



BIOINFORMÁTICA
CONTEXTO COMPUTACIONAL
E APLICAÇÕES



Organizadores:
Scheila de Avila e Silva
Daniel Luis Notari
Gabriel Dall'Alba

BIOINFORMÁTICA

CONTEXTO COMPUTACIONAL E APLICAÇÕES

Organizadores

Scheila de Avila e Silva

Daniel Luis Notari

Gabriel Dall'Alba



FUNDAÇÃO UNIVERSIDADE DE CAXIAS DO SUL

Presidente:

José Quadros dos Santos

UNIVERSIDADE DE CAXIAS DO SUL

Reitor:

Evaldo Antonio Kuiava

Vice-Reitor:

Odacir Deonísio Gracioli

Pró-Reitor de Pesquisa e Pós-Graduação:

Juliano Rodrigues Gimenez

Pró-Reitora Acadêmica:

Nilda Stecanela

Diretor Administrativo-Financeiro:

Candido Luis Teles da Roza

Chefe de Gabinete:

Gelson Leonardo Rech

Coordenadora da Educs:

Simone Côrte Real Barbieri

CONSELHO EDITORIAL DA EDUCS

Adir Ubaldo Rech (UCS)

Asdrubal Falavigna (UCS) – presidente

Cleide Calgaro (UCS)

Gelson Leonardo Rech (UCS)

Jayme Paviani (UCS)

Juliano Rodrigues Gimenez (UCS)

Nilda Stecanela (UCS)

Simone Côrte Real Barbieri (UCS)

Terciane Ângela Luchese (UCS)

Vania Elisabete Schneider (UCS)

© dos organizadores

Revisão: Izabete Polidoro Lima

Dados Internacionais de Catalogação na Publicação (CIP)
Universidade de Caxias do Sul
UCS – BICE – Processamento Técnico

B615 Bioinformática [recurso eletrônico]: contexto computacional e aplicações / org. Scheila de Avila e Silva, Daniel Luis Notari, Gabriel Dall’Alba. – Caxias do Sul, RS: Educs, 2020.
Dados eletrônicos (1 arquivo)

ISBN 978-65-5807-001-6
Apresenta bibliografia.
Modo de acesso: World Wide Web.

1. Bioinformática. 2. Biologia. 3. Computação. 4. Tecnologia. I. Silva, Scheila de Avila e. II. Notari, Daniel Luis. III. Dall’Alba, Gabriel.

CDU 2. ed.: 57:004

Índice para o catálogo sistemático:

1. Bioinformática	57:004
2. Biologia	54
3. Computação	004
4. Tecnologia	62

Catalogação na fonte elaborada pela bibliotecária
Carolina Machado Quadros – CRB 10/2236.

Direitos reservados à:



EDUCS – Editora da Universidade de Caxias do Sul

Rua Francisco Getúlio Vargas, 1130 – Bairro Petrópolis – CEP 95070-560 – Caxias do Sul – RS – Brasil

Ou: Caixa Postal 1352 – CEP 95020-972 – Caxias do Sul – RS – Brasil

Telefone/Telefax: (54) 3218 2100 – Ramais: 2197 e 2281 – DDR (54) 3218 2197

Home Page: www.ucs.br – E-mail: educs@ucs.br

Revisores

Dr. Cícero Zanoni (UCS)

Dr. Guilherme Holsbach (UCS)

Dr. Leandro Corso (UCS)

Dr. Luís Fernando Saraiva Macedo Timmers (UNIVATES)

Me. Matheus Emerick de Magalhães (UFRJ)

Ma. Raquel Cristina Balestrin (UCS)

Dr. Ricardo Dornelles (UCS)

Dr^a. Taiana Haag (Hospital Moinhos de Vento/PROADI-SUS)

Biografia dos autores

Apresentação

A ERA DA INFORMAÇÃO

Helena Graziottin Ribeiro (hgrib@ucs.br)

Associada da Sociedade Brasileira de Computação (SBC) e da ACM (SIGMOD). Possui graduação em Bacharelado em Informática pela Pontifícia Universidade Católica do Rio Grande do Sul (1989), Mestrado em Ciências da Computação pela Universidade Federal do Rio Grande do Sul (1993) e Doutorado em Informática pela Université de Grenoble I (*Scientifique et Medicale – Joseph Fourier*), França (2000).

1 PORTAIS E BANCO DE DADOS: DEFINIÇÕES COMPUTACIONAIS

Gabriele Dani (gdani@ucs.br)

Bacharela em Sistemas de Informação pela Universidade de Caxias do Sul, Mestra em Bioinformática pela *Georgetown University*, Washington, DC, USA e doutoranda em Biotecnologia também pela Universidade de Caxias do Sul.

Leonardo Pelizzon (leonardo.pelizzoni@gmail.com)

Graduado em Sistemas de Informação pela UCS em 2016. Mestre em Ciências da Saúde (2019) pelo programa de pós-graduação da Universidade de Caxias do Sul (UCS). Trabalha com desenvolvimento de *software* desde 2008.

Gustavo Sganzerla Martinez (sganzerlagustavo@gmail.com)

Graduado em Sistemas de Informação pela Universidade de Caxias do Sul e atualmente doutorando em Biotecnologia pela mesma Universidade; desenvolve pesquisa na Bioinformática, trabalhando ativamente com ferramentas computacionais voltadas para a Biologia desde 2015.

2 BIOESTATÍSTICA

Cintia Paese Giacomello (cintia.paese@ucs.br)

Sua formação foi realizada na UFRGS, onde cursou Bacharelado em Estatística, Mestrado em Engenharia de Produção e Doutorado em Administração. Professora de Estatística na Universidade de Caxias do Sul, tanto em cursos de graduação quanto de pós-graduação.

3 DATA MINING

Gabriele Dani (gdani@ucs.br)

Bacharela em Sistemas de Informação pela Universidade de Caxias do Sul, Mestra em Bioinformática pela *Georgetown University*, Washington, DC, USA e doutoranda em Biotecnologia também pela Universidade de Caxias do Sul.

Marcelo Sachet (marcelosachet@gmail.com)

Formado em Sistemas de informação e está na Coordenação da TI. Atualmente busca novos conhecimentos em práticas e ferramentas de Data Mining.

Scheila de Avila e Silva (sasilva6@ucs.br)

Graduada em Gestão da Tecnologia da Informação pela Unisinos (2014) e em Ciências Biológicas pela UCS (2004). Possui mestrado em Computação Aplicada pela Unisinos (2007) e doutorado em Biotecnologia pela UCS (2011). Possui experiência em análise de dados, integração de bases de dados biológicas e aplicação de técnicas de inteligência artificial em dados genômicos.

4 REDES NEURAIS ARTIFICIAIS: INTRODUÇÃO E DEFINIÇÕES

Scheila de Avila e Silva (sasilva6@ucs.br)

Graduada em Gestão da Tecnologia da Informação pela Unisinos (2014) e em Ciências Biológicas pela UCS (2004). Possui mestrado em Computação Aplicada pela Unisinos (2007) e doutorado em Biotecnologia pela UCS (2011). Possui experiência em análise de dados, integração de bases de dados biológicas e aplicação de técnicas de inteligência artificial em dados genômicos.

Rafael Vieira Coelho (rafael.coelho@farroupilha.ifrs.edu.br)

Graduado em Engenharia de Computação, em 2008, pela Fundação Universidade Federal do Rio Grande; Mestre em Ciência da Computação pela Universidade Federal do Rio Grande do Sul em 2011; Doutor em Biotecnologia pela Universidade de Caxias do Sul, em 2018.

5 ANÁLISE POR AGRUPAMENTOS

André Gustavo Adami (agadami@ucs.br)

Graduado em Ciência da Computação pela Universidade de Caxias do Sul (1994), Mestre em Ciência da Computação pela Universidade Federal do Rio Grande do Sul (1997), Doutor em Engenharia Elétrica na *Oregon Health and Science University* (2004) e Pós-Doutor em Engenharia Biomédica pela *Oregon Health and Science University* (2006).

Adriana Miorelli Adami (amiorell@ucs.br)

Graduada em Licenciatura Plena em Matemática pela Universidade de Caxias do Sul (1994), Mestra em Matemática Aplicada pela Universidade Federal do Rio Grande do Sul (1999), Doutora em Engenharia Elétrica pela *Oregon Health and Science University* (2006).

6 INTRODUÇÃO ÀS MÁQUINAS DE VETORES DE SUPORTE

Lucas Picinini Dutra (lpduttra@ucs.br)

Graduado em Ciência da Computação pela Universidade de Caxias do Sul (2018), com ênfase em Sistemas de Informação e Inteligência Artificial. Atualmente é discente no Programa de Pós-Graduação em Engenharia de Produção na Universidade de Caxias do Sul.

Iago dos Passos (ipassos@ucs.br)

Graduado em Ciência da Computação pela Universidade de Caxias do Sul (2018), com ênfase em Sistemas de Informação e Inteligência Artificial. Atualmente é discente no Programa de Pós-Graduação em Engenharia de Produção na Universidade de Caxias do Sul.

André Luis Martinotto (almartin@ucs.br)

Graduado em Ciência da Computação pela Universidade de Caxias do Sul (2001), Mestre em Computação pela Universidade Federal do Rio Grande do Sul (2004), Doutor em Ciências dos Materiais pela Universidade Federal do Rio Grande do Sul (2012).

7 COMPUTAÇÃO PARALELA E DISTRIBUÍDA

Alex A. L. dos Santos (allsant1@ucs.br)

Graduado em Ciência da Computação pela Universidade de Caxias do Sul (2005). Em seu Trabalho de Conclusão de Curso, abordou o uso de Computação Colaborativa para o Reconhecimento de Padrões em DNA.

Felipe S. Raota (fsraota@ucs.br)

Graduado em Ciência da Computação pela Universidade de Caxias do Sul (2018). Em seu Trabalho de Conclusão de Curso, desenvolveu soluções que permitam efetuar Simulações Computacionais de Estruturas de Nanodiamante.

Guilherme T. Paz (gtelespaz@ucs.br)

Graduado em Ciência da Computação pela Universidade de Caxias do Sul (2019). Em seu Trabalho de Conclusão de Curso, desenvolveu uma infraestrutura de Nuvem Computacional para o armazenamento de arquivos.

Marcelo Brazil (marcelo@marcelo-brazil.com)

Graduado em Ciência da Computação pela Universidade de Caxias do Sul (2009). Em seu Trabalho de Conclusão de Curso, desenvolveu uma infraestrutura de Nuvem Computacional para a execução de aplicações.

André L. Martinotto (almartin@ucs.br)

Graduado em Ciência da Computação pela Universidade de Caxias do Sul (2001), Mestre em Computação pela Universidade Federal do Rio Grande do Sul (2004), Doutor em Ciências dos Materiais pela Universidade Federal do Rio Grande do Sul (2012).

8 PORTAIS E BANCOS DE DADOS BIOLÓGICOS

Gustavo Sganzerla Martinez (sganzerlagustavo@gmail.com)

Graduado em Sistemas de Informação pela Universidade de Caxias do Sul, doutorando em Biotecnologia pela mesma Universidade; desenvolve pesquisa na Bioinformática, trabalhando ativamente com ferramentas computacionais voltadas para a Biologia, desde 2015.

9 FERRAMENTAS DE ANÁLISE E PROCESSAMENTO DE METAGENOMAS

Tahila Andrighetti (tahilaandrighetti@gmail.com)

Doutora em Genética pela Unesp, iniciou na área da Bioinformática durante sua graduação em Ciências Biológicas na Universidade de Caxias do Sul. Depois da graduação, permaneceu no domínio da Bioinformática, durante seu mestrado em Genética na Universidade Estadual Paulista (Unesp) de Botucatu em 2013, onde desenvolveu pesquisas em análise de dados metagenômicos.

10 BIOLOGIA DE SISTEMAS

Daniel Luis Notari (dlnotari@ucs.br)

Graduado em Ciência da Computação, Doutor em Biotecnologia pela UCS, Mestre em Ciência da Computação pela UFRGS. Professor e coordenador de Ciência da Computação na UCS. Atua como pesquisador em análise de dados e integração de bases de dados científicas relacionados de predição de promotores e ao desenvolvimento baseado em conhecimento.

Diego Luis Bonatto (diegobonatto@gmail.com)

Doutor em Biologia Celular e Molecular pela Universidade Federal do Rio Grande do Sul (2005). Professor associado nível II da UFRGS. Experiência nas áreas de Biologia de Sistemas e Biologia Molecular, com ênfase em Bioinformática e Biologia de Sistemas. Membro afiliado da Academia Brasileira de Ciências (2011-2015).

11 BIOLOGIA ESTRUTURAL

Bruna Schuck de Azevedo (bruna.schuck4@gmail.com)

Farmacêutica formada pela UFCSPA em 2018. Atualmente é mestranda e bolsista Capes no Programa de Pós-Graduação em Biociências da UFCSPA. Tem experiência nas áreas de farmacologia molecular e biofísica molecular computacional (bioinformática estrutural, triagem virtual reversa, modelagem, docagem e dinâmica molecular).

Leticia M. Possamai (lmpossamai@gmail.com)

Graduada em Biomedicina pela Universidade do Vale do Taquari – Univates (2017). Atualmente está realizando curso de especialização em Hematologia Clínica pela FEEVALE, RS.

Luis F.S.M Timmers (luis.timmers@univates.br)

Biólogo pela Pontifícia Universidade Católica do Rio Grande Sul (PUCRS), Mestre e doutor em Biologia Celular e Molecular pela mesma Universidade. Atualmente, é professor na Universidade do Vale do Taquari (Univates) e membro do corpo permanente do Programa de Pós-Graduação em Biotecnologia (PPGBiotec).

Rafael Andrade Caceres (rafaelca@ufcspa.edu.br)

Químico pela Universidade Luterana do Brasil, Mestre pelo Programa de Biologia Celular e Molecular, Doutor pelo Programa de Pós-Graduação em Medicina, ambos pela Pontifícia Universidade Católica do Rio Grande Sul (PUCRS). Atualmente é professor adjunto na UFCSPA e membro do corpo permanente do Programa de Pós-Graduação em Biociências e Programa de Pós-Graduação em Ciências da Saúde, pela mesma Universidade.

12 APLICAÇÕES DE REDES NEURAIS

Rafael Vieira Coelho (rafael.coelho@farroupilha.ifrs.edu.br)

Graduado em Engenharia de Computação (2008) pela Fundação Universidade Federal do Rio Grande, Mestre em Ciência da Computação pela Universidade Federal do Rio Grande do Sul (2011), Doutor em Biotecnologia pela Universidade de Caxias do Sul (2018).

Gabriel Dall'Alba (gdalba@ucs.br)

Graduado em Ciências Biológicas pela Universidade de Caxias do Sul (2019). Atualmente é mestrando pela *University of British Columbia*. Possui experiência em Bioinformática, Biologia Molecular e aplicação de técnicas de inteligência artificial para predição de promotores.

Scheila de Avila e Silva (sasilva6@ucs.br)

Graduada em Gestão da Tecnologia da Informação pela Unisinos (2014) e em Ciências Biológicas pela UCS (2004). Possui mestrado em Computação Aplicada pela Unisinos (2007) e doutorado em Biotecnologia pela UCS (2011). Possui experiência em análise de dados, integração de bases de dados biológicas e aplicação de técnicas de inteligência artificial em dados genômicos.

13 ANÁLISE DE DADOS DE EXPRESSÃO GÊNICA

Ivaine Sauthier (ivaine.sauthier@gmail.com)

Graduada em Ciências Biológicas pela Universidade de Caxias do Sul, Mestre em Bioquímica e Doutora em Genética e Biologia Molecular pela Universidade Federal do Rio Grande do Sul. Tem experiência em análises de dados ômicos: genômica, transcriptômica e metiloma; busca novos biomarcadores e alvos terapêuticos em tumores gastrintestinais.

Marcos Vinicius Rossetto (mvrossetto@ucs.br)

Bacharel em Sistemas de Informação pela universidade de Caxias do sul (2016). Mestre em Biotecnologia pela Universidade de Caxias do Sul (2017-2019). A experiência acadêmica está relacionada com pesquisas de desenvolvimento de *softwares* aplicados à análise de dados biológicos. O desenvolvimento mais recente trata-se de um *software* de análise de dados de expressão gênica obtidos em repositórios públicos.

14 ANÁLISE DE IMAGENS BIOLÓGICAS

Lucas Picinini Dutra (lpduttra@ucs.br)

Graduado em Ciência da Computação pela Universidade de Caxias do Sul (2018), com ênfase em Sistemas de Informação e Inteligência Artificial. Atualmente é discente no Programa de Pós-Graduação em Engenharia de Produção na Universidade de Caxias do Sul.

Iago dos Passos (ipassos@ucs.br)

Graduado em Ciência da Computação pela Universidade de Caxias do Sul (2018), com ênfase em Sistemas de Informação e Inteligência Artificial. Atualmente é discente no Programa de Pós-Graduação em Engenharia de Produção na Universidade de Caxias do Sul.

André Luis Martinotto (almartin@ucs.br)

Graduado em Ciência da Computação pela Universidade de Caxias do Sul (2001), Mestre em Computação pela Universidade Federal do Rio Grande do Sul (2004) e Doutor em Ciências dos Materiais pela Universidade Federal do Rio Grande do Sul (2012).

15 ANOTAÇÃO GENÔMICA

Alexandre Rafael Lenz (arlenz@ucs.br)

Graduado em Ciência da Computação pela Universidade Luterana do Brasil (2007), Mestre em Informática pela Universidade Federal do Paraná (2009). É docente na Universidade do Estado da Bahia (UNEB) – Campus I, na cidade de Salvador – BA. Atualmente, é discente do Programa de Pós-Graduação em Biotecnologia na Universidade de Caxias do Sul – RS, atuando no Projeto de Sequenciamento e Análise do Genoma do Fungo *Penicillium echinulatum*.

16 APLICAÇÕES DE COMPUTAÇÃO PARALELA E DISTRIBUÍDA EM BIOINFORMÁTICA

Clodis Boscaroli (clodis.boscaroli@unioeste.br)

Professor Associado na Universidade Estadual do Oeste do Paraná, Ciência da Computação. Mestre em Informática pela Universidade Federal do Paraná (2002), Doutor em Engenharia Elétrica pela Universidade de São Paulo (2008), Especialista em Formulação e Gestão de Políticas Públicas pela Escola de Governo do Paraná, em parceria com a Universidade Estadual do Oeste do Paraná (2008).

Guilherme Galante (guilherme.galante@unioeste.br)

Professor nos cursos de Graduação e Pós-Graduação em Ciência da Computação da Unioeste, Doutor em Ciência da Computação pela UFPR (2014). Suas áreas de interesse são Computação Aplicada e Computação de Alto Desempenho.

Luiz Antonio Rodrigues (luiz.rodrigues@unioeste.br)

Doutor em Ciência da Computação, professor adjunto da Universidade Estadual do Oeste do Paraná, Ciência da Computação. Seus principais interesses: Ciência da Computação, com ênfase em Redes de Computadores, Tolerância a Falhas, Sistemas Distribuídos e Programação de Sistemas.

Sumário

<i>Prefácio</i>	12
<i>Apresentação</i>	13
A ERA DA INFORMAÇÃO Helena Graziottin Ribeiro (hgrib@ucs.br)	

Seção I – Fundamentos Computacionais

1 PORTAIS E BANCO DE DADOS: DEFINIÇÕES COMPUTACIONAIS	19
Gabriele Dani (gdani@ucs.br) Leonardo Pelizzon (leonardo.pelizzoni@gmail.com) Gustavo Sganzerla Martinez (sganzerlagustavo@gmail.com)	
2 BIOESTATÍSTICA	40
Cintia Paese Giacomello (cintia.paese@ucs.br)	
3 DATA MINING	58
Gabriele Dani (gdani@ucs.br) Marcelo Sachet (marcelosachet@gmail.com) Scheila de Avila e Silva (sasilva6@ucs.br)	
4 REDES NEURAIS ARTIFICIAIS: INTRODUÇÃO E DEFINIÇÕES	73
Scheila de Avila e Silva (sasilva6@ucs.br) Rafael Vieira Coelho (rafael.coelho@farroupilha.ifrs.edu.br)	
5 ANÁLISE POR AGRUPAMENTOS	90
André Gustavo Adami (agadami@ucs.br) Adriana Miorelli Adami (amiorell@ucs.br)	
6 INTRODUÇÃO ÀS MÁQUINAS DE VETORES DE SUPORTE	113
Lucas Picinini Dutra (lpduttra@ucs.br) Iago dos Passos (ipassos@ucs.br) André Luis Martinotto (almartin@ucs.br)	
7 COMPUTAÇÃO PARALELA E DISTRIBUÍDA	131
Alex A. L. dos Santos (allsant1@ucs.br) Felipe S. Raota (fsraota@ucs.br) Guilherme T. Paz (gtelespaz@ucs.br) Marcelo Brazil (marcelo@marcelo-brazil.com) André L. Martinotto (almartin@ucs.br)	

Seção II – Aplicações

- 8 PORTAIS E BANCOS DE DADOS BIOLÓGICOS..... 146**
Gustavo Sganzerla Martinez (sganzerlagustavo@gmail.com)
- 9 FERRAMENTAS DE ANÁLISE E PROCESSAMENTO DE METAGENOMAS.. 166**
Tahila Andrighetti (tahilaandrighetti@gmail.com)
- 10 BIOLOGIA DE SISTEMAS..... 197**
Daniel Luis Notari (dlnotari@ucs.br)
Diego Luis Bonatto (diegobonatto@gmail.com)
- 11 BIOLOGIA ESTRUTURAL 209**
Bruna Schuck de Azevedo (bruna.schuck4@gmail.com)
Leticia M. Possamai (lmpossamai1@gmail.com)
Luis F.S.M Timmers (luis.timmers@univates.br)
Rafael Andrade Caceres (rafaelca@ufcspa.edu.br)
- 12 APLICAÇÕES DE REDES NEURAIS 222**
Rafael Vieira Coelho (rafael.coelho@farroupilha.ifrs.edu.br)
Gabriel Dall’Alba (gdalba@ucs.br)
Scheila de Avila e Silva (sasilva6@ucs.br)
- 13 ANÁLISE DE DADOS DE EXPRESSÃO GÊNICA 236**
Ivaine Sauthier (ivaine.sauthier@gmail.com)
Marcos Vinicius Rossetto (mvrossetto@ucs.br)
- 14 ANÁLISE DE IMAGENS BIOLÓGICAS..... 247**
Lucas Picinini Dutra (lpduttra@ucs.br)
Iago dos Passos (ipassos@ucs.br)
André Luiz Martinotto (almartin@ucs.br)
- 15 ANOTAÇÃO GENÔMICA..... 259**
Alexandre Rafael Lenz (arlenz@ucs.br)
- 16 APLICAÇÕES DE COMPUTAÇÃO PARALELA E DISTRIBUÍDA
EM BIOINFORMÁTICA 285**
Clodis Boscarioli (clodis.boscarioli@unioeste.br)
Guilherme Galante (guilherme.galante@unioeste.br)
Luiz Antonio Rodrigues (luiz.rodrigues@unioeste.br)

Prefácio

Philip E. Bourne, no prefácio da quarta edição do livro *Bioinformatics*, cautelosamente escolhe a palavra *sinergia* para definir a Bioinformática: ao olharmos para o histórico da área, encontramos uma relação – talvez sinérgica – entre o experimento científico, o dado biológico, a computação e o desenvolvimento tecnológico. Por outro lado, há quem considere essa dinâmica como uma “competição saudável”, onde cada passo dado – cada conhecimento gerado – em cada um dos elementos acima destacados motiva o desenvolvimento dos demais

Independente da escolha de uma palavra-chave caricata da Bioinformática, podemos observar o que esses avanços relacionados à ela nos oferecem. Através do seu caráter interdisciplinar e dependente da integração de distintas áreas e competências, vemos o sequenciamento de múltiplos genomas em questão de minutos e com custos em declínio; a possibilidade de analisar e integrar mecanismos moleculares às suas rotas metabólicas, às células onde as encontramos e os seus efeitos em indivíduos de uma população; o desenvolvimento de soluções computacionais para as problemáticas de cunho biológico, refinando a confiabilidade que podemos ter ao fazer uso destas soluções. Estes são poucos exemplos dos inúmeros que podemos creditar aos esforços dos cientistas envolvidos com a Bioinformática – e que graças à eles, continua a sua expansão.

Organizamos este livro com o intuito de aproximar os estudantes e profissionais da biologia, da computação e demais correlatas áreas à Bioinformática, ofertando um panorama dos seus conceitos e aplicações através da competência de professores e pesquisadores dos diversos campos que a compõem. Dividimos o livro em duas seções: os *Fundamentos Computacionais* e as *Aplicações*. Assim, o leitor encontrará primeiro a conceitualização técnica de metodologias e abordagens computacionais, seguido da maneira como estas são empregadas na pesquisa acadêmica. Encontrará, além disso, um referencial atualizado e exemplário da literatura sobre a Bioinformática.

Desejamos que este livro fomente ainda mais o interesse pela Bioinformática, assim como sirva de exemplo e referencial teórico sobre o papel importante que esta área que vem desempenhando – e continuará a desempenhar – na pesquisa, bem como do seu potencial.

Prof. Dr. Daniel Notari
Prof. Dr^a Scheila de Avila e Silva
Gabriel Dall’Alba

A ERA DA INFORMAÇÃO

Helena Graziottin Ribeiro*

Estamos na era da informação. Ou seria era do conhecimento? Ambas, mas também era da tecnologia, que torna acessível a informação, que dissemina o conhecimento a quem quiser buscá-lo, que viabiliza mais produção de conhecimento e que diminui, ou até elimina, as distâncias físicas como obstáculo para as colaborações que possibilitam essa produção. Em tempos de redes sociais como meio usual de interação, de geração e armazenamento de grandes volumes de dados (*big data*) e popularização do uso da Inteligência Artificial, aplicada nas mais diferentes áreas, este livro aborda a bioinformática e suas aplicações. Afinal, há um volume massivo de dados biológicos armazenados e sendo gerados continuamente, brutos ou organizados em bases de dados, e produção constante de métodos, estruturas e ferramentas computacionais para tratá-los. Então é necessário explicar alguns desses temas que vêm da área da computação e da estatística, e entender suas aplicações na biologia.

Segundo (ARAUJO, 2008), apesar da estrutura do DNA ter sido desvendada em 1953 pelos ingleses Watson e Crick, foi preciso esperar até meados da década de 1980, para que fosse desenvolvida uma lente de aumento suficientemente boa (uma máquina automatizada), que permitisse a leitura em grandes quantidades do código genético contido nas biomoléculas. Com a quantidade crescente de dados gerados continuamente, há muito tempo tornou-se impraticável analisar sequências de DNA manualmente. Para um genoma grande como o genoma humano, pode-se demorar vários dias de processamento aplicando muita memória, mesmo utilizando computadores com múltiplos processadores para montar os fragmentos, e o conjunto resultante geralmente apresenta muitas lacunas que devem ser preenchidas depois. Do lado da computação, foi também necessário um amadurecimento desde os primeiros equipamentos desenvolvidos, uma evolução da tecnologia ao longo dos anos, para disponibilizar computadores acessíveis a todos os públicos, capazes de armazenar cada vez mais informação e de processá-la de modo rápido e a custos reduzidos.

Mas o que é bioinformática? Há definições, sob a ótica de diferentes autores, como a de (LESK, 2019): “A bioinformática e a biologia computacional envolvem a análise de dados biológicos, particularmente sequências de DNA, RNA e proteínas. Os dados clássicos da bioinformática incluem sequências de DNA de genes ou genomas

* Universidade de Caxias do Sul. *E-mail*: hgrib@ucs.br

completos, sequências de aminoácidos de proteínas, e estruturas tridimensionais de proteínas, de ácidos nucleicos e de complexos de proteínas-ácidos nucleicos”. De forma geral, entende-se que a bioinformática usa computação para compreender melhor a biologia (SPENGLER, 2000). Mas não é simplesmente a teoria da informação aplicada à biologia, nem é uma miscelânea de técnicas de computação para construir, atualizar e acessar dados biológicos. Em vez disso, a bioinformática incorpora esses dois conjuntos de recursos em uma ciência interdisciplinar ampla, que envolve ferramentas conceituais e práticas para o entendimento, a geração, o processamento e a propagação de informações biológicas. Como campo interdisciplinar da Ciência, a bioinformática combina diversas áreas, tais como: biologia, ciência da computação, engenharia da informação, matemática e estatística para analisar e interpretar dados biológicos.

A bioinformática experimentou um crescimento em grande escala, a partir da década de 1990, impulsionado em grande parte pelo Projeto Genoma Humano e pelos rápidos avanços na tecnologia de sequenciamento de DNA. Uma enorme quantidade de dados biológicos foi gerada na última década, particularmente após o advento das tecnologias de sequenciamento de próxima geração (GRENNE *et al.*, 2017; MARTIZ *et al.*, 2016). Em 1999, os arquivos de sequência de ácidos nucleicos continham um total de 3,5 bilhões de nucleotídeos, um pouco mais do que o comprimento de um único genoma humano; uma década depois, continham mais de 283 bilhões de nucleotídeos, o comprimento de cerca de 95 genomas humanos. Há grandes *datasets* como *The Cancer Genome Atlas* (TCGA) e *The Encyclopedia of DNA Elements* (ENCODE). Apesar de seu tamanho considerável, esses *deep datasets* produzidos por grandes consórcios são ofuscados por amplos compêndios de dados produzidos nos laboratórios de pesquisadores individuais e disponibilizados mediante publicação. Por exemplo, o *Array Express*, um compêndio de dados de expressão gênica disponíveis publicamente, contém mais de 1,3 milhão de testes genômicos de mais de 45.000 experimentos.

Na bioinformática, como em outras áreas que produzem e exploram grandes volumes de dados, utilizam-se bancos de dados para armazenar e organizar dados. Há muitas bases de dados organizadas e mantidas por consórcios internacionais, como, por exemplo, *Nucleotide Sequence Database* (EMBL-Bank) no Reino Unido; o Banco de Dados de DNA do Japão (DDBJ), e *GenBank* do Centro Nacional de Informações sobre Biotecnologia (NCBI) (LESK, 2019). Dados biológicos processados geram mais dados biológicos, o ritmo é frenético. A alta taxa de geração de dados biológicos coloca a bioinformática na era do Big Data. Técnicas e algoritmos de armazenamento e análise convencionais não são adequadas para armazenar e tratar grandes volumes de dados. Essas questões são discutidas nos **Capítulos 1: Portais e Banco de Dados e no 8: Portais e Bancos de Dados Biológicos.**

Mas, após armazenar é preciso processar e explorar esse volume enorme de dados. Analisar dados biológicos para produzir informações significativas envolve escrever e executar programas de *software* que usam algoritmos de diferentes áreas da computação: da teoria dos grafos, da inteligência artificial, do processamento de imagens, da modelagem e simulação, do processamento paralelo e distribuído.

A simulação de sistemas muitas vezes exige muito processamento de dados, sobrecarregando uma unidade de processamento, e esse processamento pode tornar-se demorado demais. Considerando que há um grande volume de dados a processar e que os modelos criados para representar os fenômenos naturais são bastante complexos, torna-se necessário utilizar algoritmos e recursos de processamento, que possam repartir, distribuir, tanto a carga de trabalho como a estrutura de processamento, para otimizar sua execução. O **Capítulo 7** aborda Computação Paralela e Distribuída.

A Inteligência Artificial e as técnicas de mineração de dados tem sido muito importantes como recursos de exploração e análise de volumes de dados. Uma das áreas da Inteligência Artificial é a de Aprendizagem de Máquina. Em 1959, Arthur Lee Samuel, professor e pesquisador na Universidade de Stanford, definiu aprendizado de máquina como o campo de estudo que dá aos computadores a habilidade de aprenderem, sem ser explicitamente programados. O aprendizado automático compreende a construção de algoritmos que podem aprender com seus erros e fazer previsões sobre dados. A exploração efetiva do grande conjunto de dados disponíveis atualmente, por meio de análise estatística direcionada (por exemplo, a construção de testes estatísticos de controle e condições experimentais), é impossível (GREENE *et al.*, 2016). A quantidade de dados a serem explorados extrapola a capacidade de um indivíduo de executar todos os testes possíveis, e muito menos de interpretar os resultados deles. As estratégias do aprendizado de máquina estão preenchendo essa lacuna. Elas compreendem algoritmos computacionais capazes de identificar padrões importantes em grandes compêndios de dados. No campo da análise de dados, o aprendizado de máquina é um conjunto de métodos e algoritmos usados para fazer previsões (análise preditiva). Esses modelos analíticos permitem que pesquisadores, cientistas de dados, engenheiros, e analistas possam “produzir decisões e resultados confiáveis e repetíveis” e descobrir os “insights escondidos” no aprendizado das relações e tendências históricas nos dados.

O aprendizado de máquina é muitas vezes confundido com mineração de dados, pois, com frequência, eles fazem uso dos mesmos métodos. Porém, enquanto o aprendizado de máquina foca em fazer previsões, baseado em propriedades já conhecidas aprendidas a partir do uso de dados de treinamento, a mineração de dados foca em descobrir as propriedades antes não conhecidas nos dados. A mineração de

dados é uma etapa do processo de descoberta de conhecimento nos dados, precedida pelas etapas de extração e preparação dos dados para serem minerados. O **Capítulo 3: Mineração de Dados**, aborda os principais métodos e algoritmos associados. A mineração de dados que pode ser feita a partir de vários métodos e algoritmos para realizá-los, como: classificação, regressão, clusterização e associação. A clusterização vai ser apresentada com detalhes no **Capítulo 5: Clusterização**.

Métodos utilizados para aprendizagem de máquina e para mineração de dados classificam-se em supervisionados e não supervisionados. O desafio de descobrir subtipos moleculares é um problema que é melhor abordado através de métodos não supervisionados (GREENE *et al.*, 2016). Por exemplo, dadas as medições de expressão gênica para um conjunto de cânceres, queremos saber se existem padrões de expressão de genes compartilhados. Muitos algoritmos de clusterização existem, mas aqueles que dividem os dados em um número predefinido de grupos (*clusters*) são mais comumente usados para descobrir subtipos de câncer.

Métodos não supervisionados têm sido usados para separar o genoma humano em diferentes segmentações funcionais com base nos padrões de modificação de histonas do genoma do projeto ENCODE (*The Encyclopedia of DNA Elements*). As modificações específicas das histonas correlacionam-se com a ligação do fator de transcrição, iniciação e alongamento da transcrição, atividade intensificadora e repressão.

Métodos supervisionados podem ser direcionados para a construção de regras de decisão que separam exemplos (por exemplo, genes) de duas ou mais classes (por exemplo, está no *pathway* ou não está no *pathway*). Essa tarefa é chamada de tarefa de “classificação” e os métodos comumente usados são a classificação por máquinas de suporte vetorial (SVM) ou a regressão logística penalizada. Os métodos podem classificar, por exemplo, em “padrões positivos”, que representam itens que queremos que o algoritmo descubra e muitas vezes “padrões negativos”, que representam itens que gostaríamos que os algoritmos evitassem. **Máquinas de Suporte Vetorial** são apresentadas em detalhes no **Capítulo 6**.

Um algoritmo de aprendizado de rede neural artificial, normalmente chamado de “rede neural” (RN), é um algoritmo de aprendizado que é inspirado na estrutura e nos aspectos funcionais das redes neurais biológicas. Normalmente, elas são usadas para modelar relações complexas entre entradas e saídas, para encontrar padrões nos dados, ou para capturar a estrutura estatística em uma distribuição de probabilidade conjunta desconhecida entre variáveis observáveis. O aprendizado automático é usado em uma variedade de tarefas computacionais onde criar e programar algoritmos explícitos é impraticável. Redes Neurais Artificiais aprendem a partir dos exemplos recebidos e

possuem uma capacidade de generalizar este aprendizado para dados ainda não tratados.

Redes Neurais Artificiais são abordadas no **Capítulo 4**.

Os diferentes algoritmos de mineração e aprendizagem, por sua vez, dependem de fundamentos teóricos, como a matemática discreta, a teoria de controle, a teoria do sistema, a teoria da informação e a estatística. Os biólogos moleculares têm criado reais oportunidades para utilizar métodos estatísticos capazes de analisar grandes quantidades de dados biológicos, de inferir funções dos genes e de estabelecer relações estruturais entre genes e proteínas. Algumas partes do aprendizado automático estão intimamente ligadas (e muitas vezes sobrepostas) à estatística computacional; uma disciplina que foca em como fazer previsões através do uso de computadores, com pesquisas focando nas propriedades dos métodos estatísticos e sua complexidade computacional. Ela tem fortes laços com a otimização matemática, que produz métodos, teoria e domínios de aplicação para este campo. A compreensão de métodos estatísticos aplicados à biologia é abordada no **Capítulo 2: Bioestatística**.

Os **Capítulos 9, 10, 11, 12, 13, 14, 15 e 16** apresentam algumas aplicações desses métodos e estruturas na área biológica, que é o que se define a bioinformática.

Referências

GREENE, C. S., TAN, J.; UNG, M.; MOORE, J. H.; CHENG, C. **Big Data bioinformatics**. Disponível em PubMed Central® (PMC) em 19 de setembro de 2017. doi: 10.1002/jcp.24662. Disponível em: <https://www.ncbi.nlm.nih.gov/pmc/articles/PMC5604462/>. Acesso em: 7 abr. 2019.

LESK, A. M. Bioinformatics. **Encyclopaedia Britannica**. Disponível em: <https://www.britannica.com/science/bioinformatics>. Acesso em: 7 abr. 2019.

MARTIZ, R.; SUPAKSHA, M A.; HEMALATHA, N. Application of big data in bioinformatics – a survey. **International Journal of Latest Trends in Engineering and Technology (IJLTET)** Special Issue SACAIM 2016, p. 206-212 e-ISSN:2278-621X. Disponível em <https://www.ijltet.org/journal/147905351034.pdf>. Acesso em: 7 abr. 2019.

SPENGLER, S. J. Bioinformatics in the information age. **Office of Scientific & Technical Information Technical Reports**, Digital Library of UNT .February 1, 2000. Disponível em: <https://digital.library.unt.edu/ark:/67531/metadc785111/>. Acesso em: 7 maio 2019.

ARAUJO, N. D. et al. A era da bioinformática: seu potencial e suas implicações para as ciências da saúde. **Revista Estudos de Biologia (PUCPR)**, 30(70/71/72), p. 143-148, jan./dez. 2008. Disponível em: <https://periodicos.pucpr.br/index.php/estudosdebiologia/article/download/22819/21922>. Acesso em: 7 maio 2019.

Seção I – Fundamentos Computacionais

1

PORTAIS E BANCO DE DADOS: DEFINIÇÕES COMPUTACIONAIS

Gabriele Dani,¹ Leonardo Pellizzoni,² Gustavo Sganzerla³

1 O que são portais e banco de dados

Portais são tipicamente tratados como *sites* disponíveis para acesso de conteúdo através de um navegador. A disponibilidade dos portais pode variar de acordo com o propósito, podendo estar disponíveis publicamente ou restritos, em um ambiente interno onde são controlados aqueles que podem ou não usá-los. De maneira abrangente, um portal pode ter serviços genéricos como *e-mail*, grupo de discussão e mecanismos de pesquisa (O BRAIN *et al.*, 2013). Existem portais desenvolvidos com propósitos específicos para um campo de pesquisa ou área; ali são criadas funcionalidades que agreguem valor especificamente ao público-alvo daquele portal. Além de agregar valor através das funcionalidades, os portais muitas vezes servem como uma ponte entre usuário e uma base que contenha dados relevantes ao contexto a ser utilizado.

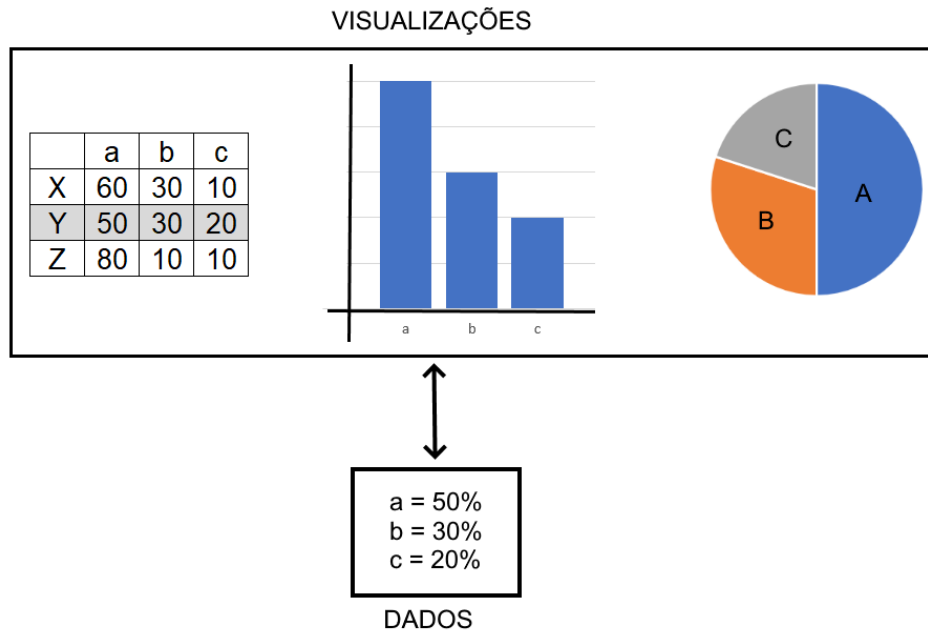
Através de portais que acessam banco de dados é possível, por meio de diferentes funcionalidades, desenvolver múltiplas visualizações das informações armazenadas, como ilustra a Figura 1. Além de diferentes visualizações dos dados, é viável em algumas situações criar processamentos automáticos e inteligentes nos portais, para facilitar o uso das informações contidas nos bancos de dados.

