou, A., Yassour, M., Grabherr, M., Blood, P. D., Bowden, J., Couger, M. B., Eccles, D., Li, B., Lieber, M., et al. (2013). De novo transcript sequence reconstruction from rna-seq using the trinity platform for reference generation and analysis. *Nature protocols*, 8(8):1494–1512.

Heer, J., Card, S. K., and Landay, J. A. (2005). Prefuse: a toolkit for interactive information visualization. In *Proceedings of the SIGCHI conference on Human factors in computing systems*, pages 421–430. ACM.

Kohl, M., Megger, D. A., Trippler, M., Meckel, H., Ahrens, M., Bracht, T., Weber, F., Hoffmann, A.-C., Baba, H. A., Sitek, B., et al. (2014). A practical data processing workflow for multi-omics projects. *Biochimica et Biophysica Acta (BBA)-Proteins and Proteomics*, 1844(1):52–62.

Latgé, J.-P. (1999). Aspergillus fumigatus and aspergillosis. *Clinical microbiology reviews*, 12(2):310–350.

Li, T., Liu, L., Zhang, X., Xu, K., and Yang, C. (2014). Provenancelens: Service provenance management in the cloud. In *Collaborative Computing: Networking, Applications and Worksharing (CollaborateCom), 2014 International Conference on*, pages 275–284. IEEE.

Li, Z., Chen, Y., Mu, D., Yuan, J., Shi, Y., Zhang, H., Gan, J., Li, N., Hu, X., Liu, B., et al. (2012). Comparison of the two major classes of assembly algorithms: overlap–layout–consensus and de-bruijn-graph. *Briefings in functional genomics*, 11(1):25–37.

Mattoso, M., Werner, C., Travassos, G., Braganholo, V., and Murta, L. (2008). Gerenciando experimentos científicos em larga escala. *SBC-SEMISH*, 8:121–135.

Moreau, L. and Missier, P. (2013). PROV-DM: The PROV Data Model.

Rosa, M., Moura, B., Vergara, G., Santos, L., Ribeiro, E., Holanda, M., Walter, M. E., and Araújo, A. (2016). Bionimbuz: A federated cloud platform for bioinformatics applications. In *Bioinformatics and Biomedicine (BIBM), 2016 IEEE International Conference on*, pages 548–555. IEEE.

Silva, W., Vilar, D., Souza, D., Walter, M. E., Brígido, M., and Holanda, M. (2016). 2path: A terpenoid metabolic network modeled as graph database. In *Bioinformatics and Biomedicine (BIBM), 2016 IEEE International Conference on*, pages 1322–1327. IEEE.

Wolstencroft, K., Haines, R., Fellows, D., Williams, A., Withers, D., Owen, S., Soiland-Reyes, S., Dunlop, I., Nenadic, A., Fisher, P., et al. (2013). The taverna workflow suite: designing and executing workflows of web services on the desktop, web or in the cloud. *Nucleic acids research*, 41(W1):W557–W561.

Zerbino, D. R. and Birney, E. (2008). Velvet: algorithms for de novo short read assembly using de bruijn graphs. *Genome research*, 18(5):821–829.

Zhang, Z., Schwartz, S., Wagner, L., and Miller, W. (2000). A greedy algorithm for aligning dna sequences. *Journal of Computational biology*, 7(1-2):203–214.

An Effective Method to Optimize Docking-Based Virtual Screening of Fully-Flexible Receptor Models

Renata De Paris[1], **Christian Vahl Quevedo**[1], **Duncan Dubugras Alcoba Ruiz**[1], **Osmar Norberto de Souza**[2]

[1]Grupo de Pesquisa em Inteligência de Negócio – GPIN
Faculdade de Informática – PUCRS
Av. Ipiranga, 6681 – Prédio 32 – Sala 628 – Porto Alegre – RS – Brasil

[2]Laboratório de Bioinformática, Modelagem e Simulação de Biossistemas – LABIO
Faculdade de Informática – PUCRS
Av. Ipiranga, 6681 – Prédio 32 – Sala 602 – Porto Alegre – RS – Brasil

{renata.paris,christian.quevedo,duncan.ruiz,osmar.norberto}@pucrs.br

Abstract. *The use of conformations obtained from molecular dynamics in docking experiments is the most accurate approach to simulate the behavior of receptors and ligands in molecular environments. However, such simulations are computationally expensive, and their execution may become an unfeasible task due to the large amount of structural information used for screening on libraries of compounds. This study presents e-FReDock, a cloud-based scientific workflow that aims to assist in high-throughput docking experiments of flexible receptors. e-FReDock is developed on a cloud-based workflow enactment system and stores docking data into a NoSQL database. Preliminary results show the costs taken to execute e-FReDock on different Azure virtual machines.*

1. Introduction

Demand for the pharmaceutical industry to offer newer and more efficient drugs has steadily grown as new diseases are surfacing [Garg et al. 2011]. Nevertheless, the development of new drugs still is a complex and challenging process since in addition to spend a long time, it requires large investments in technology resources. The search for methods that reduce the computational time involved in the molecular docking process, and to investigate accurately chemical and biological information about ligands and receptors is highly important to identify and optimize a drug candidate. In our approach, the receptor is called Fully-Flexible Receptor (FFR) model [Machado et al. 2011] since it is an ensemble of conformations generated from a Molecular Dynamics (MD) simulation trajectory [Alonso et al. 2006]. Although MD is the more accurate technique to represent the natural behavior of proteins and ligands into flexible environments, it is computationally costly [Teodoro and Kavraki 2003]. The limiting factor is the required time to perform virtual screening in an FFR model, which can have from hundreds of thousands up to millions of conformations, against small molecules databases. Several studies have been done to deal with this virtual high-throughput screening; however, it remains a challenge in the present day [Antunes et al. 2015, Buonfiglio et al. 2015, De Paris et al. 2015].

In this paper, we present e-FReDock, a flexible receptor docking-based virtual screening workflow deployed on cloud platforms. This scientific workflow was developed in e-Science Central (e-SC) web workflow enactment system [Hiden et al. 2013]

and stores docking data in a database within the NoSQL MongoDB database [Chodorow 2013]. This paper first gives a brief overview of related works and the e-SC workflow enactment system. Section 3 presents the e-FReDock conceptual specification, its two sub-workflows designed to perform the docking experiments, and describes the data model created to store docking data in MongoDB. Section 4 shows the preliminary results achieved by running e-FRedock on a set of Azure Virtual Machines (VMs). Last section concludes the paper and provides future works directions.

2. Related Works

Several computational approaches have focused on reducing the elapsed time taken to perform virtual screening of small molecules against receptors using High Performance Computing (HPC) environments, such as computing clusters [Zhang et al. 2013] or clouds [Kiss et al. 2014, Ocaña et al. 2014, Nguyen et al. 2015]. Most of these methods treat the receptor as rigid bodies to scale up the simulations based on the volume of compounds to be docked. For instance, Zhang and collaborators [Zhang et al. 2013] enhanced the performance of high-throughput virtual screening by executing simultaneous experiments on a large number of processors from a Linux cluster, using AudoDock4.2 [Morris et al. 2009]. Similarly, Kiss et al. [Kiss et al. 2014] performed virtual screening practices by using VMs from Azure cloud platform [AZURE 2017]. They compared the scalability of docking experiments using 5, 10, and 20 Azure VMs with a grid structure and analyzed the performance gains achieved in each environment. In a recent work, Nguyen et al. [Nguyen et al. 2015] compared the differences in performance by running low accuracy molecular virtual screening on a multi-site cloud environment connected through a virtual networking system. Ocaña et al. [Ocaña et al. 2014] also analyzed the performance gains obtained by scaling 10,000 receptor-ligand docking experiments out on cloud VMs, using AutoDock4 and AutoDock Vina [Morris et al. 2009].

Even though there is a large volume of published studies describing advances to facilitate large scale docking experiments, their environments were developed to execute rigid receptors without storing needed information to analyze docking results. On the other hand, our study focus on developing a cloud-based workflow able to optimize molecular docking simulations of FFR models and retrieve meaningful docking information from a NoSQL database.

3. Workflow Enactment System: e-Science Central

The e-SC platform is a cloud-based web workflow enactment system for e-Science projects [Hiden et al. 2013]. It includes essential services to support scientists and developers, who can design scientific workflows using available services or building new applications. The main virtualized services provided by e-SC are [Hiden et al. 2013]:

- Workflow enactment: the workflow services are placed on the e-SC file system and their instances are sent to be deployed on each of a pool of VMs.
- Analysis service: e-SC allows users to upload and deploy services into the platform. Services can be written in Java, JavaScript, R, and Octave.
- Security: workflows, data, and services are controlled by their owners, which grant different access for specific users or groups.
- Provenance: all system events such as workflow operation, data control, and other interactions are stored in a JSON database.

- Processing: a set of VMs are attached to the e-SC server to execute workflow invocations. These VMs can be any worker node and are also called e-SC engines.
- Data storage: all e-SC files are versioned and placed in the cloud storage system.

A scientific workflow in e-SC is composed of blocks or activities. The invocation of a workflow enables a sequence of blocks (or services) that run when all input ports have input data buffered and ready to use [Cała et al. 2013]. Data, workflows, and code are stored in logical folders. All e-SC operations can be performed by users through a web interface or using external tools on which blocks are written and deployed into the e-SC platform.

4. e-FReDock Conceptual Specification

The scientific workflow for performing docking-based virtual screening of FFR models was developed to scale docking experiments out onto cloud resources and store docking data required to run experiments in a NoSQL database. Figure 1 shows the conceptual specification of e-FReDock based on e-SC workflow enactment system. It represents how e-SC works to execute e-FReDock. The e-SC Server hosts the web server, the e-SC enactment system and the two sub-workflows from e-FReDock along with their input files that are stored in the e-SC Share Library. According to e-SC operation, the workflow enactment is executed on one of the VMs attached to the e-SC server. Each attached VM represents one of the Workflow Enactment Nodes, which contains the e-SC Engines, the Java Runtime, and the Library Directory structure. The e-SC Engines component contains the e-SC code necessary to install workflow blocks, execute workflow invocations, and communicate with the e-SC server.