¹ Universidade de Caxias do Sul. *E-mail*: gdani@ucs.br

² Universidade de Caxias do Sul. *E-mail*: leonardo.pellizzoni@gmail.com

³ Universidade de Caxias do Sul. *E-mail*: sganzerlagustavo@gmail.com

Figura 1 – Mesmos dados sendo organizados de acordo com a necessidade



Fonte: Adaptado de Gamma *et al.* (2000).

É viável afirmar que bancos de dados representam uma parte crucial na utilização, elaboração e popularização de sistemas de informação em múltiplas áreas do conhecimento. Através de sua estrutura, é possível armazenar de forma consistente, segura e íntegra os dados do mundo real organizados de forma a agregar valor aos usuários destes sistemas de informação. Bancos de dados são criados, organizados e mantidos por um conjunto de programas, conhecido como Sistema Gerenciador de Banco de Dados (SGBD). Além do conjunto inicial de operações realizadas pelo SGBD, ele é responsável pelo compartilhamento das informações entre diversos usuários, suportando diferentes níveis de segurança e controles de falhas (ELMARSÍ, NAVATHE, 2018).

Os portais podem ser caracterizados em dois segmentos que são verticais e horizontais. Os portais verticais dedicam-se principalmente aos ramos de recursos humanos, finanças, CRM (Customer relationship management) e ERP (Enterprise Resource Planning). Através desta modalidade, é viável que os portais sejam utilizados por usuários que participam do negócio direta e indiretamente, permitindo que visualizem, editem e adicionem novas informações (BALTZAN, PHILLIPS, 2012).

Um portal horizontal possui uma característica de agregador de conteúdos de diferentes sistemas; indicadores de performance são tipicamente tratados em portais horizontais. Em alguns cenários, os portais horizontais não requerem usuário e senha para acesso; nestes casos, o acesso, muitas vezes, apenas é disponibilizado em ambientes de redes internas (REYNOLDS; STAIR, 2015).

2 Evolução e os tipos de portais e bancos de dados

Os bancos de dados fazem parte da rotina diária de todos, até mesmo pessoas que não possuem um computador ou telefone celular interagem com eles regularmente. Quando tiramos dinheiro de um caixa eletrônico, verificamos nosso saldo bancário, fazemos compras *online*, visualizamos as mídias sociais ou realizamos praticamente qualquer interação digital, estamos acessando um banco de dados.

“Provavelmente o termo mais incompreendido em toda a computação empresarial é o banco de dados, seguido de perto pela palavra relacional” (HARRINGTON, 2016). Graças a uma massa de desinformação, muitos empresários e trabalhadores de tecnologia têm a falsa impressão de que projetar e implementar bancos de dados é uma tarefa simples que a equipe administrativa pode facilmente executar. Projetar e implementar um banco de dados é um grande desafio que requer análise das necessidades de uma organização e um planejamento e implementação cuidadosos.

Algumas pessoas afirmam que bancos de dados estruturados tradicionais são coisa do passado. Embora isso possa ser verdade de algumas perspectivas (por exemplo, para desenvolvedores com *sites* que têm milhões de usuários em áreas como mídias sociais), para o resto de nós, bancos de dados estruturados ainda fazem parte de nossa vida. A mudança de requisitos e a evolução da internet fizeram com que novos tipos de bancos de dados surgissem, mas eles têm usos específicos.

Bancos de dados são essencialmente aplicativos de *software*. Um sistema de gerenciamento de banco de dados (DBMS) é o nome do *software* que fornece dados para outros aplicativos, permitindo que todos os sistemas de informações digitais, com os quais interagimos hoje. Muitas vezes, um DBMS é chamado de banco de dados. Existem muitos fornecedores e soluções de *software* com diferentes licenças e usos. Os dados são compartilhados com uma variedade de padrões, mas basicamente todos eles servem ao mesmo propósito, que é fornecer dados aos aplicativos. Os aplicativos, então, processam os dados e os transformam em algo útil para os usuários: informações.

2.1 História dos portais e bancos de dados

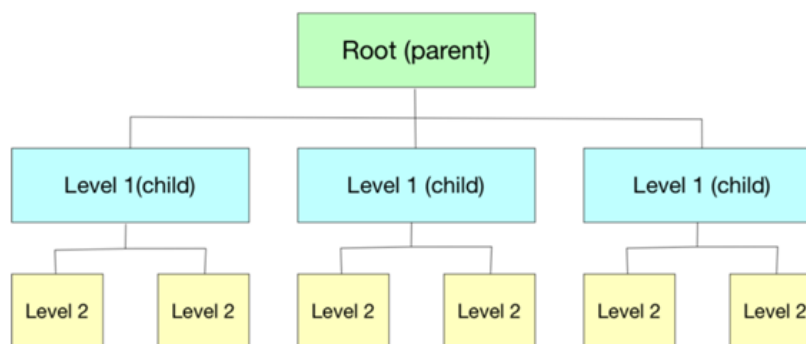
Antes da existência dos bancos de dados, tudo tinha que ser gravado no papel. Nós tínhamos listas, revistas, livros contábeis e arquivos intermináveis contendo centenas de milhares ou até milhões de registros contidos em arquivos. Quando era necessário acessar um desses registros, encontrar e obter fisicamente o registro, era uma tarefa lenta e trabalhosa. Muitas vezes, havia problemas que iam de registros equivocados a incêndios que dizimavam arquivos inteiros e destruíam a história de

sociedades, organizações e governos. Houve também problemas de segurança, porque o acesso físico era geralmente fácil de obter.

O banco de dados foi criado para tentar resolver essas limitações do armazenamento tradicional de informações baseadas em papel. Em bancos de dados, os arquivos são chamados de registros, e os elementos de dados individuais em um registro (por exemplo, nome, número de telefone, data de nascimento) são chamados de campos. A maneira como esses elementos são armazenados evoluiu desde os primeiros dias dos bancos de dados.

Os sistemas mais antigos eram chamados de modelos hierárquicos e de rede. O modelo hierárquico organizou os dados em uma estrutura de árvore, como mostrado na Figura 2. A IBM desenvolveu esse modelo na década de 1960.

Figura 2 – Modelo de um banco de dados hierárquico



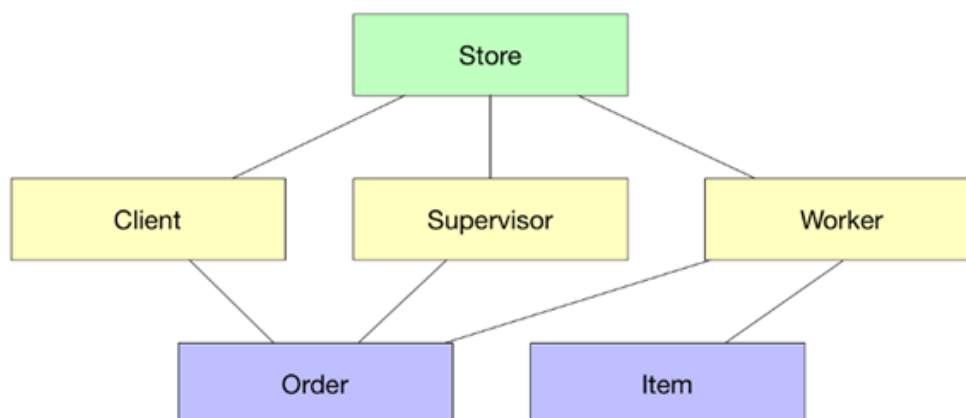
Fonte: Adaptado de Harrington (2016).

O modelo hierárquico representa dados como registros relacionados a *links*. Cada registro tem um registro-pai, começando com o registro-raiz. Este é possivelmente o modelo mais simples de entender, porque temos muitas hierarquias no mundo real – em organizações, nos militares, governos e nas escolas. Registros no modelo hierárquico continham um campo. Para acessar os dados usando este modelo, toda a árvore tinha que ser atravessada. Esses tipos de banco de dados ainda existem hoje e têm um lugar no desenvolvimento, apesar dos avanços significativos na tecnologia. Eles são, por exemplo, usados pela Microsoft no Registro do Windows e em sistemas de arquivos, e podem ter vantagens sobre modelos de banco de dados mais modernos (velocidade e simplicidade). No entanto, também há muitas desvantagens, sendo que a principal é que elas não representam facilmente relacionamentos entre tipos de dados. Isso pode ser obtido por meio de métodos bastante complexos (usando registros “fantasmas”). Para isso, o *designer* de banco de dados deve ser um especialista que compreenda o funcionamento fundamental desses sistemas.

O banco de dados hierárquico resolveu muitos dos problemas mencionados acima. Registros podem ser acessados quase instantaneamente. Ele também tinha um mecanismo completo de *backup* e recuperação que significava que o problema de arquivos perdidos devido a danos era algo do passado.

Em 1969, cientistas da Conferência sobre Linguagens de Sistemas de Dados (CODASYL) divulgaram uma publicação que descrevia o modelo de rede. Foi a próxima inovação significativa em bancos de dados. Superou as restrições do modelo hierárquico. Como mostrado na Figura 3, esse modelo permite relacionamentos e possui um “esquema” (uma representação diagramática dos relacionamentos).

Figura 3 – O modelo de banco de dados de rede



Fonte: Adaptado de Harrington (2016).

A principal diferença entre o modelo hierárquico e o modelo de rede é que o modelo de rede permite que cada registro tenha mais de um registro pai e filho. Na Figura 3, o “Cliente”, “Supervisor” e outras caixas representam o que na terminologia do banco de dados são chamadas de entidades. O modelo de rede permite que as entidades tenham relacionamentos, assim como na vida real.

O modelo de rede melhorou o modelo hierárquico, mas não se tornou dominante. A principal razão para isso é que a IBM continuou a usar o modelo hierárquico em seus produtos mais estabelecidos (IMS e DL / 1) e os pesquisadores criaram o modelo relacional. O modelo relacional era muito mais fácil para os *designers* entenderem e a interface de programação era melhor. Os modelos de rede e hierárquicos foram utilizados ao longo dos anos 60 e 70, porque ofereceram melhor desempenho. Os sistemas de computadores mainframe usados nos anos 60 e 70 precisavam das soluções mais rápidas possíveis, porque o *hardware* era extremamente limitado. No entanto, a década de 1980 assistiu a enormes avanços na tecnologia da computação e o modelo relacional começou a se tornar o mais popular.

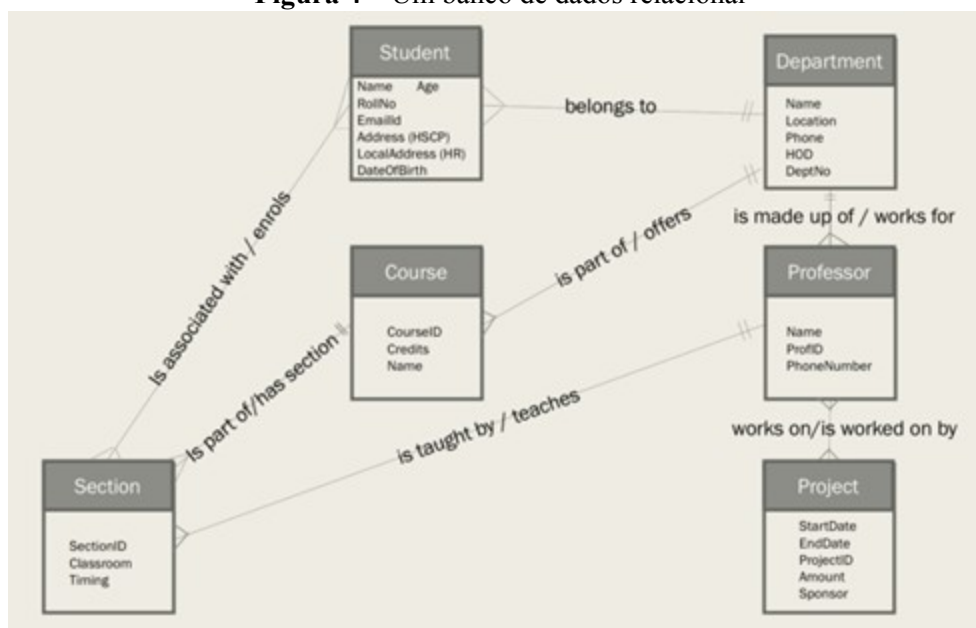
O modelo relacional foi, como o modelo de rede, descrito em uma publicação em 1969. O modelo relacional descreve os dados em um banco de dados como sendo armazenados em tabelas, cada uma contendo registros com campos.

O tipo de dados para cada campo é predeterminado (por exemplo, texto, número, data) e isso ajuda a garantir que não haja inconsistências e a saída é o que os aplicativos precisam (ajuda, por exemplo, a determinar como classificar os dados). Essas tabelas podem ter relacionamentos em um banco de dados relacional e existem diferentes tipos de relacionamentos.

Isso permite que o *designer* mostre como uma tabela se relaciona com outra. Por exemplo, um cliente provavelmente comprará muitos produtos. Portanto, um cliente pode estar associado a muitos produtos (esse é um relacionamento um-para-muitos).

Os relacionamentos podem ser obrigatórios (ou não), e isso ajuda a manter a integridade de um banco de dados. Por exemplo, se um produto deve ser associado a um fabricante para existir em um banco de dados, pode existir uma regra que permita somente a adição de produtos se eles tiverem um fabricante associado. Isso significa que há menos espaço para erro, quando o banco de dados é implantado. A Figura 4 mostra um *design* típico de banco de dados relacional.

Figura 4 – Um banco de dados relacional



Fonte: Adaptado de Harrington (2016).

A maioria dos bancos de dados relacionais usa um método padrão para acessar os dados: A Linguagem de Consulta Estruturada (SQL – Structured Query Language). O SQL permite que um aplicativo obtenha acesso aos dados necessários por um usuário. Ele pode recuperar todos os dados de uma tabela (ou até mesmo um banco de dados) ou

apenas um campo individual, determinado por um conjunto de critérios. Por exemplo, um aplicativo pode exigir apenas o nome de um professor associado a um curso (do banco de dados mostrado na Figura 4) e eles podem não precisar de mais dados das tabelas.

A principal vantagem do modelo relacional é que ele fornece consistência nos dados. O modelo implementa um conjunto de restrições e estas garantem que o banco de dados funcione conforme pretendido. As relações e restrições resultantes são desenvolvidas através do estudo do ambiente, no qual o banco de dados opera. É uma das principais razões que o *design* de banco de dados não é tão simples quanto a maioria das pessoas pensa. Os relacionamentos do mundo real entre as entidades devem ser determinados para que o banco de dados funcione corretamente. Essa análise envolve o estudo dos sistemas de registro anteriores em papel e a entrevista de funcionários e fornecedores em uma organização. Os gerentes de projeto ou analistas devem fazer uma análise rigorosa e completa dos requisitos, antes que um banco de dados possa ser preenchido e usado. Ele garante que um sistema não poderá fazer nada que cause erros ou represente incorretamente a situação real dos dados.

1980-1990

Desde que o modelo relacional foi criado, no final dos anos 1960, pouco mudou. As empresas modernas ainda usam esses sistemas para registrar suas atividades cotidianas e ajudá-las a tomarem decisões estratégicas críticas. As empresas de banco de dados estão entre as maiores e mais lucrativas organizações do mundo, e as empresas fundadas nos anos 60 e 70 ainda estão prosperando hoje.

O identificador-chave para um banco de dados tradicional é o tipo de dados que ele manipula. Ele contém dados que são consistentes e nos quais a natureza fundamental não muda significativamente com o tempo.

Em 1977, Larry Ellison, Bob Miner e Ed Oates formaram uma empresa na Califórnia. Eles pretendiam criar um banco de dados compatível com o System R. Essa empresa foi chamada de Oracle Systems Corporation em 1982. A Oracle continuaria sendo o maior e mais lucrativo fornecedor de banco de dados do mundo. Eles desenvolveram seu *software* com a linguagem de programação C, o que significava que ele era de alto desempenho e poderia ser portado para qualquer plataforma que suportasse C.

Na década de 1980, havia mais concorrência no mercado, mas a Oracle continuava a dominar. No final dos anos 80, a Microsoft desenvolveu um banco de dados para a plataforma OS / 2, chamada SQL Server 1.0. Em 1993, eles portaram isso para a plataforma Windows NT e, devido à adoção da tecnologia Windows na época,

tornou-se o padrão para pequenas e médias empresas. O ambiente de desenvolvimento que a Microsoft criou, em meados dos anos 90 (visual basic e .NET), significou que qualquer pessoa, não apenas desenvolvedores experientes de longo prazo, poderia aproveitar o poder dos bancos de dados em seus aplicativos. Em 1998, eles lançaram o SQL Server V7 e o produto estava maduro o suficiente para competir com os *players* mais estabelecidos no mercado.

No início dos anos 90, havia outro banco de dados criado que teria um efeito mais significativo do que qualquer outro, pelo menos para o mercado *online*. Em meados da década de 1990, houve uma revolução no desenvolvimento de *software*. Ele surgiu para combater o domínio da Microsoft e o rígido controle do código usado na maioria dos sistemas de PC nos anos 90, e o movimento de código aberto nasceu. Eles não acreditavam em *software* comercial proprietário e, em vez disso, desenvolviam *software* livre e distribuível (além de ter o código disponível publicamente). Em 1995, a primeira versão do MySQL foi lançada por uma empresa sueca (que financiou o projeto de código aberto) – MySQL AB. Este *software* foi o primeiro banco de dados significativo da internet e continua a ser utilizado por empresas como o Google (embora não para pesquisa), Facebook, Twitter, Flickr e YouTube. A licença de código aberto deu liberdade aos desenvolvedores de *sites* e significou que eles não precisavam depender de empresas como Oracle e Microsoft. Também funcionou bem com outros *softwares* de código aberto, que criaram a base da internet que usamos hoje (Linux, Apache, MySQL e PHP (LAMP) tornou-se a configuração mais comum para *sites*). A MySQL AB (a empresa que patrocinou o projeto MySQL) acabou sendo adquirida pela Sun Microsystems, que foi posteriormente adquirida pela Oracle.

Nos anos seguintes, muitos outros bancos de dados de código aberto foram criados. Quando a Oracle adquiriu o MySQL, um fundador do projeto MySQL fez um *fork* do projeto (ou seja, ele pegou o código e iniciou um novo projeto com um nome diferente). Este novo projeto foi chamado MariaDB. Atualmente, existem vários bancos de dados de código aberto que possuem diferentes licenças e ideologias.

Post-2000 e NoSQL

Em 1998, um novo termo foi criado, chamado NoSQL. Refere-se a bancos de dados “não SQL”, que usam outras linguagens de consulta para armazenar e recuperar dados. Esses tipos de banco de dados existem desde a década de 1960, mas foi a revolução da Web 2.0 que os fez chamar a atenção do mundo da tecnologia.

A Web 1.0 foi a primeira iteração da internet, quando os usuários recebiam o conteúdo criado por *webmasters* e suas equipes. A Web 2.0 foi a mudança para o conteúdo gerado pelo usuário e uma internet mais amigável para todos. *Sites* como o

YouTube e as mídias sociais resumem essa fase da internet. Para bancos de dados, isso significava que as necessidades dos desenvolvedores e administradores tinham mudado. Havia uma grande quantidade de dados sendo adicionados à internet por usuários a cada segundo. A computação em nuvem desbloqueou capacidades massivas de armazenamento e processamento, e a forma como usamos os bancos de dados mudou.

Nesta era da tecnologia, os requisitos mudaram para a simplicidade em relação ao *design* e à escalabilidade, devido à natureza cada vez maior da nova internet. Também era essencial ter disponibilidade 24 horas por dia e sete dias por semana, e a velocidade passou a ter extrema importância. Os bancos de dados relacionais tradicionais se esforçavam particularmente com a escalabilidade e a velocidade necessárias e, devido ao NoSQL usar estruturas de dados diferentes (ou seja, valor-chave, gráfico, documento), era geralmente mais rápido. Eles também eram vistos como mais flexíveis, porque não tinham as mesmas restrições que os bancos de dados relacionais tradicionais.

Para a nova geração de desenvolvedores da Web, o NoSQL foi melhor. Foi uma das principais razões para as inovações massivas ocorridas nas duas primeiras décadas do século XXI, porque o desenvolvimento do *site* (e mais tarde o aplicativo) foi facilitado e conseguiu lidar com a crescente natureza da World Wide Web. Os bancos de dados relacionais continuaram a ter o seu lugar, apesar do afastamento deles no mundo *online*. As empresas ainda precisavam da confiabilidade, consistência e facilidade de programação para seus sistemas de negócios.

A Web porém continuou evoluindo. A partir da data de lançamento da World Wide Web, podemos ver muita evolução técnica e infraestrutural. A quantidade de usuários de internet aumentou muito diariamente, portanto, a Web e a internet foram redesenhadas e alteradas para acomodar os diferentes usuários e diferentes dispositivos em todos estes anos, desde 1990. Já vimos a Web 1.0 na Web 3.0 e Web 4.0 e Web 5.0 serão no futuro.

Como já abordamos brevemente, a web 1.0 (1990-2000) foi o primeiro estágio da World Wide Web e foi principalmente de leitura e estática. Isso significa que os usuários podiam ler os *sites* e seu conteúdo, mas não adicionar ou interagir com a maioria deles. Essa geração de Web também era conhecida como a Web de informações, isso entre os anos 1990 e 2000 principalmente. A Web 1.0 foi usada principalmente por empresas e *sites* pessoais para mostrar suas informações. No ano 2000, houve uma transição da Web 1.0 para a Web 2.0, que ficou conhecida como “web de leitura e escrita”.

A Web 2.0 (2000-2010) foi o segundo estágio da World Wide Web. Essa etapa ou geração também era conhecida como *The Social Web*, já que agora os usuários não só

conseguiam ler os *sites*, mas também podiam interagir e se conectar com outros usuários. A maioria das mídias sociais, como Blogs, Facebook, YouTube etc., surgiram na Web 2.0 e as empresas começavam a perceber os benefícios da interação da comunidade com os sites de negócios. A maioria das pessoas também começava a colaborar em ideias, compartilhar informações e gerar ou criar informações disponíveis para todo o mundo. A Web 2.0 esteve presente dos anos 2000 até 2010 e depois disso houve uma transição e evolução da Web 2.0 para a Web 3.0.

Web 3.0 (2010-2020) foi descrita pela primeira vez em 2006 como uma geração de Web que define dados organizados ou estruturados para simplificar a automação, a integração e a descoberta em vários aplicativos. Isso implica que a Web 3.0 não é apenas uma Web de leitura e gravação, mas também uma Web voltada para usuários e máquinas individuais.

A Web 3.0 também é conhecida como “A Web Semântica”, porque tenta representar o conhecimento de forma a permitir que os computadores cheguem automaticamente a conclusões e decisões, usando algum raciocínio ou dados. Enquanto a Web 2.0 se concentra principalmente em pessoas, a Web 3.0 é uma extensão que se concentra na conexão inteligente entre pessoas e máquinas. Por exemplo, o computador pode entender o histórico de pesquisa do Google de um usuário e, em seguida, fornece um anúncio e sugestões personalizados. Este tipo de Web podemos ver agora em todos os lugares no Google, Gmail, Facebook ou em outros. Esta terceira geração de Web deve existir até 2020.

Na Web 2.0, o foco estava na conexão e interação entre as pessoas, mas o foco da Web 3.0 é principalmente conectar pessoas com dispositivos que usam a internet, por isso chamamos de “Internet das Coisas”.

IoT significa a interconexão entre todos os dispositivos e a internet, para que eles possam enviar e receber dados. É considerado como um dos desenvolvimentos mais importantes no campo da internet e, finalmente, levará da Web 3.0 para a Web 4.0. Agora podemos ver vários aplicativos móveis baseados em IoT que conectam muitos dispositivos domésticos ou de escritório que interagem entre si por meio da internet.

A Inteligência Artificial é a tecnologia que torna o computador capaz de se comunicar, pensar, raciocinar, responder e se comportar como um ser humano. Isto é o que será visto na Web 4.0 e na Web 5.0.

A interação entre humanos e máquinas tem sido a motivadora do progresso no desenvolvimento das telecomunicações, avanço na nanotecnologia no mundo e interfaces controladas utilizando a Web 4.0. Em palavras simples, as máquinas seriam inteligentes em ler o conteúdo da Web e reagir na forma de executar e decidir o que executar primeiro para carregar rapidamente os *sites* com qualidade e desempenho

superiores. Por exemplo, se você visitar o Site amazon.com mais de uma vez, ele o reconhecerá e fornecerá conselhos relevantes e personalizados. Um dos desenvolvimentos mais importantes da Web 4.0 será a migração da funcionalidade *online* para o mundo físico. Para usar um dos exemplos mais simples, imagine poder pesquisar na sua casa o Google para localizar as chaves do seu carro ou o controle remoto.

3 A estrutura computacional de portais e bancos de dados

3.1 Modelagem de dados

A modelagem de bancos de dados permite uma visão sobre como os dados serão estruturados em um sistema a ser criado. Essa etapa fornece uma visão ampla de relacionamentos que podem existir entre diferentes classes de dados presentes em um sistema. A modelagem em si trata de uma forma conceitual de estruturar dados que serão armazenados em um banco de dados. Uma das grandes vantagens de realizar uma modelagem apropriada de dados é poder obter uma representação visual do que será trabalhado. Adicionalmente, a modelagem de dados entrega consistência através de convenções de nomes, segurança e semântica, garantindo qualidade aos dados (TEOREY, 2007).

Existem duas técnicas bastante comuns na modelagem de dados:

- *Unified Modeling Language* (UML), trata-se de uma coleção de artefatos que auxilia na padronização da modelagem de dados e documentação de sistemas. UML conta com uma série de diagramas que ajudam na representação gráfica ao modelar um sistema e fornece uma forma bastante útil de comunicação entre uma equipe de desenvolvimento;
- *Modelo Entidade-Relacionamento* (ER), que atua como um fluxograma com as diferentes entidades (objetos de um sistema, pessoas, conceitos) que se relacionam entre si. Geralmente, um modelo ER é um dos primeiros passos ao modelar um banco de dados, devido à sua capacidade de rastrear variados componentes do banco de dados.

O modo de funcionamento das duas formas anteriormente exploradas segue uma modelagem de dados. As três maneiras de modelar dados são (TEOREY 2007; PRESSMAN; MAXIM, 2016):

- modelo conceitual, que define o que o sistema contém;
- modelo lógico, trata de como o sistema deve ser implementado;

- modelo físico, define a implementação do sistema incluindo um sistema gerenciador de bancos de dados.

Os três modelos serão explorados a seguir.

3.1.1 Modelo conceitual

A modelagem conceitual de banco de dados tem como objetivo estabelecer as entidades que compõem um sistema, todos os atributos que as entidades apresentem e a relação entre cada uma das entidades. Entendemos entidade como um objeto do mundo real, um atributo como uma característica que esse objeto apresenta; um relacionamento como uma associação entre duas entidades (TEOREY *et al.*, 2013; HARRINGTON, 2016).

Imaginemos um sistema cujo banco de dados precisa ter um funcionário e seu pagamento. Como demonstrado na Figura 5, as duas entidades desta modelagem são *funcionário* e a *folha de pagamento*. Os atributos da entidade *funcionário* são: nome, CPF e número, os atributos da entidade *folha de pagamento* são número e valor. E, por fim, a relação entre funcionário e a folha de pagamento é o pagamento em si. A forma de modelar um modelo conceitual de banco de dados segue o modelo ER. Uma das vantagens de desenhar um banco de dados através de um modelo conceitual é a ausência de termos técnicos; de certa forma, a linguagem usada para tais diagramas tem fácil compreensão para um público que não detenha conhecimento técnico de bancos de dados (TEOREY *et al.*, 2013).

Figura 5 – Modelo conceitual



Fonte: Adaptado de Teorey (2007).

3.1.2 Modelo lógico

A segunda forma de modelar dados é a modelagem lógica, que serve como passo intermediário entre a ideia conceitual e a implementação do modelo em um sistema gerenciador de banco de dados. Este modelo ainda trata-se de uma representação genérica dos atributos e das entidades de um sistema, porém já especifica as necessidades de cada entidade. A Figura 6 representa o mesmo esquema tratado no

modelo conceitual, porém já adiciona algum valor na modelagem, fornecendo mais compreensão sobre as entidades e os atributos e como serão implementados no modelo físico (TEOREY *et al.*, 2013; HARRINGTON, 2016).

Figura 6 – Modelo lógico

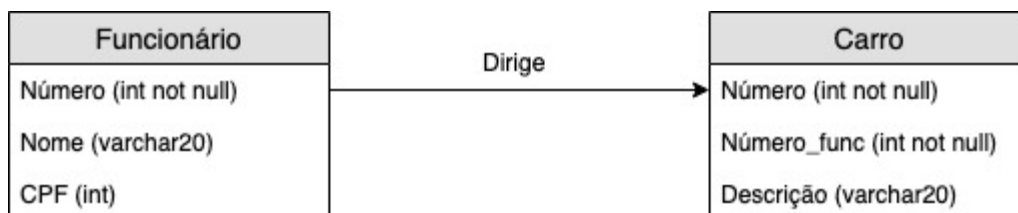


Fonte: Adaptado de Teorey (2007).

3.1.3 Modelo físico

O nosso último produto da modelagem de dados é a criação de um modelo físico. Esta é uma descrição de um banco de dados através de um sistema gerenciador de banco de dados (SGBD). Diferentes SGBDs irão gerar diferentes modelos conceituais. Componentes da estrutura de um banco de dados são especificados. Nesse estágio, são definidas as chaves primárias, que são identificadores únicos de uma entidade (um exemplo é o número do funcionário e o número de pagamento presentes na Figura 7), chaves estrangeiras, que fazem referência a uma chave primária, outra entidade, como demonstrado na Figura 7, em que um funcionário tem um carro designado a ele, porém, é necessário saber quem é o motorista do carro em questão. Para isso, é adicionada a chave estrangeira, que faz referência a um funcionário. As definições dos atributos na Figura 7 já estão detalhadas conforme o SGBD requer: *int* significa tipo de dado inteiro, *not null* especifica que o cadastro não pode ser nulo, um funcionário precisa ter um número identificador associado; *varchar20* significa que o tipo de dado suportado é até 20 caracteres.

Figura 7 – Modelo físico



Fonte: Adaptado de Teorey (2007).

Dessa forma, está concluída a primeira etapa ao modelar dados; uma grande vantagem ao fazer isso é a obtenção de uma documentação detalhada do banco de dados a ser desenvolvido, garantindo que cada passo possa facilmente ser rastreado. Os diferentes níveis proporcionam uma fácil leitura para várias pessoas que estejam envolvidas em um projeto, e a unificação da forma de modelagem com UML e diagramas ER permite uma padronização na forma de modelar dados (DATE, 2004; GILLENSON, 2006; ELMARSI, NAVATHE, 2018).

3.2 Linguagens de consulta

O modo de funcionamento de um banco de dados em nível de usuário é simples: trazer a informação que fora solicitada de maneira eficiente e organizada; em outras palavras, realizar uma consulta (*query*). Sistemas gerenciadores de bancos de dados (SGBDs) contam com o que define-se como linguagem de consulta. Essa linguagem serve para o usuário dizer ao banco de dados qual é seu desejo. Entendemos então uma linguagem de consulta como um modo de modelar todas as entidades, os atributos e relacionamentos que são apresentados na modelagem em nível físico.

O SQL (*Structured query language*) é a linguagem de pesquisa utilizada em bancos de dados. Diferentes SGBDs implementam variações nessa linguagem, então sua sintaxe tende a ser diferente conforme o SGBD que a utiliza.

A primeira etapa ao explorar SQL diz respeito à manipulação de dados (DML – data manipulation language). Esse subconjunto SQL é usado para incluir, alterar e excluir os dados presentes em uma tabela. A Tabela 1 inclui os comandos SQL referentes à manipulação de dados e exemplos da sintaxe utilizada por SQL. Os exemplos tomam como base um cenário onde existe uma tabela em um banco de dados com registros de clientes, os atributos dessa tabela são: identificador, nome, *e-mail* e endereço.

Quadro 1 – Comandos SQL

Comando SQL	Função do comando	Exemplo do comando
INSERT	Inserir registro em uma tabela	INSERT INTO clientes (id, nome, <i>e-mail</i> , endereço) VALUES (001, “Carlos Silva”, “Carlos.silva@servidor”, “Rua 3, 9870”
UPDATE	Alterar algum registro previamente inserido em uma tabela	UPDATE clientes SET endereço = “Rua da liberdade, 4590” WHERE id=001
DELETE	Remover linhas de uma tabela	DELETE from clientes WHERE id=001

Fonte: Adaptado de Elmarsi e Navathe (2018).

A segunda etapa ao trabalhar com SQL diz respeito à definição dos dados (DDL – *data definition language*). Nessa etapa, o usuário pode definir novas tabelas, alterá-las e

excluí-las. Diferentemente do comando DELETE em DML, a deleção em DDL irá deletar a tabela inteira e não apenas alguns de seus registros. Os comandos de DML são:

- CREATE TABLE, onde uma tabela é criada;
- ALTER TABLE, onde uma tabela pode ser alterada;
- DROP TABLE, onde o objeto é deletado do banco de dados.

A terceira etapa diz respeito a realizar transações com os dados. Por motivos de integridade e segurança dos dados, SQL implementa os comandos COMMIT e ROLLBACK, onde, para qualquer transação de DML ou DDL, um comando COMMIT deve ser incluído para a transação surtir efeito. Quando um comando ROLLBACK é executado, as mudanças nos dados existentes desde o último COMMIT ou ROLLBACK são descartadas.

Por fim, com o banco de dados já criado e populado, resta ao usuário realizar consultas para recuperar os dados que deseja; esse tipo de linguagem é chamado de DQL (*data query language*). Uma expressão básica em SQL compreende os comandos SELECT, FROM e WHERE, onde:

- SELECT irá buscar os atributos desejados em uma consulta. Imaginemos o cenário em que o usuário deseja obter o *e-mail* dos clientes que estão registrados em sua base de dados. O comando SELECT consistiria em SELECT *e-mail*;
- a cláusula FROM, que geralmente acompanha um comando SELECT seleciona as tabelas nas quais a informação será buscada. Pegamos o mesmo cenário anterior, porém agora nossa base de dados tem clientes e fornecedores, e o desejo do usuário é consultar o *e-mail* de todos os clientes. SELECT *e-mail* FROM clientes seria a consulta em SQL para o banco de dados retornar à informação desejada.
- por fim, o WHERE age como uma condição sobre o comando FROM. Essa cláusula pode conter expressões aritméticas (<, <=, >, >=, = e <>) e também conectores lógicos (AND, OR, NOT) a modo de filtrar a busca. O nosso mesmo cenário agora compreende uma consulta completa em SQL, em que uma empresa tem diversas filiais, e o usuário deseja reaver o *e-mail* de todos os clientes que pertencem à unidade da cidade X, em que a empresa atua. A consulta típica em SQL teria a forma:

```
select email
from clientes
where cidade = "Cidade X"
```

Diferentemente de linguagens de programação, linguagens de consulta tornam possível que um usuário de SGBD manipule, realize transações e recupere dados que deseja de uma base de dados. Essa ligação entre o SGBD e o usuário é mediada por uma linguagem de consulta, e SQL é referência em tratar dessa interação entre humano *versus* SGBD (GILLENSON, 2006; ELMARSI, NAVATHE, 2018).

3.3 Arquitetura de bancos de dados

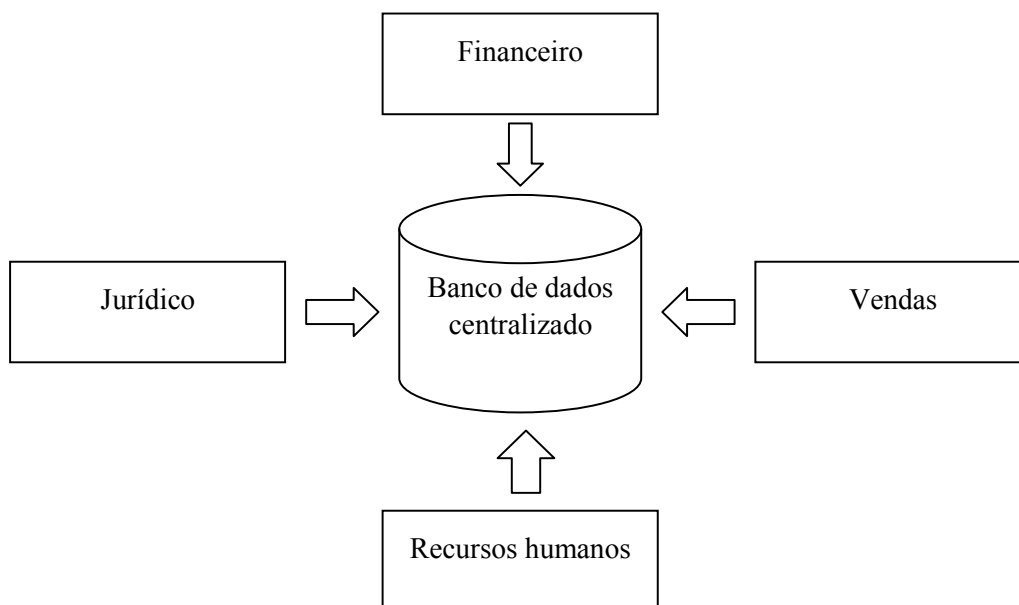
Recentemente, vivemos uma era que provavelmente, os historiadores futuros definirão como da informação. A revolução tecnológica (THORNAN *et al.*, 2009) trouxe mudanças no modo como interagimos. A economia, política, ciência e muitas áreas sofreram alterações e a tecnologia da informação vem agindo como um meio muito eficiente de entregar valor agregado às áreas citadas. Sistemas gerenciadores de bancos de dados (SGBDs) agem como um elo, quando a grande quantidade de informação que nos cerca precisa ser organizada e indexada e, sempre que possível, recuperada da maneira mais eficiente possível. Pode-se definir um SGBD como uma ferramenta que entrega o acesso e armazenamento a uma grande quantidade de dados, de maneira eficiente, segura, multiusuários e confiável (GILLENSON, 2006; THORNAN, 2009).

Podemos definir um SGBD como um intermediador entre os dados e o usuário, porém, a aplicação que faz o uso dos dados de um banco pode necessitar de diferentes configurações de um SGBD; esses diversos aspectos são chamados de arquiteturas, e cada aplicação terá uma arquitetura de dados mais apropriada para que o SGBD possa prover o acesso e armazenamento de maneira mais eficiente possível.

3.3.1 Arquitetura centralizada

Plataformas centralizadas contam com a existência de uma única máquina com uma grande capacidade de processamento. Essa máquina centralizada tem o SGBD localmente instalado em si. A partir dessa máquina, outros usuários podem manipular uma grande quantidade de dados. A Figura 8 representa uma companhia com diversos setores, e, como pode-se perceber, todos os setores têm suas informações armazenadas em um único banco de dados.

Figura 8 – Representação do modelo de arquitetura centralizado



Fonte: Adaptado de Date (2004).

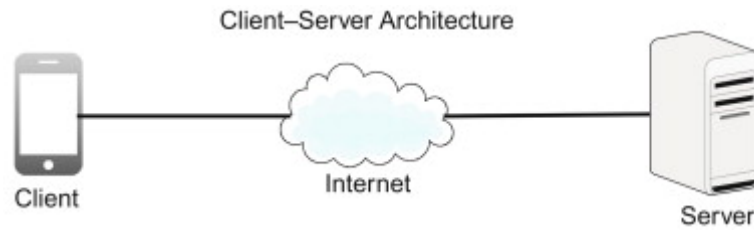
Algumas vantagens desse modelo de arquitetura são: *i*) com a maximização da integridade dos dados, devido à centralização do SGBD, torna-se mais fácil a coordenação de dados; *ii*) a segurança física e lógica ao redor de uma única unidade torna-se mais barata e fácil de ser realizada do que em uma arquitetura com bancos de dados espalhados; *iii*) toda a informação pode ser acessada do mesmo local no mesmo tempo. As desvantagens que eventualmente podem surgir ao adotar um modelo centralizado são: *i*) alta dependência de uma rede bastante rápida, devido à grande quantidade de dados em um único local, onde para uma busca ser eficiente, a rede de acesso deve ser robusta; *ii*) o tráfego de dados é bastante alto; *iii*) falhas podem ser fatais, caso não haja maneira de tratar problemas de segurança (DATE, 2004).

3.3.2 Arquitetura cliente-servidor

O modo de acesso à informação, através de uma arquitetura cliente-servidor, utiliza o conceito de que os usuários atuam como clientes e requerem informação. O servidor é responsável por entregar a requisição do cliente. Geralmente, essa arquitetura segue um conceito em que um servidor centralizado maneja diversas requisições de usuários remotos. Essa é uma das formas de arquitetura de dados mais comuns e é encontrada na maioria das tarefas que desempenhamos. A maioria dos websites que consultamos segue a arquitetura cliente-servidor (HARRINGTON, 2016).

Podemos ver na Figura 9 uma representação da arquitetura cliente-servidor: o cliente é um *smartphone*; realiza requisições à internet, que busca os dados desejados pelo cliente em um servidor.

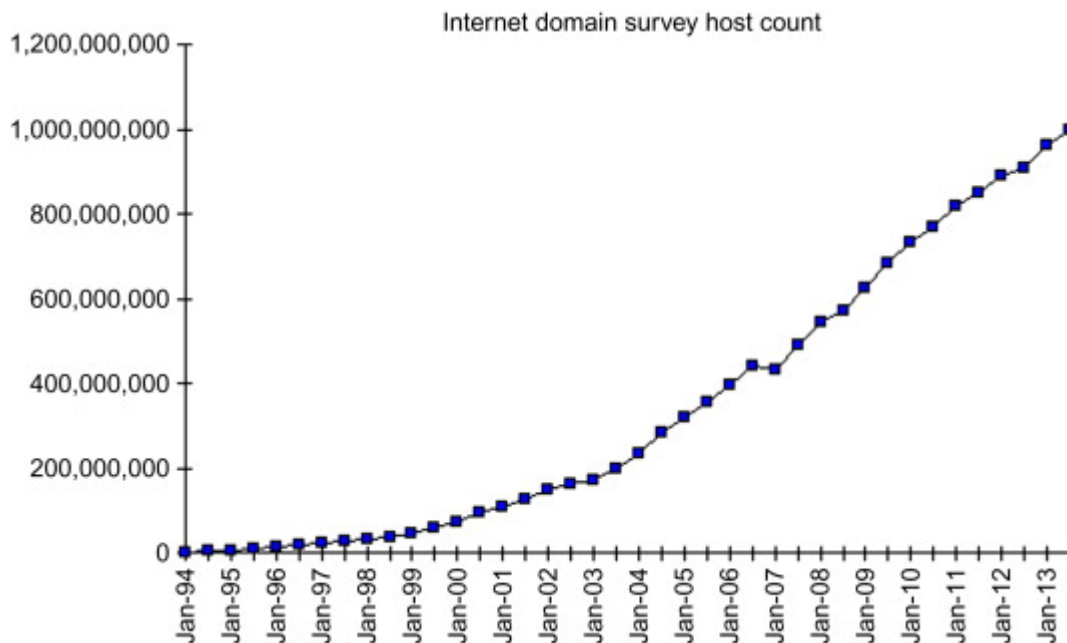
Figura 9 – Arquitetura cliente-servidor



Fonte: Adaptada de Tiwana (2014).

O que definimos no início do capítulo como a era da informação vem fazendo com que no decorrer da última , e mais aplicações baseadas na Web surjam, e isso está diretamente ligado com o uso mais acentuado de aplicações que seguem o modelo cliente-servidor. A Figura 10 representa o crescimento de domínios de internet com o passar do tempo, e podemos observar um crescimento exponencial a partir do início do século XXI, isso agiu como uma faísca para o crescimento de mais aplicações cliente-servidor (WU; BUYA, 2015).

Figura 10 – Crescimento de domínios de internet no decorrer do tempo

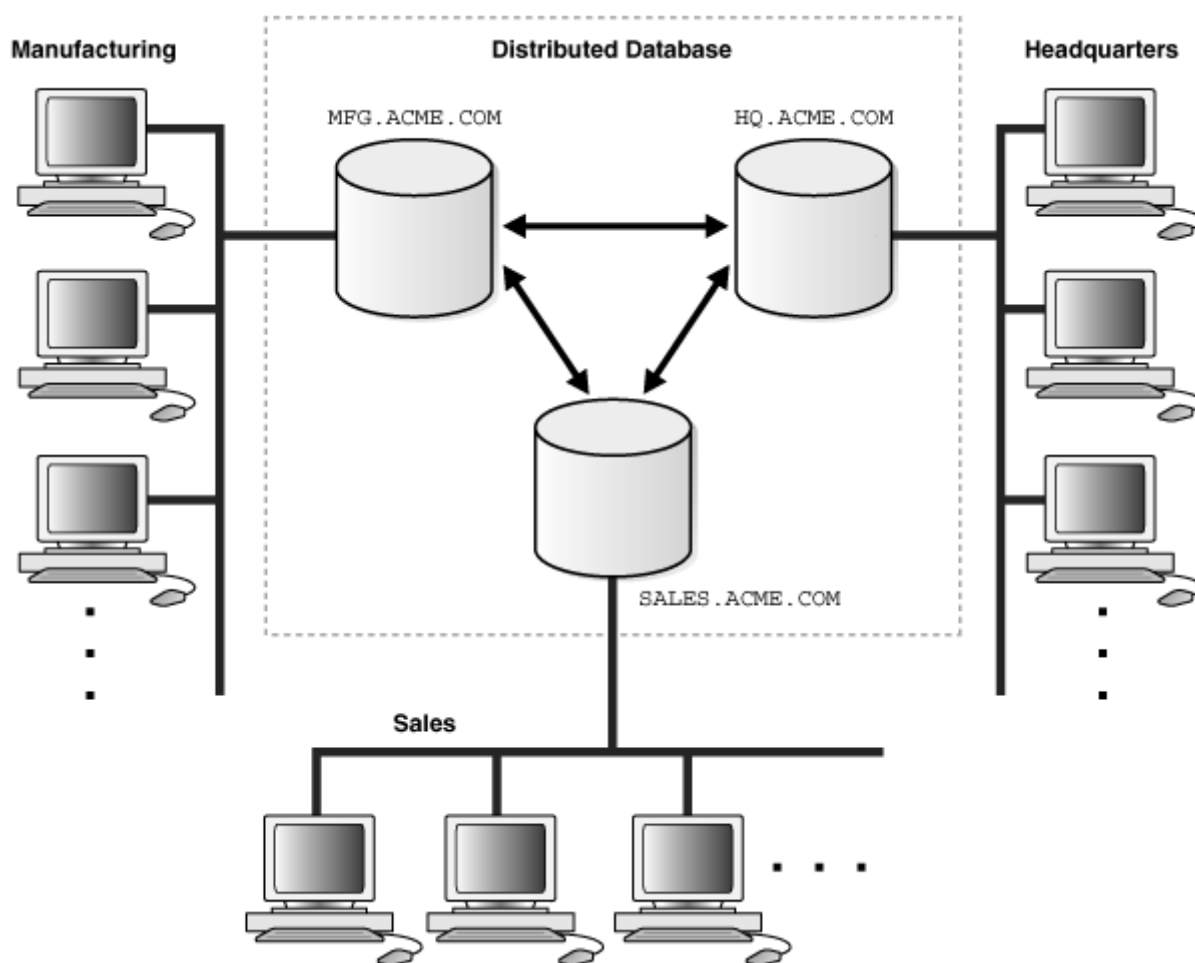


Fonte: Adaptada de Wu; Buya (2015).

3.3.3 Arquitetura distribuída

A arquitetura de bancos de dados, quanto de distribuída permite que aplicações acessem dados que estejam armazenados local e remotamente. Cada servidor presente nessa arquitetura maneja requerimentos através de uma arquitetura cliente-servidor. As consultas são requeridas para qualquer servidor, e se o servidor requisitado não dispor da informação necessária, o sistema consultará a rede e buscará a informação que o usuário deseja. A Figura 11 representa uma arquitetura de dados distribuídas. É muito comum encontrar esse modelo presente em aplicações, nas quais a requisição de dados seja muito grande, criando um grande tráfego de informação para ser cuidado por um sistema centralizado. Os bancos de dados existentes são *sales*, *manufacturing* e *headquarters*. Imaginemos uma situação em que um cliente da rede *manufacturing* necessita acessar informação do banco de dados *headquarters*. Desse modo, o banco de dados *manufacturing* irá solicitar *headquarters* e retornará a informação.

Figura 11 – Modelo de sistema distribuído de dados



Fonte: Documentação Oracle. Disponível em: https://docs.oracle.com/cd/B28359_01/server.111/b28310/ds_concepts001.htm#ADMIN12075.

3.3.4 Arquitetura paralela

Seguindo o conceito de computação paralela, essa arquitetura de banco de dados aumenta a velocidade como que os dados são tratados. Existem situações nas quais uma arquitetura centralizada apresenta gargalos ao tratar de muitas requisições ao mesmo tempo; então, através do paralelismo, as operações feitas no banco de dados são realizadas simultaneamente. Problemas complexos e grandes podem ser divididos em problemas menores e manejados por unidades de processamento comuns, eliminando a demanda de processadores de alto desempenho. Os motivos por trás da adoção de uma arquitetura de banco de dados paralelo diz respeito ao custo e à funcionalidade. Dentro do paralelismo em bancos de dados, surgem duas aplicações bastante difundidas e que podem ter suas vantagens e desvantagens (MOHAMED *et al.*, 1998):

- *arquitetura de memória compartilhada*: o sistema consiste em n processadores conectados a uma única memória. A desvantagem desse modelo é o fato de que todos os processadores competem pela mesma memória, isso limita o número de processos que podem ser realizados simultaneamente;
- *arquitetura de disco compartilhado*: nessa aplicação, cada processador tem sua memória particular, isso elimina um dos gargalos apresentados pela arquitetura de memória compartilhada. Porém, o disco é compartilhado. Entretanto, os processadores não estão à mercê de carregar instruções de um único canal de memória, que pode ser limitado. O processamento trazido pelo paralelismo tem mais disponibilidade para trabalhar com suas tarefas;
- *arquitetura de nada compartilhado*: trás aspectos da arquitetura de memória e disco compartilhado, em que o sistema é um número de máquinas autônomas, cada uma com seu disco e memória. Isso faz com que falhas em uma única unidade não impactem o funcionamento do restante do sistema.

Referências

- BALTZAN, Paige; PHILLIPS, Amy. **Sistemas de informação**. Porto Alegre: Arned, 2012.
- FUCHS, C.; HOFKIRCHNER, W.; SCHAFRANEK, M.; RAFFL, C.; SANDOVAL, M.; BICHLER, R. Theoretical Foundations of the Web: Cognition, Communication, and Co-Operation. Towards an Understanding of Web 1.0, 2.0, 3.0. **Future Internet**, n. 2, p. 41-59, 2010.
- DATE. C. J. **Introdução a sistemas de bancos de dados**. Rio de Janeiro: Elsevier, 2004.
- ELMARS, Ramez; NAVATH, Shamkant. **Sistemas de banco de dados**. 7. ed. Rio de Janeiro: Pearson, 2018.
- GAMMA, Erich *et al.* **Padrões de projeto**: soluções reutilizáveis de *software* orientado a objetos. Cidade: editora, 2000.
- GILLENSON, M. L. **Fundamentos de sistemas de gerência de banco de dados**. São Paulo: LTC, 2006.
- HARRINGTON, J. L. **Relational Database Design and Implementation**. Fourth edition. Morgan Kaufmann. 2016. SBN 9780128043998, <https://doi.org/10.1016/B978-0-12-804399-8.00041-7>.

- MOHAMED, E.; ABDEL-WAHAB, H.; EL-REWINI, HESHAM; HELAL, A. **Parallel database architectures: a comparison study**. ResearchGate, 1998.
- O BRIEN, A. J.; MARAKAS, M. G. **Administração de Sistemas de Informação**, 15. ed. cidade: editora, 2013.
- PRESSMAN, R. S.; MAXIM, B. R. **Engenharia de software: uma abordagem profissional**. 8. ed. McGrawHill, 2016. 968 p.
- REYNOLDS, R. M.; STAIR, G. W. **Princípios de sistemas de informação**. 3. ed. cidade: editora, 2015.
- TEOREY, T. J. **Projeto e modelagem de banco de dados**. Rio de Janeiro: Elsevier, 2007.
- TEOREY, T.; LIGHTSTONE, S.; NADEAU, T.; JAGADISH, H.V. **Projeto e modelagem de banco de dados**. Rio de Janeiro: Elsevier. 2013. 328 p.
- TIWANA, A. **Platform ecosystems**. MK. 2014. ISBN 9780124080669, <https://doi.org/10.1016/B978-0-12-408066-9.09982-3>.
- THORNAN, S.; BASSETT, C.; MARRIS, P. **Media studies a reader 3rd edition**. New York: University Press, 2009.
- Web 3.0: **Implications for Online Learning by Robin D Morris TechTrends**, January 2011, Volume 55, Issue 1, p. 42-46.
- Web 1.0 vs Web 2.0 vs Web 3.0 vs Web 4.0 – A bird’s eye on the evolution and definition
<http://flatworldbusiness.wordpress.com/flat-education/previously/web-1-0-vs-web-2-0-vs-web-3-0-a-bird-eye-onthe-definition/>
- WU, C.; BUYA, R. **Cloud data centers and cost modeling**. Morgan Kaufmann, 2015. ISBN 9780128014134. <https://doi.org/10.1016/B978-0-12-801413-4.00020-9>.

1 Introdução

A bioestatística é a aplicação dos métodos estatísticos para a solução de problemas relacionados à área da saúde. O uso das ferramentas da bioestatística permite compreender os fenômenos da medicina, biologia, saúde pública, dentre outras áreas. Na pesquisa muitas técnicas são utilizadas para, por exemplo, testar as novas medicações ou validar os resultados obtidos de diferentes tratamentos.

Inicialmente, é necessário compreender que a estatística é dividida em duas grandes áreas: estatística descritiva e estatística inferencial. Quando falamos de **estatística descritiva**, estamos nos referindo a todo conjunto de técnicas que tem por objetivo descrever os dados já conhecidos. Utilizamos, nesta abordagem, as medidas estatísticas (que representam a posição, dispersão ou forma de um conjunto de dados), as tabelas e os gráficos, as distribuições de frequência. Estatística descritiva não tem interesse em fazer estimativas ou testar hipóteses. Estes são os objetivos da **estatística inferencial!** Esta, por sua vez, lida com as questões que envolvem fazer afirmações sobre um conjunto maior de elementos daquele que foi analisado. Isso acontece quando uma amostra de sujeitos é submetida a um tratamento, mas deseja-se inferir sobre os resultados em uma população, por exemplo.

Um exemplo fácil para compreender o uso da estatística inferencial é o teste de uma medicação em dois grupos de doentes que aceitaram participar de um estudo: um dos grupos recebe o medicamento e o outro grupo recebe um placebo. Como podemos garantir que os resultados obtidos na amostra de doentes, que participaram do estudo, podem ser generalizados para todas as pessoas com aquela doença? Ou, ainda, como podemos provar que o medicamento é eficaz?

A inferência estatística lida com as situações, nas quais há o interesse de fazer estimativas dos parâmetros populacionais baseadas nos resultados amostrais. Para isso podemos usar os métodos de estimação ou de testes de hipóteses.

Quando se realiza um estudo, parte-se sempre dos objetivos, que podem ser expressos por uma pergunta. Para respondê-la é preciso considerar que tipo de estudo será feito: transversal ou longitudinal. Os **estudos transversais** são aqueles em que a coleta dos dados é feita em um ponto do tempo – é uma fotografia! Outros estudos, os **longitudinais**, são aqueles, nos quais há um acompanhamento, como se fosse um filme!

¹ Universidade de Caxias do Sul. *E-mail*: cintia.paese@ucs.br

Os estudos longitudinais podem ser observacionais ou experimentais. No caso de **estudos observacionais**, destacam-se os estudos de coorte ou prospectivos; acompanha-se um grupo de pacientes ao longo do tempo e os estudos caso-controle ou retrospectivos, que são baseados em dados de períodos passados. Já nos **estudos experimentais**, também chamados de ensaios clínicos, há maior controle e os pacientes são acompanhados ao longo do estudo, ou então um experimento de laboratório é realizado, as entradas são manipuladas e os resultados medidos.

Quem lida com bioestatística precisa compreender desde os conceitos básicos do *design* de experimentos biológicos até as técnicas de análise dos dados obtidos. Isto é importante para saber corretamente planejar a coleta, sumarizar e analisar os dados dos estudos.

Os passos necessários são:

- definir cuidadosamente o problema;
- formular um plano para coleta das unidades de observação;
- coletar, resumir e apresentar as unidades de observação ou de seus valores numéricos;
- analisar os resultados;
- divulgar o relatório com as conclusões, de tal modo que estas sejam facilmente entendidas por quem as for usar nas tomadas de decisão.

Ter conhecimento para identificar as técnicas de análise de dados adequadas em cada caso e saber interpretar corretamente os resultados é um dos propósitos deste capítulo. Nele você vai ter uma noção abrangente da importância do uso correto da estatística nos seus estudos!

2 Definições importantes

Antes de começarmos, é importante ter clara a diferença entre amostra e população. É definido como **população** o conjunto de todos os elementos que se deseja estudar. Podem ser todos os pacientes de um hospital, todos os moradores de uma cidade em uma faixa etária, todos os hipertensos que foram analisados por um serviço de saúde,... enfim, todos aqueles que têm alguma característica que seja de interesse do estudo que estamos desenvolvendo. Pois bem, em geral não é possível ter acesso a todos: precisamos nos restringir a uma parte deles. Esta parte é chamada de amostra. Uma **amostra** é um subconjunto de uma população! Com maior frequência, utilizamos o estudo da amostra do que da população, não só por serem menos dispendiosas e consumirem menos tempo no processamento dos dados, mas também porque muitas

vezes não dispomos de todos os elementos da população ou estamos utilizando testes destrutivos.

Vamos pensar em um exemplo! Queremos analisar a qualidade de vida relacionada a idosos acamados com a prática de fisioterapia. Para isso, dados serão coletados antes e após 10 sessões de fisioterapia. Como não é impossível analisar todos os idosos acamados, que é a população em estudo, precisamos analisar uma amostra de idosos acamados. Os resultados obtidos com os idosos que participarem do estudo serão utilizados para estimar os resultados esperados com a população de pessoas com as mesmas características.

Outra situação é no caso de ser utilizado algum tipo de teste destrutivo. Para saber a resistência de uma prótese, estudos devem ser feitos em laboratórios, aplicando uma grande força até a prótese romper. Portanto, todos os corpos de prova serão perdidos! Não é possível fazer isso com todas as próteses produzidas pela empresa, é preciso utilizar uma amostra!

Se o seu estudo é baseado nos dados populacionais, ele recebe o nome de **censo**. Algumas situações recorrem à análise de todos os casos, por exemplo, todos os transplantados que fizeram a cirurgia em um determinado hospital, todos as crianças de uma escola que apresentam diabetes, todos os membros da Sociedade Brasileira de Medicina do Exercício e do Esporte, etc. Mas, se seu estudo for com uma parcela de elementos da população, ele recebe o nome de **amostragem**.

Os principais tipos de amostragem utilizados são os probabilísticos; todos os indivíduos da população têm a mesma chance de ser selecionados. Os planos de amostragem probabilística são delineados de tal modo, que se conhecem todas as combinações amostrais possíveis e suas probabilidades, podendo-se então determinar o erro amostral.

Os métodos mais comuns de amostragem probabilística são:

- *Amostragem aleatória simples*: pode-se utilizar uma tabela de números aleatórios ou um programa de geração de números aleatórios para definir os participantes;
- *Amostragem estratificada*: subdivide-se a população em, no mínimo, dois estratos (subpopulações) que compartilham a mesma característica e, em seguida, escolhe-se aleatoriamente uma amostra de cada. Exemplo: homens e mulheres;
- *Amostragem sistemática*: sorteia-se um ponto de partida e, então, sistematicamente, selecionam-se os outros. Por exemplo: o 3º, 403º, 803º, 1203º, ... indivíduo;

- *Amostragem por conglomerados*: divide-se a população em conglomerados (áreas), em seguida sorteiam-se algumas áreas e analisam-se todos os elementos dos conglomerados escolhidos. Por exemplo: bairros.

Amostragens não probabilísticas são utilizadas quando a população em estudo é muito pequena ou de difícil obtenção. Neste caso, a análise de uma amostra aleatória poderia causar distorções. Uma pessoa familiarizada com a população pode indicar melhor as unidades amostrais. Este tipo de amostragem não permite avaliar o erro amostral. Um exemplo de amostragem deste tipo é o de seleção de pessoas com alguma doença rara.

As características dos dados coletados nos estudos são chamadas de **variáveis**. As variáveis podem ser classificadas como quantitativas ou qualitativas. As **variáveis qualitativas**, ou categóricas, não apresentam números, mas sim categorias. Podem ser classificadas em nominais, por exemplo hipertensão (sim / não) ou tipo sanguíneo (A/B/O/AB), onde não há uma ordenação entre as categorias da variável. Se houver ordem entre as categorias, são chamadas de ordinais, por exemplo estágio da doença (inicial / intermediário / final), escolaridade (Ensino Fundamental / Ensino Médio / Ensino Superior). As **variáveis quantitativas** exprimem uma quantidade. Podem ser discretas, assumindo uma quantidade enumerável de valores possíveis, por exemplo número de filhos ou número de cigarros fumados durante um dia. Podem também ser classificadas em contínuas. Neste caso, não é possível enumerar os possíveis valores que a variável pode assumir: peso, altura, IMC e colesterol podem ser citados como exemplos de variáveis quantitativas contínuas. Perceba que as discretas podem ser contadas, enquanto as contínuas precisam ser medidas utilizando algum equipamento de medição.

As variáveis ainda podem ser classificadas quanto à sua função na análise, ou seja, quanto à sua relação com as demais variáveis do estudo. De maneira geral, sempre há dois grupos: as **variáveis independentes** são aquelas que antecedem o desfecho. Também são chamadas de preditoras, de causas, de fatores, de “x”. São as que vêm antes! Já as **variáveis dependentes**, ou resposta, são o desfecho, o resultado. Matematicamente falando, são o “y”!

Por exemplo, no caso de querer avaliar o efeito do uso de um bioestimulante sobre o crescimento de uma espécie de árvore, a variável independente é a quantidade de bioestimulante utilizada, já a variável dependente, ou resultado, é o crescimento observado.

Ainda é importante perceber que outras variáveis podem interferir no estudo. Estas são chamadas de **covariáveis** ou **variáveis de exposição**. Se o experimento anterior, sobre o uso do bioestimulante, fosse realizado ao ar livre, em uma fazenda, a

temperatura, a incidência de sol, a umidade do solo poderiam interferir nos resultados. Estas covariáveis poderiam ser controladas se o experimento fosse realizado em um laboratório, sob condições controladas, retirando a influência causada por elas. Quando não há controle sobre elas, deve-se considerá-las nas análises, sob pena de haver interpretações errôneas nos resultados.

3 Análise descritiva dos dados

Antes de partir para qualquer análise, temos que garantir que não há problemas com os dados. A análise inicial pode ser simplificada e servirá para avaliar se houve algum engano na coleta, como, por exemplo, valores digitados errados, valores faltantes, dados atípicos. As tabelas de frequência têm um papel importante nesta fase.

As **tabelas de frequência** relacionam os valores das variáveis com a frequência de ocorrência. Os valores podem ser expressos em números absolutos ou em percentual. Apresenta exemplo de uma tabela de frequência para as idades dos participantes de um estudo.

Tabela 1 – Frequências para a idade

		Frequência	Porcentual	Porcentagem válida	Porcentagem acumulativa
Válido	18,00	3	11,1	11,1	11,1
	19,00	3	11,1	11,1	22,2
	20,00	3	11,1	11,1	33,3
	21,00	5	18,5	18,5	51,9
	22,00	4	14,8	14,8	66,7
	23,00	1	3,7	3,7	70,4
	24,00	1	3,7	3,7	74,1
	25,00	1	3,7	3,7	77,8
	28,00	1	3,7	3,7	81,5
	29,00	1	3,7	3,7	85,2
	30,00	1	3,7	3,7	88,9
	32,00	1	3,7	3,7	92,6
	33,00	2	7,4	7,4	100,0
	Total	27	100,0	100,0	

Fonte: Elaboração da autora.

Após termos certeza de que os dados estão corretos, queremos entender o que, resumidamente, estão nos dizendo! Neste momento são usadas medidas estatísticas que resumem o conjunto de dados e dão uma noção dos resultados.

As **medidas estatísticas** são a forma mais sintética de examinar os dados e tem o objetivo de simplificar a interpretação e permitir a comparação entre conjuntos de dados. Elas estão agrupadas em medidas de tendência central, medidas de dispersão e medidas de forma.

É importante lembrar que existe uma diferença entre os valores das medidas amostrais e das medidas populacionais. Como, em geral, são utilizados dados obtidos a partir de amostras, nessa seção serão apresentadas as medidas amostrais. Tanto em livros de estatística quanto nos sistemas computacionais estas diferenças podem ser percebidas. Os *softwares* estatísticos, IBM SPSS, R, Epi Info, Stata, etc., também consideram esta forma de cálculo, a menos que seja definida outra função.

As **medidas de tendência central** têm o objetivo de indicar simplificada e onde os dados se localizam, se situam. São chamadas de medidas de posição. As mais utilizadas são a média aritmética (a partir deste ponto chamada somente de média), mediana e moda. Se nenhum valor se repetir, o que é muito comum para variáveis contínuas, não há moda na distribuição.

Média aritmética	Mediana	Moda
Soma de todos elementos dividido pela quantidade de elementos	Valor central	Valor mais frequente
$\bar{x} = \frac{\sum_{i=1}^n x_i}{n}$	$\text{Med} = \frac{x_{n/2} + x_{n/2+1}}{2}, \text{ se } n \text{ é par}$ $\text{Med} = x_{(n+1)/2}, \text{ se } n \text{ é impar}$	

Como exemplo podemos pensar no estudo realizado com 27 pessoas do sexo masculino, que utilizaram uma medicação ao longo de três meses. Se formos informados de que a idade média dos pacientes é de 23 anos, metade dos pacientes tem até 21 anos e metade tem mais de 21 anos e a idade que mais vezes se repetiu foi 21 anos, teremos uma boa noção da característica etária destes homens. Entretanto, perceba que não sabemos nada sobre a idade mínima, máxima, se os valores das idades são próximos ou não. As medidas de tendência central somente nos dão uma ideia de onde os dados se situam!

São as **medidas de variabilidade**, ou dispersão, que indicam se os valores são próximos entre si. As principais medidas são amplitude, variância, desvio padrão e coeficiente de variação.

Amplitude Diferença entre o menor e o maior valor	Variância Diferença média quadrática	Desvio padrão Diferença média	coeficiente de variação Variabilidade relativa
$Amp = x_{m\acute{a}x} - x_{m\acute{i}n}$	$Var = \frac{\sum_{i=1}^n (x_i - \bar{x})^2}{n - 1}$	$DP = \sqrt{\frac{\sum_{i=1}^n (x_i - \bar{x})^2}{n - 1}}$	$CV = \frac{DP}{\bar{x}}$

Com base no exemplo, do estudo com 27 pessoas, agora podemos ter a informação de que o participante mais jovem tem 18 e o mais velho 33 anos. A diferença entre eles é de 15 anos. O desvio padrão é de 4,7 anos, o que indica que a distância média entre a idade média dos participantes e as idades dos 27 participantes é de aproximadamente cinco anos. O coeficiente de variação de 0,204, indica uma variabilidade de aproximadamente 20% em torno da média. Perceba que a variância foi obtida através das distâncias quadráticas, portanto, embora seja uma medida importante na estatística, ela não tem interpretação real. Mas ela não deve ser desconsiderada, pois é a base para o cálculo do desvio padrão! O coeficiente de variação é a medida que permite comparar a variabilidade entre dois conjuntos de dados, pois mede a variabilidade relativa.

Outro grupo de medidas são as de **assimetria e curtose**. Elas caracterizam a forma da distribuição dos dados. Em geral são utilizadas quando há necessidade de ver se os dados têm similaridade com a distribuição normal. A assimetria avalia se os dados estão centralizados ou se há maior frequência em um dos extremos do eixo de variação, e a curtose avalia se a distribuição está mais achatada ou não que a distribuição normal.

Voltando ao exemplo anterior, vejamos como seriam feitos os cálculos. A tabela apresenta as idades dos 27 pacientes que utilizaram uma medicação ao longo de três meses.

19	22	25	21	20	18	28	33	29
20	21	18	21	18	21	30	19	20
22	33	24	22	22	21	19	32	23

$$\text{Média} = \frac{19+22+\dots+23}{27} = 23 \text{ anos}$$

$$\text{Var} = \frac{(19-23)^2 + (22-23)^2 + \dots + (23-23)^2}{27-1} = 21,9$$

$$\text{DP} = \frac{19+22+\dots+23}{27} \text{ anos}$$

$$\text{CV} = \frac{4,7}{23} = 0,204 = 20,4\%$$

Para o cálculo das outras medidas, é necessário utilizar os valores ordenados.

18	18	18	19	19	19	20	20	20
21	21	21	21	21	22	22	22	22
23	24	25	28	29	30	32	33	33

Amplitude = 33 – 18 = 15 anos Mediana = 21 (como n é ímpar,
 achar o valor da posição $(27+1)/2=14$) Moda = 21 (o mais frequente)

A Tabela 2 apresenta os resultados para a análise realizada no *software* IBM SPSS.

Tabela 2 – Estatísticas descritivas

	N	Mínimo	Máximo	Média	Desvio padrão	Assimetria		Kurtosis	
	Estatística	Estatística	Estatística	Estatística	Estatística	Estatística	Modelo padrão	Estatística	Modelo padrão
Idade	27	18,00	33,00	23,0000	4,68221	1,105	,448	,035	,872
N válido (de lista)	27								

Fonte: Elaboração da autora.

Além destas medidas, são úteis as **medidas separatrizes**, como os quartis, decis e percentis. Os quartis separam o conjunto de dados em quatro partes de 25% dos dados cada; os decis em dez partes de 10% dos dados cada e os percentis indicam que 1% está em cada intervalo. Por exemplo, de acordo com padrões internacionais,² crianças com peso ao nascer abaixo do percentil 10 para a idade gestacional são consideradas pequenas para a idade gestacional; entre percentis 10 e 90, são adequadas para a idade gestacional e, acima do percentil 90, são grandes para a idade gestacional. Noutro exemplo, referente ao peso ao nascer das crianças do Estado de São Paulo,³ tem-se que $P_{10}=2690$, $P_{50}=3205g$ e $P_{90}=3765g$ para a idade gestacional entre 37 e 41 semanas, indicando, respectivamente, que 10% dos bebês nasceram com peso inferior a 2690g, metade das crianças nasceu com peso abaixo de 3.205g e 10% das crianças nasceram com peso superior a 3.765g, nesta idade gestacional.

As medidas estatísticas apresentadas até aqui têm condições de fornecer uma descrição dos dados coletados, entretanto, se houver interesse em obter conclusões sobre dados além daqueles da amostra, e se sua amostra for probabilística, deve-se utilizar as técnicas de inferência, que serão apresentadas na sequência.

4 Inferência estatística

O ramo da estatística responsável por fazer afirmações sobre os parâmetros populacionais, quando se utilizam amostras, é chamado de *inferência estatística*, cujas

² Tabelas da OMS e Intergrow <http://portalms.saude.gov.br/saude-de-a-z/microcefalia/tabelas-da-oms-e-intergrowth>

³ BERTAGNON, J.R.D.; ARMOND, J.E.; RODRIGUES, C.L.; JABUR, V.A.; KURAIM, G.A.; NOVO, N.F.; SEGRE, C.A.M. Distribuição do peso ao nascer da população do Hospital Geral do Grajaú comparada à da população da cidade de São Paulo. *Einstein*, v. 8, n. 1, p. 1-4, 2010.

principais técnicas são as estimações e os testes de hipótese. A inferência é útil para determinar se as diferenças observadas entre duas amostras são devidas a uma variação casual ou são verdadeiramente significativas. Por exemplo, queremos testar se os tempos de absorção de duas substâncias são diferentes ou não. Neste caso, vamos decidir comparando os resultados obtidos em duas amostras independentes!

Perceba que, ao utilizar dados de uma amostra para estimar parâmetros populacionais, dependendo das características da amostra que participou do estudo, os resultados das estimativas podem variar. É a teoria da amostragem que se ocupa do estudo das relações entre a população e as amostras dela extraídas! Para cada amostra extraída, em teoria, é possível calcular valores amostrais, como média, desvio padrão... e o conjunto das inúmeras amostras possíveis gera uma distribuição de probabilidade adequada a cada tipo de variável ou característica dos dados que estão sendo analisados. São exemplos de distribuições de probabilidade a distribuição Normal, a t de Student, a Qui-Quadrado, a F, a Uniforme, a Binomial, dentre outras. Para realizar as estimativas e os testes, deve-se associar aos resultados amostrais as características das distribuições de probabilidade.

A **distribuição de probabilidade** mais importante na estatística é a distribuição normal, pois, à medida que o tamanho da amostra vai se tornando suficientemente grande, a distribuição da média vai se aproximando da distribuição normal, independentemente do formato da distribuição dos valores individuais da população. Isto é indicado, algumas vezes, dizendo-se que a população é assintoticamente normal. No caso de uma população ser normalmente distribuída, a distribuição amostral das médias também o será, mesmo para amostras pequenas ($n < 30$). Na prática, a distribuição amostral da média pode ser considerada como normal, sempre que o tamanho da amostra for maior que 30.

Como dito anteriormente, as estimações, ou intervalos de confiança, e os testes de hipótese são duas aplicações da inferência estatística.

4.1 Testes de hipóteses

Os testes de hipóteses são procedimentos estatísticos que permitem tomar uma decisão sobre os dados populacionais, a partir dos dados de uma (ou mais) amostras. São chamados de testes de hipóteses, pois partem de uma afirmação sobre os parâmetros populacionais (a chamada hipótese). Como não conhecemos os verdadeiros valores dos parâmetros populacionais, construímos hipóteses sobre o que acreditamos que sejam os valores!

Os testes são construídos considerando uma hipótese nula (H_0) e uma hipótese alternativa (H_1). A hipótese nula refere-se a uma afirmação de igualdade, já a hipótese

alternativa dá uma alternativa à hipótese nula (que pode ser maior, menor ou diferente). Com o tamanho da amostra utilizado, a distribuição de probabilidade correspondente e os resultados amostrais, o pesquisador calcula uma estatística de teste e toma a decisão se deve aceitar ou rejeitar o H_0 , através da comparação do valor calculado com um valor tabelado (para gl graus de liberdade e nível de significância α). Em todos os testes, quando o valor absoluto calculado for menor que o valor tabelado (também em número absoluto), a hipótese nula não poderá ser rejeitada. O nível de significância (α) é o risco máximo admitido para o erro de afirmar que existe uma diferença, quando ela efetivamente não existe. Ele deve ser estabelecido antes de realizar o teste de hipóteses. Entre os testes de hipóteses, os mais utilizados são os testes de médias e os testes de proporções.

Vejamos um exemplo: considere o caso em que um pesquisador acredita que o tempo médio para absorção da água em sementes é de 15 minutos, mas alguns estudos sugerem que pode ser superior a 15 minutos. Será que o pesquisador está certo ou não? Ele realiza um experimento, medindo o tempo de absorção. Neste caso teríamos a hipótese nula como sendo $H_0: \mu=15$ e a hipótese alternativa $H_1: \mu>15$. Após a coleta dos dados e do cálculo da estatística de teste, considerando o nível de significância, ele deve tomar uma decisão. Se o resultado sugerir que H_0 deve ser rejeitado, então o pesquisador (que acreditava que o tempo médio era igual a 15 minutos) não deve estar certo. Por outro lado, se os resultados obtidos com base na amostra sugerirem que H_0 deve ser aceito, então poderemos dizer, com determinada probabilidade de acerto, que o pesquisador deve estar correto.

Vejamos um exemplo: considere o caso em que um pesquisador acredita que o tempo médio para absorção da água em sementes é de 15 minutos, mas alguns estudos sugerem que pode ser superior a 15 minutos. Será que o pesquisador está certo ou não? Ele realiza um experimento, medindo o tempo de absorção. Neste caso, teríamos a hipótese nula como sendo $H_0: \mu=15$ e a hipótese alternativa $H_1: \mu>15$. Após a coleta dos dados e do cálculo da estatística de teste, considerando o nível de significância, ele deve tomar uma decisão. Se o resultado sugerir que H_0 deve ser rejeitado, então o pesquisador (que acreditava que o tempo médio era igual a 15 minutos) não deve estar certo. Por outro lado, se os resultados obtidos com base na amostra sugerirem que H_0 deve ser aceito, então poderemos dizer, com determinada probabilidade de acerto, que o pesquisador deve estar correto.

4.1.1 Teste de hipóteses para médias

Os testes de hipóteses para médias são utilizados sempre que a variável coletada for do tipo quantitativa e desejamos testar relações entre as médias. Para uma ou duas

amostras podemos utilizar testes baseados na distribuição de probabilidade t de *student* ou testes baseados na distribuição normal. Como a distribuição t tende à distribuição normal à medida que a amostra aumenta, podemos utilizar os testes t em todos os casos (amostras pequenas ou grandes) sempre que não tivermos disponível o desvio padrão populacional (o que ocorre em geral!). Quando o interesse testar mais de dois grupos, deve-se utilizar a análise de variância, que é um teste de médias que utiliza como base a distribuição de probabilidade F de Snedecor.

A seguir apresentamos algumas explicações, exemplos e fórmulas para cada teste.

Teste para uma média: deve ser utilizado quando queremos testar se a média populacional é igual a um valor pré-estipulado. Por exemplo, quando queremos saber se o crescimento das plantas é igual ou maior que 30 cm por ano.

$$H_0: \mu = \mu_0 \quad t_{\text{calc}} = \frac{\bar{x} - \mu_0}{s / \sqrt{n}}$$

$H_1: \mu \neq \mu_0$ onde \bar{x} é a média, s o desvio padrão e n o tamanho da amostra

Comparar com valor tabelado de t , com $n-1$ graus de liberdade

Teste para médias de duas amostras independentes: usamos nos casos nos quais há interesse em saber se as médias de dois grupos são iguais, como no caso de querer saber se o ganho de massa magra foi igual ou diferiu entre atletas de duas modalidades esportivas diferentes. Os testes utilizados dependem se as amostras possuem variâncias iguais ou não.

Variâncias iguais assumidas

$$t_{\text{calc}} = \frac{\bar{x}_A - \bar{x}_B}{\sqrt{s_0^2 \left(\frac{1}{n_A} + \frac{1}{n_B} \right)}}$$

$H_0: \mu_A = \mu_B$
 $H_1: \mu_A \neq \mu_B$

Onde:

$$s_0^2 = \frac{(n_A - 1)s_A^2 + (n_B - 1)s_B^2}{n_A + n_B - 2}$$

Comparar com valor tabelado de t com $n_A + n_B - 2$ graus de liberdade

Variâncias iguais não assumidas

$$t_{\text{calc}} = \frac{\bar{x}_A - \bar{x}_B}{\sqrt{\frac{s_A^2}{n_A} + \frac{s_B^2}{n_B}}}$$

$$v = \frac{\left(\frac{s_A^2}{n_A} + \frac{s_B^2}{n_B} \right)^2}{\frac{\left(\frac{s_A^2}{n_A} \right)^2}{n_A - 1} + \frac{\left(\frac{s_B^2}{n_B} \right)^2}{n_B - 1}}$$

Comparar com valor tabelado de t com v graus de liberdade

Se v não for inteiro, arredondar para o inteiro mais próximo

Teste para médias de duas amostras em par: nos casos onde a amostra é a mesma, porém os dados foram coletados em dois momentos. Por exemplo, para saber se houve perda de peso após 30 dias de dieta de restrição calórica, as mesmas pessoas serão pesadas antes e depois da dieta e, portanto, não podemos perder o vínculo entre os dados dos dois períodos. Outro exemplo pode ser a medição de uma mesma característica, utilizando dois equipamentos diferentes. Cada corpo de prova vai ser analisado pelos dois métodos, e então os resultados obtidos não podem ser separados. O teste consiste em ver se a diferença média entre as duas medidas é nula ou não.

$$t_{\text{calc}} = \frac{\bar{d}}{s_d / \sqrt{n}}$$

$$H_0: \mu_d = 0$$

$$H_1: \mu_d \neq 0$$

onde \bar{d} é a média das diferenças entre os dois momentos e s_d é o desvio padrão das diferenças

Comparar com valor tabelado de t com $n-1$ graus de liberdade

Testes para mais de duas médias: o teste utilizado quando se quer comparar mais de dois grupos é conhecido por análise de variância (ANOVA). Deve-se analisar uma estatística F , que é obtida a partir de uma tabela da ANOVA. Tanto planilhas eletrônicas quanto *softwares* estatísticos calculam os valores que compõem as tabelas.