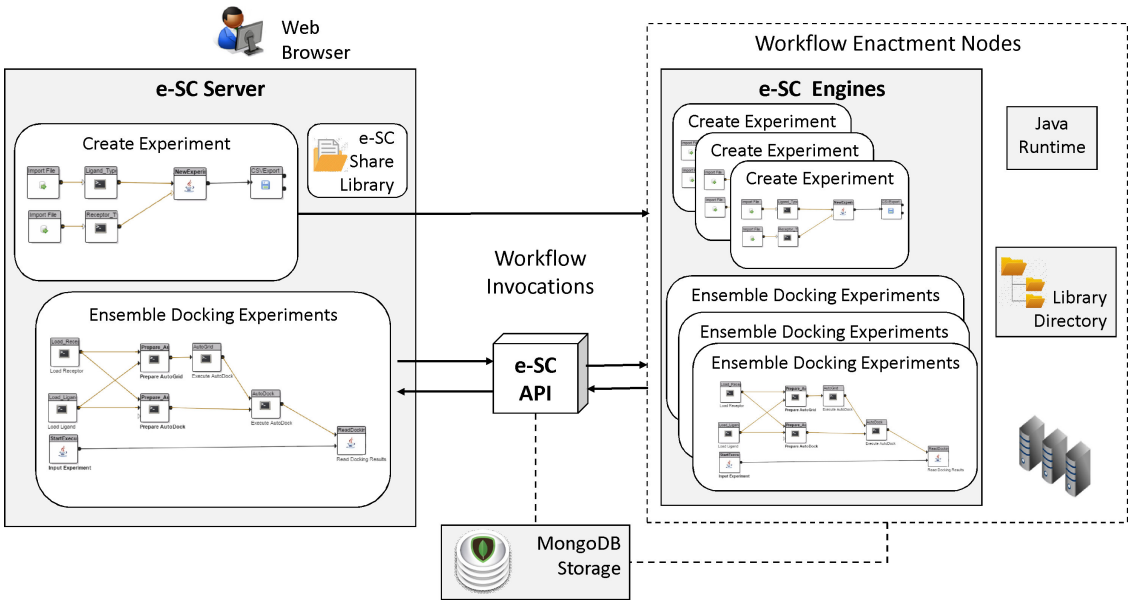


Figure 1. Conceptual architecture of e-FReDock scientific workflow based on e-SC. A workflow instance starts on the e-SC server and its invocation is sent to be executed on one of the enactment nodes.

There are a number of events performed by the e-SC Server from start to finish of a workflow invocation. The e-SC system initiates by placing the workflow invocation

onto a queue with its parameters and settings required. When an idle node is found, the e-SC engine installed on this node removes the execution request from the queue and starts to execute a workflow block. During workflow execution, the e-SC engine automatically deploys workflow blocks into the library on the enactment node as follows: (1) the block code and its dependencies (software, packages, and files) are downloaded from e-SC Server and installed within the Library Directory; (2) the block operations are initialized; and (3) the main and post processing routines of the block are executed.

It is worth mentioning that e-SC performs the deployment of workflow blocks on the enactment node only when the workflow invocation is executed for the first time. One advantage of transferring all block files from e-SC server to the e-SC engine at once is that it avoids the high-throughput data transfer produced at runtime, thereby minimizing delays or failures caused by network connection issues, and data transfer costs charged by public cloud platforms. The e-SC Share Library was used to store all snapshots from the FFR model and ligands used to perform docking experiments.

The e-SC engine creates a file directory for each workflow invocation and stores input and output files during the workflow execution. By default, e-SC eliminates temporary files from the enactment node immediately after ending a workflow execution; unless one or more target folders are indicated to save required files. This is especially useful for performing docking-based virtual screening experiments since a simple molecular docking simulation executed on AutoDock4.2 [Morris et al. 2009] generates approximately 3.15 MB files, varying depending on the ligand size. For instance, an exhaustive docking simulation between an FFR model with 19,500 conformations and a small ligand would produce approximately 60 GB of input and output files, the majority of which are temporary files. This clearly indicates the importance of having a database to store essential information from docking results, and then delete unessential files produced by docking experiments.

4.1. The MongoDB Storage Component

The MongoDB Storage component is the database created to store and manipulate data generated during the e-FReDock execution. To control the docking-based virtual screening experiments on e-FReDock scientific workflow, we created the following collections: *DockingConf*, *Experiment*, *FFRConformations*, *ReceptorAtomType* and *Docking*.

The *Experiment* collection stores every ensemble docking experiments submitted to the e-FReDock workflow, including ligand information, the date and time of experiment generation, and attributes used to control experiments. All information required to create AutoGrid and AutoDock input files are stored in the *DockingConf* collection. To store all AutoDock4.2 inputs in one collection, the *type* attribute indicates "G" when a document has AutoGrid input parameters and, "D" when a document has Lamarckian Genetic Algorithm (LGA) input parameters. Thus, each experiment is related to two documents of the *DockingConf* collection.

Essential docking results are extracted from the AutoDock output file and stored in *Docking* collection. Such data include: (i) the best predicted FEB value, the RMSD value from the best FEB, and the 3D coordinates from the best ligand pose. The latter allows users to retrieve a final docking pose and plot it on a molecular graphical visualization tool. The RMSD stores the distance between the ideal ligand position and its final docking

pose when a known ligand position is used as reference. The period when a docking experiment starts and ends is also captured and stored in the database.

4.2. The Create Experiment Sub-Workflow

The Create Experiment sub-workflow is responsible for submitting a new experiment on e-FReDock and arranging all data needed to execute the Ensemble Docking Experiments sub-workflow. All collections, except *Docking*, are updated when a new experiment is added to e-FReDock. Figure 2 displays the design of the Create Experiment sub-workflow. With the exception of *Import File* and *CSVExport* blocks, which are e-SC services, all blocks were developed for the purposes of this study.

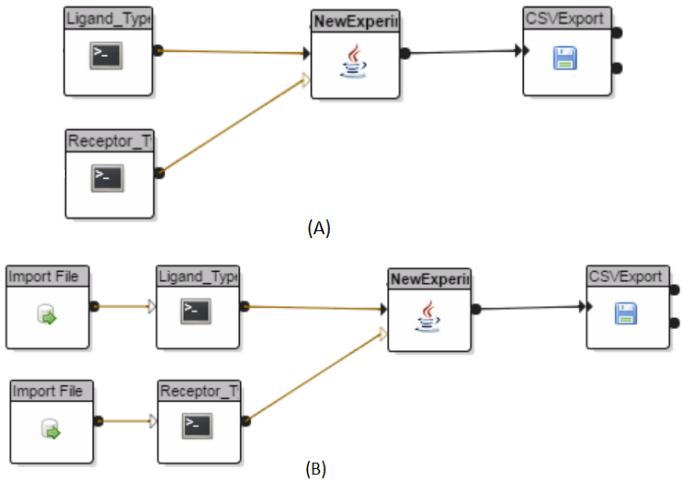


Figure 2. Create Experiment sub-workflows. Both sub-workflows insert new ensemble docking experiments to e-FReDock. The main difference is that sub-workflow (A) search for receptor and ligands in the e-SC Share Library, while sub-workflow (B) searches for receptor and ligands in the e-SC file system. Picture captured from e-SC web service.

The *Import File* block (Figure 2-B) downloads ligands and receptor files from e-SC file system. If a ligand or receptor file is taken from the e-SC Share Library (Figure 1), the *Import File* block should be disconnected as shown Figure 2-A. In this case, the physical file name should be set in the bock parameters corresponding to the file. The *CSVExport* block generates a CSV file format containing the experiment number.

Receptor_Type and *Ligand_Type* blocks were developed to assign the atom types from receptor and ligand PDBQT files. Besides extracting atom types from the ligand file, the *Ligand_Type* block also obtains the number of rotatable bonds. Each block creates a TXT file and places it in the output port. When all input ports have data ready to use, the *newExperiment* block creates a new docking experiment following the grid, dock and the LGA input parameters provided by users.

4.3. The Ensemble Docking Experiments Sub-Workflow

The Ensemble Docking Experiments sub-workflow (Figure 3), developed to scale onto cloud VMs, contains a set of blocks for performing molecular docking simulations based on Autodock4.2 features [Morris et al. 2009]. Even though this sub-workflow can be executed through the e-SC interface, we created an external program by using e-SC API.

The e-SC API contains features to control workflow invocations, retrieve necessary information from MongoDB database and send them to every workflow's blocks.

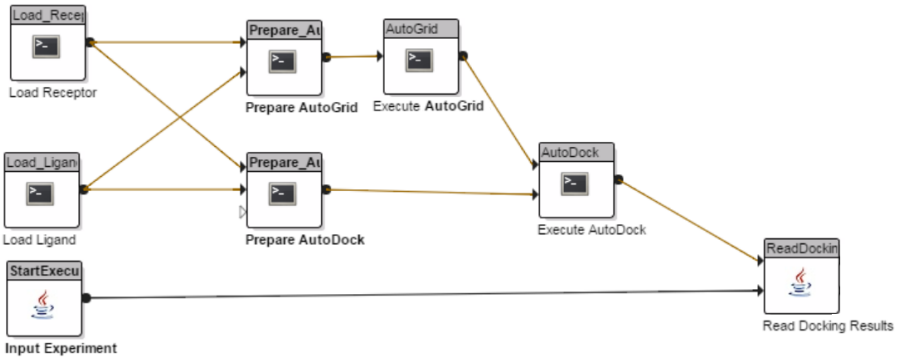


Figure 3. Ensemble Docking Experiments sub-workflow deployed on e-FreDock. Picture captured from e-SC web service.

The *Load Receptor* and *Load Ligand* blocks download PDBQT files from the Library Directory to the workflow invocation folder and generate a TXT file to be placed in the block output port. The physical PDBQT file name and the atom types from both, ligand and receptor, are sent by the e-SC API as a parameter. The *Input Experiment* block generates a CSV file containing the experiment identification, and the date and time the experiment starts.