4.1.2 Teste de hipótese para proporções

Este tipo de teste é apropriado, quando os dados sob análise consistem de contagem ou frequências de itens em duas ou mais classes. A finalidade é avaliar afirmações sobre a proporção (ou percentagem) de uma população. O teste foca na diferença entre o número esperado de ocorrências (supondo-se verdadeira uma afirmação) e o número efetivamente observado. A diferença é então comparada com a variabilidade prescrita por uma distribuição amostral baseada na hipótese de que H_0 é realmente verdadeira.

Teste para uma proporção: utilizamos quando queremos testar se a proporção populacional é tal como especificado no H_0 . Se um pesquisador acredita que mais de 30% das plantas de um viveiro estão contaminadas por um fungo, ele pode coletar uma amostra aleatória de tamanho n e realizar o teste para ver se aceita ou rejeita H_0 .

$$H_0: p = p_0$$

$$H_1: p \neq p_0$$

$$Z_{\text{teste}} = \frac{\frac{x}{n} - p_0}{\sqrt{p_0(1-p_0)/n}}$$

Comparar com valor tabelado de z

Teste para duas ou mais proporções: o teste utilizado neste caso deve ser o teste qui-quadrado (χ^2). A finalidade do teste é avaliar se as proporções de k amostras

independentes provêm de populações que contêm a mesma proporção de determinado item. Se todas as amostras têm a mesma proporção populacional, então se diz que as variáveis são independentes. Por este motivo, este teste também é conhecido como teste de associação. Com a utilização do teste qui-quadrado foi feita a comparação de 34 pacientes com diagnóstico de bronquiolite obliterante pós-infecciosa e 34 controles.⁴ Para garantir que a proporção de crianças do sexo masculino e feminino não diferia utilizaram o teste qui-quadrado e concluíram que a proporção de crianças de cada gênero era igual nos dois grupos.

$H_0: p_A = p_B = p_C = \dots$
 H_1 : pelo menos uma das proporções difere

$$\chi_0^2 = \sum_{i=1}^k \frac{(O_i - E_i)^2}{E_i}$$

Comparar com valor tabelado de χ^2

Em todos os testes apresentados, partimos de uma hipótese. Mas, e se não tivermos nenhuma ideia sobre os dados populacionais? Se não tivermos uma teoria subjacente que justifique a construção das hipóteses? Se estamos iniciando um estudo e precisamos fazer as estimativas populacionais? Nestes casos podemos recorrer aos intervalos de confiança.

4.3 Intervalos de confiança

A estimativa de um parâmetro populacional de uma variável pode ser obtida a partir da estatística de uma amostra coletada daquela população. Tal estimativa é chamada de estimativa pontual, porque origina apenas uma única estimativa do parâmetro. Em virtude da variabilidade amostral, é usual incluir uma estimativa intervalar, para acompanhar a estimativa pontual. Esta nova estimativa considera a variação, a distribuição de probabilidade envolvida e o tamanho da amostra, gerando assim uma estimativa com a margem de erro associada, permitindo que o pesquisador conheça o risco envolvido na estimação. A esta estimativa se dá o nome de intervalo de confiança, pois é um intervalo que contém o verdadeiro valor populacional da média, com uma determinada probabilidade predefinida.

Os intervalos mais comuns, assim como os testes, se referem às médias e proporções, e é função da característica dos dados: variáveis quantitativas ou qualitativas, respectivamente. Como, em geral, não dispomos da variabilidade populacional, neste artigo nos detemos apenas no uso da distribuição t de *student* para a estimativa de médias. Da mesma forma, focamos nas estimativas construídas com base

⁴ SARRIA, E.E.; MUNDSTOCK, E.; MACHADO, D.G.; MOCELIN, H.T.; FISCHER, G.B.; FURLAN, S.P.; ANTONELLO, R.S.; MATTIELLO, R. Health-related quality of life in patients with bronchiolitis obliterans. *Jornal de Pediatria*, v. 94, n. 4, p. 374-379, 2018.

em populações infinitas ou muito grandes. Detalhes sobre ajustes no caso de estimações para populações finitas podem ser obtidos em Devore (2018).

4.3.1 Estimativas para médias

Quando estamos interessados em estimar a média populacional de uma variável, coletamos uma amostra de tamanho n e calculamos a média e o desvio padrão amostral. Com base neles e no valor da estatística t de *student* (com $n-1$ graus de liberdade), podemos calcular a margem de erro. Podemos ter intervalos de confiança unilateral ou bilateral (estes são os mais utilizados). No caso de intervalos unilaterais, a margem de erro é utilizada somente em um dos sentidos.

Como exemplo, podemos querer saber o peso médio dos peixes criados em um tanque. Retiramos aleatoriamente e pesamos 20 peixes. A média amostral dos 20 peixes servirá como estimativa pontual do peso médio de todos os peixes do tanque. O cálculo da margem de erro será feito considerando o desvio padrão, o tamanho da amostra e a confiança determinada pelo pesquisador. A estimativa pontual, mais ou menos a margem de erro, determinará o intervalo de confiança.

Percebemos que intervalos construídos com 99% de confiança terão amplitude maior do que intervalos com 90% de confiança (porque o valor tabelado será maior). É importante que o pesquisador defina qual o nível de confiança necessário para ter uma boa segurança nas suas conclusões. O valor mais utilizado é o de 95% de confiança nas estimativas (o que corresponde a um nível de significância α de 0,05).

$$\bar{x} - t_{\frac{\alpha}{2};n-1} \frac{s}{\sqrt{n}} \leq \mu \leq \bar{x} + t_{\frac{\alpha}{2};n-1} \frac{s}{\sqrt{n}}$$

Como percebemos, o tamanho da margem de erro é uma função do tamanho da amostra utilizada. Quanto maior o tamanho da amostra, menor será a margem de erro da estimativa. Então pode-se pensar em determinar o tamanho da amostra necessário para desenvolver uma estimativa com margem de erro máxima (e), determinada pelo pesquisador. É necessário utilizar o valor da variabilidade dos dados, neste caso representada pelo desvio padrão S . Se o pesquisador não souber S , pode utilizar uma amostra piloto somente para defini-lo.

$$n = \left(Z \frac{S}{e} \right)^2$$

4.3.2 Estimativa das proporções

No caso de proporções, o raciocínio é análogo, porém utilizamos a distribuição normal e sempre amostras maiores que 30, pois estamos trabalhando com variáveis categóricas.

Vejamos este exemplo: o secretário de Saúde de um município quer investigar o índice de obesidade infantil em estudantes de 6 a 10 anos da rede pública. Ele tem duas opções: fazer um censo com todos os estudantes desta faixa etária, ou selecionar uma amostra aleatória de estudantes e utilizar o resultado da amostra para estimar a proporção de crianças obesas.

$$\hat{p} - z_{\alpha/2} \sqrt{\frac{\hat{p}(1-\hat{p})}{n}} \leq p \leq \hat{p} + z_{\alpha/2} \sqrt{\frac{\hat{p}(1-\hat{p})}{n}}$$

Por outro lado, se o secretário de Saúde estiver disposto a fazer uma pesquisa com margem de erro máxima (e), ele pode querer saber quantas crianças precisam ser analisadas, ou seja, ele quer saber o tamanho da amostra. Ele pode calcular utilizando a fórmula a seguir.

$$n = \frac{Z^2 \hat{p}(1-\hat{p})}{e^2}$$

5 Técnicas de análise multivariada importantes em bioestatística

Quando falamos em técnicas de análise, derivam dos testes e das estimativas outras possibilidades de análises, que dependem basicamente dos objetivos e das características das variáveis e de suas relações. Basicamente, existem técnicas de dependência e técnicas de interdependência. Nas **técnicas de dependência**, existe uma relação onde uma ou mais variáveis independentes ou preditoras (aquelas que antecedem o desfecho) explicam o resultado de uma ou mais variáveis dependentes (o desfecho). Por outro lado, nas **técnicas de interdependência** o objetivo é compreender a estrutura que existe entre as variáveis de interesse. Selecionamos para este capítulo três das mais importantes técnicas: regressão múltipla, regressão logística e análise fatorial.

Se houver interesse em aprofundar seus conhecimentos em modelagem multivariada, sugerimos a leitura dos livros de Hair *et al.* (2009) e Johnson e Wichern (2007).

5.1 Correlação e análise de regressão

A análise de regressão e de correlação compreende a análise de dados amostrais para saber se e como duas ou mais variáveis estão relacionadas em uma população. A correlação linear é uma medida da relação entre duas variáveis. Podemos dizer que duas variáveis são correlacionadas se mudanças nos valores de uma variável estão associadas

a mudanças no valor da outra variável. O coeficiente de correlação mais conhecido é o r de Pearson, calculado para duas variáveis métricas. Ele pode assumir valores entre -1 e 1. Se o valor for negativo, podemos dizer que há uma relação inversa entre as variáveis. Se for positivo, a relação será direta. Quanto mais próximo dos extremos for o valor, mais forte será a correlação. Se o valor for próximo de zero, podemos dizer que há ausência de correlação.

O coeficiente de correlação de Pearson é calculado com base na fórmula a seguir, para cada par de variáveis, denominadas de x e y .

$$r = \frac{\sum xy - \frac{\sum x \sum y}{n}}{\sqrt{\left(\sum x^2 - \frac{(\sum x)^2}{n}\right) \left(\sum y^2 - \frac{(\sum y)^2}{n}\right)}}$$

Considerando que existe alguma relação entre as variáveis, deve-se utilizar a análise de regressão para obter como resultado uma equação matemática que descreva este relacionamento. Se for utilizada somente uma variável dependente (VD – o desfecho – métrica) e uma variável independente (VI – variável preditora – também métrica ou passível de transformação), dizemos que se trata de um modelo de regressão simples. Entretanto, é mais comum utilizar mais de uma VI, resultando num modelo multivariado. Neste caso, analisamos a relação entre uma VD e uma ou mais VI's.

São objetivos da regressão múltipla prever valores de uma variável com base em valores conhecidos de outra variável (usado nas situações em que as duas variáveis medem aproximadamente a mesma situação, mas uma delas é relativamente dispendiosa ou difícil de lidar, enquanto a outra não), ou explicar o relacionamento entre as variáveis.

Como exemplo pode-se citar um estudo⁵ que teve por objetivo avaliar os efeitos das características antropométricas como estatura, massa corporal, índice de massa corporal, dobra cutânea da panturrilha, comprimento da tíbia (VI's) na velocidade de corrida de estudantes (VD). Os autores concluíram que as medidas antropométricas podem prever o desempenho da corrida, sendo a dobra cutânea da panturrilha a variável mais importante para prever a velocidade.

Uma vez coletados os dados, o cálculo do modelo de análise de regressão pode ser feito utilizando *softwares* estatísticos. Cada VI é testada no modelo através da significância estatística dos coeficientes da regressão. Devem permanecer no modelo as variáveis que contribuírem para a explicação da variabilidade da VD. O ajuste do

⁵ BORBA, D.A.; FERREIRA-JÚNIOR, J.B.; BRANT, V.M.; GUIMARÃES, J.B.; VIEIRA, C.A. Qual a contribuição das características antropométricas na velocidade de corrida de curta distância? *Pensar a Prática*, v. 19, n. 2, p. 423-431, 2016.

modelo de regressão é avaliado através do coeficiente de determinação, que é a medida da variabilidade da VD que é explicada pelas VI's. Quanto mais próximo de 1 for o valor, maior será o poder de explicação da equação de regressão e, portanto, melhor a previsão da VD.

5.2 Regressão logística

A regressão logística, em sua forma básica, é utilizada quando a variável dependente é binária, ou seja, assume somente dois valores. Pode-se utilizar para prever se o paciente terá ou não uma doença, se um medicamento provocará melhora no paciente ou não, se uma variedade sobreviverá ou não, etc. Na essência, a VD deve ter dois grupos, embora haja formulações que permitem lidar com mais de dois grupos. A regressão logística pode ser comparada com a análise discriminante; entretanto, a regressão logística é mais robusta em situações nas quais algumas suposições inerentes ao uso da análise discriminante não são atendidas (HAIR *et al.*, 2009, p. 225). Um exemplo de aplicação da regressão logística consiste em analisar a associação entre multimorbidade em idosos (VD com dois grupos: com e sem multimorbidade) e as VI's depressão e qualidade de vida.⁶

5.3 Análise fatorial

A análise fatorial é uma técnica que visa a definir a estrutura subjacente em uma matriz de dados; em outras palavras, analisa a estrutura entre as correlações de um grande número de variáveis, reduzindo-as a um conjunto menor de variáveis latentes, chamadas de fatores. Ela ajuda o pesquisador a saber que variáveis devem ficar juntas; quais virtualmente medem a mesma coisa; em outras palavras, o quanto mede a mesma coisa. O uso da análise fatorial serve, dentre outros exemplos, para mensurar fenômenos que não podem ser diretamente observados.

Os dois métodos principais da análise fatorial são a análise de componentes principais (ACP) e a análise fatorial comum (AF). A ACP leva em conta a variância total dos dados. Deve ser usada quando a preocupação maior é determinar o número mínimo de fatores que respondem pela máxima variância nos dados, para utilização em análises multivariadas subsequentes. Os fatores são chamados de componentes principais. Já na AF, os fatores são estimados com base apenas na variância comum. Este método é adequado quando a preocupação principal é identificar as dimensões

⁶ AMARAL, T.L.M.; AMARAL, C.A.; LIMA, N.S.; HERCULANO, P.V.; PRADO, P.R.; G.T.R. MONTEIRO. Multimorbidade, depressão e qualidade de vida em idosos atendidos pela Estratégia de Saúde da Família em Senador Guiomard, Acre, Brasil. *Ciência & Saúde Coletiva*, v. 23, n. 9, p. 3077-3084, 2018.

subjacentes e a variância comum é um elemento de interesse. Este método é conhecido também como fatoramento no eixo principal.

O número de fatores a serem extraídos pode ser determinado *a priori* ou com base em autovalores, gráficos de declive (*scree plot*), percentagem de variância, confiabilidade meio a meio, ou testes de significância. Em geral, a matriz de fatores rotada constitui a base para a interpretação dos fatores, pois, embora a matriz inicial (não rotada) de fatores indique a relação entre os fatores e as variáveis individuais, ela raramente resulta em fatores que possam ser interpretados, porque os fatores são correlacionados com muitas variáveis.

É muito comum o uso da análise fatorial no desenvolvimento de escalas, de tal forma que os itens de um questionário sejam agrupados e representem a estrutura subjacente, ou os construtos. Por exemplo, para desenvolver a escala de suporte social para pessoas vivendo com HIV/Aids, os autores identificaram, a partir de 26 questões, utilizando a análise fatorial exploratória, a existência de dois fatores de primeira ordem: suporte social emocional e suporte social instrumental.⁷ Outro exemplo é o estudo que visa a identificar os fatores que possam influenciar a autogestão da saúde e a qualidade de vida da pessoa diabética. Os autores identificaram três dimensões: sofrimento psicológico, alimentação desinibida e barreiras à atividade.⁸

6 Conclusão

Este capítulo teve por objetivo apresentar desde as noções básicas e definições importantes da bioestatística como as várias possibilidades de análises. Destacamos, para cada técnica estatística, alguns exemplos reais recentes de diferentes ramos da aplicação: biologia, fisioterapia, medicina, agronomia, saúde pública, etc.

A compreensão e o uso correto das ferramentas de análise, associadas à bioestatística, fornecem subsídios para que os pesquisadores e gestores tomem decisões com maior segurança em ambientes de grande variabilidade.

Referências

DEVORE, Jay L. **Probabilidade e estatística para engenharia e ciências**. 3. São Paulo: Cengage Learning, 2018.

HAIR, J.F.; BLACK, W.C.; BABIN, B.J.; ANDERSON, R.E.; TATHAM, R.L. **Análise multivariada de dados**. 6. ed. Porto Alegre: Bookman, 2009.

JOHNSON, R.A.; WICHERN, D. W. **Applied multivariate statistical analysis**. 6. ed. Upper Saddle River, N.J.: Pearson, 2007.

⁷ SEIDL, E.M.; TRÓCCOLI, B .T. Desenvolvimento de escala para avaliação do suporte social em HIV/Aids. *Psicol Teor Pesqui*, v. 22, p. 317-326, 2006.

⁸ CRUZ, R.S.; LEITÃO, C.E.; FERREIRA, P.L. Determinantes do estado de saúde dos diabéticos. *Revista Portuguesa de Endocrinologia, Diabetes e Metabolismo*, n.11, v.2, p. 188-196, 2016.

1 O que é mineração de dados e descoberta de conhecimento em banco de dados

Também conhecida como “descoberta do conhecimento”, a mineração de dados é o processo de escavação de grandes conjuntos de dados, analisando-os e extraindo significado a partir de métodos assistidos por computador. As ferramentas de mineração de dados preveem tendências e comportamentos futuros e, assim, permitem tomadas de decisão proativas e orientadas pelo conhecimento.

As ferramentas de mineração de dados são capazes de dar respostas a questões de negócios que, de outra forma, levariam muito tempo para serem resolvidas tradicionalmente. Essas ferramentas essencialmente vasculham bancos de dados para encontrar informações preditivas, bem como “desenterrar” padrões ocultos que os especialistas podem perder, devido ao fato de não estarem dentro do esperado.

1.1 Knowledge Discovery in Databases (KDD)

Conforme Fayyad *et al.* (1996, p 50), “KDD é o processo não trivial de identificação de padrões, a partir de dados, que sejam válidos, novos, potencialmente úteis e compreensíveis”. Neste contexto, o resultado deve ser fácil de ser verificado e validado para a tomada de decisão. O autor destaca ainda que o KDD possui estrita relação com a pessoa que analisa estes dados, visto que o processamento do volume dos dados é uma tarefa computacional, mas a significação das análises requer habilidades humanas. O processo de KDD é compreendido por seis fases: seleção dos dados, limpeza de dados, enriquecimento, transformação, mineração de dados e a visualização dos resultados (FAYYAD, 1996; ELMASRI; NAVATHE, 2011).

Dentre os processos do KDD, destaca-se a mineração de dados, que consiste na aplicação de técnicas em um determinado conjunto de dados, contendo informações do mundo real. Essas análises ocorrem por meio de algoritmos que resultam em padrões e tendências. Assim, a mineração de dados compreende a extração de conhecimento inovador, que tenha valor ao domínio em que é aplicado (SILVA *et al.*, 2016; CARVALHO *et al.*, 2011).

¹ Universidade de Caxias do Sul. *E-mail:* gdani@ucs.br

² Universidade de Caxias do Sul. *E-mail:* marcelosachets@gmail.com

³ Universidade de Caxias do Sul. *E-mail:* sasilva6@ucs.br

Como mencionam Elmasri e Navathe (2011), a mineração de dados é considerada uma parte do processo de descoberta do conhecimento, ou seja, está inserido no processo do KDD, conforme a Figura 1 demonstra. Muitos autores consideram a mineração de dados como sendo o sinônimo do KDD. Essa discordância é comum no meio acadêmico, pois não há consenso entre os autores (WANG, 2008; HAND *et al.*, 2001).

Figura 1 – Relação da Mineração de dados e o KDD



Fonte: Adaptado de Elmasri e Navathe (2011).

1.2 Organização e Estrutura dos Dados

Conforme Silva *et al.* (2016), os dados são a fonte para a aplicação dos métodos de mineração de dados. Eles podem estar apresentados de duas formas, dados estruturados e dados não estruturados. Os dados estruturados estão dispostos em estruturas tabuladas, sendo que as linhas armazenam a ocorrência de determinado evento e as colunas representam as características do exemplar, na qual denomina-se de instância. Esses dados podem ser resultantes de processos de medição e observação de determinado ambiente. Outra forma de representação dos dados é a não estruturada. Sendo este formato representado por textos, imagens, vídeos e sons (SILVA *et al.*, 2016; AMARAL, 2016).

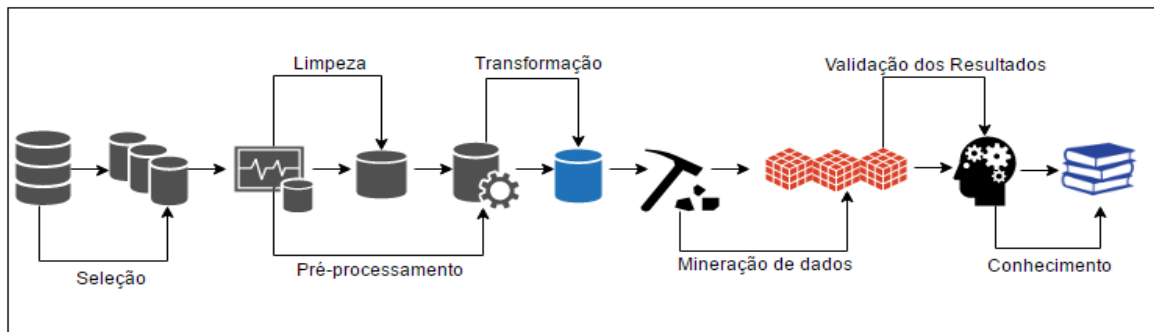
A técnica de mineração de dados muitas vezes é aplicada tanto em banco de dados transacionais quanto em *Data Warehouses*. Para Elmasri e Navathe (2011), o processo de mineração de dados em *Data Warehouses* auxilia na tomada de decisões de diferentes naturezas. O processo de organização e estruturação de dados é a primeira etapa do processo para a criação de um *Data Warehouse*. Esse processo organiza a base de dados que contém as informações para a análise. Os dados passam por uma série de procedimentos de pré-processamento que inclui a criação de um repositório único que pode ser denominado *Data Warehouse*. O objetivo do *Data Warehouse* é dar suporte à tomada de decisão com dados (ELMASRI; NAVATHE, 2011; SILVA *et al.*, 2016).

O *Data Warehouse* pode ser considerado um depósito de dados integrados de múltiplas fontes. A diferença do banco de dados transacionais que normalmente são heterogêneos é a grande quantidade de dados históricos que é mantido. Essa base histórica é utilizada para auxiliar o apoio em tomadas de decisão. O *Data Warehouse* são bancos não voláteis, ou seja, as informações contidas nele não mudam com tanta frequência, como um banco de dados transacional (ELMASRI; NAVATHE, 2011).

1.3 Etapas do Kdd

O KDD inclui atividades multidisciplinares. Isso abrange armazenamento e acesso a dados, algoritmos de dimensionamento para conjuntos de dados massivos e interpretação de resultados. A limpeza de dados e o processo de acesso aos dados incluídos no *data warehousing* facilitam o processo do KDD. A inteligência artificial também suporta o KDD, descobrindo leis empíricas de experimentação e observações. Os padrões reconhecidos nos dados devem ser válidos em dados novos e devem possuir algum grau de certeza. Esses padrões são considerados novos conhecimentos. As etapas envolvidas em todo o processo de descoberta de conhecimento em banco de dados KDD, estão ilustradas na Figura 2.

Figura 2 – Processos do KDD



Fonte: Adaptado de Silva *et al.* (2016).

O primeiro processo do KDD é a seleção dos dados. Segundo Elmasri e Navathe (2011), esse procedimento seleciona as informações que serão utilizadas para a elaboração do processo. O método de extração dos dados pode ser realizado diretamente no banco de dados através de instruções SQL. Após a seleção dos dados-alvo, ocorre o pré-processamento. A etapa compreende a limpeza e transformação dos elementos. Essa etapa tem o objetivo de filtrar e selecionar conteúdo ruidosos, inconsistentes e também ausentes, que podem afetar a qualidade do processo de MD, podendo até anular todo o processo (ELMASRI; NAVATHE, 2011; SILVA, 2016; GOLDSCHMIDT; PASSOS, 2015).

Quando os atributos não apresentam valores, este conjunto de dados pode conter relações nulas, tornando assim os atributos inválidos. Problemas de ausência de valores podem ser solucionados com o preenchimento manual das informações. Para aplicar-se esse método manual, deve-se verificar se é possível executar a aquisição dos dados novamente e se o conjunto de preenchimentos não são relativamente grandes. Outra forma de solucionar este problema é com o preenchimento automático; para isso deve-se verificar os valores mais frequentes, o valor médio ou o valor mediano (SILVA *et al.*, 2016).

Outro problema tratado no processo de transformação são os valores ruidosos, isto é, elementos que estão fora do conjunto de informações esperado, um valor consideravelmente diferente da maioria. Conforme Silva *et al.* (2016), esses ruídos são facilmente percebidos quando possuímos conhecimento sobre os valores resultantes dos atributos. O autor destaca ainda que esses valores podem ser preenchidos com a inspeção e correção manual ou com uma identificação e limpeza automática. A inspeção e correção manual podem ser feitas através de uma análise investigativa dos dados, sendo este em pequena quantidade. Em quantidades maiores, pode ser utilizado o procedimento de suavização, que é endereçado por medidas de posição, que são analisadas pela criação de faixas de valores. Já a identificação e limpeza automática pode ser aplicada por meio de algoritmos usados para suavizar ou anular ruídos (SILVA *et al.*, 2016).

A etapa após a transformação dos dados é a mineração de dados, na qual são aplicados os algoritmos que irão gerar regras e padrões. O processo seguinte a MD é a validação dos resultados e apresentação, por meio de gráficos tabelas ou listagens, que geram o conhecimento (ELMASRI; NAVATHE, 2011).

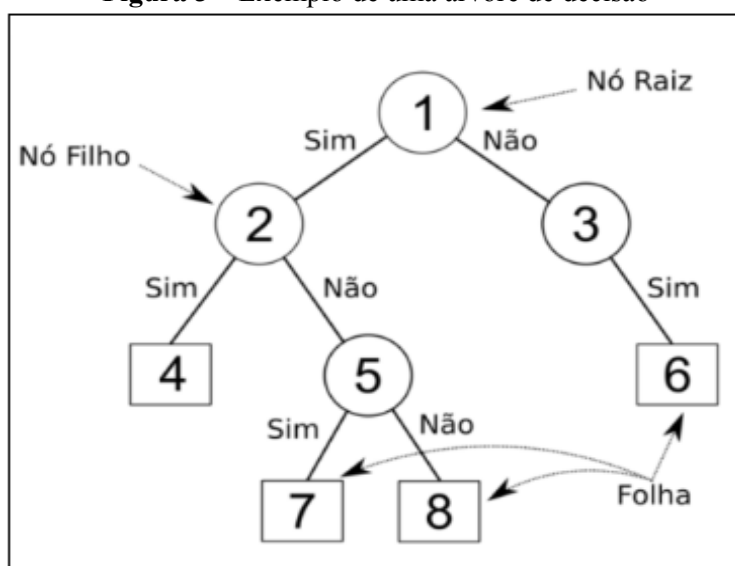
2 Tipos de técnicas e algoritmos

A técnica de classificação é uma forma de descrever determinado exemplar através de modelos de previsão. Neste caso, a base de dados deve estar rotulada em duas ou mais classes, já que o seu principal objetivo é determinar a qual classe um novo exemplar irá pertencer. Esse novo exemplar pode ser classificado em uma classe já existente, ou poderá gerar uma nova classe com suas novas características. As técnicas de classificação possibilitam a criação dos modelos de previsão. Dentre os algoritmos destacam-se as Árvores de Decisão, *Naive Bayes*, Redes Bayesianas e Redes Neurais (QUILICI-GONZALEZ; DE ASSIS ZAMPIROLI, 2014; AMARAL, 2016).

A mineração de dados possui um enfoque em conhecimento indutivo descobrindo regras e associações em dados desestruturados e estruturados, sendo que o último pode

ser representado através de árvores de decisão. Essas técnicas são modelos classificadores que consistem em uma estrutura de árvore com nós folhas e arcos. Cada nó interno da árvore representa um determinado teste em uma característica de uma instância, e os arcos representam os resultados. A árvore é percorrida de cima para baixo, iniciando por um único nodo raiz que vai sendo dividido até levar a classe conhecido como nó-folha. O nó-folha contém a classificação da instância. A figura 3 apresenta um exemplo de árvore de decisão (AMARAL, 2016; SILVA *et al.*, 2016, CARVALHO, 2015; ELMASRI; NAVATHE, 2011).

Figura 3 – Exemplo de uma árvore de decisão



Fonte: Adaptado de Carvalho (2015).

Segundo Silva *et al.* (2016) apontam, o modelo de árvore de decisão pode ser interpretado como um modelo SE ENTÃO. Ao percorrer a árvore regras SE ENTÃO e SE ENTÃO SENÃO são criadas para determinar o caminho. Na Figura 3 é possível verificar essa regra, sendo que SE 1 for sim ENTÃO 2; se 2 for sim ENTÃO 4; SENÃO 5, SE 5 for sim ENTÃO 7, SENÃO 8; já SE 1 for não ENTÃO 3; e 3 só pode ser sim resultando 6. O exemplo da Figura 3 pode ser considerado muito simples, mas é de fácil compreensão para entendimento do funcionamento da árvore de decisão.

Para o processo de criação da árvore de decisão, o conceito mais utilizado é o de dividir e conquistar. A árvore criada busca o nó-raiz, que é considerado o nó inicial e recursivamente trabalha em cada nó-filho, buscando criar uma única classe a partir de cada subconjunto. O fator mais crítico para a criação de uma árvore de decisão é a escolha da característica do nó árvore que será utilizado. Esse processo deve buscar a divisão mais “pura”, ou seja, que o número máximo possível de instâncias, em cada

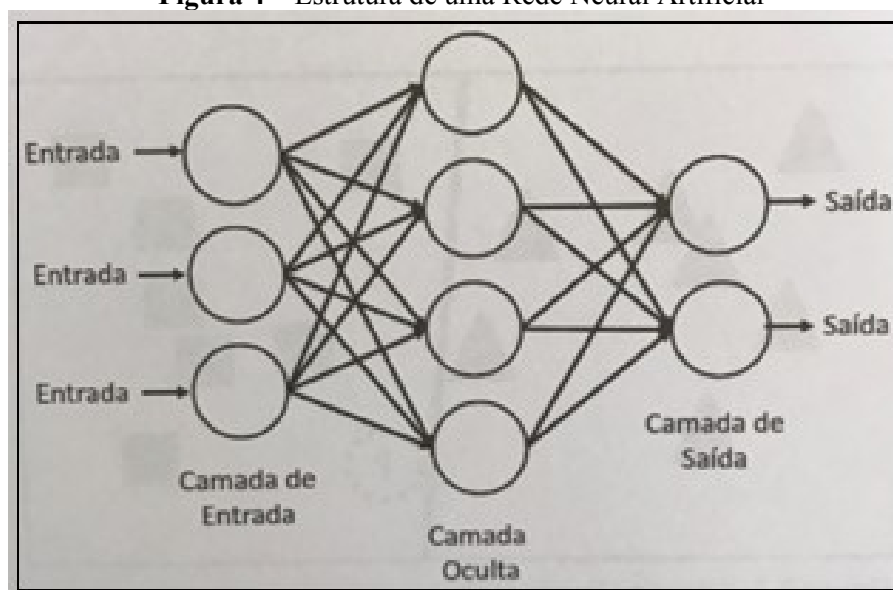
subconjunto, pertença a uma única classe. Para a criação de um critério para a escolha do nó raiz é necessário medir, a impureza da partição dos dados que será utilizado. Uma forma de medição para esse processo é a entropia (CARVALHO, 2015; SILVA *et al.*, 2016).

O algoritmo *Naive Bayes* baseia-se na teoria da probabilidade. Este algoritmo apresenta um grande desempenho tanto em dados categóricos como em dados numéricos (SILVA *et al.*, 2016; AMARAL, 2016). Amaral (2016, p 41) destaca ainda que, “na criação do modelo este classificador constrói uma tabela mostrando o quanto cada categoria de cada atributo contribui para cada classe”. Ou seja, assim que for atribuída uma nova instância aos classificadores, serão verificados os pesos criados na tabela e, para saber qual classe será vitoriosa, somam-se os pesos; a classe de maior valor é a vitoriosa. O algoritmo *Naive Bayes* pode classificar um dado em uma classe antes mesmo de rotulá-las (AMARAL, 2016; MANTUANI *et al.*, 2016).

Outro algoritmo baseado na teoria dos grafos são as Redes Bayesianas. Amaral (2016) descreve que uma Rede Bayesiana é formada por grafo, sendo que cada nó representa uma variável, e para as ligações os nós são formados por arcos que representam a probabilidade condicional. Galvão e Marin (2009, p 689) destacam que Redes Bayesianas “[...] fornecem representações gráficas de distribuição probabilística derivadas de contagem da ocorrência dos dados num determinado conjunto, representando um relacionamento de variáveis”.

As Redes Neurais Artificiais também podem ser consideradas algoritmos de classificação. Elas são modelos inspirados no funcionamento dos neurônios humanos, sendo capazes de adquirir, armazenar e utilizar conhecimento. As Redes Neurais são formadas por um conjunto de entradas e saídas. O conjunto de entradas são ponderadas e somadas; a soma equivale ao potencial de processamento da rede. A saída é calculada por uma função que pode ser uma simples função linear ou uma função de grau. Essas camadas de entrada são consideradas os atributos, sendo um neurônio para cada entrada. Existe ainda uma camada oculta que é a responsável pelo processamento. A camada de saída é o resultado do processamento, conforme a Figura 4 (GOLDSCHMIDT, 2010; AMARAL, 2016; VILELA NETO; PACHECO, 2012).

Figura 4 – Estrutura de uma Rede Neural Artificial



Fonte: Adaptado de Goldschmidt (2010).

Outra técnica de mineração de dados é o agrupamento, que pode ser definido como uma relação de características existentes entre vários atributos de um conjunto de dados, possibilitando identificar semelhanças entre estes atributos. O principal objetivo desta tarefa é buscar a similaridade entre os elementos de um conjunto de dados. A grande diferença para o método de classificação é que o agrupamento identifica, automaticamente, a qual grupo de dados determinado elemento deverá pertencer. O autor destaca ainda que os principais algoritmos utilizados nesta tarefa são *K-means* e *K-medoid* (GALVÃO; MARIN, 2009; SILVA *et al.*, 2016; AMARAL, 2016).

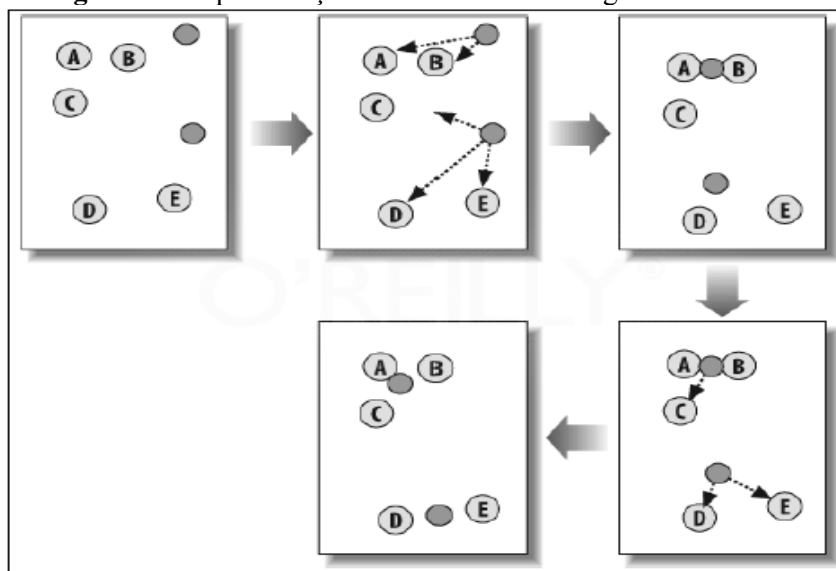
O algoritmo *K-means* e *K-medoid* são os algoritmos agrupadores, também conhecidos como *clustering*. A seu respeito Amaral (2016) declara:

São baseados em centroides ou medoides. Um centroide é um ponto aleatório que pode ou não coincidir com uma instância da relação e que depois vai ser recalculado a partir da média dos elementos próximos. O cálculo entre as distâncias usa, por exemplo, a distância euclidiana, embora outras medidas de distância possam ser utilizadas. Já um medoide é um ponto que coincide com algum elemento dos dados. São chamados de algoritmos baseados em protótipos, pois os centros são deslocados para ficarem mais próximos das instâncias a um certo número de interações, normalmente definido através de um parâmetro do algoritmo (AMARAL, 2016, p. 104).

Este algoritmo tem como principal característica a definição prévia por parte do usuário do número de agrupamentos desejados, sendo esta uma limitação da técnica, pois o usuário é quem informa o número *k* de grupos que se deseja encontrar. Outra característica é que os agrupadores determinam que todas as instâncias serão

congregadas. Aleatoriamente, vários pontos chamados de k-centroides são escolhidos para representar os centroides dos grupos. Os centroides são as médias calculadas e a cada nova interação ela é recalculada novamente, esse processo é realizado várias vezes até que os centroides permaneçam estáveis. Cada elemento é alocado ao centroide que se encontrar mais próximo. Porém não se pode garantir que se encontre o estado ótimo, neste caso o algoritmo é executado várias vezes, até que se encontre o melhor particionamento possível dos dados. Pode-se definir que o algoritmo aloca os dados que estão próximos de um dado central chamado de centroide, podendo assim formar uma relação entre os itens. A Figura 5 apresenta um exemplo do algoritmo *K-means* (AMARAL, 2016; COSTA *et al.*, 2013; CASTRO; FERRARI, 2016).

Figura 5 – Representação do funcionamento algoritmo K-means



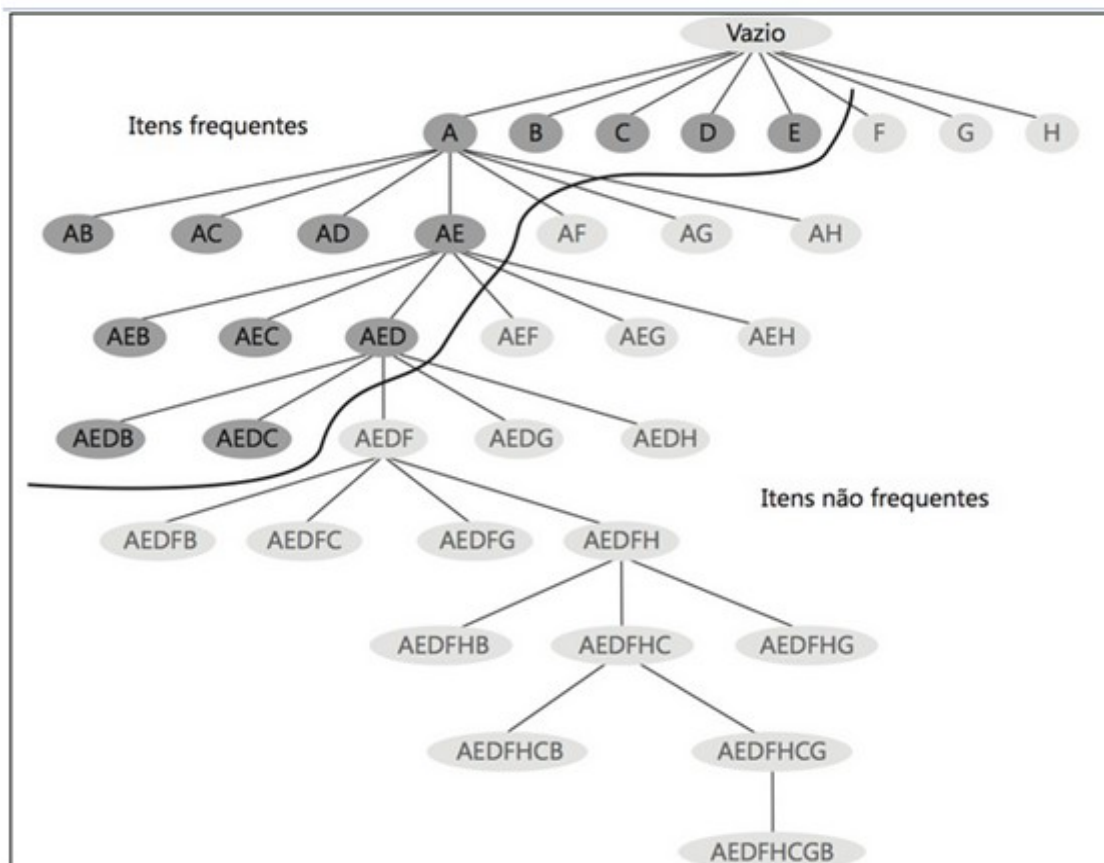
Fonte: Adaptado de Castro e Ferrari (2016).

Um dos algoritmos utilizados na associação é o algoritmo *A priori*, o qual destaca-se por utilizar o princípio de que a presença de um conjunto de itens implica um subconjunto destes itens. Este algoritmo executa várias análises, a fim de encontrar os dados transacionais mais frequentes na base de dados analisada. Com o grande número de itens em uma base de dados, o número de itens frequente também será grande. Os itens mais frequentes são considerados os itens candidatos e os menos frequentes são eliminados. A Figura 6 apresenta um exemplo de criação de um conjunto de itens mais frequentes e menos frequentes (AMARAL, 2016; SILVA, 2016; CASTRO; FERRARI, 2016; SILVA *et al.*, 2016).

Outro algoritmo associador é conhecido como *FP-Growth*. Este algoritmo realiza uma primeira análise na base de dados e descarta os dados menos frequentes, criando

uma tabela com os mais frequentes. Em um segundo procedimento, outras diversas tarefas são realizadas para ordenar esses valores em uma estrutura hierárquica (SILVA *et al.*, 2016).

Figura 6 – Relação de frequência entre itens algoritmo *A priori*



Fonte: Adaptado de Castro e Ferrari (2016).

2.1 Ferramentas de Mineração de Dados

O *SAS Enterprise Miner* é uma ferramenta criada pela empresa SAS e distribuída comercialmente. Foi criada para atender a todas as etapas do processo de mineração. Fornece *insights* que levam à melhor tomada de decisão. É composta por um conjunto de ferramentas interativas de preparação de dados, que possibilita tratar valores ausentes, filtros e possibilidade de desenvolver regras de segmentação (SAS, 2016). A ferramenta comercial *Oracle Data Mining*, foi desenvolvida para trabalhar diretamente no banco de dados da Oracle. Os algoritmos são implementados como funções SQL. Através da implantação das etapas de mineração através de um *dashboard* arrastando as funções, é possível serem gerados os códigos das funções SQL. É uma ferramenta de uso comercial (ORACLE, 2016). Já a ferramenta comercial *IBM Intelligent Miner*, trabalha diretamente no banco de dados da IBM DB2. Apresenta funções de

classificação com algoritmos de redes neurais e árvores de decisão, além de técnicas de regressão e funções estatísticas (IBM, 2016).

Outra ferramenta é o Weka. Conforme Amaral (2016), o Weka é uma ferramenta desenvolvida em Java e mantida pela Universidade de Waikato na Nova Zelândia e distribuída livremente. A grande popularidade do Weka se dá pelo fato de utilizar interface gráfica. O Weka é uma ferramenta *open source* desenvolvida pela Universidade de Waikato, na Nova Zelândia, muito popular no meio acadêmico desenvolvido em Java. A ferramenta pode ser utilizada por linha de comando, mas sua maior popularidade é através da possibilidade de ser utilizada setando valores e não utilizando linha de comando. A interface gráfica auxilia para que esse processo seja interativo e intuitivo com os objetivos propostos. A ferramenta trabalha com um arquivo próprio denominado *Attribute-Relation File Format*, com sua extensão ARFF. O ARFF tem uma forma de tabela relacional simples, com metadados em seu cabeçalho, e os dados separados por vírgula, podendo ser lido em qualquer editor de texto. Além do formato padrão de arquivo do Weka, outros formatos são aceitos pela ferramenta, como *csv* e *JSON* (AMARAL, 2016; COSTA *et al.*, 2013).

O Weka oferece suporte a todo processo de mineração de dados, que compreende as tarefas para pré-processamento de dados, classificação, regressão, agrupamento, regras de associação e visualização dos resultados. A ferramenta conta ainda com mais de 50 técnicas, dentre elas *bayes*, *functions*, *a priori*, *simple k-means*, *lazy*, entre outras (WEKA, 2017).

Outra ferramenta é o RapidMiner; contempla desde a preparação de dados à modelagem até a implantação de negócio. Possui um ambiente de programação visual, com suporte a dados estruturados e não estruturados. É uma ferramenta comercial, sendo possível utilizar funções básicas de forma gratuita. (RAPIDMINER, 2016). Uma ferramenta de destaque também é a IBM SPSS Modeler. Originalmente nomeado Clementine, foi adquirido pela IBM e hoje é denominado IBM SPSS Modeler. É um *software* de Data Mining, com distribuição comercial. O IBM SPSS Modeler suporta diversas extensões e fontes de dados. Foi projetado para auxiliar nas tomadas de decisão. Com técnicas de algoritmos avançadas em análise de textos, de entidades, gerenciamento de decisão e otimização. O IBM SPSS Modeler é uma ferramenta para uso comercial (IBM, 2016).

O Quadro 1 proporciona melhor visão sobre a abrangência das ferramentas e seus algoritmos de aplicação. A ferramenta Oracle DataMining e IBM intelligentMiner se restringe à utilização em banco de dados desenvolvidos pelo próprio fabricante, e não possui uma interface intuitiva perante as outras. Já o Weka se destaca, pois, além de atender a uma gama de algoritmos, aceita diversas fontes de dados e é *Open Source*. A

ferramenta RapidMiner possui uma grande abrangência de algoritmos e fonte de dados, porém a utilização dos seus recursos é limitada, e solicita a aquisição de licença para alguns recursos mais avançados.

Quadro 1 – Comparativo de ferramentas de mineração de dados

Nome	Desenvolvedor	Funções				Licenciamento	
		Algoritmos	Fontes de dados	Suporte a Pré-processamento	Interface Intuitiva	Open Source	Comercial
SAS Enterprise Miner	SAS	Clustering, Regressão Linear, árvores de decisão, Redes Neurais, Associação, Extração de recursos, Detecção de anomalias etc.	Excel, Txt, SAS Data Files, Banco de dados diversos, etc.	X	X		X
Oracle Data Mining	Oracle	Classificação, Regressão, Associação, Clustering, Extração de recursos, Detecção de erros, etc.	Oracle	X			X
IBM Intelligent Miner	IBM	Classificação, árvore de decisão, Regressão, Clustering, Redes Neurais, Previsões, etc.	IBM DB2	X			X
Weka	Universidade de Waikato	Classificação, árvores de decisão, Clustering, Regressão, Redes Neurais, Previsões, Associações, etc.	ARFF, csv, JSON, Acesso a banco de dados via JDBC	X	X	X	
RapidMiner	RapidMiner	Clustering, Regressão, Associação, Árvore de decisão, Similaridade, Redes Neurais, etc.	Excel, SQL, Postgres, Oracle, Etc..	X	X	X (Acesso básico)	X
IBM SPSS Modeler	IBM	Regressão, Classificação, Agrupamento, Associação, Clustering, Árvore de decisão, etc.	SQL, Excel, diversas fontes de dados	X	X		X

Fonte: Adaptado de Costa *et al.* (2013).

3 Potencialidades e limitações

O potencial da mineração de dados é uma inspiração para a maioria das organizações. A mineração de dados é definida como a extração de informações produzidas em diferentes momentos de nossa vida. Quando trabalhamos com dados, começamos a descobrir os benefícios de encontrar padrões e seu significado real.

A capacidade de extrair informações ocultas, mas úteis, dos dados tornou-se crítica no mundo moderno. Quando os dados são usados na previsão, isso torna as características futuras de um negócio claras.

Com o surgimento de desenvolvimentos na indústria tecnológica, tem havido um enorme crescimento nas indústrias de *hardware* e *software*. Bancos de dados complexos foram desenvolvidos e ajudaram a armazenar grandes conjuntos de dados. Isso trouxe a necessidade de minerar dados em diferentes ambientes. Os diferentes contextos incluem coleta de dados, aprendizado de máquina, descrição, previsão e análise.

Hoje em dia, muitas pessoas estão interessadas em inteligência e precisam entender os enormes *terabytes* de dados armazenados nos bancos de dados e desenvolver padrões importantes a partir deles.

A mineração de dados é um processo com amplitude de emprego. É um processo que permite transformar dados brutos em informações úteis. Com a ajuda de alguns *softwares* especializados, é possível obter um grande lote de dados e, em seguida, pesquisá-los para encontrar padrões. A mineração de dados dependerá do fato de poder coletar, efetivamente, os dados, o armazenamento adequado e o processamento do computador.

Há muitas informações boas que podem ser coletadas desses dados, mas pode ser motivo de preocupação, quando as informações erradas, ou as informações que não são representativas do grupo de amostra geral, são usadas para formar a hipótese.

Quando as empresas decidem centralizar todos os dados que coletam em um programa ou banco de dados, estão passando por um processo conhecido como *data warehousing*. Com isso, uma empresa pode dividir partes ou segmentos dos dados para usuários específicos utilizarem e analisarem.

No entanto, há outras ocasiões em que o analista pode iniciar o processo com o tipo de dados desejado e, em seguida, usar essas especificações para criar um *warehouse*. Independentemente de como se planeja organizar dados, eles serão usados para apoiar os processos de tomadas de decisão.

Os programas de mineração de dados são responsáveis por analisar os relacionamentos e padrões nos dados, com base no que os usuários solicitam. Por exemplo, este *software* pode ser usado para ajudar a criar diferentes classes de informação.

3.1 Os benefícios da mineração de dados

Na verdade, existem muitos benefícios que podem ser aproveitados ao se trabalhar com mineração de dados. Na verdade, quase todos os setores podem se beneficiar dessa técnica, desde que aprendam a usá-la corretamente.

Embora outras soluções possam favorecer os prestadores de serviço em saúde, ou as seguradoras, a mineração de dados beneficia todos os envolvidos, desde as organizações de assistência médica até as seguradoras e os pacientes.

Os pacientes recebem serviços de saúde mais acessíveis e melhores. Isso acontece quando os funcionários da área de saúde usam programas de mineração de dados, para identificar e observar pacientes de alto risco e doenças crônicas e planejar as intervenções corretas necessárias. Esses programas também reduzem o número de sinistros e internações hospitalares, agilizando ainda mais o processo.

Os provedores de serviços em saúde usam mineração de dados e análise de dados para encontrar as melhores práticas e os tratamentos mais eficazes. Essas ferramentas comparam sintomas, causas, tratamentos e efeitos negativos e, em seguida, analisam qual ação será mais eficaz para um grupo de pacientes. Esta é também uma forma de os fornecedores desenvolverem os melhores padrões de atendimento e melhores práticas clínicas.

As seguradoras agora são capazes de detectar melhor abusos e fraudes em seguros médicos, por causa da mineração de dados. Padrões de reivindicações incomuns são mais fáceis de identificar com essa ferramenta e podem identificar encaminhamentos inadequados e solicitações médicas e de seguro fraudulentas. Quando as seguradoras reduzem suas perdas devido a fraudes, o custo dos cuidados em saúde também diminui.

Hospitais e grupos de serviços em saúde usam ferramentas de mineração de dados para alcançar melhores decisões relacionadas ao paciente. A satisfação do paciente é aprimorada, porque a mineração de dados fornece informações que ajudarão a equipe nas interações do paciente, reconhecendo padrões de uso, necessidades atuais e futuras, e preferências do paciente.

Como se pode ver, a mineração de dados é capaz de beneficiar a todos. Se está no comando de uma grande corporação ou de uma empresa menor, descobrirá que há algum uso de mineração de dados para você.

3.2 Por que mineração de dados?

Vivemos em um mundo onde temos muitos dados gerados e coletados todos os dias. Além disso, a necessidade de analisar estes tipos de dado é importante. Portanto, com a mineração de dados, temos a chance de transformar um grande conjunto de dados em conhecimento.

O mecanismo de pesquisa tem milhões de consultas inseridas todos os dias. Podemos ver cada consulta como uma transação em que o usuário pode explicar as informações ou suas necessidades. Curiosamente, temos certos padrões nas consultas de pesquisa do usuário, que podem revelar conhecimento crucial que ninguém encontra lendo apenas itens de dados individuais.

Existem muitas razões pelas quais trabalhar com mineração de dados é interessante. Neste momento, o volume de dados que está sendo produzido está

dobrando a cada dois anos, e a taxa provavelmente aumentará no futuro. Dados não estruturados, por si sós, são capazes de representar 90% do nosso universo digital. Mas mais informações nem sempre se traduzem em mais conhecimento.

Com a ajuda da mineração de dados, se pode analisar todo o ruído que é repetitivo e caótico nos dados que se tem à frente; entender o que é relevante, e fazer bom uso dessas informações para ajudar a avaliar os resultados prováveis que acontecerão. Pode acelerar a rapidez com que se é capaz de tomar decisões que são inteligentes, graças aos dados que estão disponíveis.

Como vimos, os métodos analíticos aplicados na mineração de dados são algoritmos e técnicas amplamente matemáticos. Muito importante é a maneira como as técnicas são usadas. Em suma, a mineração de dados tem muitos benefícios no mundo atual. As perspectivas de mineração de dados a longo prazo são amplamente diversas.

Referências

- AMARAL, Fernando. **Aprenda mineração de dados: teoria e prática**. Rio de Janeiro: Alta Book, 2016.
- BRAGA, Luis Paulo Vieira. **Introdução à mineração de dados**. 2. ed. ampl. e rev. Rio de Janeiro: E-Papers, 2005.
- CARVALHO, André Carlos Ponce de Leon de; FACELI, Katti; LORENA, Ana Carolina; GAMA, João. **Inteligência artificial: uma abordagem de aprendizado de máquina**. Rio de Janeiro: LTC, 2011.
- CARVALHO, Hialo Muniz. **Aprendizado de máquina voltado para mineração de dados: árvores de decisão**. 2015. 98 f. Trabalho de Conclusão de Curso – Universidade de Brasília – UnB Faculdade UnB Gama – FGA, Brasília, 2014.
- CASTRO, Leandro Nunes de; FERRARI, Daniel Gomes. **Introdução à mineração de dados: conceitos básicos, algoritmos e aplicações**. São Paulo: Saraiva, 2016.
- COSTA, Evandro *et al.* Mineração de dados educacionais: conceitos, técnicas, ferramentas e aplicações. **Jornada de Atualização em Informática na Educação**, v. 1, n. 1, p. 1-29, 2013.
- DA COSTA, Nattane Luíza. **Mineração de dados para classificação e caracterização de alguns vinhos Vitis Vinífera da América do Sul**. 2016. 99 f. Dissertação (Mestrado em Ciências da Computação) – Universidade Federal de Goiás, Instituto de informática (INF), Programa de Pós-Graduação em Ciências da Computação, Goiânia, 2016.
- ELMASRI, Ramez; NAVATHE, Shamkant B. **Sistema de banco de dados**. 6. ed. São Paulo: Pearson Addison Wesley, 2011.
- FAYYAD, Usama M.; HAUSSLER, David; STOLORZ, Paul E. **KDD for Science Data Analysis: Issues and Examples**. In: KDD-96 Proceedings, p. 50-56, 1996.
- GALVÃO, Noemi Dreyer; MARIN, Heimar de Fátima. Técnica de mineração de dados: uma revisão da literatura. **Acta Paulista de Enfermagem**, v. 22, n. 5, p. 686-690, 2009.
- GAMA, João *et al.* **Extração do conhecimento de dados: data mining**. 2. ed. Lisboa: Editora Silabo, 2015.
- GOLDSCHMIDT, Ronaldo Ribeiro. **Uma introdução à inteligência computacional: fundamentos, ferramentas e aplicações**. Rio de Janeiro: IST-Rio, 2010.
- GOLDSCHMIDT, Ronaldo; PASSOS, Emmanuel. **Data Mining: conceitos, técnicas, algoritmos, orientações e aplicações**. 2. ed. Rio de Janeiro: Elsevier, 2015.

HAND, David J.; MANNILA, Heikki; SMYTH, Padhraic. **Principles of data mining**. London: MIT Press, 2001.

IBM. **IBM DB2 Intelligent Miner for Data**. Disponível em: https://www.ibm.com/support/knowledgecenter/en/SSEPGG_9.5.0/com.ibm.im.overview.doc/c_ibm_db2_intelligent_miner_for_data.html. Acesso em: 16 maio 2017.

IBM. **IBM SPSS Modeler**. Disponível em: <http://www03.ibm.com/software/products/pt/spss-modeler>. Acesso em: 17 maio 2017.

JONES, Herbert. **Data Analytics: the ultimate guide to big data analytics for business, data mining techniques, data collection, and business intelligence concepts**. CreateSpace Independent Publishing Platform, 2018.

MANTUANI, Silvia Ribeiro; ALMEIDA, Edson de; ROCHA, José Carlos Ferreira da. Classificação de fatores que influenciam no crescimento, desenvolvimento e produtividade da cultura de soja. **Revista de Engenharia e Tecnologia**, v. 8, n. 3, p. 175-180, 2016.

QUILICI-GONZALEZ, José Artur; ZAMPIROLI, Francisco de Assis. **Sistemas inteligentes e mineração de dados**. Santo André: Triunfal Gráfica e Editora, 2014.

RAPIDMINER. **Rapidminer**. Disponível em: <https://rapidminer.com/products/>. Acesso em: 17 maio 2017.

SATO, Luciane Yumie *et al.* Análise comparativa de algoritmos de árvore de decisão do sistema WEKA para classificação do uso e cobertura da terra. *In: SIMPÓSIO BRASILEIRO DE SENSORIAMENTO REMOTO*, 16., 2013, Foz do Iguaçu. **Anais [...]**, Foz do Iguaçu, 2013.

SILVA, Leandro Augusto; PERES, Sarajane Marques; BOSCARIOLI, Clodis. **Introdução à mineração de dados com aplicação em R**. Rio de Janeiro: Elsevier, 2016.

VILELA NETO, Omar Paranaíba; PACHECO, Marco Aurélio Cavalcanti. **Nanotecnologia computacional inteligente: concebendo a engenharia em nanotecnologia**. Rio de Janeiro: Interciência: PUC-Rio, 2012.

WANG, John (ed.). **Data Warehousing and Mining: concepts, methodologies, tools, and applications: concepts, methodologies, tools, and applications**. IGI Global, 2008.

WEKA, Weka. Disponível em: <http://www.cs.waikato.ac.nz/ml/weka/book.htm>. Acesso em: 27 maio 2017.

REDES NEURAIS ARTIFICIAIS: INTRODUÇÃO E DEFINIÇÕES

Scheila de Avila e Silva,¹ Rafael Vieira Coelho²

Este capítulo apresenta conceitos fundamentais sobre as Redes Neurais Artificiais. Além da definição, apresentam-se conceitos relacionados com neurônios, arquiteturas e o processo de aprendizado que ocorre em dois tipos de arquiteturas de rede: *Perceptrons* e redes multicamadas (incluindo o algoritmo *back-propagation*).

1 Definições fundamentais

As Redes Neurais Artificiais (RN) modelam o processamento de informação e aprendizagem do cérebro. Trata-se de um modelo computacional aplicável a uma ampla variedade de áreas como engenharia, economia e biologia. Elas podem ser aplicadas na síntese e no reconhecimento de fala, *interface* adaptativa entre humanos e sistemas físicos complexos, aproximação de funções, etc. Mais especificamente na área da biologia, elas são aplicadas principalmente em problemas de análise de sequências e no reconhecimento de padrões (BALDI; BRUNAK, 2001).

Uma RN é um processador paralelamente distribuído constituído de unidades simples, com a propensão natural de armazenar conhecimento experimental e torná-lo disponível para o uso (HAYKIN, 1999). Em um nível mais elementar, ela consiste de redes com unidades interconectadas envolvidas no tempo. A conexão de uma unidade i com uma unidade j começa com um peso sináptico denotado por W_{ij} . Assim, pode-se representar uma RN como um grafo direcionado com peso ou arquitetura (MOUNT, 2000).

Uma RN é formada de camadas de unidades de processamento com conexões entre as mesmas. A unidade básica de uma camada é um neurônio artificial. Estas unidades, como os neurônios reais, têm conexões de entrada (dendritos) e conexões de saída (axônios). Assim como neurônios reais, as unidades da rede neural artificial têm alguma forma de processamento interno que cria um sinal de saída a partir do sinal de entrada.

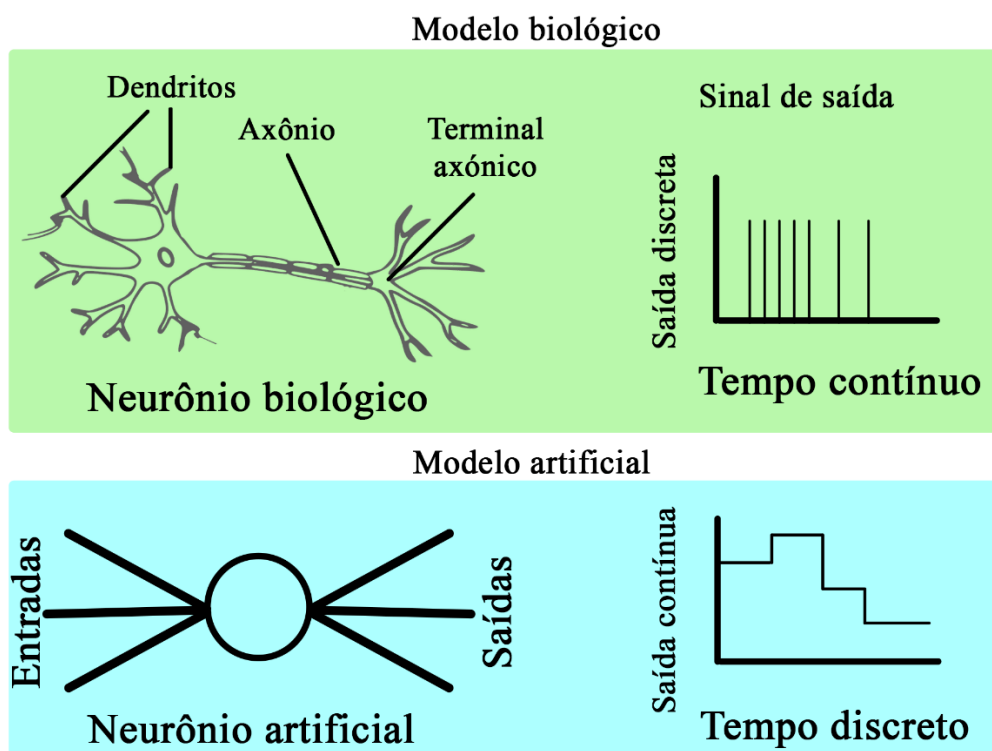
No entanto, existem duas diferenças fundamentais entre neurônios reais e artificiais: (i) a saída de um neurônio biológico é um pulso de sinal modulado (série de pulsos de amplitude fixa) com sua frequência mudando em resposta aos sinais recebidos pelos dendritos, enquanto o neurônio artificial tem como saída um número; e (ii) a saída de um neurônio biológico muda continuamente com o tempo. Já a de um artificial

¹ Universidade de Caxias do Sul. *E-mail*: sasilva6@ucs.br

² Instituto Federal do Rio Grande do Sul. *E-mail*: rafael.coelho@farroupilha.ifrs.edu.br

apresenta mudanças somente em um intervalo discreto de tempo, conforme é ilustrado na Figura 1 (WU; MCLARTY, 2000).

Figura 1 – Analogia entre neurônios biológicos e artificiais. Aqui compara-se a estrutura e o sinal de saída



Fonte: Wu e Mclarty, (2000).

2 Arquiteturas de redes neurais

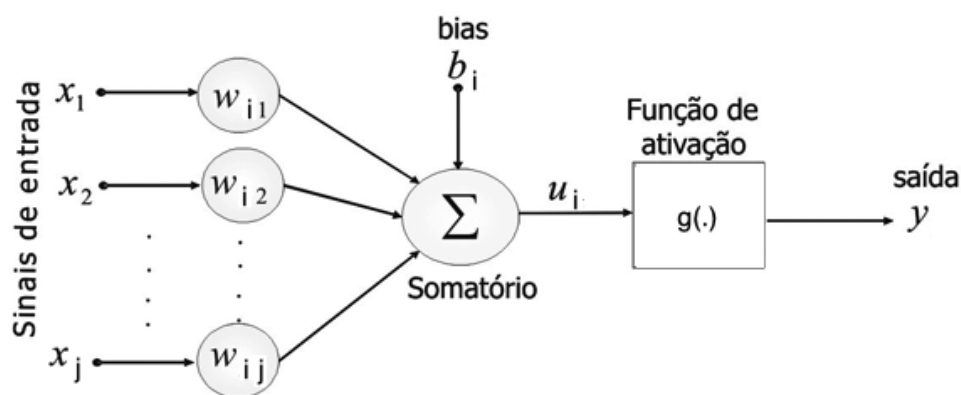
Uma RN é caracterizada pelo (i) padrão de conexões entre os neurônios (chamado de arquitetura), (ii) método de determinação de pesos nas conexões (chamado de treinamento ou aprendizagem); e (iii) sua função de ativação.

Os neurônios são conectados por **vínculos** orientados (Figura 2). Um vínculo da unidade j para a unidade i serve para propagar a **ativação** x_j desde j até i . Cada vínculo também tem um peso numérico W_{ij} associado a ele, o qual determina a intensidade e o sinal da conexão. Especificamente, um sinal x_j na entrada da sinapse i conectada ao neurônio j é multiplicada pelo peso sináptico W_{ij} (RUSSELL, 2003; HAYKIN, 1999). Após, cada unidade j calcula inicialmente uma soma ponderada de suas entradas (Equação 1). Então ela aplica uma função de ativação g a essa soma para derivar a saída (Equação 2).

Figura 2 – Modelo de um neurônio artificial. A ativação da saída da unidade é $X_i = g(\sum_{j=0}^n W_{ji}a_j)$, onde x_j é a ativação de saída da unidade j e W_{ij} é o peso no vínculo da unidade j até essa unidade

$$in_i = \sum_{j=0}^n W_{ji}a_j . \quad (1)$$

$$a_i = g(in_i) = g\left(\sum_{j=0}^n W_{ji}a_j\right) . \quad (2)$$



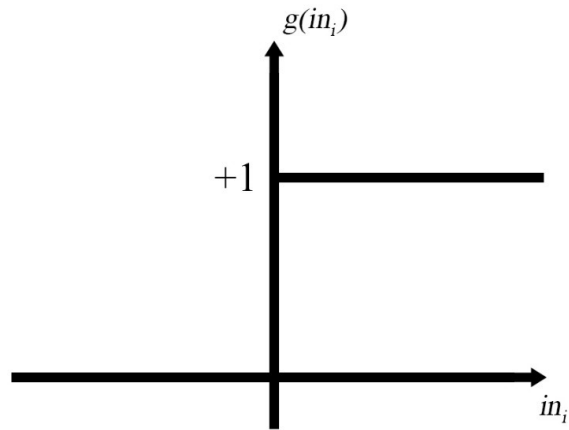
Fonte: Adaptado de Russell (2003).

A função de ativação g é projetada para atender a duas aspirações: primeiro, a unidade de estar “ativa” (próxima de +1), quando as entradas positivas forem recebidas e negativas (próxima a 0) quando as entradas “erradas” forem recebidas. Em segundo lugar, a ativação precisa ser não-linear, caso contrário a RN inteira entrará em colapso, tornando-se uma função linear simples (RUSSELL, 2003; HAYKIN, 1999). Segundo Haykin (1999), existem três tipos básicos de funções de ativação, que são detalhados a seguir e podem ser visualizados nas Figuras 3 até 5. A função de ativação, representada por g , define a saída de um neurônio em termos do campo local induzido (in_i), ou seja, entrada dos neurônios. São elas:

Função de Limiar: conforme a Figura 3, para este tipo de função de ativação (Equação 3), esta função assume apenas valores 0 ou 1.

Figura 3 – Função de limiar

$$g(in_i) = \begin{cases} 1 & \text{se } in \geq 0 \\ 0 & \text{se } in < 0 \end{cases} \quad (3)$$

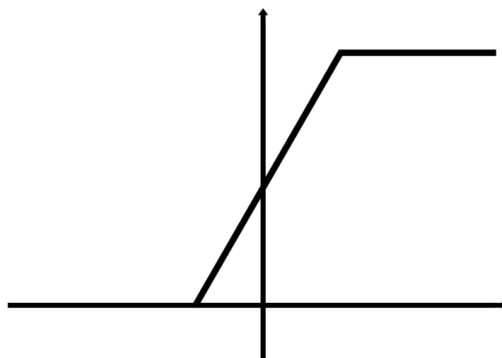


Fonte: Adaptado de Russell (2003).

Função Linear por partes: para este tipo de função de ativação assume que o fator de amplificação dentro da região linear de operação é a unidade (vide Equação 4). Esta forma de função de ativação pode ser vista como uma aproximação de um amplificador não-linear. A forma desta função de ativação pode ser vista na Figura 4.

Figura 4 – Função linear por partes

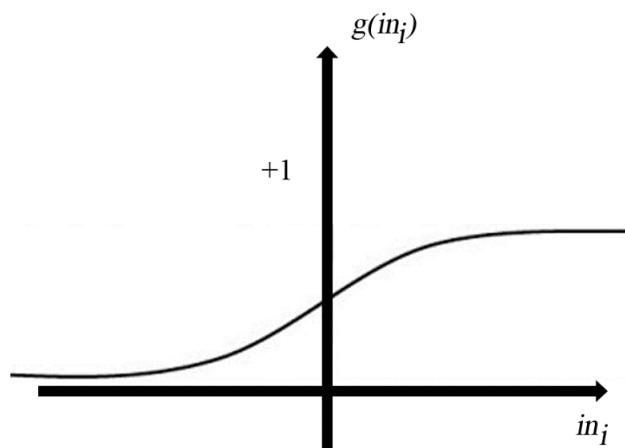
$$g(in_i) = \begin{cases} 1, & in \geq +1/2 \\ in, + \frac{1}{2} > & in > -1/2 \\ 0, & in \leq -1/2 \end{cases} \quad (4)$$



Fonte: Adaptado de Russell (2003).

- *Função Sigmoide*: esta é a forma mais comum de função de ativação utilizada em RN. Ela é definida como uma função estritamente crescente, que exibe um balanceamento adequado entre comportamento linear e não linear (Equação 5). Um exemplo de função sigmoide é a função logística, conforme ilustrado na Figura 5. Enquanto a função de limiar assume apenas valores 0 e 1, a função sigmoide assume valores contínuos entre 0 e 1 e a inclinação final desta função se dá pelos ajustes dos pesos.

Figura 5 – Função Sigmoide

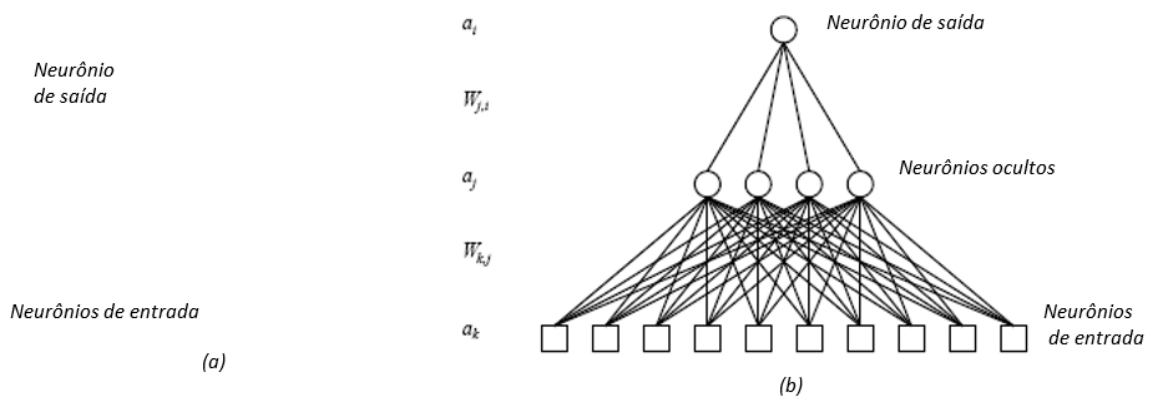


Fonte: Adaptado de Russell (2003).

2.1 Perceptrons e Redes Neurais Multicamadas

Perceptron é um modelo de rede na qual todas as entradas são conectadas diretamente às saídas (vide Figura 6a). Cada peso afeta apenas a sua respectiva saída, pois cada unidade de saída é independente (HAYKIN, 1999). Minsky e Papert (1969) provaram matematicamente que este tipo de estrutura de processamento apresenta limitações e só pode ser aplicada com sucesso em problemas linearmente separáveis.

Figura 6 – (a) Modelo de rede com duas camadas (*Perceptron*) e, (b) rede com três camadas – *Multilayer Perceptron (MLP)*



Fonte: Adaptado de Russell (2003).

Uma arquitetura com maior capacidade de generalização é composta de múltiplas camadas na rede (*Multilayer Perceptron* – MLP). Na Figura 6b é apresentado um exemplo com uma camada oculta. Segundo Hornik (1989), as RN com uma única camada oculta são aproximadores universais, pois aproximam qualquer função com precisão arbitrária. A função dos neurônios ocultos é intervir entre a entrada externa e a saída de maneira útil.

As vantagens do uso de camadas ocultas é o aumento do espaço de hipóteses que a rede pode representar e, por consequência, a capacidade de extrair estatísticas de ordem elevada. Isto é particularmente valioso, quando o tamanho da camada de entrada é grande. No entanto, o custo computacional é proporcional ao número de camadas ocultas adicionadas. Além disso, redes com múltiplas camadas ocultas apresentam menor capacidade de generalização, se comparadas a redes com uma única camada oculta. Por fim, a extração das regras da rede se torna mais difícil (MOUNT, 2000).

2.2.1 Treinamento de redes neurais

Segundo Wu e McLarty (2000), o treinamento de uma RN consiste em atribuir valores a um conjunto de pesos (inicializado normalmente de forma aleatória). A partir da aplicação dos dados de entrada à rede, deve-se verificar como esta responde a determinados conjuntos de pesos. Se o desempenho não for satisfatório, então os pesos são modificados pelo algoritmo específico da arquitetura e repete-se o procedimento. Este deve ser repetido até que algum critério de parada pré-especificado seja atingido.

A passagem de todos os vetores dos dados de entrada através da rede é chamado de *época*. Alterações nos pesos podem ser feitas a cada padrão processado (treinamento *online*) ou após uma época inteira (treinamento em lote), sendo este o procedimento mais utilizado. O objetivo do treinamento é encontrar o conjunto de parâmetros (número de camadas, número de neurônios nas camadas e pesos entre as camadas), que minimize a diferença entre os valores de saída da rede e os valores desejados.

Outro fator que deve ser considerado, durante o treinamento, é a estrutura da rede. Se ela tiver uma arquitetura com camadas ocultas em excesso ou for treinada por muitas épocas (*overtraining*), ela será capaz de memorizar todos os exemplos. Assim, ela forma uma extensa tabela de busca, mas não realiza generalizações aceitáveis para entradas não testadas. Existem alguns métodos de testar a exatidão da rede que podem ser utilizados: (i) *holdout*; (ii) *k-fold-cross-validation* (*k*-FCV); e (iii) *jackknife*.

O método *holdout* consiste em separar, de forma aleatória, o arquivo de padrões em dois arquivos. O de treinamento tipicamente conterá dois terços dos dados, e o de validação o terço restante. Já a técnica de validação cruzada ou *k-fold-cross-validation* (*k*-FCV) consiste em particionar aleatoriamente o arquivo de padrões em *k* partes de

mesmo tamanho. As etapas de treinamento e validação são repetidas k vezes, sendo utilizados para treinamento $k-1$ arquivos e para validação o k -ésimo não utilizado no treinamento. A cada interação, o arquivo de validação possui um k diferente. Por fim, o método *jackknife*, conhecido como *leave-one-out*, é semelhante ao k -FCV, mas k é igual ao número de linhas do arquivo de padrões. Com isto, cada arquivo de validação conterá somente uma linha em cada etapa do processo (BALDI; BRUNAK, 2001).

2.2.2 Algoritmos de aprendizado

Os algoritmos de aprendizado podem ser supervisionados ou não supervisionados, embora aspectos de cada um possam coexistir em uma arquitetura. O treinamento supervisionado é acompanhado pela apresentação de uma sequência no vetor de treinamento, associada com um vetor de saída-alvo. Este tipo de treinamento exige que um especialista externo faça ajustes iterativos, para minimizar o erro de acordo com o algoritmo de aprendizado (WU, 1997). Já um algoritmo de aprendizado supervisionado tem o objetivo de minimizar a diferença entre o valor de saída da rede e o valor desejado. Uma típica função de erro a ser minimizada, pode ser observada na Equação 6.

$$E = \sum_{i=1}^n (y_i - h_w(x))^2 \quad (6)$$

Na Equação 6, n é o número de padrões de entrada, y_i é a saída da rede (para um dado conjunto de parâmetros w) e $h_w(x)$ o valor esperado de saída. Se uma rede possui mais que uma unidade na camada de saída, então a Equação 6 se torna a Equação 7.

$$E = \sum_{i=1}^n \sum_{j=1}^k (y_{ij} - h_w(x))^2 \quad (7)$$

2.2.3. Treinamento de Perceptrons

Segundo Russell (2003), os algoritmos de aprendizagem de RN visam ao ajuste dos pesos da rede para minimizar alguma medida do erro, no conjunto de treinamento. Desse modo, a aprendizagem é formulada como uma busca de otimização no espaço de pesos, sendo a medida tradicional a soma dos erros quadráticos. O erro quadrático para um único exemplo de treinamento com entrada x e saída verdadeira y é descrito na Equação 8.

$$E = \frac{1}{2} Err^2 \equiv \frac{1}{2} (y - h_w(x))^2 \quad (8)$$

É possível usar o declínio do gradiente para reduzir o erro quadrático, calculando a derivada parcial de E , em relação a cada peso. O ajuste nos pesos é dado pelas Equações 9 a 12.

$$\Delta w(j) = \frac{-\alpha \partial E}{\partial w(j)} \quad (9)$$

$$\frac{\partial E}{\partial W_j} = Err \times \frac{\partial Err}{\partial W_j} \quad (10)$$

$$= Err \times \frac{\partial}{\partial W_j} g \left(y - \sum_{j=0}^n W_j x_j \right) \quad (11)$$

$$= -Err \times g'(in) \times x_j \quad (12)$$

A taxa de aprendizagem é dada por α e g' . A taxa de aprendizagem é dada por α e g' é a derivada da função de ativação. No algoritmo de declínio do gradiente, onde se quer reduzir E , o peso é atualizado conforme a Equação 13.

$$W_j \leftarrow W_j + \alpha \times Err \times g'(in) \times x_j \quad (13)$$

Se o erro $Err = y - h_w(x)$ for positivo, então a saída da rede é pequena demais. Portanto, os pesos devem ser aumentados para as entradas positivas e diminuídos para as entradas negativas. Acontece o oposto quando o erro é negativo.

O algoritmo do aprendizado de declínio do gradiente para *perceptrons* é mostrado a seguir (RUSSELL, 2003):

Algoritmo do aprendizado de declínio do gradiente para *perceptrons*

função APRENDIZAGEM-DE-PERCEPTRON (*exemplos, rede*) **retorna** uma hipótese de perceptrons.

entrada: *exemplos*, um conjunto de exemplos, cada um com entrada $\mathbf{x} = x_1, \dots, x_n$ e saída y

rede, um perceptron com pesos $W_j, j = 0 \dots n$ e função de ativação g

repita

para cada e **em** *exemplos* **faça**

$$in \leftarrow \sum_{j=0}^n W_j x_j[e]$$

$$Err \leftarrow y[e] - g(in)$$

$$W_j \leftarrow W_j + \alpha \times Err \times g'(in) \times x_j x_j[e]$$

até algum critério de parada ser satisfeito

retornar HIPÓTESE-DA-REDE NEURAL (*rede*)

Fonte: Adaptado de Russell (2003).

2.2.4 Treinamento de Redes Neurais Multicamadas

Redes multicamadas apresentam as seguintes características:

- o modelo de cada neurônio da rede inclui uma *função de ativação não linear*. A presença da não linearidade é importante porque, do contrário, a relação entrada-saída da rede poderia ser reduzida àquela de *perceptron* de camada única;
- a rede contém uma ou mais camadas de neurônios ocultos, que não são parte da entrada ou da saída da rede. Estes neurônios capacitam a rede a aprender tarefas complexas, extraíndo progressivamente as características mais significativas dos vetores de entrada;
- a rede exhibe um alto grau de conectividade, determinado pelas sinapses da rede.

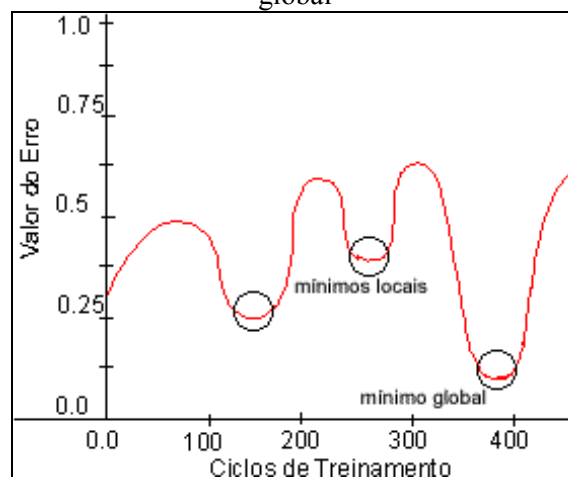
Rumelhart *et al.* (1986) definiram o método *Back-Propagation* (BP), cujo erro é propagado da saída para a entrada da rede. Ou seja, a propagação do erro pode ser efetuada da camada de saída para a camada oculta e desta para a camada de entrada. O processo de propagação de retorno emerge diretamente de uma derivação do gradiente

de erro global e da aplicação da regra da cadeia (WU; McLARTY, 2000; RUSSELL, 2003). O treinamento com o BP envolve três estágios: (i) o *feedforward* dos padrões dos dados de entrada do treinamento; (ii) o cálculo e a *back-propagation* do erro associado; e (iii) o ajuste dos pesos.