The *Prepare AutoGrid* and *Prepare AutoDock* blocks receive the receptor and ligand files through their input port. Autogrid and Autodok parameters are taken from database through e-SC API. The output file from *Prepare AutoGrid* and *Prepare AutoDock* blocks are a GPF and a DPF file, respectively. The *Prepare AutoDock* block has an optional input port to load the ligand reference file. While the *Execute AutoGrid* block receives a GPF file through the input port and executes AutoGrid program from AutoDock4.2 toolkit, the *Execute AutoDock* block waits until the end of its execution. When all *Prepare AutoDock* and *Execute AutoGrid* output files arrive to the input port, the *Execute AutoDock* block executes AutoDock software from AutoDock4.2 toolkit. After receiving the *Input Experiment* and *Execute AutoDock* output files, the *Read Docking Results* block extracts the best predicted FEB value along with its 3D coordinates pose and, when applicable, the RMSD value from the DLG file and store these information in the *Docking* collection from MongoDB database. The *Read Docking Results* block also stores in the *Experiment* collection the date and time the experiment started and ended.

5. e-FReDock Performance Analyses on Azure Cloud

In an effort to better understand which choices to make regarding price and performance when the e-FReDock scientific workflow is deployed on the Azure cloud, we performed a preliminary set of docking experiments. A total of 19,500 Ensemble Docking Experiments sub-workflow invocations with identical ligand and docking parameters were used to estimate the overall time and cost to complete the docking experiments in different Dv2-series Ubuntu 14.04 instances located in the North Europe data center. The Dv2-series instances were used to scale the Ensemble Docking Experiments sub-workflow, since they are based on the 2.4 GHz Intel Xeon E5-2673 v3 processor with

Intel Turbo Boost Technology 2.0 that can go up to 3.2 GHz. According to Azure website [AZURE 2017], Dv2-series instances carry more powerful CPUs which are on average about 35% faster than D-series instances for the same memory and disk configuration.

In these experiments, the LGA and its parameters were used to execute the molecular docking simulations between 19,500 conformations from the InhA FFR model [Gargano 2009] and the TCL ligand (PDB ID 2B35) [Sullivan et al. 2006] with 2 rotatable bonds. Twenty-five LGA independent runs were executed with a maximum of 500,000 energy evaluations. The other LGA parameters were kept at default values. Table 1 lists the costs and time taken to execute e-FReDock on different Azure instances configurations. The values show that as the number of nodes decrease the time and price to execute e-FReDock increase. This finding was unexpected and suggests that running e-FReDock on a greater number of VMs with few cores is faster, cheaper and, therefore, likely that such configuration will scale effectively to hundreds of machines.

Table 1. Comparative analysis on the price and time needed to execute 19,500 molecular docking simulations on e-FReDock with different Dv2 Azure instances.

Instance	Cores	Nodes	Threads (total)	Time(hours)	Price ($)
D2v2	2	16	32	20.18	22.44
D3v2	4	8	32	20.66	22.90
D4v2	8	4	32	21.48	26.63
D5v2	16	2	32	22.97	27.56

6. Conclusion

The contribution of this study was to make progress on reducing the computational costs involved in using FFR models to perform practical virtual screening in databases of small molecules. Although this study focuses on presenting e-FReDock conceptual specification, preliminary results may well have a bearing on the choice of the appropriate Azure cloud instance. As future steps, we intend to investigate the scalability and throughput when more worker nodes would be added to the e-FReDock workflow. Such experiments will use new FFR models and a larger number of different ligands.

References

Alonso, H., Bliznyuk, A. A., and Gready, J. E. (2006). Combining docking and molecular dynamic simulations in drug design. *Medicinal Research Reviews*, 26(5):531–568.

Antunes, D. A., Devaurs, D., and Kavraki, L. E. (2015). Understanding the challenges of protein flexibility in drug design. *Expert Opinion on Drug Discovery*, 10(12):1301–1313.

AZURE (2017). Microsoft Azure: The cloud for modern business.

Buonfiglio, R., Recanatini, M., and Masetti, M. (2015). Protein flexibility in drug discovery: From theory to computation. *ChemMedChem - Chemistry Enabling Drug Discovery*, 10(7):1141–1148.

Cała, J., Hiden, H., Woodman, S., and Watson, P. (2013). Cloud computing for fast prediction of chemical activity. *Future Generation Computer Systems*, 29(7):1860–1869.

Chodorow, K. (2013). *MongoDB: the definitive guide*. O'Reilly Media, Sebastopol, CA.

De Paris, R., Quevedo, C. V., Ruiz, D. D., and Noberto de Souza, O. (2015). An effective approach for clustering InhA molecular dynamics trajectory using substrate-binding cavity features. *PloS one*, 10(7):1–25.

Garg, V., Arora, S., and Gupta, C. (2011). Cloud computing approaches to accelerate drug discovery value chain. *Combinatorial Chemistry & High Throughput Screening*, 14(10):861–871.

Gargano, F. (2009). *Efeito da temperatura na enzima 2-trans-enoil-ACP(CoA) redutase (EC 1.3.1.9) de Mycobacterium tuberculosis em complexo com o NADH: um estudo por simulação por dinâmica molecular*. PhD thesis, Programa de Pós-Graduação em Biologia Celular e Molecular, PUCRS, Porto Alegre, RS, Brasil.

Hiden, H., Woodman, S., Watson, P., and Cala, J. (2013). Developing cloud applications using the e-science central platform. *Philosophical Transactions of the Royal Society of London A: Mathematical, Physical and Engineering Sciences*, 371(1983):1–12.

Kiss, T., Borsody, P., Terstyanszky, G., Winter, S., Greenwell, P., McEldowney, S., and Heindl, H. (2014). Large-scale virtual screening experiments on Windows Azure-based cloud resources. *Concurrency and Computation: Practice and Experience*, 26(10):1760–1770.

Machado, K. S., Winck, A. T., Ruiz, D. D., and Norberto de Souza, O. (2011). Mining flexible-receptor molecular docking data. *Wiley Interdisciplinary Reviews: Data Mining and Knowledge Discovery*, 1(6):532–541.

Morris, G. M., Huey, R., Lindstrom, W., Sanner, M. F., Belew, R. K., Goodsell, D. S., and Olson, A. J. (2009). AutoDock4 and AutoDockTools4: Automated docking with selective receptor flexibility. *Journal of Computational Chemistry*, 30(16):2785–2791.

Nguyen, A., Matsunaga, A., Tsugawa, M., Ichikawa, K., Haga, J. H., et al. (2015). Deployment of a multi-site cloud environment for molecular virtual screenings. In *International Conference on e-Science*, pages 145–154. IEEE.

Ocaña, K., Benza, S., De Oliveira, D., Dias, J., and Mattoso, M. (2014). Exploring large scale receptor-ligand pairs in molecular docking workflows in HPC clouds. In *IEEE International Parallel & Distributed Processing Symposium Workshops*, pages 536–545. IEEE.

Sullivan, T. J., Truglio, J. J., Boyne, M. E., Novichenok, P., Zhang, X., Stratton, C. F., Li, H.-J., Kaur, T., Amin, A., Johnson, F., et al. (2006). High affinity InhA inhibitors with activity against drug-resistant strains of *Mycobacterium tuberculosis*. *ACS Chemical Biology*, 1(1):43–53.

Teodoro, M. L. and Kavraki, L. E. (2003). Conformational flexibility models for the receptor in structure based drug design. *Current Pharmaceutical Design*, 9(20):1635–1648.

Zhang, X., Wong, S. E., and Lightstone, F. C. (2013). Message passing interface and multithreading hybrid for parallel molecular docking of large databases on petascale high performance computing machines. *Journal of Computational Chemistry*, 34(11):915–927.

A Study of Index Structures for K-mer Mapping

Elvismary M. de Armas[1], Marcos V. Marques da Silva[1] and Sérgio Lifschitz[1]

[1]Departamento de Informática
Pontifícia Universidade Católica do Rio de Janeiro (PUC-Rio)

{earmas, mvmsilva, sergio}@inf.puc-rio.br

Abstract. *K-mer Mapping is an internal process for de novo genome fragments assembly methods. It constitutes a computational challenge due to its high main memory consumption. We present in this paper a study of index-based methods to deal with this problem. We consider a RDBMS environment for plant genome assembling and a particular version of Velvet program. We propose an ad-hoc I/O cost model in order to analyze the performance of both B+-tree and hash-based index structures. We present experimental results considering an actual RDBMS implementation with a sugarcane dataset and we show that we may obtain considerable performance gains while reducing RAM requirements.*

Resumo. *O mapeamento K-mer é um processo interno dos métodos de novo de montagem de fragmentos de genoma. Há um desafio computacional devido ao alto consumo de memória principal. Nós apresentamos neste artigo um estudo de métodos de indexacão que lidam com este problema. Consideramos um ambiente de SGBD para montagem de fragmentos de genoma de plantas e uma versão do programa Velvet. É proposto um modelo de custo ad-hoc para analisar o desempenho de árvores B+ e estruturas de índice baseadas em hash. Alguns resultados experimentais implementados em um SGBD real e um conjunto de dados da cana de acúcar mostram que é possível obter bons ganhos de desempenho ao mesmo tempo que se reduz requisitos de RAM.*

1. Introduction

Fragment assembly (e.g. [El-Metwally et al. 2013]) is a fundamental problem in bioinformatics. As current DNA sequencing technologies cannot read whole genomes in a single run, we need to reconstruct the original sequences considering the available short reads. The Next-Generation Sequencing (NGS, e.g [Metzker 2010]) technologies commonly deliver these small fragments. When there is no reference genome, one needs to consider a *de novo* [Schatz et al. 2010] assembling technique, which consists of combining short reads in order to create full-length sequences with correct biological semantics.

To address the computational challenges brought by *de-novo* assembly procedures, there exists genome assemblers and corresponding implementations that are based on *de Bruijn* graph structure [Pevzner et al. 2001]. These help with the assembly computation, particularly dealing with overlaps. Assemblathon [Earl et al. 2011][Bradnam et al. 2013] and Gage [Salzberg et al. 2012] present a preliminary evaluation for those assemblers.

In order to build a *de Bruijn* graph, firstly we decompose the reads into *k-mers* (substrings with specific *k* length) that eventually become graph nodes. We create an arc between two nodes when the *k-mers* in these nodes occur consecutively in at least one read. Since we must map identical *k-mers* into the same node, there is a need to identify duplicate *k-mers*, a process known as K-mer Mapping. Once this process is accomplished, the (directed) edges set for the *de Bruijn* graph is created by scanning the short reads. However, the *de Bruijn* graph approach requires an very large amount of RAM for its construction and processing. Some assembling experiments [Li et al. 2009],[Cook and Zilles 2009] and supplementary results in [Bradnam et al. 2013] confirm that the construction of the *de Bruijn* graph is step that consumes more memory and CPU time in those assemblers. *De Bruijn* assemblers, such as Velvet [Zerbino and Birney 2008], do not scale for large genomes.

The K-mer Mapping process defines the main memory requirements for *de Bruijn* graph construction. The memory requirements are higher for more complex organisms, like plants, when compared to bacterial genomes. The large complex plant genomes remain a particularly difficult challenge for *de novo* assembly methods due to a variety of biological, computational and bio molecular reasons. Plant genomes can be nearly 100 times larger than some of the currently sequenced animals, like mammalian genomes. They also have higher rates of heterozygosity and repeats than their counterparts for other species [Schatz et al. 2012].

VelvetH-DB is an implementation for Velvet in a Relational Data Base Management System (RDBMS) of the K-mer mapping process [Silva 2016]. The idea is to overcome the high main memory consumption using persistent structures in external memory like RDBMS do. In this work we investigate the impact of considering different index-based structures that aim at reducing the RAM computational requirements, while enabling improvements for the whole execution time. VelvetH-DB can be used to replace *VelvetH* - Velvet's first phase - during genome assembly.

This paper is organized as follows: first we give an overview of the current implementation of VelvetH-DB with its main characteristics in Section 2. An overview of related works is exposed in Section 3. Then, in Section 4,we present an analytical and experimental comparison of traditional indexes for VelvetH-DB. We propose the K-mer Mapping I/O cost model and briefly discuss the *k-mer* codification. Finally, our results and conclusions are listed in Section 5.

2. VelvetH-DB overview

VelvetH-DB implements K-mer Mapping following the Velvet program, an open-source assembler developed by the European Bioinformatics Institute (EMBL-EBI), designed for a *de novo* genome assembly. It has stood out over others assemblers due to its method for removing errors, identify tips and bubbles in the graph, and apply a coverage cut-off.

VelvetH is responsible for the execution of the K-mer Mapping. It starts preprocessing all sequences files, creating a normalized file, called *Sequences*, with the contents of all incoming sequence files. Next, it executes the K-mer Mapping process, looking for overlapped regions into the reads and writing annotations of overlaps into the Roadmaps file. The Roadmaps and the Sequence files are the entry points to build *de Bruijn* graph implemented by *VelvetG*, Velvet's second phase. A research study [Cook and Zilles 2009]

shows that the Roadmap generation consumes 25% of the time of all assembler execution in Velvet, and that this time consumption is dominated by the *k-mer* search operation. Velvet uses an in-memory optimized data structure for local search and keep the unique *k-mers* information. However, it fails when the amount of RAM is not enough for the number of unique *k-mers* for a specific *k*.

An important step for *VelvetH* execution corresponds to the duplication test for *k-mer* occurrences. First, a hash function is applied to each *k-mer* in order to obtain the bucket number to which it belongs. Then, the *k-mer* is searched in the *splay* tree of that bucket. Since *VelvetH* tries to minimize the number of annotations, it finds not only duplicate *k-mers* but also duplicate chains of *k-mers*. Therefore, when the current *k-mer* K is found, the program checks if the left adjacent *k-mer* has also an overlap in the same read as K, as well as if it's the left adjacent k-mer of K in the overlapped read. If it is, the current *k-mer* is included into the same map A to be written when no more consecutive *k-mers*, in the current read, are overlapped in a continuous chain. The mapping, pending to be written (if there is one), is persisted and a new mapping is created including the current *k-mer*. When a *k-mer* is not found into the corresponding splay tree, it is stored as a new node and the map pending to be written (if there is one) is achieved. It is important to note that the size of a hash table is proportional to the number of distinct *k-mers* in the sequences dataset.

As the original *VelvetH* algorithm does, VelvetH-DB generates the *Roadmap* file with all *k-mers* mapping duplicated. We have implemented VelvetH-DB into PostgreSQL 9.5 as RDBMS functions. Those functions were implemented using 4 different entities, which eventually are mapped into corresponding 4 physical relations (or tables): *sequence*, *presequence*, *identification* and *roadmap*.

There are other two temporal relations designed to get the statistics and generate the final Roadmaps file. The K-mer Mapping process implemented in VelvetH-DB visits each *k-mer* in the Sequences relation and may or not classify it as a duplicate *k-mer*. Each *k-mer* must be checked whether (or not) it already exists in the Identification relation. Only two kinds of operations are executed over the Identification relation: search operations and insertions. Both the number of search operations $((m - k + 1) \times n)$, where m is the sequence length and n the number of reads, and the number of insertions $(O(4^k))$ influence significantly the execution time for the K-mer Mapping process.

VelvetH-DB have been implemented in 5 main functions. The first two load the reads from ".fastq" file into a Sequences table to preprocess the sequences replacing $'N'$ character for a valid nitrogenous base character, and load it into the Presequences table. The third function executes the K-mer Mapping over sequences in the Presequences relation, searching for duplicate *k-mers* over the Identification relation, inserting unique *k-mers* or mapping duplicate *k-mers* into the Roadmap relation. The last function generates maps into files similar to the Velvet Roadmaps file. All of these functions are orchestrated by an extra function that also run the transactions checkpoints to persist data during the process.

3. Overview of Related Works

The authors in [Constantin et al. 2016] propose the AS-Index, which is an index designed for full matching string for variable pattern size. It uses a hashing function based on

algebraic signatures of *n-grams*. The authors argue for a constant search time based on limiting disk accesses to the two buckets associated to the first and last *n-grams* signatures of the pattern. Thus, the search runs independently from pattern's size. The hash directory itself can often be cached into RAM or needs at most two additional disk accesses. The pattern's size is set by *k*, which is fixed during the whole execution. Therefore, an additional complex processing and structures focused to search variable pattern size in a constant time (under some ideals conditions) is out scope here. Also the approach of AS-index is to index every *n-grams* previously, which does not make sense for K-mer Mapping.

Other approaches have used the *n-grams* strategy for indexing DNA and proteins sequences like the *q-gram* index. The authors in [Cao et al. 2005], present a two level hash index based (hash over *q-grams* clusters and *c-trees* index) on the fact that two sequences share a certain number of *q-grams* if the edit distance between them is within a certain threshold. In [Wang et al. 2013] a two-level inverted index is presented. It allows the use of longer but approximate *n-gram* matching for searching *k*-nearest neighbors (KNN), based on edit distance. In [Tan et al. 2003] a new index structure, called *ed-tree*, is proposed to support probe-based homology search in DNA sequence databases. However, the above indexes focus on similarity search, with approximate sequence matching for long sequences, instead of exact search matching.

The work in [Greenfield and Roehm 2013] presents a study about *k-mer* uniqueness across genomes. The authors have developed a 'shared k-mers' metric to the 'relatedness' of two organisms. This work is based on fast exact matches of *k-mer* strings using a database, rather than conventional alignment based on inexact matches of much longer strings. These *k-mers* were stored in a conventional relational database and indexed to support efficient exact match operations. The authors used SQL for answering biological questions over a *k-mer* database. They first load all *k-mers* and use a non-clustered composite index to execute parallel full matching search over them. The databases are implemented using SQL Server 2008 and the index is stored in a separate file group located on a solid-state disc to improve the process time. Although some index configuration was briefly mentioned in the paper, no details were given about these indexes.

4. Analytical and experimental comparison of traditional indexes for VelvetH-DB

There exists two data structures commonly used as indexes in relational DBMS [Ramakrishnan and Gehrke 2003]: tree-based indexes and hash-based indexes.

Tree-based indexes arrange data entries in a sorted order by search key value into a binary search tree. The leaf nodes contain the data entries and the contents of pages in non-leaf levels direct the search to correct leaf pages. Since all search operations begin at the root node, the number of disk I/Os is equal to the length of a path to a leaf, plus the number of leaf pages with qualifying data entries. The B+-tree is a tree-based index structure that is always balanced. Also, all leaf pages in a B+-tree are maintained in a double-linked list as a way to optimize range queries.

Hash-based indexes use a hash table organization to maintain the data entries grouped into buckets. A bucket consists of a primary page and possibly additional pages

for larger records or collision management. Given a bucket number, a hash-based structure allows us to retrieve the primary page for the bucket in one or two disk I/O operations.

The selection of a good index to maintain fast search operations and insertions over the set of unique *k-mers* highly affects the run time for the K-mer Mapping process. In order to obtain good performances, we present in this work a study of two basic not clustered index configurations for *k-mers*: a) *B+-tree as a primary key index* over the *k-mer* string (not null and uniqueness constrains) and b) *Hash index* over *k-mer* string.

The I/O cost of K-mer Mapping will dominate the execution time due to the large number of search operations and insertions involved. The I/O cost function can be modeled as a sequence of operations with a corresponding I/O cost for random access. Since buffer hits save one read page operation, we take into account the number of buffer hits that decreases the I/O cost for searches and insertions. The hits number at each step is determined by the state of the DBMS buffer, given a buffer replacement policy, and the previously sequence of searches and insertions executed. For our analysis we use the K-mer Mapping cost model showed in Table 1

Table 1. K-mer Mapping I/O Cost Model

Index	K-mer Mapping cost model
Unclustered tree index	$(log_F 0.15B + 1 - \textit{mean}(hits_{search}))D \times numb\_searches\ +$ $(log_F 0.15B + 3 - \textit{mean}(hits_{insert}))D \times numb\_insertions$
Unclustered hash index	$(2 - \textit{mean}(hits_{search}))D \times numb\_searches\ +$ $(4 - \textit{mean}(hits_{insert}))D \times numb\_insertions$

We assume a constant buffer size and a fixed replacement policy (LRU or MRU) based on the order in which the pages were used, corresponding to the sequences of *k-mers* searched and inserted. The shared buffer is dedicated to that process.

4.1. Experimental settings

To study the index behavior, we have executed a same group of experiments using B+-tree and hash index varying the *k* value for the same dataset. All experiments were executed in the same machine: CPU Core i7-3770 3.40GHz, 8GB RAM and 1TB HDD, on a Ubuntu 14.0.4 LTS Linux distribution. The experiments ran over a sugarcane DNA dataset with 2 million reads. The read length is 100 and we have used a sufficiently large *k* variation, over 39 to 71.