Na fase *feedforward*, os pesos permanecem inalterados através da rede, e os sinais de função da rede são computados neurônio por neurônio até produzir um conjunto de saídas como a resposta real da rede. Já na fase *back-propagation*, os sinais de erro são computados recursivamente para cada neurônio, iniciando da camada de saída e passando de trás para frente através da rede, para produzir o erro das unidades ocultas. Por fim, os pesos são ajustados para diminuir a diferença entre a saída obtida pela rede e a saída desejada, de acordo com uma regra de correção de erro. Para isto, ele utiliza um método de descida do gradiente (WU, 1997; WU; McLARTY, 2000; HAYKIN, 1999).

Russell (2003) afirma que o gradiente de uma função está na direção e no sentido em que a função tem taxa de variação máxima. Isto garante que a rede caminha na superfície, na direção que vai reduzir mais o erro obtido. Para superfícies simples, este método certamente encontra a solução com erro mínimo. Para superfícies complexas, esta garantia não mais existe, podendo levar o algoritmo a convergir para mínimos locais, conforme a Figura 7.

Figura 7 – Gráfico de uma possível superfície de erro indicando mínimos locais e mínimo global



Fonte: Adaptado de Russell (2003).

O cálculo da propagação do erro $y - h_w$ da camada de saída para as camadas ocultas emerge diretamente do gradiente de erro global. Na camada de saída, a regra de atualização dos pesos é idêntica à Equação 9. Assim, no BP, os pesos são atualizados conforme a Equação 14.

$$W_{ji} \leftarrow W_{ji} + \Delta W_{ji} \quad (14)$$

ΔW_{ji} é dado pela Equação 15.

$$\Delta W_{ji} = -\alpha \frac{\partial E}{\partial W_{ji}} \quad (15)$$

A expressão $\partial E / \partial W_{ji}$ parte do erro quadrático, definido pela Equação 16.

$$E = \sum_{i=1}^n (y_i - a_i)^2 \quad (16)$$

Na Equação 17, a ativação a_i é expandida.

$$\begin{aligned} \frac{\partial E}{\partial W_{ji}} &= (y_i - a_i) \frac{\partial a_i}{\partial W_{ji}} = -(y_i - a_i) \frac{\partial g(in_i)}{\partial W_{ji}} & (17) \\ &= -(y_i - a_i) g'(in_i) \frac{\partial g(in_i)}{\partial W_{ji}} - (y_i - a_i) g'(in_i) \frac{\partial}{\partial W_{ji}} \left(\sum_j W_{ji} a_j \right) \\ &= -(y_i - a_i) g'(in_i) a_j = -a_j \times \Delta_i \end{aligned}$$

Finalizando a regra de atualização dos pesos da camada de saída para a camada oculta, conforme a Equação 18.

$$W_{ji} \leftarrow W_{ji} + \alpha \times a_j \times \Delta_i \quad (18)$$

Para atualizar as conexões entre as unidades de entrada e as unidades ocultas, efetua-se a propagação de retorno do erro. O neurônio oculto j é responsável por alguma fração do erro Δ_i em cada um dos nós de saída aos quais ele está conectado. Deste modo, os valores Δ_i são divididos de acordo com a intensidade da conexão entre o nó oculto e o nó de saída. E são propagados de volta para fornecer os valores Δ_j referentes à camada oculta.

Desta forma, a regra de atualização de pesos correspondente aos pesos entre as entradas e a camada oculta é quase idêntica à regra de atualização para a camada de saída (Equação 19).

$$W_{kj} \leftarrow W_{kj} + \Delta W_{kj} \quad (19)$$

ΔW_{kj} é dado pela Equação 20.

$$\Delta W_{kj} = -\alpha \frac{\partial E}{\partial W_{kj}} \quad (20)$$

E $\partial E / \partial W_{kj}$ origina-se da expansão das ativações a_i e a_j , conforme as equações 21 e 22.

$$\begin{aligned} \frac{\partial E}{\partial W_{kj}} &= (y_i - a_i) \frac{\partial a_i}{\partial W_{kj}} = -\sum_i (y_i - a_i) \frac{\partial g(in_i)}{\partial W_{kj}} & (21) \\ &= -\sum_i (y_i - a_i) g'(in_i) \frac{\partial g(in_i)}{\partial W_{kj}} = -\sum_i \Delta_i \frac{\partial}{\partial W_{kj}} \left(\sum_j W_{ji} a_j \right) \\ &= -\sum_i \Delta_i W_{ji} \frac{\partial a_i}{\partial W_{kj}} = \sum_i \Delta_i W_{ji} \frac{\partial g(in_j)}{\partial W_{kj}} \\ &= -\sum_i \Delta_i W_{ji} g'(in_j) \frac{\partial g(in_j)}{\partial W_{kj}} \\ &= -\sum_i \Delta_i W_{ji} g'(in_j) \frac{\partial}{\partial W_{kj}} \left(\sum_k W_{kj} a_k \right) \\ &= -\sum_i \Delta_i W_{ji} g'(in_j) a_k = -a_k \times \Delta_j \end{aligned}$$

$$\frac{\partial E}{\partial W_{kj}} = -a_k \times \Delta_j \quad (22)$$

Onde Δ_j é definido pela Equação 23.

$$\Delta_j = g'(in_j) \sum_i W_{ji} \Delta_i \quad (23)$$

A Equação 24 define a regra de atualização dos referidos pesos.

$$W_{kj} \leftarrow W_{kj} + \alpha \times a_k \times \Delta_j \quad (24)$$

O processo de propagação de retorno pode ser resumido da seguinte forma:

1. calcular os valores Δ para as unidades de saída, usando o erro observado.
2. começando pela camada de saída, repetir as etapas a seguir para cada camada na rede, até ser alcançada a camada oculta conectada à camada de entrada: (i)

propagar os valores Δ de volta até a camada anterior; (ii) atualizar os pesos entre as duas camadas.

O algoritmo BP é apresentado a seguir (RUSSELL, 2003):

Algoritmo BP
Função APRENDIZAGEM-POR-PROPAGAÇÃO-DE-RETORNO (<i>exemplos, rede</i>) retorna uma rede neural
Entradas: <i>exemplos</i> , um conjunto de exemplos, cada um com vetor de entrada x e vetor de saída y <i>rede</i> , uma rede de várias camadas com L camadas, pesos W_{ji} e função de ativação g
Repita
Para cada e em <i>exemplos</i> faça
Para cada neurônio j na camada de entrada faça $a \leftarrow x_j[e]$
Para $l = 2$ até M faça
$in_l \leftarrow \sum_j W_{jl} a_j$
$a_l \leftarrow g(in_l)$
Para cada neurônio i na camada de saída faça
$\Delta_i \leftarrow g'(in_i) \times (y_i[e] - a_i)$
Para $l = M - 1$ até 1 faça
Para cada neurônio j na camada $l + 1$ faça
$W_{ji} \leftarrow W_{ji} + \alpha \times a_j \times \Delta_i$
Até algum critério de parada ser satisfeito
Retornar HIPÓTESE-DA-REDE NEURAL (<i>rede</i>)

Fonte: Adaptado de Russell (2003).

O desenvolvimento do algoritmo BP representa um marco nas RNs, pois fornece um método computacional eficiente para o treinamento de MLPs. Apesar de não podermos afirmar que o algoritmo forneça uma solução ótima para todos os problemas resolúveis, ele acabou com o pessimismo sobre a aprendizagem em máquinas de múltiplas camadas.

2.2 Extração de conhecimento das redes neurais treinadas

RN não requerem conhecimento prévio da aplicação do problema para a construção do modelo. Sendo assim, para tornar esta tecnologia compreensível ao

usuário, deve-se extrair conhecimento, a partir das redes treinadas para definir regras biológicas essenciais (WU; MCLARTY, 2000). Estas regras incluem: regras de inferência (*if-then-else*), árvores de decisão, regras difusas, entre outras.

Apesar da classe de regras geradas, algumas características devem ser buscadas, conforme descrito em Tickle *et al.* (1998): (i) poder expressivo ou formato da regra; (ii) qualidade; (iii) translucidez; (iv) complexidade algorítmica da regra ou da técnica de extração das regras; e (v) portabilidade.

Poder expressivo sugere três grupos de formatos de regras: regras simbólicas convencionais (Booleana, proposicional), regras baseadas em lógica *fuzzy* e as regras expressas na forma de lógica de primeira ordem. A qualidade da regra é dada por um conjunto de quatro medidas: (i) acurácia (o grau com o qual o conjunto de regras extraídas é capaz de classificar exemplos “não vistos” de forma correta); (ii) fidelidade (grau de similaridade entre as regras extraídas e a RN da qual se originaram); (iii) consistência (indica o grau com que, sob diferentes treinamentos, a RN gera regras que produzam a mesma classificação para os casos “não vistos”) e (iv) compreensibilidade (tamanho do conjunto de regras extraídas – número de regras e de antecedente por regra) (ARBATLI; AKIN, 1997; ANDREWS *et al.*, 1995; TICKLE *et al.*, 1998).

Já o critério de translucidez busca categorizar uma técnica de extração de regras baseada na granularidade da RN, a qual pode ser implícita ou explícita. Conforme Andrews *et al.* (1995), existem três identificadores-chave (decomposicional, eclética e pedagógica), para definir pontos de referência no espectro de tais níveis de granularidade percebidos.

Por fim, a portabilidade define a extensão de que uma dada técnica possa ser aplicada através de um grupo de arquiteturas de RNs e regimes treinados. A necessidade para este critério foi estabelecida com base na característica das técnicas para extração de regras, a partir de RNs treinadas foi a preponderância.

Segundo Andrews *et al.* (1995), a extração de regras pode oferecer benefícios: descoberta de novos relacionamentos e/ou características importantes a partir das regras extraídas; expressão do conhecimento de modo formal; capacidade de gerar explicações para as decisões tomadas internamente pela RN, de modo que facilite a aceitação do uso da rede pelos usuários; integração com sistemas simbólicos e, assim, a possibilidade de descobrir em que situações a rede pode cometer erros de generalização; e identificação de regiões no espaço de entrada que não se fizeram representar no conjunto de treinamento.

A extração de regras é baseada no comportamento dos neurônios, sendo a relação entre as entradas e as saídas usualmente analisada (CLOETE; ZURADA, 2000; HUANG; XING, 2005). Elas podem ser apresentadas para um especialista, que as

analisa e verifica se há incorreções. Assim, as regras corretas podem ser usadas para gerar padrões de treinamento adequados, os quais podem melhorar a capacidade de generalização da rede (CLOETE; ZURADA, 2000).

A extração de regras é realizada através da interpretação dos pesos da rede neural e pode ser feita utilizando lógica *fuzzy*. Esta também permite que as regras sejam expressas na estrutura *if-then-else*. Neste tipo de regra, a parte SE especifica um conjunto de condições sobre valores de atributos previsores e a parte ENTÃO especifica um valor previsto para o atributo de saída. Os atributos previsores são as premissas da regra que devem ser obedecidas, para assim obter um atributo classe.

IF < condição> *THEN* < conclusão> (<confidência>)

A condição é uma expressão lógica que contém variáveis relevantes e das quais os valores podem ser inferidos a partir das bases de fatos ou fornecidos pelo usuário. A conclusão determina o valor de alguma variável que corresponde para a condição ser satisfeita. O grau de certeza ou validade da regra é expresso pelo seu percentual de confidência (CLOETE; ZURADA, 2000).

As regras do tipo *if-then* podem ser utilizadas posteriormente em um sistema de inferência lógica para a resolução de problemas. Um segundo uso destas regras pode ser a geração de regras para um sistema baseado em conhecimento. Deve-se observar, também, que quanto mais curtas as regras (em termos de números de cláusulas) melhor, pois regras curtas geralmente podem ser aplicadas a mais situações (CLOETE; ZURADA, 2000).

Segundo Andrews *et al.* (1995), a extração de regras pode ser feita pelas seguintes categorias de técnicas: decomposicional, pedagógica e eclética. Esta combina elementos das duas categorias básicas.

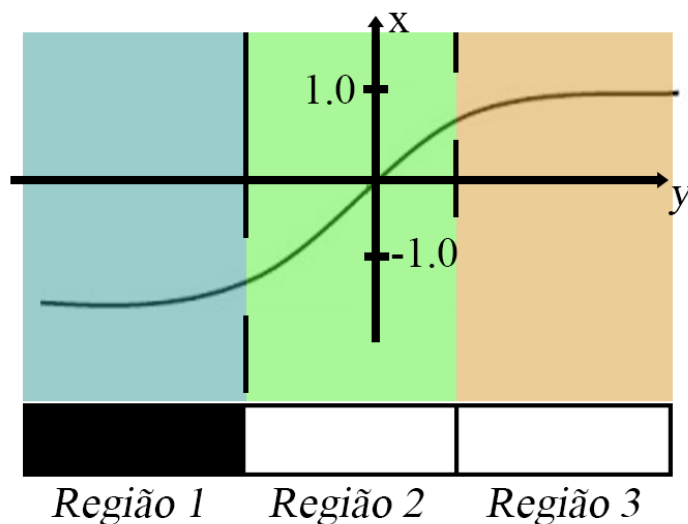
Na técnica decomposicional, o foco da extração de regras está no nível de unidades individuais (camada oculta e saída). Assim, a estrutura da rede é a principal fonte de regras. Um requerimento básico para a extração de regras desta abordagem é que a saída computada em cada unidade oculta e de saída pode ser mapeada em um resultado binário (sim/não). Já a técnica pedagógica visa à extração de regras como uma tarefa de aprendizado, na qual o conceito alvo é a função computada pela rede. As características de entrada são simplesmente as características das entradas da rede. Portanto, esta técnica objetiva a extração de regras que mapeiam as entradas diretamente com as saídas, sem se preocupar com os passos intermediários (ANDREWS *et al.*, 1995).

Por fim, a técnica eclética inclui abordagens que utilizam o conhecimento sobre a arquitetura interna e/ou vetores de pesos para complementar um algoritmo de

aprendizado simbólico, que utiliza dados de treinamento (ANDREWS *et al.*, 1995). Portanto, a técnica não é estritamente decomposicional, porque não extrai regras de neurônios individuais com subsequente agregação para formar uma relação global; não pode ser eclética porque não há aspecto que o enquadre no perfil pedagógico (TICKLE *et al.*, 1998).

Para a obtenção de regras, a partir dos neurônios da camada oculta da RN treinada, o programa denominado FAGNIS (CECHIN, 1998) analisa o valor de ativação dos neurônios na camada oculta e os classifica em três regiões, conforme ilustrado na Figura 8. Para cada entrada da rede, verifica-se em qual das regiões a ativação dos neurônios ocultos se enquadra. O número máximo de combinações possíveis é 3^n , onde n simboliza o número de neurônios na camada oculta. No entanto, nem todas estas combinações ocorrem e, somente as combinações mais frequentes são consideradas, porque elas representam melhor os dados. Como resultado, temos o protótipo da regra. Por protótipo, definimos a média das entradas de cada grupo (combinação das regiões). Assim, a escrita formal da regra possui a forma de uma equação linear: “SE $X \cong$ protótipo ENTÃO $Y =$ constante da equação linear + (os coeficientes da equação linear) * X .”

Figura 8 – Ilustração das três regiões definidas na função sigmoide para análise dos dados de entrada e extração de regras



Fonte: Russel (2003).

Referências

ANDREWS, Robert; DIEDERICH, Joachim; TICKLE, Alan B. A survey and critique of techniques for extracting rules from trained artificial neural networks. **Knowledge-Based Systems**, v. 8, n. 6, p. 373-389, 1995.

ARBATLI, A. D.; AKIN, H.L. Rule extraction from trained neural networks using genetic algorithms. **Non linear analysis, Theory, Methods e Applications**, v. 30, n. 3, p. 1639-1648, 1997.

- BALDI, Pierre; BRUNAK, Soren. **Bioinformatics: the machine learning approach**. 2. ed. Cambridge: MIT, 2001. 351 p.
- CECHIN, Adelmo Luis. **The extraction of Fuzzy Rules from Neural Networks**. 1998. 149 f. Tese (Doutorado em Informática) – Aachen: Shaker, [1998].
- CLOETE, Ian.; ZURADA, Jacek M. **Knowledge-Based neurocomputing**. Cambridge: MIT, 2000. 486 p.
- COTIK, V.; ZALIZ, R. Romero; ZWIR, I. A hybrid promoter analysis methodology for prokaryotic genomes. **Fuzzy Sets and Systems**, v.152, n. 1, p. 83-102, 2005.
- HAYKIN, Simon. **Neural networks: a comprehensive foundation**. 2. ed. New Jersey: Prentice-Hall, 1999.
- HORNIK, Kurt. Multilayer feedforward networks are universal approximators. **Neural Networks**, v. 2, n. 5, p. 359-366, 1989.
- HUANG, Samuel H.; XING, Hao. Extract intelligible and concise fuzzy rules from neural networks. **Fuzzy Sets and Systems**, v.132, p. 233-243, 2005.
- MOUNT, David W. **Bioinformatics: sequence and genome analysis**. New York: Cold Spring Harbor Laboratory, 2000.
- RUMELHART, David E; HINTON, Geoffrey E.; WILLIAMS, RONALD J Learning representations by backpropagating errors. **Nature**, v. 323, n. 6088, p. 533-536, 1986.
- RUSSELL, Stuart J.; NORVIG, Peter. **Artificial intelligence: a modern approach**. 2. ed. New Jersey:Prentice Hall International, 2003.
- WU, Cathy H. Artificial neural networks for molecular sequence analysis. **Computers & Chemistry**, v. 21, n. 4, p. 237-256, 1997.
- WU, Cathy H.; MCLARTY, Jerry Wayne. **Neural networks and genome informatics**. New York: Elsevier, 2000. 205 p.

1 Introdução

Diariamente somos desafiados a reconhecer objetos, situações, sentimentos, pessoas utilizando as mais diversas fontes de informações. Apesar de não termos sido expostos a todas as possibilidades, utilizamo-nos de experiências passadas para identificar as informações percebidas e relacionar com o que temos em memória. De certa forma, a quantidade e a variedade de experiências nos permitirão refinar este processo de identificação e, assim, reconhecer mais e melhor objetos, situações, sentimentos, pessoas, entre outros.

O reconhecimento de padrões baseado em experiências é replicado nas máquinas através do aprendizado supervisionado (DUDA; HART; STORK, 2001; MITCHELL, 1997; THEODORIDIS; KOUTROUMBAS, 2008). Utilizando-se de padrões previamente coletados, algoritmos são desenvolvidos, ou de forma análoga treinados, para reconhecer tais padrões. É importante ressaltar que todo este processo de aprendizado se utiliza da categoria ou classe dos padrões a serem reconhecidos para obter um modelo que descreve a estrutura ou similaridades entre os padrões. Isto é, todo aprendizado é realizado apresentando a possível informação de entrada e a saída que deve ser produzida pelo algoritmo. Mas como explicar o reconhecimento de padrões quando novas experiências são reconhecidas, mesmo sem ter experiências passadas?

O processo cognitivo de reconhecimento de padrões é mais complexo do que uma comparação de um padrão com referências armazenadas em memória (*template matching*). Padrões podem sofrer alterações com o passar do tempo ou o ambiente (por exemplo, envelhecimento) ou podem não conter todas as informações (por exemplo, oclusão de parte da imagem de uma pessoa). A psicologia cognitiva possui diversas teorias que tentam explicar esta capacidade, como a do protótipo (uma característica abstrata do padrão de um tipo de objeto) que permite lembrar de um avião quando vimos um cilindro com asas e a das características na qual o objeto é descrito através de características em vez de um protótipo ou uma referências (PI; LU; LIU; LIAO, 2008). Por isso, pode-se afirmar que, no caso de padrões que ainda não são conhecidos (sem experiências passadas), o processo cognitivo deve encontrar alguma forma de capturar as similaridades e/ou diferenças, a fim de produzir classes ou categorias de padrões.

¹ Universidade de Caxias do Sul. *E-mail*: agadami@ucs.br

² Universidade de Caxias do Sul. *E-mail*: amiorell@ucs.br

Métodos de agrupamento (*clustering*) são utilizados para descobrir características ou estruturas que permitam organizar padrões em grupos ou categorias, sem informação *a priori*. Em tais problemas, os padrões não possuem informação sobre a qual classe ou categoria pertencem (aprendizado não supervisionado). Em uma primeira análise, parece que não há nada para se aprender a partir de tais padrões, mas a organização de tais padrões permite derivar conclusões com base nas similaridades e diferenças encontradas entre os padrões. O avanço em diferentes áreas do conhecimento depende da compreensão dos relacionamentos que existem entre os padrões, como o que existe entre as espécies na biologia e os elementos químicos na Química (KEMP; TENENBAUM, 2008).

2 Análise por agrupamento

A análise por agrupamento tem por objetivo arranjar os dados em grupos distintos, nos quais os membros de um grupo compartilham semelhanças entre si, mas não entre dados de grupos diferentes. Conforme Fukunaga (1990), são algoritmos iterativos que recebem pontos de dados (geralmente vetores) e, baseados em critérios que definem medidas para separar as classes, agrupam estes dados.

Em linhas gerais, todos os métodos de agrupamento produzem a partição de um conjunto de dados em grupos não vazios e mutuamente exclusivos (WEBB; COPSEY, 2011). Para um conjunto formado por n elementos $\mathcal{X} = \{x_1, x_2, \dots, x_n\}$, a partição de \mathcal{X} em k grupos, representada por $\mathcal{P} = \{C_1, \dots, C_k\}$, deve satisfazer as seguintes condições

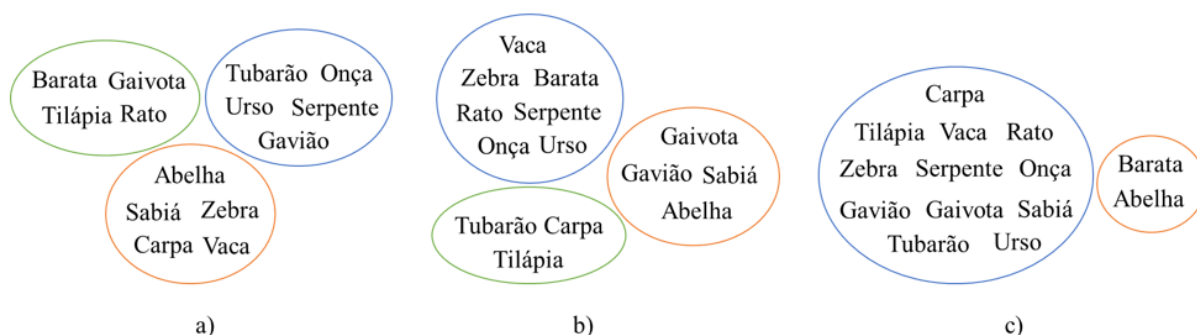
- $C_i \neq \emptyset, i = 1, \dots, k$
- $\bigcup_{i=1}^k C_i = \mathcal{X}$
- $C_i \cap C_j = \emptyset, i \neq j, i, j = 1, \dots, k.$

A análise por agrupamento, conhecida também por *clustering*, é geralmente utilizada para exploração de dados (WEBB; COPSEY, 2011). A análise é utilizada para obter agrupamentos naturais dos dados que permitam identificar algum tipo de estrutura, a fim de produzir hipóteses sobre o conjunto de dados em questão ou até mesmo ainda não vistos (base para as aplicações baseadas em aprendizado não supervisionado).

Encontrar o melhor agrupamento para um conjunto de dados não é uma tarefa simples, pois, dependendo do critério de agrupamento, diferentes partições do conjunto de dados podem ser produzidas. Considere os seguintes animais: barata, gaivota, tilápia, rato, tubarão, onça, serpente, urso, gavião, abelha, sabiá, zebra, carpa e vaca. Ao utilizar um critério de agrupamento pelo tipo de alimentação, os animais serão agrupados em três grupos (Figura 1.a): carnívoros (tubarão, onça, serpente, urso e gavião), herbívoros (abelha, sabiá, zebra, carpa e vaca) e onívoros (barata, gaivota, tilápia e rato). Se forem

classificados pelo seu habitat, os animais serão também agrupados em três tipos (Figura 1.b): terrestre (vaca, zebra, barata, rato, serpente, onça e urso), aquático (tubarão, carpa e tilápia) e aéreo (gaviota, gavião, sabiá e abelha). Se forem classificados por possuir coluna vertebral, os animais serão também agrupados em dois grupos (Figura 1.c): a barata e a abelha formariam um grupo e os demais animais o outro grupo.

Figura 1 – Grupos de animais resultantes para diferentes critérios de agrupamento: a) alimentação (carnívoros, herbívoros e onívoros); b) habitat (terrestre, aquático e aéreo); e c) coluna vertebral (vertebrado e invertebrados)



Fonte: Autor.

Como busca-se a melhor partição que permita algum tipo de conclusão relevante, um algoritmo do tipo força bruta poderia ser utilizado em um conjunto de dados, para encontrar as possíveis partições e selecionar a que melhor atende a um determinado critério. Entretanto, isto não é possível mesmo para um pequeno conjunto de dados. O número exato de maneiras de partições possíveis do conjunto \mathcal{X} em k grupos não vazios e mutuamente exclusivos é dado pelo número de Stirling de segunda espécie:

$$S(N, k) = \frac{1}{k!} \sum_{i=0}^k (-1)^{k-i} \binom{k}{i} i^N.$$

Por exemplo, para o caso onde temos 15 objetos e queremos obter 3 grupos, pode-se produzir 2.475.101 partições. Se forem 100 objetos e 5 grupos, podem ser produzidas 10^{68} partições! Isto mostra que a enumeração de todas as partições e avaliação delas pode ser computacionalmente impraticável. Além de haver tantas partições, o problema é agravado quando o número de grupos é desconhecido.

3 Etapas de um problema de agrupamento

O processo de análise por agrupamento deve ser realizado com muito cuidado, pois diferentes partições de um conjunto de dados podem levar a diferentes

interpretações. A própria definição das propriedades que um grupo de dados deve apresentar precisa ser levada em conta.

Assumindo que os elementos do conjunto \mathcal{X} são vetores de características que formam um espaço d -dimensional, as etapas a serem seguidas para realizar uma análise por agrupamento são (THEODORIDIS; KOUTROUMBAS, 2008):

1. seleção de características: deve-se selecionar as características que mais contribuem para a discriminação dos grupos. Esta etapa tem recebido muita atenção na área de bioinformática (análise de dados de *microarray* de DNA) (ANG; MIRZAL; HARON; HAMED, 2016; SAHA; ALOK; EKBAL, 2016; WEBB; COPSEY, 2011). A redução de características também simplifica a complexidade dos modelos dos dados, facilita a interpretação dos resultados e reduz o custo computacional (tanto em processamento como em armazenamento);
2. medida de proximidade: como o problema é achar uma partição natural em um conjunto de dados, é necessário definirmos alguma medida que quantifique o quão similar (ou dissimilar) são dois vetores de características;
3. critério de agrupamento: uma vez definida a medida de proximidade, deseja-se saber como o conjunto de dados deve ser particionado, de tal maneira que cada grupo contenha dados com maior similaridade. Os algoritmos de agrupamento utilizam uma função de critério, tais como a soma dos quadrados das distâncias, para encontrar um agrupamento que otimize a função de critério;
4. algoritmo de agrupamento: esta etapa refere-se à escolha do algoritmo que deve ser utilizado para descobrir a estrutura do conjunto de dados. Como não há uma definição precisa de grupo (*cluster*), existe uma grande variedade de algoritmos de agrupamento (ESTIVILL-CASTRO, 2002);
5. validação dos resultados: com os agrupamentos definidos, deve-se avaliar se o particionamento está correto. Um dos problemas recorrentes é decidir quantos grupos existem no conjunto de dados (DUDA; HART; STORK, 2001). Uma solução é utilizar alguma medida de qualidade de ajuste, que expresse o quão bem a partição encontrada expressa o conjunto de dados;
6. interpretação dos resultados: esta etapa demanda um especialista no problema em questão a fim de gerar conclusões ou hipóteses com base na partição produzida. Esta etapa também fornece subsídios para se reavaliar todo o processo de agrupamento.

Pode-se perceber que a escolha de características, a medida de proximidade, o critério de agrupamento e o algoritmo de agrupamento influencia no resultado do particionamento do conjunto de dados. Além disso, é importante notar que os algoritmos de agrupamento produzirão uma partição do conjunto de dados, mesmo que não exista agrupamento natural dos dados (WEBB; COPSEY, 2011).

4 Medidas de proximidade

Como uma partição é resultado do agrupamento de elementos similares de um conjunto de dados, é necessário definir alguma medida que permita determinar se dois elementos são similares ou dissimilares. As medidas podem ser classificadas em dois tipos: medidas de dissimilaridade e medidas de similaridade.

4.1 Medidas de dissimilaridade

Uma medida de dissimilaridade $d(x_i, x_j)$ mede a diferença entre dois objetos x_i e $x_j \in \mathcal{X}$. A medida $d(x_i, x_j)$ é uma função tal que é positivamente definida (*i.e.*, $d(x_i, x_j) \geq 0$) e é simétrica (*i.e.*, $d(x_i, x_j) = d(x_j, x_i)$). Assim, $d(x_i, x_j)$ é próximo a 0 (*i.e.*, $d(x_i, x_j) = 0$, quando $i = j$) quando os objetos são mais semelhantes (ou estão mais próximos um do outro) e torna-se maior ($d(x_i, x_j) < \infty$) conforme eles mais diferem (ou estão mais afastados).

Muitos métodos de agrupamento utilizam medidas de distâncias para determinar a dissimilaridade entre objetos (DOUGHERTY, 2013; DUDA; HART; STORK, 2001; JASKOWIAK; CAMPELLO; COSTA, 2014; THEODORIDIS; KOUTROUMBAS, 2008; WEBB; COPSEY, 2011). Além de atender às propriedades descritas para dissimilaridades, uma medida de distância deve satisfazer a propriedade de desigualdade triangular, *i.e.*,

$$d(x_i, x_j) \leq d(x_i, x_k) + d(x_k, x_j), \quad \forall x_i, x_j, x_k \in \mathcal{X}.$$

Uma classe geral de medidas de distâncias para vetores d -dimensionais é a métrica Minkowski, definida por

$$d(x_i, x_j) = \left(\sum_{k=1}^d |x_{ik} - x_{jk}|^p \right)^{1/p},$$

também conhecida por norma L_p . Dois casos especiais da métrica Minkowski muito conhecidos são as distâncias:

- Euclideana ($p = 2$): é a medida (norma L_2) mais utilizada na tarefa de agrupamento em diversas áreas do conhecimento (JASKOWIAK; CAMPELLO; COSTA, 2013; MOONSAP; LAKSANAVILAT; TASANASUWAN; KATE-NGAM *et al.*, 2019), representada por

$$d_{\text{Euclideana}}(x_i, x_j) = \sqrt{\sum_{k=1}^d |x_{ik} - x_{jk}|^2};$$

- Manhattan ($p = 1$): também conhecida por *city block*, a medida (norma L_1) é representada por

$$d_{\text{Manhattan}}(x_i, x_j) = \sum_{k=1}^d |x_{ik} - x_{jk}|;$$

- Chebyshev ($p = \infty$): também conhecida por *city block*, a medida (norma L_∞) é representada por

$$d_{\text{Chebyshev}}(x_i, x_j) = \max_{1 \leq k \leq d} |x_{ik} - x_{jk}|.$$

As normas L_1 e L_∞ podem ser vistas como uma superestimação e uma subestimação da norma L_2 , respectivamente (THEODORIDIS; KOUTROUMBAS, 2008). Apesar de sua vasta utilização, as métricas de Minkowski são aplicadas somente a vetores de valores reais ($d : \mathcal{X} \times \mathcal{X} \rightarrow \mathbb{R}$).

Como estas medidas tratam todas as variáveis igualmente, vetores de características que possuem variáveis com grandezas diferentes têm os seus valores transformados para intervalos comuns (HAN; KAMBER; PEI, 2012; THEODORIDIS; KOUTROUMBAS, 2008; WEBB; COPSEY, 2011). Variáveis com intervalos grandes de valores podem influenciar no cálculo da distância mais do que variáveis com intervalos menores, quando isto não necessariamente reflete a sua significância na distância entre os vetores. A transformação, conhecida por normalização, altera a escala das variáveis, a fim de evitar a dependência em tais tipos de variáveis. Por exemplo, quando o peso é alterado de quilogramas para gramas pode levar a resultados diferentes, pois com a unidade de medida menor aumenta o intervalo daquela variável, resultando em uma dimensão com maior peso ou importância. Ao normalizar, o intervalo das variáveis é reduzido a um intervalo comum, tipicamente $[0;1]$ ou $[-1;1]$.

4.2 Medidas de similaridade

As medidas de distância tradicionais não são apropriadas para vetores esparsos e de grandes dimensões (HAN; KAMBER; PEI, 2012). Apesar de que muitas dimensões em comum possuem valores 0, isso não quer dizer que os vetores são similares em tais dimensões. Por isso, é necessário escolher uma medida que não utilize tais dimensões. Uma medida muito utilizada é a similaridade Cosseno (JASKOWIAK; CAMPELLO; COSTA, 2014). Esta medida calcula o cosseno do ângulo entre dois vetores:

$$s_{\text{Cosseno}}(x_i, x_j) = \frac{x_i^T x_j}{\|x_i\| \|x_j\|}$$

onde $\|x_i\|$ é a norma euclidiana do vetor x_i definida como por $\sum_{k=1}^d x_{ik}^2$ e $\|x_j\|$ é a norma euclidiana do vetor x_j . Um cosseno de valor 0 quer dizer que os vetores estão em 90 graus um do outro (ortogonais) e por isso não possuem similaridade. Quanto mais próximo de 1, menor é o ângulo e maior é a similaridade entre os vetores. Esta medida é invariante a rotações, mas não a transformações lineares. A medida de dissimilaridade do cosseno pode ser obtido por

$$d_{\text{cosseno}}(x_i, x_j) = 1 - s_{\text{cosseno}}(x_i, x_j),$$

O coeficiente de correlação de Pearson estima a medida entre as diferenças dos vetores, o que permite identificar correlações lineares entre os vetores. Semelhante à medida do cosseno, o coeficiente de correlação de Pearson é expresso por

$$s_{\text{pearson}}(x_i, x_j) = \frac{\tilde{x}_i^T \tilde{x}_j}{\|\tilde{x}_i\| \|\tilde{x}_j\|},$$

onde $\tilde{x} = [x_1 - \bar{x}, \dots, x_d - \bar{x}]$ e $\bar{x} = \frac{1}{d} \sum_{k=1}^d x_k$. Esta medida é muito utilizada em análise de dados expressões de genes (JASKOWIAK; CAMPELLO; COSTA, 2013). Outras medidas baseadas em correlação podem ser encontradas em Jaskowiak, Campello e Costa (2014).

5 Algoritmos de agrupamento

Devido à falta de uma definição clara do que é um grupo de dados, diversos algoritmos de agrupamentos foram propostos. Diversos autores (DOUGHERTY, 2013; JAIN, 2010; KAUFMAN; ROUSSEEUW, 2005) dividem os algoritmos em dois grandes grupos: hierárquicos e particionais. Os métodos hierárquicos produzem uma série de possíveis partições que possuem uma relação hierárquica recursiva entre eles (as partições se diferenciam pelo agrupamento de dois grupos de um nível da hierarquia para outro). Os métodos particionais encontram todos os grupos simultaneamente, como uma partição dos dados, sem impor uma estrutura hierárquica. Estes métodos movem os dados iterativamente de um grupo para outro, começando de uma partição inicial.

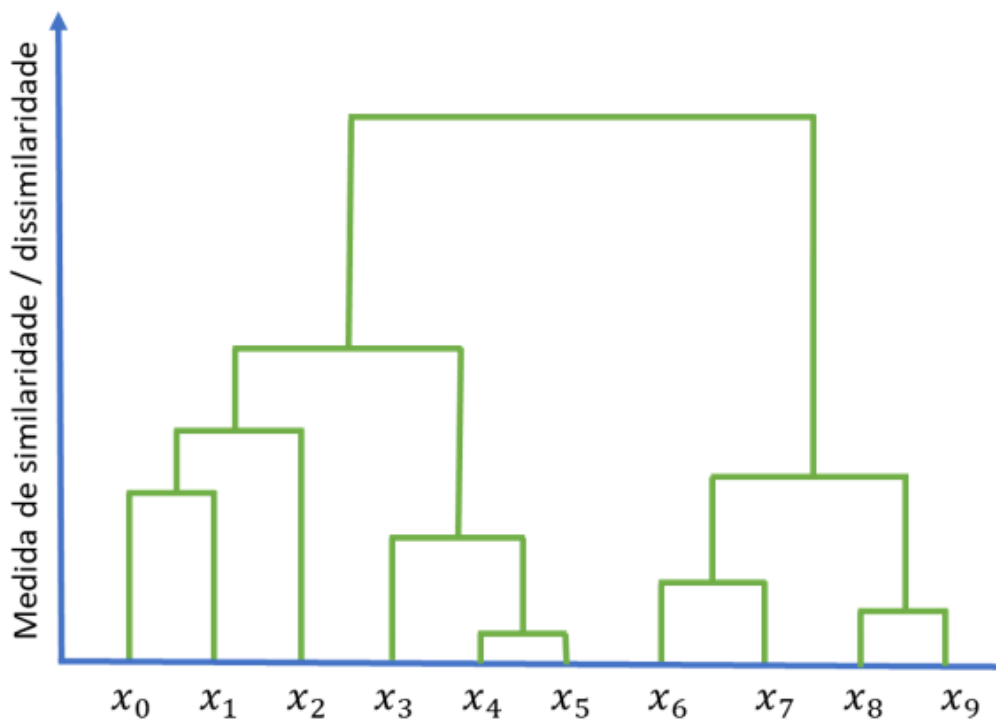
Han; Kamber e Pei (2012) expandem a categorização dos métodos de agrupamento em baseados em densidade, em grade e em modelos. Os métodos baseados em densidade buscam descobrir grupos de formas arbitrárias, ao contrário da maioria dos métodos buscam grupos com formas esféricas. A ideia destes métodos é permitir um grupo crescer enquanto a densidade (número de elementos) na “vizinhança” exceder algum limiar. Os métodos baseados em grade quantizam o espaço de características em um número finito de células que formam uma estrutura em grade. Todas as operações são executadas na estrutura em grade. Uma vantagem destes métodos é o rápido tempo

de processamento, pois não depende do tamanho do conjunto de dados, mas do tamanho da estrutura em grade. Os métodos baseados em modelo assumem que os dados vêm de uma mistura de distribuições de probabilidade, cada um representando um grupo (FRALEY; RAFTERY, 1998). Um critério de máxima verossimilhança é utilizado para unir grupos.

5.1 Métodos hierárquicos

Uma das técnicas mais frequentemente utilizadas, devido sua simplicidade conceitual (DUDA; HART; STORK, 2001), o agrupamento hierárquico produz uma hierarquia de particionamentos. Estes algoritmos produzem um novo particionamento dos dados com base no particionamento anterior. As partições produzidas em cada iteração podem ser visualizadas de forma efetiva por meio de uma árvore hierárquica ou dendrograma (do grego *dendron* = árvore, *gramma* = desenho), a qual mostra a sequência de divisões ou agrupamentos em seu nível de similaridade (ao longo do eixo vertical), conforme ilustrado na **Figura 2**. Uma grande disparidade na medida de similaridade (ou dissimilaridade) para sucessivos níveis de particionamento geralmente indica o número natural de grupos. Por exemplo, na **Figura 2**, uma partição seria $\{x_0, x_1, x_2, x_3, x_4, x_5\}$ e $\{x_6, x_7, x_8, x_9\}$. Assim, estes tipos de algoritmos não exigem que se defina um número de grupos a serem obtidos.

Figura 2 – Dendrograma representando o resultado de um agrupamento hierárquico



Fonte: Autor.

Os algoritmos de agrupamento hierárquico podem ser divididos em dois tipos: aglomerativo e divisivo. A principal diferença entre eles está na ordem na qual os grupos são formados. Algoritmos divisivos são geralmente computacionalmente ineficientes (exceto para os casos onde a maioria das variáveis são binárias) (WEBB; COPSEY, 2011).

5.1.1 Agrupamento hierárquico aglomerativo

Os métodos de agrupamento hierárquico aglomerativo também conhecidos como método *bottom-up*, partem de n grupos, cada um contendo somente um elemento, agrupando os dois grupos mais similares em um único grupo, reduzindo o número de grupos por um. O processamento é realizado até que todos os dados do conjunto estejam agrupados em um único grupo (THEODORIDIS; KOUTROUMBAS, 2008; WEBB; COPSEY, 2011).