4.2. *K-mer* codification.

During the K-mer Mapping, the number of search operations (one search per *k-mer*) is the main factor that impact the execution time. These occur over the *k-mer* attribute in the Identification relation. This attribute contains the unique *k-mer* strings that have been identified during the process. It is a string of length k restricted to $\Sigma = \{'A', 'T', 'C', 'G'\}$.

Therefore, some codifications may be applied to *k-mer* data. It should be noted that we have encoded the *k-mer* string as a byte array (bytea PostgreSQL datatype) using 2 bits for each character in order to reduce space, since $|\Sigma| = 4$. However, the *k-mer* view as a character array is very attractive due to its simplicity and better comprehension for biologists. Our experiments check the cost of modeling the *k-mer* string as an encode byte array compared to an ASCII character array. Figure 1 shows the execution time for a 2 million sequence reads dataset. As it can be observed, the time cost is really expensive for our experimentation in the case of B+-tree. For *k-mer* string codification as a bytea PostgreSQL datatype, using 2 bits for each character, that execution time is halved.

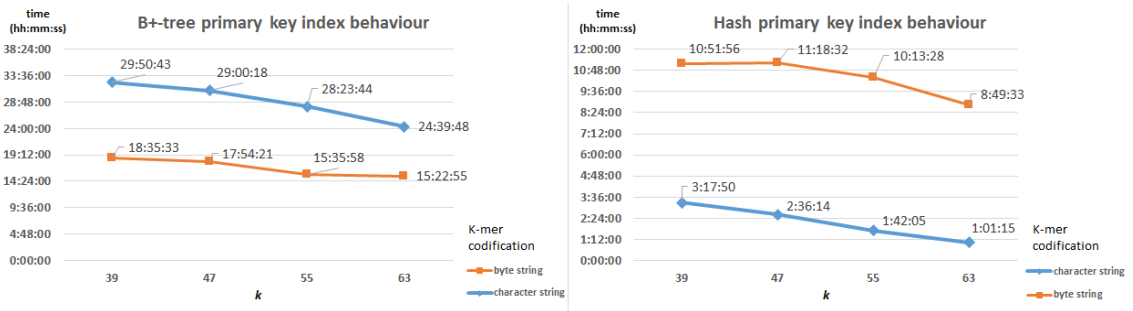


Figure 1. Time comparison for different *k-mer* string codification

Consequently, for B+-tree index we select byte array codification for *k-mer*. We implemented the codification methods inside the PL/PSQL function that execute the mapping. However, as we can see in Figure 1, the behavior for hash index is the opposite, due to the internal implementation of hash index into PostgreSQL. The ASCII character array codification showed a very high impact in improving the execution times.

4.3. Indexes evaluation

Table 2 presents the K-mer Mapping runtime variations against k value for both indexes, showing that while k increases, the execution time decreases. The main cause is that the total *k-mers* number decreases while k value increases.

To measure the performances of these indexes we gathered the EXPLAIN ANALYZE query plan output. We selected the following values related with the shared buffer behavior: the number of buffer hits blocks and the number of blocks that have to be read from the disk. The average of those values were analyzed for search operations and insertions to find relationships with the execution time. Results are given in Table 2.

The average of the number of hits in the shared buffer for search operations was higher using B+-tree primary key index for all experimental executions. This might suggests that less blocks from the disk were necessary. However, the number of disk pages involved using the B+-tree primary key index was higher than the number involved using a hash index, as it is shown by the average of the number of shared buffer reads. Similarly, the average of the number of hits in the shared buffer for insertions was higher for executions using B+-tree primary key index (Table 2). In contrast, the average blocks read does not have the same behavior with respect to search operations. It was higher for hash index in the case of $k = 47$ and $k = 55$, twice in order of hundredths, and it was equal for the remaining k values. However, the execution time difference during the K-mer Mapping, between B+-tree primary key and hash index, was affected by the insertions times.

Table 2. Indexes comparison. Execution time and buffer metrics statistics.

k	B+-tree index					Hash index				
	$Time$	searches		insertions		$Time$	searches		insertions	
		hit $blocks$	$read$ $blocks$	hit $blocks$	$read$ $blocks$		hit $blocks$	$read$ $blocks$	hit $blocks$	$read$ $blocks$
39	18:35:33	4.092	0.3852	4.8	0.01	3:17:50	2.515	0.2502	3.22	0.01
47	17:54:21	4.077	0.3478	4.78	0.01	2:36:14	2.523	0.2216	3.23	0.02
55	15:35:58	4.066	0.3005	4.77	0.01	1:42:05	2.535	0.1958	3.25	0.02
63	15:22:55	4.068	0.3378	4.88	0.02	1:01:15	2.561	0.1659	3.27	0.02
71	13:09:21	3.991	0.3389	4.85	0.02	0:37:03	2.504	0.1894	3.29	0.02

The average of shared dirty blocks that represents the value of $mean(2 - hits_{update})$ for B+-tree was almost twice (in order of tenths) the corresponding values for the hash index. The B+-tree primary key average values for shared dirty blocks was more than 1.5 times the corresponding values for hash index.

Our experiments validate the model and support the selection of hash index for VelvetH-DB. As proof cases, we have tried to execute the K-mer Mapping using other two sugar cane libraries, one with 10.151.440 reads and another with 13.413.501 reads, both with read length 100. The K-mer Mapping in *VelvetH* failed for both libraries at 7 million reads and 6 million reads, respectively. An estimation for the memory size required by *VelvetH* to be able to process that amount of reads gives an approximation of 22,3 GB, for the first experiment, and 29,4 GB, for the second, to persist only the hash table. However, using our approach, we were able to execute the process successfully in 7,46 days for the first and 11,46 days for the second using 8GB of RAM.

5. Conclusions

We have presented here a study of index performances for the VelvetH-DB implementation. It constitutes an approach for K-mer Mapping into RDBMS for *de novo* plant genome assembly, as an strategy to avoid the high main memory consumption.

Looking forward for good performances, we explore two basic non clustered index configurations for *k-mers*: (i) *B+-tree as a primary key index* over the *k-mer* and (ii) *Hash index* over *k-mer* string. Better results were obtained for hash-based indexes. Based on an analytically study, we have proposed an I/O cost model for K-mer Mapping taking into account variables associated to buffer behavior, such as the number of hits. Our experiments show that the selection of a good index configuration over the set of unique *k-mers* in an actual DBMS highly affects the K-mer Mapping execution time. Our practical experiments also show that this solution overcomes the memory limitations for VelvetH K-mer Mapping. Indeed, we managed to process a dataset that the original *VelvetH* could not complete in the same hardware configuration. Therefore, it becomes suitable as an input parameter to *VelvetG* - the 2nd Velvet phase - in order to complete the graph.

References

Bradnam, K. R., Fass, J. N., et al. (2013). Assemblathon 2: evaluating de novo methods of genome assembly in three vertebrate species. *GigaScience*, 2(1):1–31.

Cao, X., Li, S. C., and Tung, A. K. H. (2005). *Indexing DNA Sequences Using q-Grams*, pages 4–16. Springer Berlin Heidelberg, Berlin, Heidelberg.

Constantin, C., du Mouza, C., Litwin, W., Rigaux, P., and Schwarz, T. (2016). As-index: A structure for string search using n-grams and algebraic signatures. *Journal of Computer Science and Technology*, 31(1):147–166.

Cook, J. J. and Zilles, C. (2009). Characterizing and optimizing the memory footprint of de novo short read dna sequence assembly. In *Performance Analysis of Systems and Software, 2009. ISPASS 2009. IEEE International Symposium on*, pages 143–152.

Earl, D., Bradnam, K., et al. (2011). Assemblathon 1: A competitive assessment of de novo short read assembly methods. *Genome Research*, 21(12):2224–2241.

El-Metwally, S., Hamza, T., Zakaria, M., and Helmy, M. (2013). Next-generation sequence assembly: Four stages of data processing and computational challenges. *PLoS Comput Biol*, 9(12):1–19.

Greenfield, P. and Roehm, U. (2013). Answering biological questions by querying k-mer databases. *Concurrency and Computation: Practice and Experience*, 25(4):497–509.

Li, R., Zhu, H., et al. (2009). De novo assembly of human genomes with massively parallel short read sequencing. *Genome Research*.

Metzker, M. L. (2010). Sequencing technologies - the next generation. *Nature reviews. Genetics*, 11(1):31–46.

Pevzner, P. A., Tang, H., and Waterman, M. S. (2001). An eulerian path approach to dna fragment assembly. *Proceedings of the National Academy of Sciences*, 98(17):9748–9753.

Ramakrishnan, R. and Gehrke, J. (2003). *Database Management Systems*. McGraw-Hill, Inc., New York, NY, USA, 3 edition.

Salzberg, S. L. et al. (2012). Gage: A critical evaluation of genome assemblies and assembly algorithms. *Genome Research*, 22(3):557–567.

Schatz, M. C., Delcher, A. L., and Salzberg, S. L. (2010). Assembly of large genomes using second-generation sequencing. *Genome Research*, 20(9):1165–1173.

Schatz, M. C., Witkowski, J., and McCombie, W. R. (2012). Current challenges in de novo plant genome sequencing and assembly. *Genome Biology*, 13(4):1–7.

Silva, M. (2016). Master thesis. VelvetH-DB: Uma abordagem robusta de banco de dados no processo de montagem de fragmentos de sequencias biologicas.

Tan, Z., Cao, X., Ooi, B. C., and Tung, A. K. H. (2003). The ed-tree: an index for large dna sequence databases. In *Scientific and Statistical Database Management, 2003. 15th International Conference on*, pages 151–160.

Wang, X., Ding, X., Tung, A. K. H., and Zhang, Z. (2013). Efficient and effective knn sequence search with approximate n-grams. *Proc. VLDB Endow.*, 7(1):1–12.

Zerbino, D. R. and Birney, E. (2008). Velvet: Algorithms for de novo short read assembly using de bruijn graphs. *Genome Research*, 18(5):821–829.