Para um conjunto de dados \mathcal{X} , os principais passos em um agrupamento aglomerativo estão descritos no algoritmo da Figura 3. O algoritmo inicia atribuindo cada elemento do conjunto de dados a um grupo (linha 1). Note que a formação dos grupos está baseada na medida de proximidade, como mostra a linha 2 do algoritmo. No caso de uma medida de dissimilaridade, a proximidade é representada pelo menor valor possível, enquanto no caso de uma similaridade, ela é representada pelo maior valor possível. Como geralmente são utilizadas medidas de dissimilaridade em algoritmos aglomerativos, consideraremos somente tais medidas no restante do capítulo. Argumentos similares podem ser aplicados a medidas de similaridade. Em seguida, os dois grupos mais próximos são utilizados para gerar um novo grupo (linha 3).

Figura 3 – Algoritmo de agrupamento hierárquico aglomerativo

Algoritmo de Agrupamento Hierárquico Aglomerativo	
Entrada:	$\mathcal{X} = \{\mathbf{x}_1, \mathbf{x}_2, \dots, \mathbf{x}_N\}$
Saída:	hierarquia de n partições \mathcal{P}
Início	
1) Definir a partição inicial, onde \mathcal{P}_N é uma partição com n grupos $\mathcal{P}_N \leftarrow \{C_i = \{\mathbf{x}_i\}, i = 1, \dots, N\} \quad l \leftarrow N$	
Repita	
2) Encontrar os dois grupos mais próximos entre todos os grupos de P_l .	
$a, b =$	$\begin{cases} \arg \min_{i,j=1,\dots,l \text{ e } i \neq j} d(C_i, C_j) & (\text{medida de dissimilaridade}) \\ \arg \max_{i,j=1,\dots,l \text{ e } i \neq j} d(C_i, C_j) & (\text{medida de similaridade}) \end{cases}$
3) Criar o grupo $C_r = C_a \cup C_b$ produzindo a nova partição P_{L-1} .	
$\mathcal{P}_{l-1} = (\mathcal{P}_l - \{C_a, C_b\}) \cup \{C_r\}$	
$l \leftarrow l - 1$	
até que $L = 1$	
(i.e., todos os elementos estejam em um único grupo)	
fim	

Fonte: Autor.

Na aplicação do algoritmo aglomerativo, é necessário, inicialmente, estimar alguma medida de dissimilaridade, d , entre cada par de vetores, \mathbf{x}_i e \mathbf{x}_j . Como d é uma função semimétrica (DUBIEN; WARDE, 1979), o conjunto de medidas pode ser denotado por

$$D = \{d_{ij} | i < j, i = 1, 2, \dots, N - 1, j = 2, 3, \dots, N\}.$$

Assim, a matriz triangular estritamente superior D é o conjunto de todas as distâncias da partição P na hierarquia, e deve ser atualizada a cada união de dois grupos. Quando uma nova partição é gerada a partir de N grupos, o tamanho da matriz D passa a ser $(N - 1) \times (N - 1)$. Em termos práticos, as duas linhas e colunas são excluídas (as quais correspondem aos grupos mais próximos C_a e C_b) e uma nova linha e coluna serão criadas contendo as distâncias entre o mais novo grupo C_r (resultado da união dos grupos mais próximos C_a e C_b) e os demais grupos que não fizeram parte da união (isto é, $\mathcal{P}_L - \{C_a, C_b\}$). Os métodos mais simples de atualização da matriz (THEODORIDIS; KOUTROUMBAS, 2008) são:

- vizinho mais próximo (*single link*): um dos mais antigos (WEBB; COPSEY, 2011), este método define que a distância dentre dois grupos, C_i e C_j , é a distância entre os seus membros mais próximos, *i.e.*,

$$d(C_i, C_j) = \min_{m \in C_i, n \in C_j} d(m, n).$$

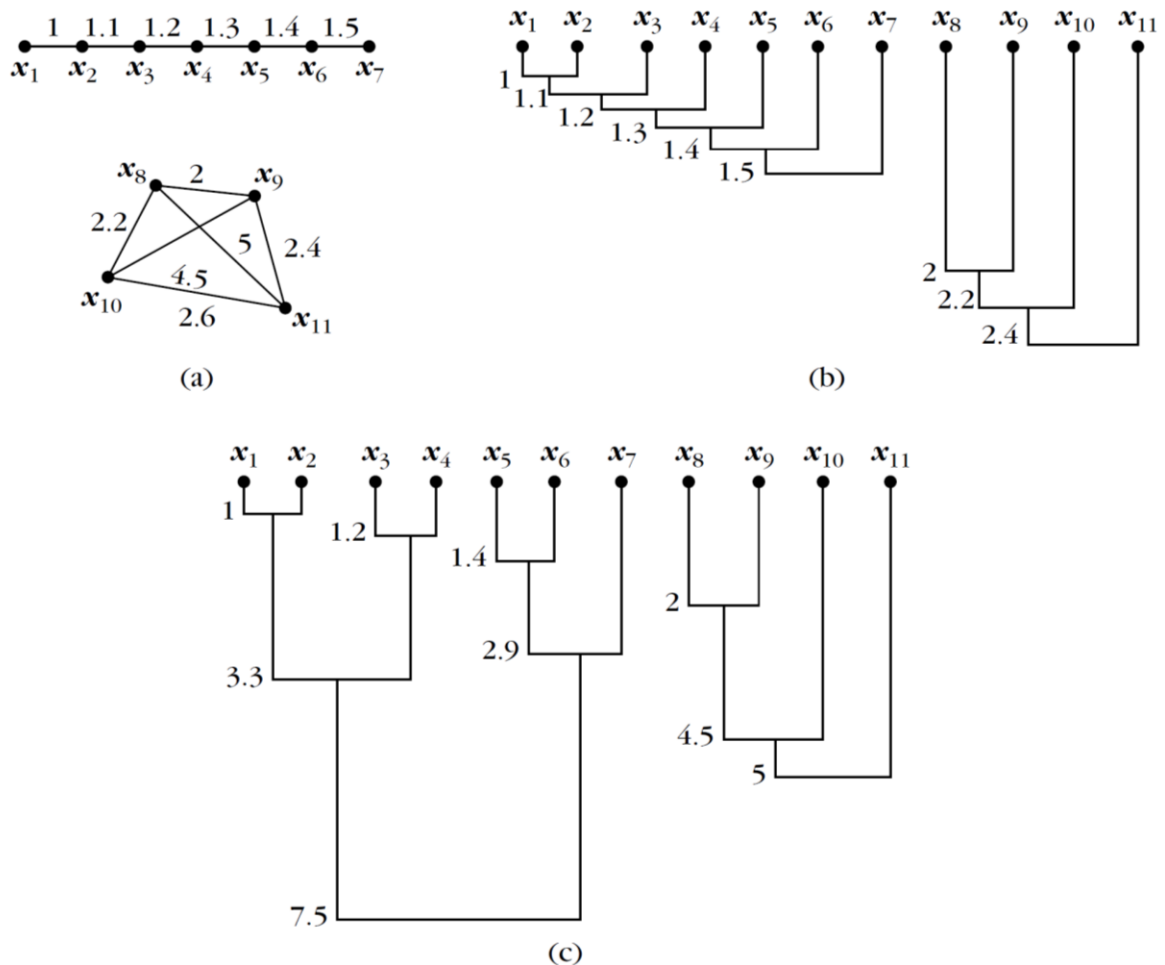
Assim, como o mínimo das distâncias é utilizado, os grupos são formados a partir das menores dissimilaridades de um dendrograma. Este método favorece a produção de grupos longos e finos, pois dois elementos devem estar no mesmo grupo se existir um encadeamento de elementos intermediários entre eles, que são mais próximos (daí o nome em inglês *single link* ou *single linkage*). (WEBB; COPSEY, 2011). Esta característica é uma desvantagem, pois dois grupos são unidos com base em dois únicos pontos (um de cada grupo), enquanto os demais podem estar muito afastados uns dos outros (DOUGHERTY, 2013; DUDA; HART; STORK, 2001). Nota-se que na **Erro! Fonte de referência não encontrada**.b, o dendrograma produzido primeiramente obtém o grupo alongado e depois recupera o segundo grupo (mais compacto) nos níveis mais altos de dissimilaridade.

- Vizinho mais distante (*complete link*): este método utiliza a distância entre os membros mais afastados entre eles, *i.e.*,

$$d(C_i, C_j) = \max_{m \in C_i, n \in C_j} d(m, n).$$

Ao contrário do método vizinho mais próximo, os grupos são formados a partir das maiores dissimilaridades de um dendrograma, favorecendo os grupos mais compactos de diâmetros aproximadamente iguais (DOUGHERTY, 2013; DUDA; HART; STORK, 2001). Por isso, ele evita a desvantagem do encadeamento do método vizinho mais próximo. Na Figura 4, o dendrograma produzido primeiramente obtém o grupo mais compacto.

Figura 4. (a) Conjunto de dados X; (b) dendrograma de dissimilaridades produzido pelo método do vizinho mais próximo; (c) dendrograma de dissimilaridades produzido pelo método do vizinho mais distante



Fonte: Theodoridis e Koutroumbas (2008).

Estes dois algoritmos não são os únicos e existem diversas opções que não são tão restritivas como o mínimo e máximo das distâncias.

Com o objetivo de suavizar o efeito dos máximos e mínimos, diversos trabalhos vêm adotando a média das distâncias, como forma de atualização ((MOONSAP; LAKSANAVILAT; TASANASUWAN; KATE-NGAM *et al.*, 2019). Estes métodos consideram que a distância entre dois grupos é igual à média da distância de qualquer membro de um grupo para qualquer membro de outro grupo. Um método das médias, chamado de *weighted pair group method average* (WPGMA), é definido por

$$d(C_i \cup C_j, C_l) = \frac{1}{2} (d(C_i, C_l) + d(C_j, C_l)).$$

E outro método leva em conta o número de membros de cada grupo (*unweighted pair group method average* – UPGMA), produzindo uma média ponderada descrita por

$$d(C_i \cup C_j, C_l) = \frac{n_i}{n_i + n_j} d(C_i, C_l) + \frac{n_j}{n_i + n_j} d(C_j, C_l).$$

5.1.2 Limitações e complexidade

Para casos, nos quais o conjunto de dados é grande, os algoritmos hierárquicos não são recomendados, devido à complexidade de tempo e de espaço. Para um conjunto de dados com N elementos, a complexidade de tempo do algoritmo aglomerativo é de $\mathcal{O}(N^3)$ e a complexidade de espaço é de $\mathcal{O}(N^2)$. A complexidade de tempo resulta da busca exaustiva da matriz de distâncias $N \times N$ em cada uma das $N - 1$ iterações do algoritmo. A complexidade de espaço resulta do armazenamento das distâncias entre os N elementos do conjunto de dados. O algoritmo pode ser aperfeiçoado mantendo as distâncias em uma estrutura de dados de forma ordenada, facilitando a busca pela distância mínima. Esta otimização reduz a complexidade do algoritmo aglomerativo para $\mathcal{O}(N^2 \log N)$.

Uma desvantagem dos métodos aglomerativos é que não existe como recuperar de uma união “errada” de dois grupos. Isto é, se um novo grupo é formado a partir de dois grupos que não deveriam unir-se “naturalmente” no nível de hierarquia, este grupo espúrio pode comprometer os demais níveis na hierarquia. (GOWER, 1967).

Como os algoritmos de atualização da matriz de distâncias envolvem medidas de mínimos e máximos, eles tendem a ser muito sensíveis a valores atípicos (DUDA; HART; STORK, 2001). Neste caso, uma opção para dirimir tais situações é utilizar a média das distâncias ou a distância entre as médias dos grupos.

5.2. Métodos particionais

Métodos particionais particionam o conjunto de dados em K grupos. O número de grupos K é definido *a priori*. Certamente, nem todo valor de K produz agrupamentos “naturais”. Por isso, é aconselhável que sejam avaliados diferentes valores de K , para selecionar o mais adequado ao problema (FUKUNAGA, 1990; HAN; KAMBER; PEI, 2012). A necessidade de especificar o número de grupos, K , para executar o algoritmo pode ser considerado uma desvantagem.

5.2.1 K-means

Um dos métodos não hierárquicos mais populares (DUDA; HART; STORK, 2001), o *K-means* particiona o conjunto de dados em K grupos tal que o erro quadrado entre o

centro de um grupo e os dados deste grupo é minimizada (JAIN, 2010; LLOYD, 1982). Este tipo de método trabalha bem com grupos compactos e isolados (JAIN; MURTY; FLYNN, 1999). Erro quadrado para uma partição \mathcal{P} (contendo K grupos) de um conjunto de dados \mathcal{X} é

$$e^2(\mathcal{X}, \mathcal{P}) = \sum_{k=1}^K \sum_{x_i \in \mathcal{X}} \|x_i - c_k\|^2$$

onde c_k é o centroide do k -ésimo grupo.

O método transforma a tarefa de agrupamento em um problema de otimização ao minimizar uma função custo J (DUDA; HART; STORK, 2001; THEODORIDIS E KOUTROUMBAS, 2008). O custo J é uma função de vetores do conjunto de dados \mathcal{X} e é parametrizado em termos de um vetor de parâmetros desconhecidos, θ . O objetivo é encontrar $\theta = [m_1^T, m_2^T, \dots, m_k^T]$, onde m_i é um vetor de d -dimensões que corresponde ao grupo C_i , o qual caracteriza melhor a partição de \mathcal{X} . No caso do K -means, o centro de um grupo é definido pela média μ_i dos elementos que o compõe, e a função custo pode ser definida como

$$J(\theta, U) = \sum_{i=1}^N \sum_{j=1}^K u_{ij} \|x_i - \mu_j\|^2$$

onde $u_{ij} \in U$, uma matriz $N \times K$ com elementos (i, j) que denotam o grau de pertencimento do vetor x_i ao j -ésimo grupo. O processo de minimização está sujeito à restrição

$$\sum_{j=1}^K u_{ij} = 1, \quad i = 1, \dots, N$$

onde

$$u_{ij} \in \{0, 1\}, \quad i = 1, \dots, N, j = 1, \dots, K,$$

$$0 < \sum_{i=1}^N u_{ij} < N, \quad j = 1, \dots, K$$

Como o K -means atribui cada elemento a um único grupo, $u_{ij} \in \{0, 1\}$, isto é, $u_{ij} = 1$ quando x_i pertence ao grupo j e $u_{ij} = 0$ caso contrário.

O algoritmo do método K -means pode ser dividido em três principais operações, conforme ilustrado na Figura 5. A primeira é a inicialização dos centroides que representam os grupos (linha 1). O conjunto de centroides pode ser estimado aleatoriamente, a partir dos próprios dados ou através de alguma heurística. É importante notar que o resultado da partição é sensível à escolha dos centroides iniciais (JAIN; MURTY; FLYNN, 1999). Uma vez definidos os centroides, é necessário estimar a

qual grupo cada \mathbf{x}_i pertence (linha 2), onde $u_{ij} = 1$ se \mathbf{x}_i é mais próximo do centroide do grupo j e zero para os casos contrários (consequentemente, minimizando $J(\theta, U)$). A distância euclidiana é utilizada como medida de proximidade, minimizando a variabilidade dentro do centroide e maximizando a variabilidade entre os centroides (DUDA; HART; STORK, 2001). A última operação realiza a atualização dos centroides (linha 3). O algoritmo termina quando um critério de convergência é encontrado, como por exemplo, nenhuma (ou mínima) relocação de vetores entre os grupos ou mínima redução no erro quadrado (JAIN; MURTY; FLYNN, 1999).

Figura 5 – Algoritmo de Agrupamento K-means

Algoritmo de Agrupamento K-means

Entrada: $\mathcal{X} = \{\mathbf{x}_1, \mathbf{x}_2, \dots, \mathbf{x}_N\}$, K

Saída: Partição \mathcal{P} com K grupos

Início

1) Selecione $\mu_j(0)$ como estimativa inicial de $\mu_j, j = 1, \dots, K$

$t = 1$

Repita

2) Determinar a qual grupo pertence cada elemento \mathbf{x}_i

Para $i = 1, \dots, N$

Para $j = 1, \dots, K$

$$u_{ij}(t) = \begin{cases} 1, & \text{se } d(\mathbf{x}_i, \mu_j(t)) = \min_{m=1, \dots, k} d(\mathbf{x}_i, \mu_m(t)) \\ 0, & \text{caso contrário} \end{cases}$$

fim para

fim para

$t = t + 1$

3) Atualização dos centroides

Para $j = 1, \dots, k$

$$\mu_j = \frac{\sum_{i=1}^N u_{ij}(t-1) \mathbf{x}_i}{\sum_{i=1}^N u_{ij}(t-1)}$$

fim para

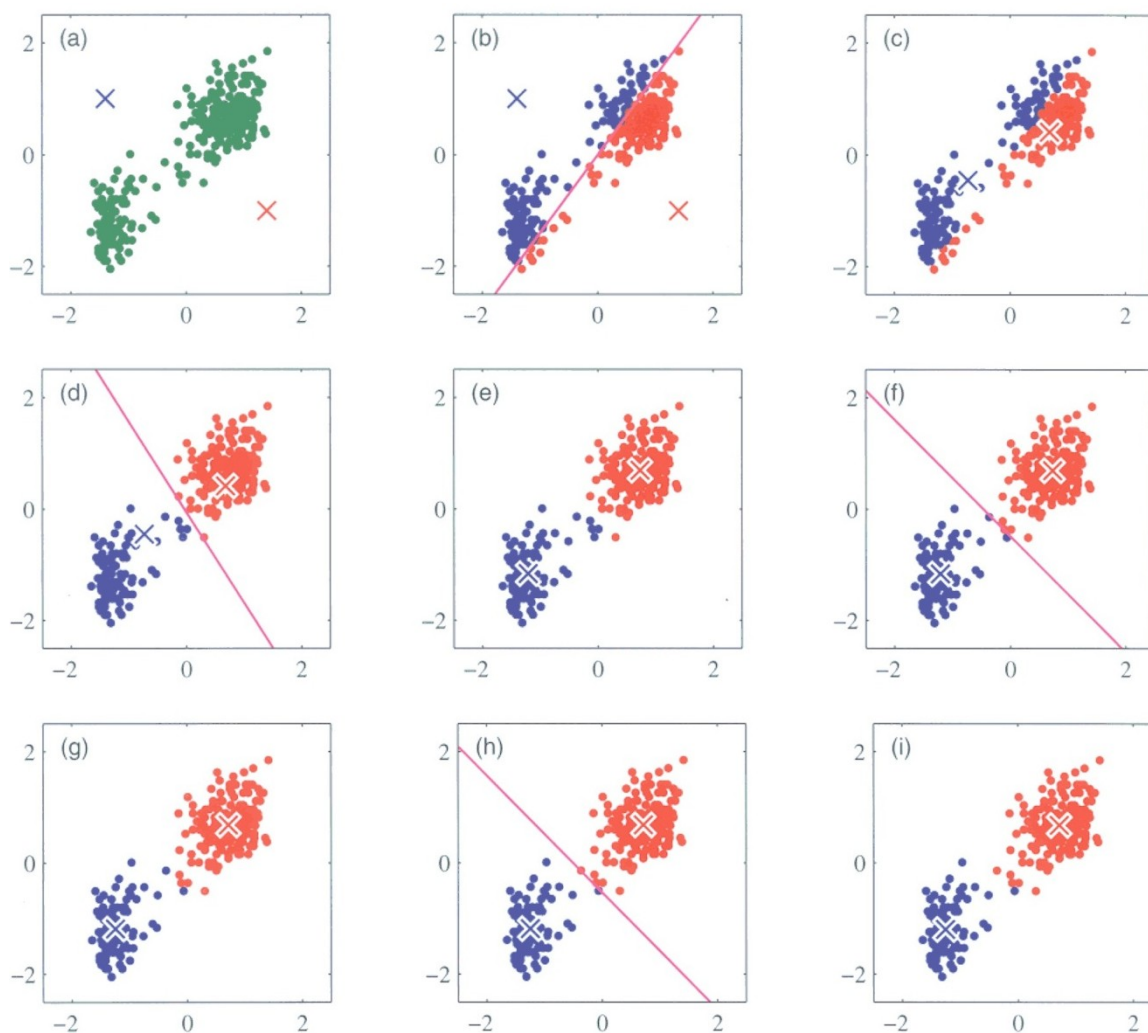
até que algum critério é satisfeito

fim

Fonte: Elaboração do autor.

A Figura 6 mostra um exemplo de agrupamento utilizando o algoritmo *K-means*. A distribuição inicial dos dados é apresentada na Figura 6.a, onde o “x” representa os centroides e os pontos representam os dados. A Figura 6 (b) mostra o resultado da operação de determinar a qual grupo cada elemento pertence (a linha representa a divisão dos dados de acordo com o seu centroide mais próximo). A Figura 6 (c) apresenta o resultado da atualização dos centroides (linha 3 do algoritmo da Figura 5). Os demais gráficos da Figura 6 (d - i) mostram as iterações sucessivas das linhas 2 e 3 do algoritmo, até convergir.

Figura 6 – Ilustração do processo de agrupamento de um conjunto de dados utilizando *K-means*, onde $K = 2$



Fonte: <https://www.mghassany.com/MLcourse/clustering.html#k-means>

A maior vantagem do algoritmo *K-means* é a sua simplicidade computacional. A complexidade computacional do algoritmo é $O(NKT)$, onde T é o número de iterações

(geralmente muito menor do que o número de amostras) necessários para a sua convergência. (DUDA; HART; STORK, 2001). Como $K \ll N$ e $T \ll N$, *K-means* é essencialmente linear no número de elementos do conjunto de dados e, por isso, torna-o um excelente candidato para o processamento de grandes bases de dados.

Um grande problema com este método é que é sensível a seleção da partição inicial. Se a seleção inicial não é realizada apropriadamente, o método pode convergir para um mínimo local da função-critério. Uma maneira de superar este problema é executar o algoritmo várias vezes com diferentes partições iniciais e selecionar a partição com o menor erro quadrado. (JAIN, 2010; LIKAS; VLASSIS; J. VERBEEK, 2003). Webb e Copsey (2011) apresentam diversas técnicas de inicialização.

K-means é sensível a ruído e valores atípicos. Os valores atípicos influenciam na estimação dos centroides e como consequência na partição final. Grupos com poucos elementos podem ser formados por valores atípicos. Uma maneira de lidar com este problema é eliminar grupos com poucos elementos (THEODORIDIS; KOUTROUMBAS, 2008). Esta desvantagem motivou a criação dos algoritmos chamados de *K-medoides*, onde cada grupo é representado por um dos seus vetores (KAUFMAN; ROUSSEEUW, 2005). Estes algoritmos também possuem a vantagem de trabalhar com vetores binários, pois cada grupo é representado por um dos seus vetores.

K-means e suas variações possuem uma variedade de limitações para certos tipos de grupos (DOUGHERTY, 2013b). O método enfrenta dificuldades em encontrar partições de conjunto de dados, quando os grupos possuem formas não esféricas ou possuem grandes diferenças de tamanhos ou densidades. Estas dificuldades podem ser superadas superestimando o número de grupos e posteriormente unir alguns dos grupos.

5.2.2 *Fuzzy c-means*

A suposição de que cada elemento pertence a um único grupo, como *K-means*, pode ser um tanto restritiva, especialmente para grupos que não são compactos. A flexibilização desta condição é definida no algoritmo *Fuzzy c-means* (FCM), em que um elemento pode pertencer a mais de um grupo ao mesmo tempo, com diferentes graus de pertinência (BEZDEK, 1981). Esta característica é útil quando as fronteiras entre os grupos são bem separadas ou ambíguas (RUI; WUNSCH, 2005). O algoritmo FCM busca minimizar a seguinte função custo;

$$J(\theta, U) = \sum_{i=1}^N \sum_{j=1}^K (u_{ij})^m \|x_i - c_j\|^2$$

onde $m \in [1, \infty]$ controla o nível de fuzzificação,³ que indica quão nebuloso os conjuntos serão, e c_j é o centro do grupo j . O algoritmo FCM é apresentado na Figura 7, em que as principais mudanças em relação ao algoritmo *K-means* estão na função $u_{ij}(t)$ e atualização do centro do grupo c_j . Note-se que, quando m se aproxima a 1, o algoritmo FCM produz valores de pertencimento menos *fuzzy* (*i.e.*, tende ao *K-means*).

Figura 7 – Algoritmo de agrupamento *Fuzzy c-means*

Algoritmo de Agrupamento *Fuzzy c-means*

Entrada: $\mathcal{X} = \{\mathbf{x}_1, \mathbf{x}_2, \dots, \mathbf{x}_N\}$, K, m

Saída: Partição \mathcal{P} com K grupos

Início

1) Seleccione $c_j(0)$ como estimativa inicial de $c_j, j = 1, \dots, K$

$t = 1$

Repita

2) Determinar a qual grupo pertence cada elemento \mathbf{x}_i

Para $i = 1, \dots, N$

Para $j = 1, \dots, K$

$$u_{ij}(t) = \frac{1}{\sum_{l=1}^K \left(\frac{d(\mathbf{x}_i, c_j(t))}{d(\mathbf{x}_i, c_l(t))} \right)^{\frac{1}{m-1}}}$$

fim para

fim para

$t = t + 1$

3) Atualização dos centroides

Para $j = 1, \dots, k$

$$c_j(t) = \frac{\sum_{i=1}^N (u_{ij}(t-1))^m \mathbf{x}_i}{\sum_{i=1}^N (u_{ij}(t-1))^m}$$

fim para

até que algum critério é satisfeito

fim

Fonte: Autor.

³ Tradução livre de *fuzzyfication*.

Um critério de parada inclui terminar o algoritmo quando ocorrer uma mudança relativamente pequena, isto é, $\|c_j(t) - c_j(t-1)\| < \varepsilon$, onde ε é um limiar definido pelo usuário (RUI; WUNSCH, 2005; WEBB; COPSEY, 2011). Outros critérios de parada são baseados nos valores da função de pertencimento u_{ij} ou da função-custo $J(\theta, U)$.

O algoritmo minimiza a variância intragrupo (grupos mais compactos) e é menos suscetível a valores atípicos. Apesar disso, apresenta os mesmos problemas que o *K-means*, e o mínimo da função-custo é mais provável ser um mínimo local do que um mínimo global, e os resultados dependem da escolha inicial dos valores da função de pertencimento u_{ij} (DOUGHERTY, 2013).

A complexidade computacional do algoritmo *Fuzzy c-means* é $\mathcal{O}(NK^2T)$. Quando o conjunto de dados é grande, a atualização dos valores da função de pertencimento u_{ij} e do centro de cada grupo impõe um custo computacional muito alto. HAVENS; BEZDEK; LECKIE; HALL *et al.* (2012) analisam a eficácia de três implementações de técnicas que visam a estender FCM para grandes bases de dados e produz um conjunto de recomendações para a sua utilização neste tipo de aplicação.

Outro algoritmo que também flexibiliza as condições de partição de um conjunto de dados \mathcal{X} em grupos é baseado em modelos de probabilidade. (FRALEY; RAFTERY, 1998). Modelos de misturas finitas têm sido propostos para o contexto de agrupamento. (ANDREOPOULOS; AN; WANG; SCHROEDER, 2009). A ideia é que cada componente de distribuição de probabilidade corresponda a um grupo. A adição de componentes para lidar com valores atípicos pode ser utilizada em tais algoritmos.

6 Validação do grupo

O procedimento de avaliar os resultados de um algoritmo de agrupamento é chamado de validação de grupos. Esta etapa é importante, pois um algoritmo de agrupamento encontrará uma partição, mesmo que não existam grupos “naturais” no conjunto de dados (WEBB; COPSEY, 2011). Assim, o procedimento de validação é cheio de dificuldades e raramente simples.

A partição obtida pode ser analisada visualmente ou de forma comparativa. No caso de dados bidimensionais, pode-se visualizar o resultado. No caso de dados com maiores dimensões, pode-se visualizar a partição de uma representação dos dados em uma dimensão menor, utilizando métodos de projeção linear e não linear de dados. Uma outra abordagem é aplicar diferentes métodos de agrupamento e comparar os resultados, para ver se a estrutura obtida é o efeito de um método em particular.

O problema é que uma estrutura de um conjunto de dados encontrada de forma incorreta pode levar a incorretas conclusões sobre os dados. Por isso, é importante

definir métodos adequados para realizar uma avaliação quantitativa do resultado do particionamento dos dados. Note que tais métodos são apenas ferramentas nas mãos do pesquisador, para a avaliação dos resultados (THEODORIDIS; KOUTROUMBAS, 2008).

Existem três tipos de abordagens ou direções para a validação de grupos (HAN; KAMBER; PEI, 2012; THEODORIDIS; KOUTROUMBAS, 2008; WEBB; COPSEY, 2011):

- externos: empregam critérios que não são inerentes ao conjunto de dados. Assumem que a verdade sobre o particionamento dos dados está disponível e pode ser utilizado para comparar com o estimado. O problema é que nem todo o problema possui a verdade (por exemplo, área da biologia). No contexto da Biologia, outros tipos de fonte de informações biológicas podem ser utilizados (por exemplo, ontologias) (BRUNO; FIORI, 2014).
- internos: empregam distâncias entre os elementos de um grupo, permitindo avaliar os resultados em termos de coesão intragrupo e separação intergrupos (ou a combinação de ambos). Não utiliza nenhuma informação externa, além dos próprios dados;
- Relativos: utilizam outros agrupamentos (normalmente obtidos por diferentes algoritmos ou trabalhos anteriores) para validar os resultados.

6.1 Medidas internas

O coeficiente silhuete mede se o elemento pertence mais ao seu grupo (coesão) do que qualquer outro grupo (separação). Para um elemento x , pode-se medir o quão bem x é atribuído a seu grupo (quanto menor o valor, melhor a atribuição), calculando a distância média entre x e qualquer outro elemento do grupo utilizando

$$a(x) = \frac{1}{|C_i| - 1} \sum_{y \in C_i, x \neq y} d(x, y).$$

Similarmente, pode-se medir a separação entre o elemento x e os demais grupos (quanto maior o valor, mais separado é dos demais grupos), estimando a distância média mínima entre x e qualquer outro grupo ao qual ele não pertence, utilizando

$$b(x) = \min_{C_j: 1 \leq j \leq K, j \neq i} \frac{1}{|C_j|} \sum_{y \in C_j} d(x, y).$$

O coeficiente silhuete de x pode ser então definido como

$$s(x) = \frac{b(x) - a(x)}{\max\{b(x), a(x)\}}.$$

O valor do coeficiente $s(\mathbf{x})$ está entre -1 e 1. Quando o valor se aproxima de 1, o grupo de \mathbf{x} é compacto e \mathbf{x} está longe dos demais grupos (o que é desejável). Caso contrário, \mathbf{x} é mais próximo de elementos de outros grupos do que elementos do seu grupo (o que deve ser evitado). No caso de avaliar o grupo ou a partição inteira, pode-se utilizar o valor médio dos coeficientes dos elementos em um grupo ou em todos os grupos, respectivamente (THEODORIDIS; KOUTROUMBAS, 2008). O coeficiente silhuete é a medida interna mais frequentemente utilizada para avaliar resultados de agrupamentos de expressões gênicas (BRUNO; FIORI, 2014).

Outra medida que avalia a coesão e a separação é o índice Dunn (BRUNO; FIORI, 2014). O objetivo é maximizar as distâncias intergrupos, enquanto minimiza as distâncias intragrupos. O índice é definido como a razão entre a menor distância entre os elementos de diferentes grupos e a maior distância intergrupo, conforme

$$Dunn = \min_{1 \leq i \leq K} \left\{ \min_{1 \leq j \leq K, j \neq i} \left\{ \frac{d(C_i, C_j)}{\max_{C_m \in \mathcal{P}} diam(C_m)} \right\} \right\}$$

onde $diam(C_m)$ é a distância máxima entre elementos no grupo C_m (distância intragrupos). Esta medida pode assumir valores de 0 a ∞ , em que valores altos correspondem a bons grupos. Diversas outras medidas podem ser consultadas em Bruno e Fiori (2014) e Theodoridis e Koutroumbas (2008).

7 Considerações finais

Apesar de todos os algoritmos de agrupamento existentes, não existe um método que é universalmente aplicável em descobrir estruturas em conjunto de dados multidimensionais (JAIN; MURTY; FLYNN, 1999). As diferentes suposições sobre a forma dos agrupamentos, a sua distribuição no espaço de características e critérios de agrupamento utilizados podem afetar na produção de uma partição que permita a extração de conhecimento. O profundo conhecimento das limitações e suposições de cada método deve ser primordial no trabalho com tais ferramentas.

É importante que antes de selecionar o método de agrupamento, deva-se buscar a melhor representação dos dados. Além de facilitar o descobrimento das estruturas que definem o espaço de características, e consequentemente dos dados, deve permitir a aplicação do maior número de algoritmos e técnicas disponíveis. Nem todos os algoritmos trabalham com valores categóricos (por exemplo, *K-means*) ou trabalhar com dados de tipos misturados (binomiais e numéricos). A alta dimensionalidade dos vetores de características ou a inexistência de valores em algumas dimensões podem produzir resultados que levarão a conclusões errôneas sobre o problema em questão.

Além disso, a alta dimensionalidade aumenta a complexidade computacional e de espaço dos algoritmos de agrupamento.

A crescente capacidade de processamento disponibilizada tem permitido o processamento de problemas que não era possível em tempos passados. Em consonância com esta evolução, a quantidade de dados também tem crescido de forma exponencial. Por exemplo, a extração de informações de imagens, de documentos e de genes (BRUNO; FIORI, 2014; JAIN; MURTY; FLYNN, 1999) são alguns exemplos de aplicações, em que uma das características é a imensa quantidade de dados. Neste caso, o *K-means* parece ser um excelente candidato devido à sua baixa complexidade. Entretanto, como a grande quantidade de elementos obriga o sistema a utilizar uma memória secundária para armazenamento (e.g., disco rígido), o algoritmo deve realizar diversos acessos aos elementos aumentando o tempo de processamento. Os agrupamentos hierárquicos precisam trabalhar com uma matriz de distâncias de $N \times N$, proibitivo para representação em memória. Novos algoritmos têm sido desenvolvidos para lidar com grandes bases de dados (HAN; KAMBER; PEI, 2012; HAVENS; BEZDEK; LECKIE; HALL *et al.*, 2012; JAIN; MURTY; FLYNN, 1999).

Enfim, existem diversos desafios a serem enfrentados no problema de agrupamento de dados. Devido à sua complexidade desde a seleção das características, passando pelo método de agrupamento até a validação dos resultados, muita atenção tem sido dada a este problema pelos resultados que podem ser alcançados. Mas a disponibilidade de bases de dados e de ferramentas, que permitem realizar os mais diferentes experimentos e avaliar os resultados, garante que tais desafios sejam sobrepujados. E por todas estas razões que existe muito material produzido e em produção sobre este tema.

Referências

ANDREOPOULOS, B.; AN, A.; WANG, X.; SCHROEDER, M. A roadmap of clustering algorithms: finding a match for a biomedical application. **Briefings in Bioinformatics**, 10, n. 3, p. 297-314, 2009.

ANG, J. C.; MIRZAL, A.; HARON, H.; HAMED, H. N. A. Supervised, Unsupervised, and Semi-Supervised Feature Selection: A Review on Gene Selection. **IEEE/ACM Transactions on Computational Biology and Bioinformatics**, 13, n. 5, p. 971-989, 2016.

BEZDEK, J. C. **Pattern recognition with fuzzy objective function algorithms**. New York: Plenum press, 1981. 0306406713.

BRUNO, G.; FIORI, A. Spread of evaluation measures for microarray clustering. *In*: ELLOUMI, M. e ZOMAYA, A. Y. (Ed.). **Biological Knowledge Discovery Handbook: Preprocessing, Mining, and Postprocessing of Biological Data**. Hoboken, New Jersey: John Wiley & Sons, Inc, 2014. p. 569-590. (Bioinformatics: Computational Techniques and Engineering).

DOUGHERTY, G. Unsupervised Learning. *In*: **Pattern Recognition and Classification: An Introduction**. New York: Springer Publishing Company, Incorporated, 2013. cap. 8, p. 143-155.

DUBIEN, J. L.; WARDE, W. D. A mathematical comparison of the members of an infinite family of agglomerative clustering algorithms. **The Canadian Journal of Statistics / La Revue Canadienne de Statistique**, 7, n. 1, p. 29-38, 1979.

- DUDA, R. O.; HART, P. E.; STORK, D. G. **Pattern classification**. 2nd ed. New York ; Chichester: Wiley, 2001. 0471056693.
- ESTIVILL-CASTRO, V. Why so many clustering algorithms: a position paper. **SIGKDD Explor. Newsl.**, 4, n. 1, p. 65-75, 2002.
- FRALEY, C.; RAFTERY, A. E. How Many Clusters? Which Clustering Method? Answers Via Model-Based Cluster Analysis. **The Computer Journal**, 41, n. 8, p. 578-588, / 1998.
- FUKUNAGA, K. **Introduction to statistical pattern recognition**. Academic Press, 2nd ed., 1990.
- GOWER, J. C. A Comparison of some methods of cluster analysis. **Biometrics**, 23, n. 4, p. 623-637, 1967.
- HAN, J.; KAMBER, M.; PEI, J. **Data Mining: Concepts and Techniques**. Waltham, Mass.: Morgan Kaufmann Publishers, 2012. Text.
- HAVENS, T. C.; BEZDEK, J. C.; LECKIE, C.; HALL, L. O. *et al.* Fuzzy c-Means Algorithms for Very Large Data. **IEEE Transactions on Fuzzy Systems**, 20, n. 6, p. 1130-1146, 2012.
- JAIN, A. K. Data clustering: 50 years beyond K-means. **Pattern Recognition Letters**, 31, n. 8, p. 651-666, 2010.
- JAIN, A. K.; MURTY, M. N.; FLYNN, P. J. Data clustering: a review. **ACM Computing Surveys**, 31, n. 3, p. 264-323, 1999.
- JASKOWIAK, P. A.; CAMPELLO, R. J.; COSTA, I. G. On the selection of appropriate distances for gene expression data clustering. **BMC Bioinformatics**, 15, n. 2, p. S2, January 24 2014. journal article.
- JASKOWIAK, P. A.; CAMPELLO, R. J. G. B.; COSTA, I. G. Proximity measures for clustering gene expression microarray data: a validation methodology and a comparative analysis. **IEEE/ACM Transactions on Computational Biology and Bioinformatics**, 10, n. 4, p. 845-857, 2013.
- KAUFMAN, L.; ROUSSEEUW, P. J. **Finding groups in data: an introduction to cluster analysis**. Wiley, 2005. 9780471735786.
- KEMP, C.; TENENBAUM, J. B. The discovery of structural form. **Proceedings of the National Academy of Sciences**, 105, n. 31, p. 10687-10692, 2008.
- LIKAS, A.; VLASSIS, N.; J. VERBEEK, J. The global k-means clustering algorithm. **Pattern Recognition**, 36, n. 2, p. 451-461, 2003/02/01/ 2003.
- LLOYD, S. Least squares quantization in PCM. **IEEE Transactions on Information Theory**, 28, n. 2, p. 129-137, 1982.
- MITCHELL, T. M. **Machine learning**. McGraw-Hill, 1997. 9780071154673.
- MOONSAP, P.; LAKSANAVILAT, N.; TASANASUWAN, P.; KATE-NGAM, S. Assessment of genetic variation of 15 Thai elite rice cultivars using InDel markers. **Crop Breeding and Applied Biotechnology**, v. 19, p. 15-21, 2019.
- PI, Y.; LU, J.; LIU, M.; LIAO, W. **Theory of cognitive pattern recognition**. INTECH Open Access Publisher, 2008. 9789537619244.
- RUI, X.; WUNSCH, D. Survey of clustering algorithms. **IEEE Transactions on Neural Networks**, 16, n. 3, p. 645-678, 2005.
- SAHA, S.; ALOK, A. K.; EKBAL, A. Use of semisupervised clustering and feature-selection techniques for identification of co-expressed genes. **IEEE Journal of Biomedical and Health Informatics**, 20, n. 4, p. 1171-1177, 2016.
- THEODORIDIS, S.; KOUTROUMBAS, K. **Pattern recognition**. Elsevier Science, 2008. 9780080949123.
- WEBB, A. R.; COPSEY, K. D. **Statistical pattern recognition**. Wiley, 2011. 9781119961406.

INTRODUÇÃO ÀS MÁQUINAS DE VETORES DE SUPORTE

Lucas Picinini Dutra,¹ Iago dos Passos,² André Luis Martinotto³

A técnica de aprendizado de Máquinas de Vetores de Suporte (SVMs, do inglês *Support Vector Machines*) tem recebido crescente atenção da comunidade de aprendizado de máquina devido, principalmente, aos resultados obtidos com o uso dessa técnica em problemas de classificação. De fato, os resultados da aplicação desta técnica são comparáveis aos resultados obtidos por outros algoritmos de aprendizado, como, por exemplo, as Redes Neurais Artificiais (RNAs) (LORENA; CARVALHO, 2007).