VelvetH-DB: Persistência de Dados no Processo de Montagem de Fragmentos de Sequências Biológicas

Marcos Vinicius Marques da Silva[1], Maristela Holanda[2], Edward Hermann Haeusler[1], Elvismary Molina de Armas[1], Sérgio Lifschitz[1]

[1]Dep. de Informática – Pontifícia Universidade Católica do Rio de Janeiro (PUC-Rio)

[2]Dep. de Ciência da Computação – Universidade de Brasília (UNB)

{msilva,hermann,earmas,sergio}@inf.puc-rio.br, mholanda@unb.br

Resumo. *A montagem da cadeia completa de uma sequência biológica envolve a leitura de um grande volume de fragmentos do genoma (short reads). Os principais programas de montagem de fragmentos existentes têm como gargalo principal seu alto consumo de memória principal. Neste artigo estudamos e avaliamos a implementação de um destes programas, Velvet, amplamente usado e recomendado. É proposta uma solução, chamada VelvetH-DB, com o intuito de reduzir o consumo de memória principal, considerando persistência de dados de maneira a obter eficácia e robustez.*

1. Introdução e Motivação

Diversas espécies e organismos já foram sequenciados e montados usando tecnologias de sequenciamento tradicionais, como o método Sanger [Liu et al. 2012]. Nos últimos anos, uma nova geração de sequenciadores trouxe grandes avanços, como Roche 454 System e Illumina [Liu et al. 2012]. Houve redução de custo e aumento da eficiência através do emprego de variadas técnicas de sequenciamento massivamente paralelas [Henson J et al. 2012]. Assim, torna-se possível o mapeamento de genomas cada vez maiores a um custo cada vez menor.

A nova geração de sequenciadores [Cook and Zilles 2009] tem como característica a obtenção de fragmentos com tamanho pequeno (*short reads*), quando comparados aos métodos mais tradicionais. Entretanto, há também uma maior quantidade de erros de sequenciamento. Estas duas características somadas aumentam significativamente o volume de dados a ser processado. Os programas de montagem, por sua vez, requerem um alto consumo de memória principal [Alkan et al. 2011]. Quando executados em máquinas ditas convencionais, e para determinadas espécies com grande volume de *short reads*, não somente sua eficiência é reduzida como a eficácia nem mesmo é garantida.

Esta pesquisa está relacionada com projetos correntes, em parceria com o Instituto de Bioquímica da UFRJ, que fazem sequenciamento e montagem de genomas de plantas, como é o caso da cana-de-açúcar. No caso, não há genoma de referência para avaliação dos resultados e existe uma estrutura complexa, com diversos genes homólogos. Consequentemente, é necessário sequenciar uma grande quantidade de material genético, tornando o problema da montagem dos fragmentos ainda maior em relação a outros genomas bastante estudados, como o do ser humano, vírus e bactérias.

Um dos programas de montagem de fragmentos mais utilizados é o Velvet [Zerbino and Birney 2008] devido à sua alta qualidade de resultados de montagem. O Velvet tem um módulo inicial, chamado VelvetH, que permite retirar boa quantidade de erros de pré-

processamento antes da montagem propriamente dita. Este módulo inicial enfrenta problemas de consumo de memória RAM, dificultando ou impedindo seu uso. Há estudos que visam reduzir o consumo de memória principal em alguns destes programas [Alkan et al. 2011] [Li et al. 2013], [Georganas et al. 2014], [Quitzau and Stoye 2008] , [Surget-Groba and Montoya-Burgos 2010].

Este artigo discute o funcionamento do programa Velvet e as causas do alto consumo de memória principal, em particular, do módulo VelvetH. Nós propomos uma solução, chamada VelvetH-DB, que traz robustez e eficácia ao processo de montagem.

Cabe observar que não se considera aqui como adequadas as soluções baseadas exclusivamente em tecnologia como, por exemplo, aqueles programas que assumem memória RAM virtualmente infinita, envolvendo *hardware* com baixa relação custo-benefício. Buscamos manter-nos no contexto de computadores com componentes de prateleira (*off-the-shelf*), presentes na grande maioria dos laboratórios de bioinformática.

Este artigo está organizado da seguinte forma: na Seção 2 explicamos um pouco mais sobre o processo de montagem de fragmentos de sequências biológicas. Ilustramos o problema de eficácia para grandes conjuntos de *short reads*. Já na Seção 3 explicamos o problema da geração do Roadmap e propomos uma solução para persistir dados e trazer robustez e eficácia para a solução. Na Seção 4 discutimos a implementação realizada com uso de um SGBD relacional e testes práticos preliminares. Por fim, concluímos na Seção 5 com comentários finais, trabalhos em andamento e futuros.

2. Programa Velvet: VelvetH e consumo de memória RAM

Com a tecnologia que temos atualmente, é inviável fazer a leitura de um genoma por completo, já que a sua cadeia de nucleotídeos pode possuir bilhões de pares de bases nitrogenadas. Portanto, para realizar a leitura o gene precisa ser lido por partes. Ou seja, o cromossomo é fragmentado, cada parte é lida e ao final é realizada a montagem dos fragmentos, obtendo desta forma o sequenciamento completo do genoma.

A montagem de fragmentos é um dos principais problemas estudados em biologia computacional [Fang et al. 2005] e a complexidade está relacionada ao tamanho do genoma, à quantidade de material genético sequenciado e à cobertura realizada. Existem basicamente duas abordagens para a montagem de fragmentos: (i) *Mapeamento*: quando existe um genoma de referência na tomada de decisão ao encontrar ambiguidades e erros de sequenciamento; (ii) *De Novo*: a montagem do genoma é feita exclusivamente a partir das sequências geradas, sem usar como base um genoma de referência.

Os maiores desafios estão na abordagem *De Novo*, também utilizada pela maioria dos programas montadores, com foco na construção do grafo de De Bruijn, estrutura auxiliar na montagem. Este é o caso do programa Velvet, aqui estudado. Procura-se obter o relacionamento entre subsequências exatas de tamanho k extraídas dos fragmentos (*short reads*) através de grafos. Os nós representam os *k-mers* e as arestas indicam os *k-mers* adjacentes com a sobreposição exata de k-1, conforme ilustrado na Figura 1.

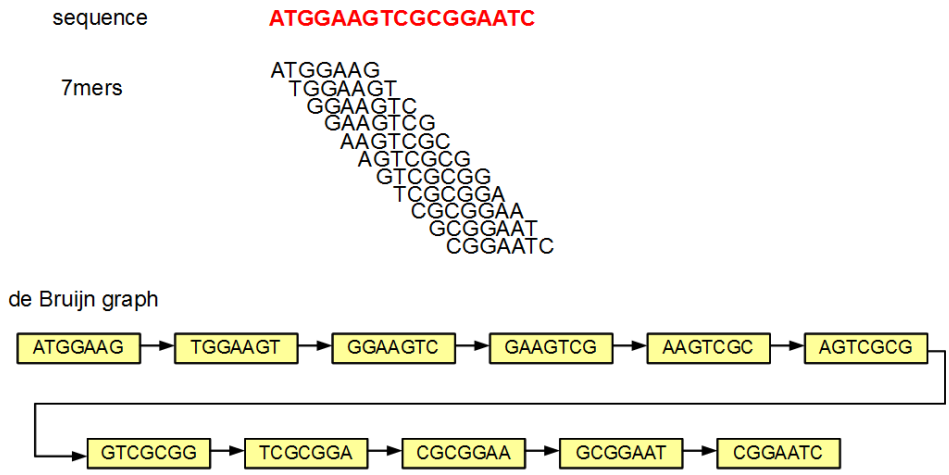


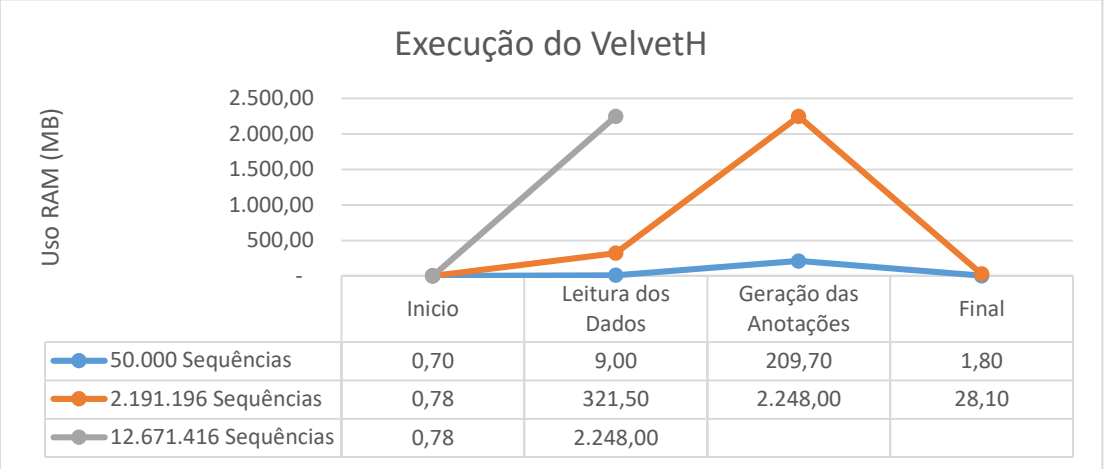
Figura 1. Extração dos k-mers (k=7) e construção do grafo de De Bruijn.

A questão sobre qual deve ser o tamanho de k é algo que, em termos práticos, só é descoberto executando-se o programa diversas vezes, com diferentes tamanhos para k e escolhendo ao final o melhor resultado analisando a quantidade e a qualidade dos *contigs* gerados [Sterky and Lundeberg 2000]. Este problema está fora do escopo deste artigo.

Velvet é um conjunto de programas criados para a manipulação de grafos de Bruijn para a montagem de sequências muito curtas (NGS) [Zerbino and Birney 2008]. Estes programas podem ser estudados em 7 fases distintas, e por sua vez agrupadas em dois subprogramas: VelvetH, com a fase 1 de padronização das sequências e a fase 2 das anotações de sobreposições das sequências; e o VelvetG, com as 5 fases seguintes: criação do pré-grafo; inclusão de informações dos nós do grafo; heurísticas de remoção de erros de sequenciamento e cálculo do caminho euleriano para obter os *contigs*. Detalhes maiores sobre Velvet e seus módulos podem ser encontrados em [Silva, 2016].

Com o objetivo de entender o consumo de memória principal da aplicação, decidimos realizar alguns testes experimentais preliminares e medir o consumo de memória RAM do módulo VelvetH. Para isso, foi criada uma máquina virtual, através do Virtual Box (V4.3.34), com uma instalação de sistema operacional Ubuntu 64 (V14.04 LTS) e 8GB de memória RAM disponíveis, além de 2 processadores (Intel® Core™ i7-4510U CPU @ 2.00GHz × 2) e 1TB de espaço em disco.

Usamos 3 bibliotecas de dados de sequências, uma com 50.000 (10MB), outra com 2.191.196 (500MB) e mais uma com 12.671.416 (3.9GB) de sequências. A primeira biblioteca é um conjunto de sequências de teste do próprio programa Velvet, e as duas últimas são subconjuntos de sequências de um total de mais de 275 milhões de sequências. São dados do projeto de mapeamento do genoma da cana-de-açúcar do Instituto de Bioquímica da UFRJ, nossa parceira e principal usuária com respeito à esta pesquisa.



	Inicio	Leitura dos Dados	Geração das Anotações	Final
50.000 Sequências	0,70	9,00	209,70	1,80
2.191.196 Sequências	0,78	321,50	2.248,00	28,10
12.671.416 Sequências	0,78	2.248,00		

Figura 2. Consumo de memória Principal pelo VelvetH.

Como observado na Figura 2, o conjunto de dados com mais de 12 milhões de *short reads* não conseguiu ser executado pela aplicação. Esta execução foi interrompida após 12h de processamento e todo o trabalho foi perdido. Mesmo o VelvetH sendo um módulo que visa a redução do consumo de memória RAM para as etapas seguintes, quando o conjunto de dados é grande existe problema de consumo de memória principal.

3. VelvetH-DB: versão robusta e persistente para VelvetH

Neste artigo propomos a implementação do VelvetH-DB, que é uma abordagem baseada em banco de dados para a geração dos Roadmaps, que tem como objetivo criar uma solução robusta e eficaz da fase de geração de anotações do VelvetH.

O fluxo para a geração do *Roadmap* através da implementação do VelvetH original, como descrito na Figura 3, é relativamente simples. O usuário, através de linha de comando, executa a aplicação com seus parâmetros e a lista dos arquivos contendo as sequências a serem processadas. O programa executa as etapas já descritas, salvando em disco os arquivos de saída, *Sequences* e *Roadmap*.

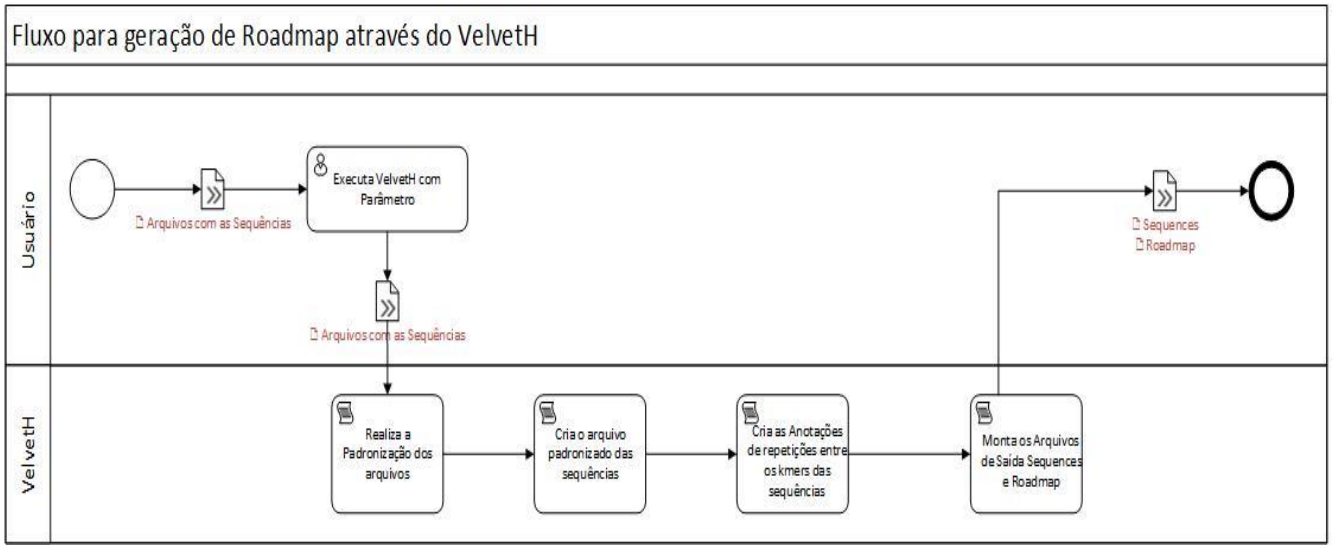


Figura 3: Fluxo de geração do *Roadmap* através do VelvetH sem modificações.

A solução proposta nesta pesquisa, o VelvetH-DB, utiliza na fase inicial de padronização das sequências uma versão alterada do VelvetH, com a saída deste sendo a entrada para a versão com persistência em banco de dados, ilustrada na Figura 4.

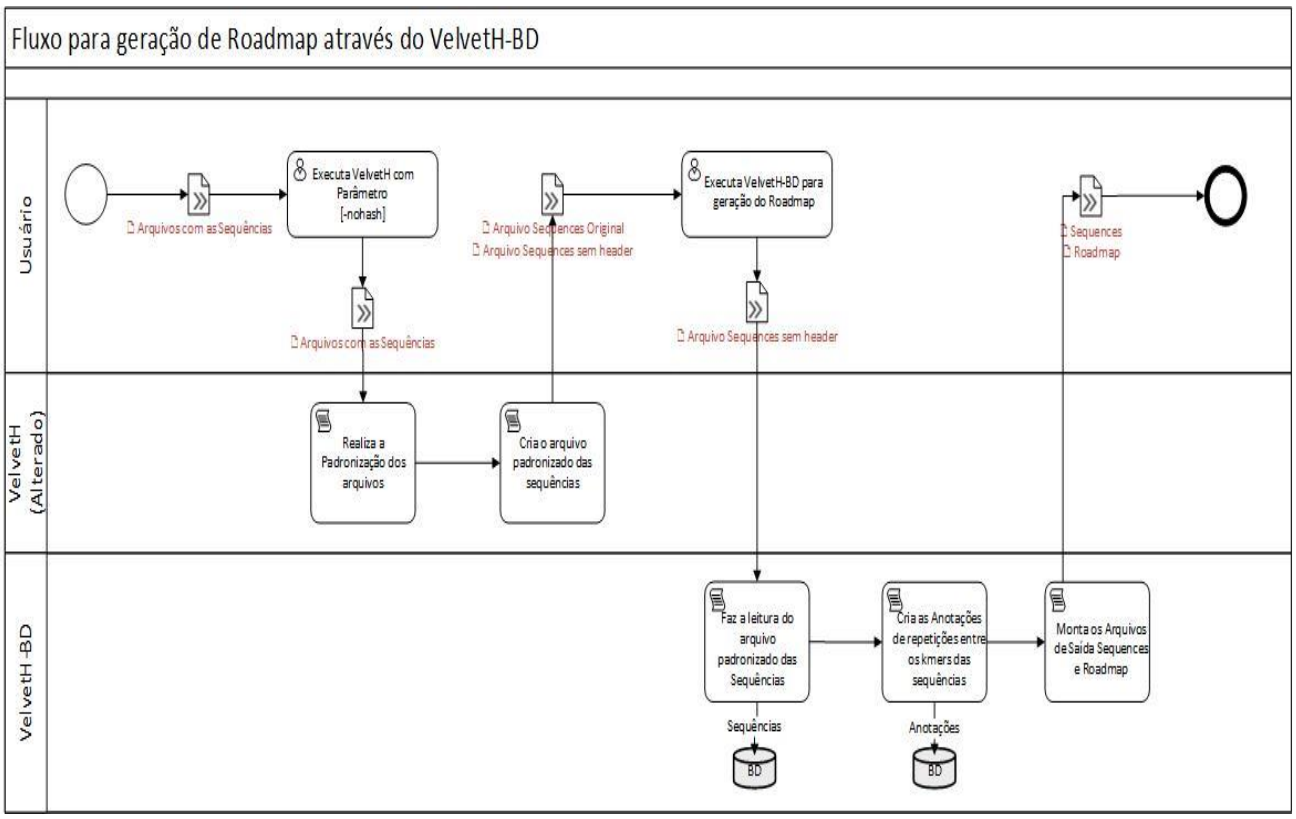


Figura 4: Fluxo para a geração do *Roadmap* através do VelvetH-BD.

Quando o volume de dados a ser processado pelo VelvetH não cabe em memória principal, o sistema operacional passa a fazer o uso do *swap* em disco, tornando a aplicação drasticamente mais lenta. Em caso de falha, todo o processamento realizado, que pode corresponder a vários dias de trabalho [Li et al. 2010], é perdido e o processo não pode ser reiniciado a partir do ponto de interrupção. No VelvetH-DB, conseguimos obter a robustez desejada, garantindo desta forma a continuidade no processamento mesmo em casos onde há falha na máquina.

Outra funcionalidade diz respeito ao tratamento dado quando surgem novas bibliotecas de sequências para serem processadas, todo o trabalho de geração do *Roadmap* tem que ser feito novamente, não sendo possível processar apenas o conjunto novo de dados e incluir ao final do processamento já realizado. Na abordagem em banco de dados, iremos encurtar o tempo de processamento de novas bibliotecas evitando desperdício de tempo de processamento de análises já realizadas, procedendo de maneira incremental.

4. VelvetH-DB: implementação e avaliação preliminar

Para a implementação deste trabalho foi escolhido o SGBD de código aberto *PostgreSQL*, versão 9.3. Outros SGBDs poderiam ter sido usados, mas o PostgreSQL, além de ser um

software robusto e bem difundido, é fruto de experiências em implementações e alterações no código-fonte deste SGBD no grupo de pesquisas BioBD da PUC-Rio.

A implementação aqui proposta consiste em uma alteração no VelvetH original para gerar um outro formato para o arquivo padronizado de sequências e funções de banco de dados, escritas em *PL/pgSQL*, que é a linguagem estruturada do próprio *PostgreSQL*.

A modelagem relacional define um esquema com 5 tabelas:

- **SEQUENCES**: Contém as informações originais das sequências.
- **PRESEQUENCES**: Armazena as informações das sequências ajustadas por substituições de caracteres, conforme realizado no VelvetH original.
- **IDENTIFICATION:** Guarda as informações dos k-mers únicos identificados.
- **ROADMAP**: Armazena as informações das anotações de sobreposições encontradas entre as sequências
- **TEMPFILE**: Armazena temporariamente a estrutura de dados dos arquivos.

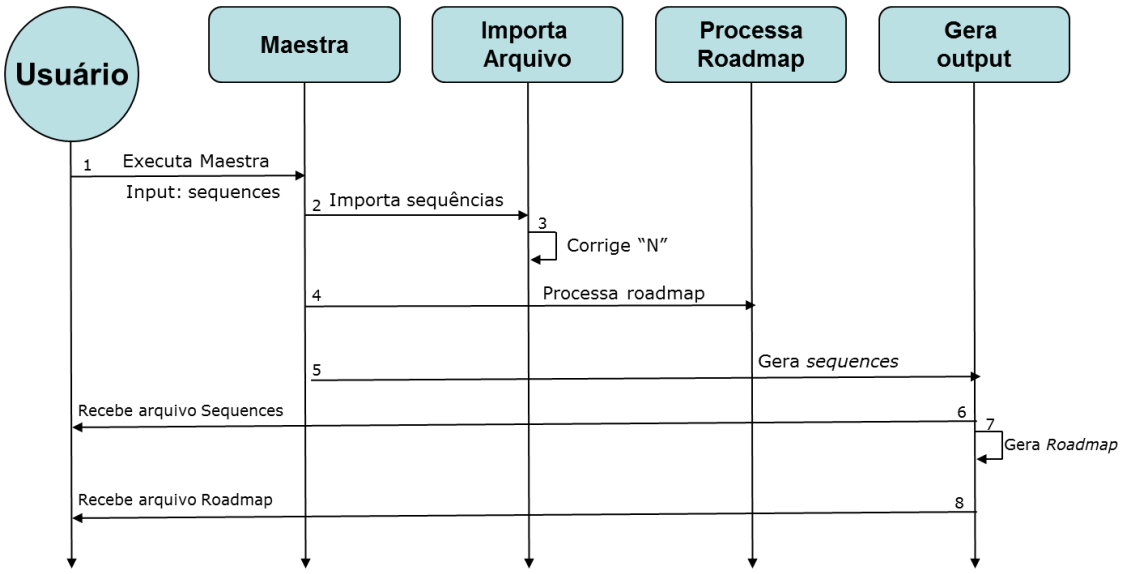


Figura 5. Diagrama de Sequência das Funções do VelvetH-DB.

Para fazer o processamento das sequências e gerar o *Roadmap*, foram criadas 4 funções no SGBD relacional PostgreSQL, todas escritas em PL/PgSQL, a linguagem estruturada do próprio SGBD. Devido às restrições de espaço, o leitor poderá consultar os códigos respectivos na referência de pesquisa base para este artigo [Silva, 2016].

Ilustramos na Figura 5 o diagrama de sequências para a execução da aplicação. A função *Maestra* tem a responsabilidade de fazer as chamadas às outras funções da aplicação. Tem como parâmetros de entrada o arquivo padronizado gerado pelo VelvetH por nós modificado e o tamanho do k-mer, retornando os arquivos do *Roadmap* e *Sequences*. A função *Importa Arquivo* é responsável pela leitura do arquivo padronizado das sequências, persistindo os dados na tabela *Sequences* e gerando a tabela *PreSequences*, como feito no Velvet original. Basicamente há um filtro para caracteres "N" em uma sequência: trata-se de base nitrogenada que não pode ser lida no processo de

sequenciamento. A aplicação assume como um erro e altera, arbitrariamente, para o caractere "A". Já a função *Processa Roadmap* acessa as sequências na tabela *PreSequences*, extraindo os k-mers e incluindo-os na tabela *Identification*. Quando o k-mer já se encontra na tabela, uma anotação é gravada na tabela *Roadmap*. Ao final, a função *Gera Output* cria na tabela *TempFile* as tuplas no formato do arquivo *Sequences,* e gera as tuplas no formato do *Roadmap*, exportando um arquivo texto em disco, finalizando assim o processo.

As funções criadas cobrem a fase 2 do Velvet, processando o arquivo padronizado com as sequências, dividindo cada sequência em k-mers e gerando as anotações das repetições entre as sequências já processadas. Ao final escreve-se um arquivo *Roadmap* que possuiu a mesma estrutura do *Roadmap* gerado pela aplicação original e pode ser utilizado pelo VelvetG, para a geração do grafo e completitude da montagem.

Por fim, com o objetivo preliminar de demonstrar a eficácia da aplicação VelvetH-DB, realizamos um teste de execução com duas bibliotecas de dados: uma pequena e que poderia ser executada por uma máquina sem muitos recursos; e outra com volume de dados tal que os recursos existentes não seriam suficientes.

Utilizamos a mesma máquina virtual antes mencionada, porém configurada para rodar sem SWAP ativo no sistema operacional, para medir o limite de uso da memória RAM.

Tabela 1. Resultados da Execução VelvetH original e VelvetH-DB.

Número de Sequências	VelvetH			VelvetH-DB		
	Tempo de execução	Qtd. sequências processados	Execução sucedida	Tempo de execução	Qtd. sequências processados	Execução sucedida
25.000	<1s	25 K	Sim	10 m	25 K	Sim
10.151.440	N/A	7 M	Não	156h	10 M	Sim

O resultado da execução das duas aplicações, apresentado na Tabela 1, expõe que a nossa solução proposta e a respectiva implementação é, como esperado, menos eficiente que a aplicação tradicional quando o conteúdo cabe na RAM. Entretanto, a aplicação VelvetH-DB se mostra eficaz nos casos quando a memória RAM não é suficiente nem mesmo para completar a 1ª fase da montagem de fragmentos.

No caso de teste, foram 156 horas para a montagem do arquivo *Sequences* e o arquivo *Roadmap* mas, pelo menos, o processo foi realizado até o final: com o VelvetH original, o processamento foi interrompido com erro de memória após 3/4 das sequências processadas. Mais detalhes dos testes podem ser encontrados em [Silva, 2016].

5. Comentários finais

Neste trabalho buscamos um entendimento detalhado do processo de montagem de fragmentos estudando um programa bastante utilizado e referenciado, chamado Velvet. Foi essencial identificar as razões para o gargalo no consumo de memória RAM, em particular, já presente no primeiro módulo, chamado de VelvetH. Para pequenos volumes de dados (e.g. genomas de bactérias), não apresenta muitos problemas. Entretanto, para volumes maiores, como é o caso do genoma da cana de açúcar, o processamento não é somente lento como inviável.

Nesta pesquisa buscamos uma alternativa para permitir eficácia e também robustez, mesmo que, no momento, comprometendo-se a eficiência nos casos mais simples. Como trabalhos em andamento e futuros, estamos realizando testes experimentais para análise de eficácia e eficiência, buscando uma solução completa que inclua também melhorias para o módulo VelvetG.

Referências Bibliográficas

Alkan, C.; Sajjadian, S.; Eichler, E. E. (2011) "Limitations of next-generation genome sequence assembly". Nature 8(1), p. 61–65.

Cook, J. J.; Zilles, C. (2009) "Characterizing and optimizing the memory footprint of de novo short read DNA sequence assembly." Anais do ISPASS - International Symposium on Performance Analysis of Systems and Software, p. 143–152.

Fang, S.-C.; Wang, Y.; Zhong, J. A (2005) "Genetic Algorithm Approach to Solving DNA Fragment Assembly Problem". In: Journal of Comp. and Theoretical Nanoscience, 2(4), p. 499–505.

Georganas, E.; Buluç, A.; Chapman, J.; et al. (2014) "Parallel de Bruijn Graph Construction and Traversal for de Novo Genome Assembly". Anais da Intl Conf. for High Performance Computing, Networking, Storage and Analysis, p.437–448.

Henson J, Tischler G, Ning Z. (2012) "Next-generation sequencing and large genome assemblies". In: *Pharmacogenomics* 13(8): p. 901-915.

Li, R.; Zhu, H.; Ruan, J.; et al. (2010) De novo assembly of human genomes with massively parallel short read sequencing. *Genome Research* 20(2), p. 265–72.

Li, Y.; Kamousi, P.; Han, F.; et al. (2013) "Memory efficient minimum substring partitioning". Proceedings of the VLDB Endowment 6(3), p. 169–180.

Liu, L., Li,Y., Li,S. et al. (2012) "Comparison of next-generation sequencing systems", Journal of Biomedical Biotechnology, Article ID 251364, 11 pages.

Quitzau, J.; Stoye, J. (2008) "A space efficient representation for sparse de Bruijn subgraphs". Universität Bielefeld technical report.

Silva, M.V.M. da (2016) "VelvetH-DB: Uma abordagem robusta de banco de dados no processo de montagem de fragmentos de sequências biológicas", Dissertação de Mestrado, PUC-Rio.

Sleator, D. D.; Tarjan, R. E. (1985) "Self-adjusting Binary Search Trees". Journal of the ACM 32(3), p. 652–686.

Sterky, F.; Lundeberg, J. (2000) "Sequence analysis of genes and genomes". In: Journal of biotechnology 76, p. 1–31.

Surget-Groba,Y;Montoya-Burgos,J. (2010) "Optimization of de novo transcriptome assembly from next-generation sequencing data". *Gen. Research*, 20(10), p.1432–40.

Zerbino, D. R.; Birney, E. (2008) "Velvet: Algorithms for de novo short read assembly using de Bruijn graphs", *Gen. Research* 18(5), p. 821–829.

REALIZAÇÃO

SBC Sociedade Brasileira de Computação

ORGANIZAÇÃO

UFU

Faculdade de Computação

APOIO

ProPP

UBERLÂNDIA CONVENTION & VISITORS BUREAU

FOMENTO

CAPES

FAPEMIG

CNPq

PATROCÍNIO DIAMANTE

Itaú

NVIDIA

PATROCÍNIO OURO

Algar Telecom

PATROCÍNIO PRATA

Google

IBM Research | Brasil

PATROCÍNIO BRONZE

SEBRAE

neppo

SANKHYA GESTÃO DE NEGÓCIOS

ClickPerformance®