As SVMs baseiam-se na teoria de aprendizado estatístico desenvolvida por Vladimir Vapnik, a partir de estudos iniciais realizados em conjunto com Alexey Chervonenkis no ano de 1971 (FACELI *et al.*, 2011). Essas foram desenvolvidas para resolução de problemas de classificação binária (MEYER; WIEN, 2015), onde uma amostra deve ser associada a uma de duas possíveis classes. Por exemplo, dado um conjunto de pontos (x, y) , sendo x um vetor de características de uma amostra e y a classe à qual a amostra pertence (1 ou -1), uma regra é definida para atribuir corretamente a classe y à outras amostras, de classes desconhecidas (FRADKIN; MUCHNIK, 2006).

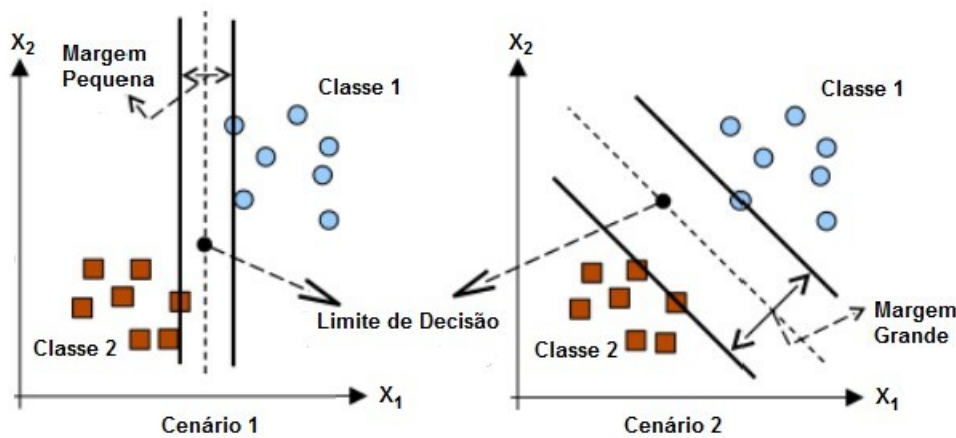
A criação da regra é realizada através da definição de um hiperplano para a separação das classes. Este hiperplano é definido com a finalidade de obter a maior margem entre os pontos mais próximos das duas classes (LORENA; CARVALHO, 2007). Na Figura 1 tem-se 2 hiperplanos que dividem um conjunto de amostras. No Cenário 1, o limite de decisão não é o melhor possível, visto que a margem obtida entre as duas classes não é a maior dentre as possibilidades existentes. Já no Cenário 2, pode se observar a definição da maior margem viável para separação das classes, sendo assim, tem-se um hiperplano mais adequado para a classificação das amostras. A definição do hiperplano do segundo cenário é o objetivo que a técnica das SVMs procura atingir.

¹ Universidade de Caxias do Sul. *E-mail*: lpdutra@ucs.br / ld.lucasdutra@gmail.com

² Universidade de Caxias do Sul. *E-mail*: ipassos@ucs.br / iago.dpassos@gmail.com

³ Universidade de Caxias do Sul. *E-mail*: almartin@ucs.br

Figura 1 – Hiperplano de divisão



Fonte: Elaboração dos autores.

A técnica das SVMs é útil para a análise e classificação de amostras, principalmente, para casos onde não existe uma regularidade nos dados, ou seja, quando os dados não são uniformemente distribuídos ou ainda quando não se tem uma distribuição conhecida. Apesar da sólida fundamentação matemática, a sua aplicação não requer alto conhecimento técnico, visto que esses conceitos encontram-se implementados em diversas ferramentas.

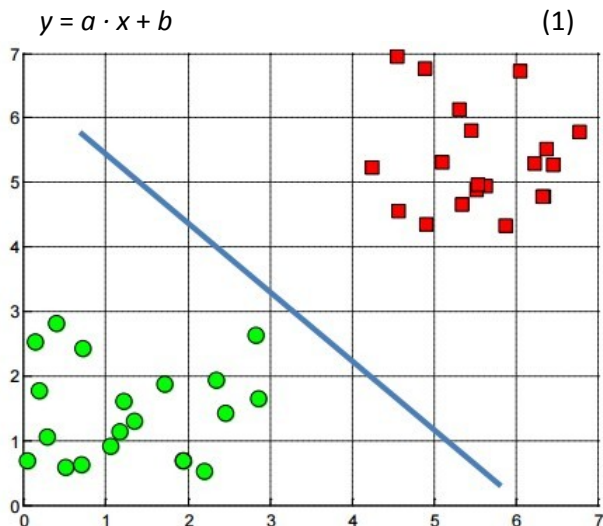
Neste contexto, o presente capítulo descreve primeiramente a fundamentação matemática das SVMs, partindo-se do conceito de como uma reta pode ser usada para a divisão de amostras em um espaço de duas dimensões, assim como um hiperplano pode ser utilizado para dividir as amostras em espaços multidimensionais. Desta maneira, apresenta-se inicialmente a equivalência da equação de reta com a equação de hiperplanos na qual as SVMs baseiam-se para a construção do modelo de separação de classes. Em seguida, são apresentadas as restrições que devem ser respeitadas a fim de definir um modelo adequado para a separação das classes. Além disso, apresenta-se o cálculo da margem de separação das classes, para, em seguida, descrever como esta margem pode ser maximizada. Por fim, são apresentados os principais *kernels* disponíveis na literatura, além de ferramentas computacionais que implementam as SVMs.

1 Equivalência de uma reta e um hiperplano

Tendo como base um espaço de duas dimensões, pode-se visualizar na Figura 2, que a superfície de decisão entre um conjunto de amostras será uma reta. A equação da

reta é apresentada na Equação 1, onde a é o coeficiente angular⁴ e b é o coeficiente linear.⁵

Figura 2 – Superfície de decisão em espaço de 2 dimensões

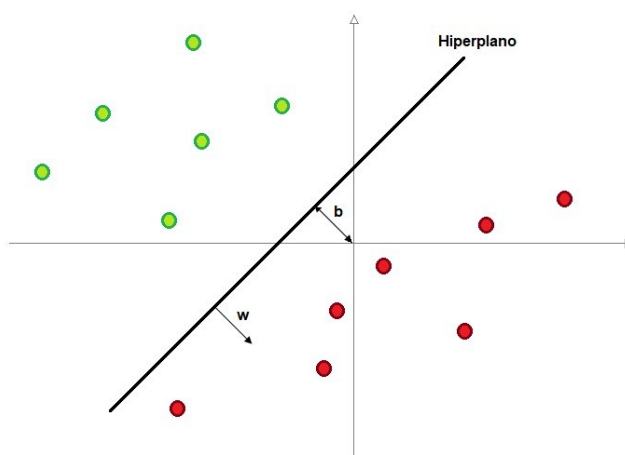


Fonte: Elaboração dos autores.

Porém, tendo em vista que as SVMs possuem como objetivo separar amostras em espaços multidimensionais, torna-se matematicamente conveniente utilizar a Equação 2 para representação de um hiperplano para a separação das classes. Nesta equação, \vec{w} é um vetor perpendicular a esse hiperplano e b um valor escalar que representa a distância entre o hiperplano e a origem (Figura 3).

$$\vec{w} \cdot \vec{x} + b = 0 \quad (2)$$

Figura 3 – Representação de \vec{w} e b



Fonte: Elaboração dos autores.

⁴ O coeficiente angular denota a inclinação da reta em relação ao eixo das abscissas.

⁵ O coeficiente linear denota o valor numérico em que a reta cruza o eixo das ordenadas.

A distância b é determinada através da Equação 3, onde $\|\mathbf{w}\|$ é a norma⁶ do vetor \vec{w} .

$$\frac{|b|}{\|\mathbf{w}\|} \quad (3)$$

Para demonstrar a relação entre a equação do hiperplano (Equação 2) e a equação da reta (Equação 1), podemos considerar uma reta que passa pela origem, ou seja, que possui um coeficiente linear b igual a zero (Equação 4).

$$y = a \cdot x + 0 \quad (4)$$

Subtraindo-se $a \cdot x$ de ambos os lados da Equação 4, obtemos a Equação 5.

$$\begin{aligned} y - a \cdot x &= a \cdot x + 0 - a \cdot x \Rightarrow \\ y - a \cdot x &= 0 \end{aligned} \quad (5)$$

A Equação 5 pode ser representada em um hiperplano através do produto escalar entre os vetores \vec{w} e \vec{x} , conforme pode ser observado na Equação 6.

$$\begin{aligned} \vec{w} \cdot \vec{x} &= \begin{bmatrix} 1 \\ -a \end{bmatrix} \cdot \begin{bmatrix} y \\ x \end{bmatrix} \Rightarrow \\ \vec{w} \cdot \vec{x} &= 1 \cdot y + (-a) \cdot x \Rightarrow \\ \vec{w} \cdot \vec{x} &= y - a \cdot x \end{aligned} \quad (6)$$

Adicionando a constante b nos dois lados da Equação 6 obtém-se a Equação 7. Uma vez que no lado direito da igualdade temos a equação da reta, podemos concluir que a equação da reta e a equação do hiperplano são equivalentes.

$$\vec{w} \cdot \vec{x} + b = y - a \cdot x + b \quad (7)$$

Dado que a equação da reta e dos hiperplanos são equivalentes, torna-se mais conveniente a utilização de vetores e hiperplanos para a representação e a resolução de problemas de classificação em espaços multidimensionais (KOWALCZYK, 2017).

2 Restrições para definição do hiperplano

Partindo de um hiperplano H_0 , definido pela Equação 2, e que divide o conjunto de classes no espaço, podemos selecionar outros 2 hiperplanos H_1 e H_2 que também

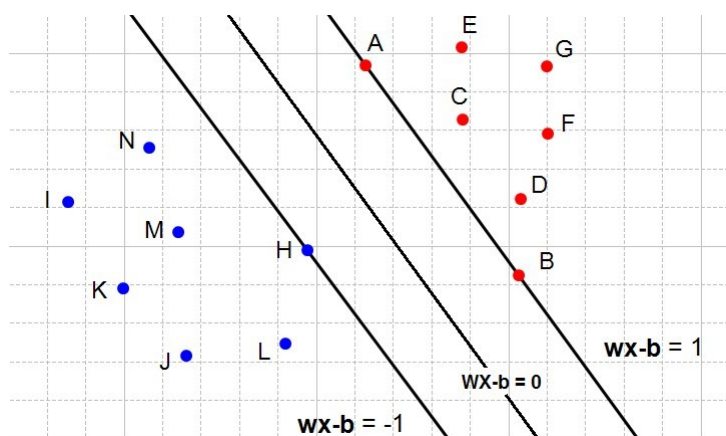
⁶ A norma de um vetor é definida pela raiz quadrada do produto escalar do vetor em relação a ele mesmo.

separam as classes e que são descritos pelas Equações 8 e 9. O hiperplano H_0 é equidistante de H_1 e H_2 e, portanto, está localizado entre eles (Figura 4).

$$\vec{w} \cdot \vec{x} + b = 1 \quad (8)$$

$$\vec{w} \cdot \vec{x} + b = -1 \quad (9)$$

Figura 4 – Hiperplanos de divisão



Fonte: Elaboração dos autores.

A seleção dos hiperplanos H_1 e H_2 deve assegurar que não exista nenhum ponto entre eles. Para isto, algumas restrições devem ser respeitadas. Utilizando a Figura 4 como exemplo, sabemos que os pontos vermelhos pertencem a Classe 1 e os pontos azuis pertencem a Classe -1. Para garantir que nenhum ponto vermelho encontre-se entre os hiperplanos, a Equação 10 deve ser respeitada para todos os pontos \vec{x} da cor vermelha.

$$\vec{w} \cdot \vec{x} + b \geq 1 \quad (10)$$

Da mesma forma, a Equação 11 deve ser respeitada para todos os pontos \vec{x} da cor azul.

$$\vec{w} \cdot \vec{x} + b \leq -1 \quad (11)$$

Tendo as duas equações como base, pode-se definir uma terceira equação que é matematicamente equivalente as Equações 10 e 11 e que transforme elas em uma única equação com ambas as restrições. Para isto, multiplica-se ambos os lados destas equações por y , obtendo-se assim as Equações 12 e 13.

$$y(\vec{w} \cdot \vec{x} + b) \geq y(1) \quad (12)$$

$$y(\vec{w} \cdot \vec{x} + b) \leq y(-1) \quad (13)$$

A Equação 12 representa as restrições que devem ser respeitadas para as amostras da Classe 1 (pontos vermelhos) e a Equação 13 representa as restrições que devem ser respeitadas para as amostras da Classe -1 (pontos azuis). Sabe-se que no conjunto de

treinamento, as classes de cada amostra são representadas pela variável y , isto é, para todas as amostras de Classe 1, y é igual a 1 e para todas as amostras de Classe -1, y é igual a -1. Dessa forma, substituindo y por 1 e -1 no lado direito das Equações 12 e 13, respectivamente, tem-se as Equações 14 e 15.

$$y (\vec{w} \cdot \vec{x} + b) \geq 1 \quad (14)$$

$$y (\vec{w} \cdot \vec{x} + b) \geq -1 \quad (15)$$

Pode-se observar que multiplicando-se o lado direito das Equações 14 e 15 tem-se, em ambos os casos, o valor 1. Desta forma, as restrições para as duas classes podem ser representadas através da Equação 16.

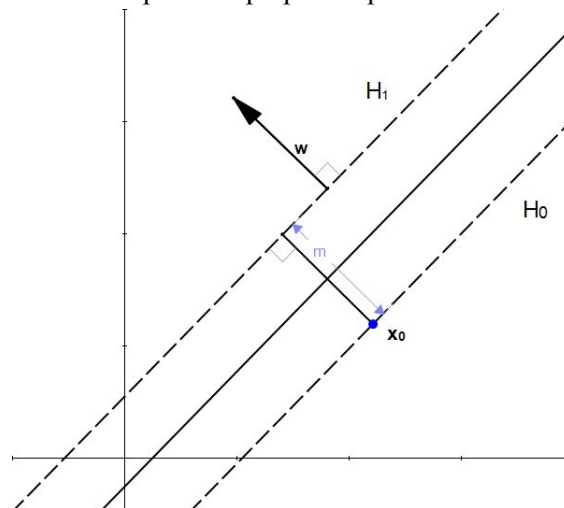
$$y(\vec{w} \cdot \vec{x} + b) \geq 1 \quad (16)$$

Esta restrição será abordada novamente na Seção 4 para a definição do hiperplano mais adequado para a classificação das amostras.

3 Cálculo da margem

O objetivo do método SVM é definir um hiperplano em que seja obtida a maior margem de separação entre as classes. Como exemplo, temos dois planos H_0 e H_1 em que necessita-se definir a margem m entre eles, sendo que, sobre o plano H_0 existe um ponto x_0 que pertencente a Classe -1 (Figura 5).

Figura 5 – Exemplo de hiperplanos para cálculo da margem



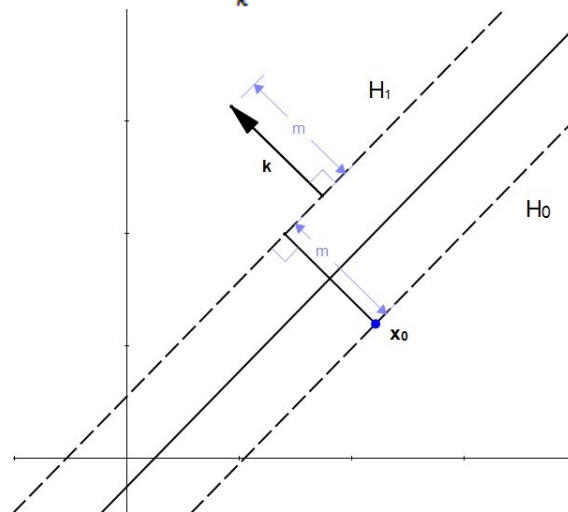
Fonte: Elaboração dos autores.

Tendo em vista que a margem m é um valor escalar, não é possível somá-lo ao ponto x_0 de forma a obter um ponto localizado sobre o hiperplano H_1 . Para isto, faz-se

necessário calcular um vetor \vec{k} de magnitude igual ao valor da margem m e de direção perpendicular aos hiperplanos. O vetor \vec{k} pode ser calculado a partir da Equação 17, onde o vetor unitário de \vec{w} é multiplicado pelo valor escalar m . Em outras palavras, transforma-se o vetor \vec{w} em outro vetor de mesma direção, porém com uma magnitude igual a 1 ($\frac{\vec{w}}{\|\vec{w}\|}$) e, em seguida, este novo vetor deve ser multiplicado pelo valor de m , resultando no vetor \vec{k} (Figura 6).

$$\vec{k} = m * \frac{\vec{w}}{\|\vec{w}\|} \quad (17)$$

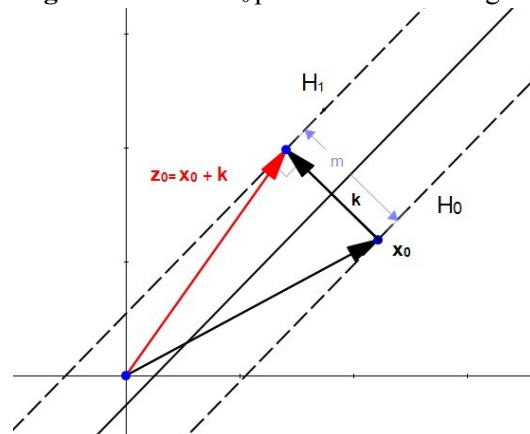
Figura 6 – Vetor \vec{k} para cálculo da margem



Fonte: Elaboração dos autores.

Obtido um vetor \vec{k} de tamanho igual a margem m , podemos somá-lo ao ponto x_0 e obter um novo ponto z_0 que encontra-se sobre o hiperplano H_1 e pertencente a Classe 1 (Figura 7).

Figura 7 – Ponto z_0 para cálculo da margem



Fonte: Elaboração dos autores.

Sabendo que z_0 pertence a Classe 1 e está localizado exatamente sobre o plano de divisão H_1 , pode-se definir a Equação 18.

$$\vec{w} \cdot z_0 + b = 1 \quad (18)$$

A partir da substituição de z_0 por $x_0 + \vec{k}$ na Equação 18 (Figura 7), obtém-se a Equação 19.

$$\vec{w} \cdot (x_0 + \vec{k}) + b = 1 \quad (19)$$

A Equação 19 pode ser reescrita substituindo-se o vetor \vec{k} pela sua definição (Equação 17), obtendo-se a Equação 20.

$$\vec{w} \cdot (x_0 + m \cdot \frac{\vec{w}}{\|\vec{w}\|}) + b = 1 \quad (20)$$

A Equação 20 pode ser reescrita aplicando-se a operação distributiva, obtendo-se a Equação 21.

$$\vec{w} \cdot x_0 + m \cdot \frac{\vec{w} \cdot \vec{w}}{\|\vec{w}\|} + b = 1 \quad (21)$$

Uma vez que o produto escalar de um vetor com ele mesmo é igual ao quadrado de sua magnitude ($\vec{w} \cdot \vec{w} = \|\vec{w}\|^2$) pode-se reescrever a Equação 21, obtendo-se a Equação 22.

$$\begin{aligned} \vec{w} \cdot x_0 + m \cdot \frac{\|\vec{w}\|^2}{\|\vec{w}\|} + b &= 1 \Rightarrow \\ \vec{w} \cdot x_0 + m \cdot \|\vec{w}\| + b &= 1 \Rightarrow \\ \vec{w} \cdot x_0 + b &= 1 - m \cdot \|\vec{w}\| \end{aligned} \quad (22)$$

Dado que o ponto x_0 pertence a Classe -1 e está localizado exatamente sobre o plano H_0 , a restrição definida pela Equação 9 deve ser respeitada. Desse modo, o lado esquerdo da Equação 22 pode ser substituído pela constante -1, definindo-se a Equação 23.

$$\begin{aligned} -1 &= 1 - m \cdot \|\vec{w}\| \Rightarrow \\ -1 - 1 &= -m \cdot \|\vec{w}\| \Rightarrow \\ -2 &= -m \cdot \|\vec{w}\| \Rightarrow \\ 2 &= m \cdot \|\vec{w}\| \Rightarrow \\ m &= \frac{2}{\|\vec{w}\|} \end{aligned} \quad (23)$$

Sendo assim, conclui-se a partir da Equação 23 que a maximização da margem m pode ser obtida através da minimização da norma de \vec{w} , ou seja, a minimização de $\|\vec{w}\|$.

4 Maximização da margem

Sabe-se que o hiperplano é definido pela Equação 2 e que a margem máxima é obtida minimizando a norma do vetor \vec{w} (Equação 23). Além disso, as restrições definidas pela Equação 16 devem ser respeitadas a fim de assegurar que não haja dados de treinamento entre as margens de separação das classes. Esta questão recorre a um problema de otimização, onde deve-se minimizar $\|\vec{w}\|$ aplicando as restrições a todas as amostras de treinamento (Equação 24).

$$\begin{aligned} & \text{Minimizar } \|\vec{w}\| \\ & \text{Com as restrições: } \{ y_i(\vec{w} \cdot \vec{x}_i + b) \geq 1 \end{aligned} \quad (24)$$

Visto que a resolução deste problema é baseada na derivação de funções, é interessante representar a Equação 24 na forma integral, como apresentado na Equação 25. Observa-se que quando aplicada a derivação sobre a Equação 25, é obtida novamente a função original do problema ($\|\vec{w}\|$).

$$\int \|\vec{w}\| \, d\vec{w} = \frac{1}{2} \|\vec{w}\|^2 \quad (25)$$

Assim, o problema de otimização representado pela Equação 24 pode ser representado através da Equação 26.

$$\begin{aligned} & \text{Minimizar } \frac{1}{2} \|\vec{w}\|^2 \\ & \text{Com as restrições: } \{ y_i(\vec{w} \cdot \vec{x}_i + b) \geq 1 \end{aligned} \quad (26)$$

Este problema de otimização pode ser resolvido através do método de Multiplicadores de Lagrange, que foi criado pelo matemático italiano Joseph Louis Lagrange no ano de 1806 (KOWALCZYK, 2017). Este método baseia-se na definição de uma função Lagrangiana, a qual, tendo uma função objetivo, engloba-se as restrições a essa de forma a atingir o objetivo desejado (FACELI *et al.*, 2011).

Para a incorporação das restrições à função objetivo, associa-se a cada restrição um conjunto de parâmetros α_i , denominados multiplicadores de Lagrange (Equação 27).

No contexto das SVM, estes multiplicadores podem ser vistos como a influência de cada restrição na definição do hiperplano (FACELI *et al.*, 2011).

$$\alpha_i(y_i(\vec{w} \cdot \vec{x}_i + b)) \geq \alpha_i(1) \quad (27)$$

Isolando-se os termos da Equação 27, obtém-se a Equação 28.

$$\begin{aligned} \alpha_i(y_i(\vec{w} \cdot \vec{x}_i + b)) &\geq \alpha_i(1) \Rightarrow \\ \alpha_i(y_i(\vec{w} \cdot \vec{x}_i + b)) - \alpha_i &\geq 0 \Rightarrow \\ \alpha_i(y_i(\vec{w} \cdot \vec{x}_i + b) - 1) &\geq 0 \end{aligned} \quad (28)$$

Tendo em vista que a restrição definida na Equação 28 deve ser repetida para todas amostras de treinamento, a função Lagrangiana das SVMs é definida conforme a Equação 29, onde i representa o índice de cada uma das amostras da fase de treinamento. Nesta função, a soma das restrições de todas amostras de treinamento é subtraída do valor ao qual deseja-se minimizar ($\frac{1}{2} \|\mathbf{w}\|^2$).

$$L(\vec{w}, b, \alpha) = \frac{1}{2} \|\mathbf{w}\|^2 - \sum_{i=1}^n \alpha_i (y_i(\vec{w} \cdot \vec{x}_i + b) - 1) \quad (29)$$

Na Equação 29, as variáveis do problema (w e b) devem ser minimizadas enquanto os multiplicadores de Lagrange (variáveis α_i) precisam ser maximizados. Essa questão implica na utilização de uma formulação denominada de forma *dual*. Sendo assim, o problema original (Equação 29), também denominado de forma *primal*, é transformado em um segundo caso, referenciado como forma *dual*. A forma *dual* tende a apresentar as restrições de maneira mais simples a fim de facilitar a resolução do problema (FACELI *et al.*, 2011).

A forma *dual* é construída considerando-se que a maximização das variáveis da função Lagrangiana é obtida nos pontos em que a derivada parcial desta função em relação as demais variáveis são nulas. Isto é, tendo em vista que procura-se minimizar os valores w e b na forma *primal* (Equação 29), a forma *dual* considerará apenas os locais onde $\partial L / \partial \vec{w}$ (derivada parcial de L com relação a \vec{w}) e $\partial L / \partial b$ (derivada parcial de L com relação a b) são iguais a zero.

Neste contexto, define-se inicialmente a derivada parcial de L em relação à b (Equação 30) e, em seguida, a derivada parcial de L em relação à \vec{w} (Equação 31).

$$\frac{\partial L}{\partial b} = \sum_{i=1}^n \alpha_i y_i \quad (30)$$

$$\frac{\partial L}{\partial \vec{w}} = \vec{w} - \sum_{i=1}^n \alpha_i y_i \vec{x}_i \quad (31)$$

Posteriormente, ambas as derivadas parciais (Equações 30 e 31) são igualadas a zero, obtendo-se as Equações 32 e 33.

$$\sum_{i=1}^n \alpha_i y_i = 0 \quad (32)$$

$$\vec{w} - \sum_{i=1}^n \alpha_i y_i \vec{x}_i = 0 \quad (33)$$

A partir da reestruturação da Equação 33, obtém-se a definição de \vec{w} para o problema *dual* (Equação 34).

$$\vec{w} = \sum_{i=1}^n \alpha_i y_i \vec{x}_i \quad (34)$$

A forma *dual* é obtida substituindo \vec{w} na equação *primal* e inserindo a restrição imposta pela Equação 32 como uma restrição do problema *dual*. Deste modo, obtém-se o problema de otimização representado pelas Equações 35 e 36.

$$\text{Maximizar}_{\alpha} \sum_{i=1}^n \alpha_i - \frac{1}{2} \sum_{i,j=1}^n \alpha_i \alpha_j y_i y_j (\vec{x}_i \cdot \vec{x}_j) \quad (35)$$

$$\text{Com as restrições: } \begin{cases} \alpha_i \geq 0, \forall i = 1, \dots, n \\ \sum_{i=1}^n \alpha_i y_i = 0 \end{cases} \quad (36)$$

Atualmente, o algoritmo de minimização sequencial SMO (do inglês, *Sequential Minimal Optimization*) é o método mais empregado para resolver o problema de otimização apresentado nas Equações 35 e 36 (ZENG *et al.*, 2008). Este método foi criado pelo cientista da computação John Carlton Platt em 1998 (PLATT, 1998).

No trabalho intitulado como “*Sequential minimal optimization: A fast algorithm for training support vector machines*”, Platt detalha o funcionamento deste algoritmo, destacando o principal diferencial do método, que consiste em separar o problema inicial em problemas menores, resultando em uma diminuição na quantidade de memória computacional necessária e uma redução no tempo de execução, quando comparado a outros métodos.

Os valores de α são determinados pelo algoritmo SMO e, a partir destes valores, w_{ecw} pode ser obtido através da Equação 34. A definição do valor de b é baseada nas condições KKT (Karush-Kuhn-Tucker), que foram inicialmente definidas pelo matemático William Karush em 1939 (KARUSH, 1939) e complementadas pelos matemáticos Harold William Kuhn e Albert William Tucker em 1951 (KUHN; TUCKER, 1951). Estes definiram que para a obtenção da solução ótima em problemas de otimização, caso do problema tratado as SVMs, as condições KKT devem ser respeitadas.

Diversas condições KKT são descritas, porém a condição utilizada como base para o cálculo de valor de b define que, no ponto ótimo, o produto entre as variáveis duais (de Lagrange) e as restrições do problema *primal* deve ser nulo (Equação 37).

$$\alpha_i(y_i(\vec{w} \cdot \vec{x}_i + b) - 1) = 0 \quad (37)$$

Através da Equação 37 pode-se observar que α_i pode ser diferente de 0 somente para as amostras que se encontram sobre os hiperplanos H_1 e H_2 , isto é, para os pontos em que a Equação 38 é verdadeira. Nos demais casos a Equação 37 só é válida quando $\alpha_i = 0$. As amostras associadas a α com valores maiores de 0, são chamadas de vetores de suporte e são os objetos mais informativos do conjunto de treinamento, uma vez que esses são os dados localizados mais próximos do limite de separação entre as classes e, portanto, o hiperplano de decisão será construído utilizando apenas estes vetores (FACELI *et al.*, 2011).

$$y_i(\vec{w} \cdot \vec{x}_i + b) = 1 \quad (38)$$

O valor do escalar b é calculado a partir da Equação 37, conforme demonstrado na Equação 39.

$$\begin{aligned} \alpha_i(y_i(\vec{w} \cdot \vec{x}_i + b) - 1) &= 0 \\ \alpha_i(y_i\vec{w} \cdot \vec{x}_i + y_ib - 1) &= 0 \\ \alpha_i y_i \vec{w} \cdot \vec{x}_i + \alpha_i y_i b - \alpha_i &= 0 \\ \alpha_i y_i b &= \alpha_i - \alpha_i y_i \vec{w} \cdot \vec{x}_i \\ b &= \frac{\alpha_i}{\alpha_i y_i} - \frac{\alpha_i y_i \vec{w} \cdot \vec{x}_i}{\alpha_i y_i} \\ b &= \frac{1}{y_i} - \vec{w} \cdot \vec{x}_i \end{aligned} \quad (39)$$

A Equação 39 deve ser aplicada para todos os vetores de suporte. Assim, esse procedimento pode ser representado através da Equação 40, ou seja, através da média entre os valores de b para cada vetor de suporte.

$$b = \frac{1}{n_{VS}} \sum_{i=1}^{n_{VS}} \frac{1}{y_i} - \vec{w} \cdot \vec{x}_i \quad (40)$$

Uma vez que o valor de \vec{w} e b são calculados, tem-se definido o hiperplano ótimo para a separação das amostras entre as duas possíveis classes. Dessa forma, a classe de novas amostras pode ser determinada baseando-se na localização do vetor de características recebido (vetor \vec{x}) com relação ao hiperplano, isto é, verificando o sinal retornado pela Equação 41.

$$\vec{w} \cdot \vec{x} + b \quad (41)$$

Neste contexto, o classificador das SVMs é definido pela Equação 42, onde a função sgn retorna 1 para valores positivos (Classe 1) e -1 para valores negativos (Classe -1).

$$y = sgn(\vec{w} \cdot \vec{x} + b) \quad (42)$$

A Equação 42 pode ser reescrita substituindo-se o vetor \vec{w} pela sua definição (Equação 34), obtendo-se a definição final do classificador (Equação 43).

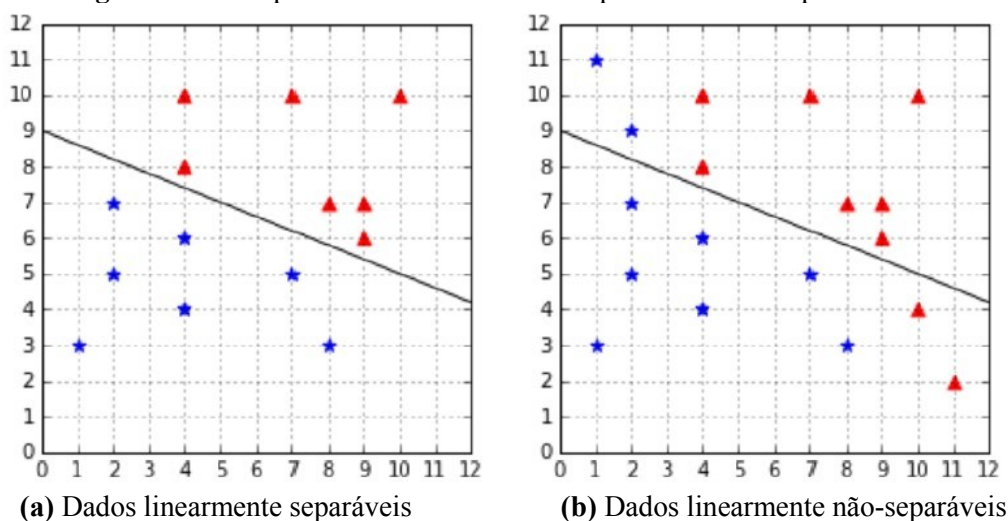
$$y = sgn\left(\sum_{i=1}^{n_{SV}} \alpha_i y_i \vec{x}_i \cdot \vec{x} + b\right) \quad (43)$$

5 Funções de *Kernel*

O classificador da Equação 43 aplica-se às SVMs lineares, os quais são eficazes na classificação de conjuntos de dados linearmente separáveis (Figura 8a). No entanto, em grande parte das ocasiões, não é possível dividir satisfatoriamente os dados de treinamento através de um hiperplano (Figura 8b). Sendo assim, para a abordagem de problemas não lineares, são utilizadas funções denominadas *kernels*. Estas funções mapeiam o conjunto de dados de seu espaço original para um espaço com um número maior de dimensões.

Este mapeamento tem como objetivo facilitar a separação das amostras.

Figura 8 – Exemplo de dados linearmente separáveis e não-separáveis



Fonte: Elaboração dos autores.

Os *kernels* são incorporados ao classificador das SVMs conforme a Equação 44, onde K denota a função *kernel*, a qual recebe como entrada $x_{\sim i}$ o vetor de suporte i e o vetor $\sim x$ denotando a amostra a ser classificada.

$$y = \text{sgn}\left(\sum_{i=1}^{n_{SV}} \alpha_i y_i K(\vec{x}_i, \vec{x}) + b\right) \quad (44)$$

A escolha do *kernel* mais apropriado depende fundamentalmente do problema a ser abordado. Inicialmente, a própria disposição dos dados pode sugerir a escolha de um *kernel*. Entretanto, em situações onde a complexidade dos dados não permite a visualização explícita desta disposição, a literatura sugere a definição do *kernel* através de testes. Os *kernels* mais utilizados na prática são o Polinomial (BOSER; GUYON; VAPNIK, 1992) e o Gaussiano ou RBF (do inglês, *Radial-Basis Function*) (KEERTHI; LIN, 2003).

5.1 Kernel Polinomial

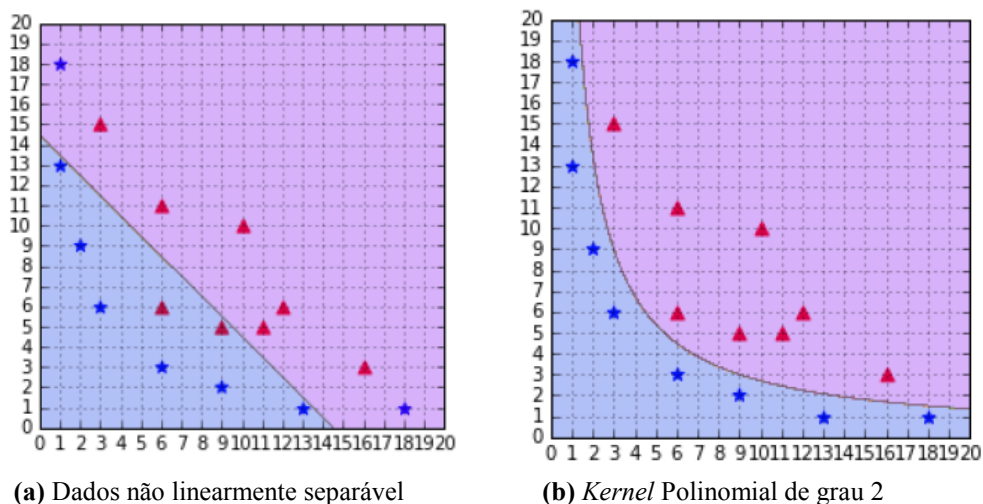
O *kernel* Polinomial permite modelar hiperplanos para separação dos dados a partir de funções de ordem polinomial. A função da Equação 45 representa esse tipo de *kernel*, sendo que essa função possui 2 valores a serem definidos, onde o valor c corresponde a uma constante e o valor d representa o grau do polinômio.

$$K(x, x') = (x \cdot x' + c)^d \quad (45)$$

Na Figura 9a tem-se um exemplo de um conjunto de dados que não é linearmente separável em um espaço de duas dimensões, isto é, em que não é possível separá-los

através de uma reta. Neste contexto, o emprego do *kernel* Polinomial de grau 2 mostra-se mais adequado (Figura 9b).

Figura 9 – Separação de dados através do *kernel* Polinomial de grau 2



Fonte: Elaboração dos autores.

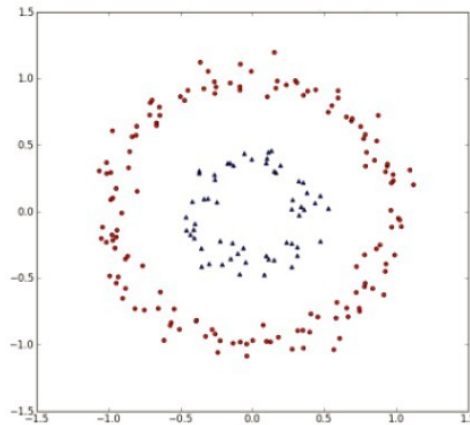
5.2 *Kernel* Gaussiano ou RBF

O *kernel* Gaussiano, também conhecido como RBF, permite a separação dos dados a partir de círculos ou hiperesferas. Este *kernel* mapeia o espaço de entradas para um novo espaço com um número maior de dimensões (Equação 46), buscando possibilitar a separação dos dados através de um hiperplano.

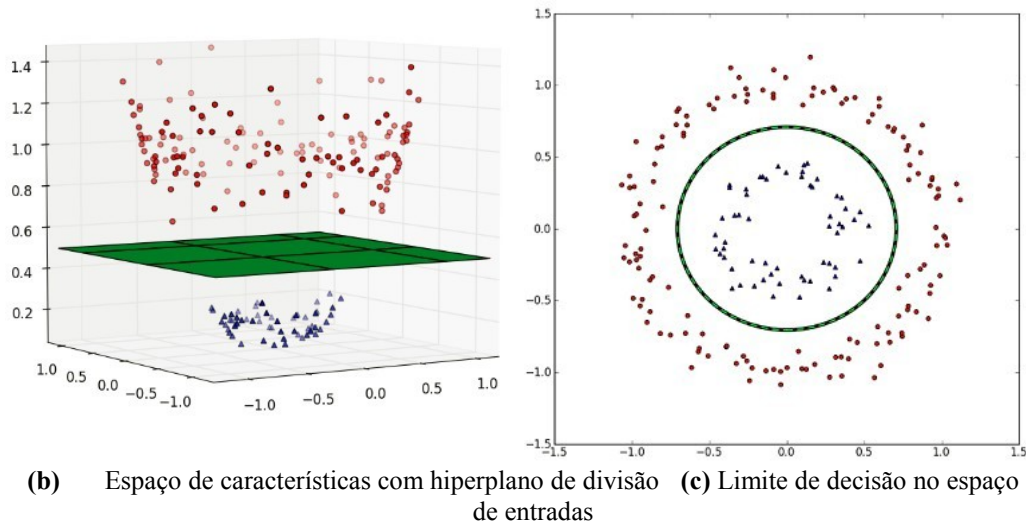
$$K(x, x') = \exp(-\gamma \|x - x'\|^2) \quad (46)$$

Na Figura 10a tem-se um conjunto de dados em um espaço de duas dimensões, em que não é possível separar os dados por um hiperplano ou por uma função polinomial. Aplicando o *kernel* RBF, estes dados podem ser mapeados em um espaço de 3 dimensões, por exemplo, e, desta forma, torna-se possível definir um hiperplano para a divisão das classes (Figura 10b). A Figura 10c apresenta o limite de decisão no espaço de entradas, isto é, no espaço original.

Figura 10 – Separação de dados através do *kernel* RBF



(a) Conjunto de dados não separáveis pelo *kernel* Polinomial



(b) Espaço de características com hiperplano de divisão (c) Limite de decisão no espaço de entradas

6 Ferramentas que implementam SVMs

Apesar da rigorosa fundamentação matemática, as SVMs podem ser facilmente utilizadas através do uso de bibliotecas ou ferramentas que as implementam. Dentre estas, destacam-se as bibliotecas mySVM (RUPING, 2000), SVM Light (JOACHIMS, 1999) e LibSVM (CHANG; LIN, 2011).

A biblioteca mySVM é um pacote de código aberto desenvolvido na linguagem de programação C++. Além disso, ela apresenta ainda uma implementação na linguagem de programação Java com o nome de JmySVM (HOFMANN; KLINKENBERG, 2013). Essa biblioteca destaca-se pelo suporte à diversos formatos de arquivos de entrada, bem como a disposição de diversos tipos de *kernels* para utilização (SIANG *et al.*, 2015). Como um exemplo de sua aplicação, pode-se citar o trabalho de (WEATHERS *et al.*,

2004), onde a biblioteca mySVM é utilizada para reconhecimento de regiões de desordem em proteínas intrinsecamente desordenadas.

A biblioteca SVM-Light é uma implementação de código aberto desenvolvida na linguagem de programação C por pesquisadores da Universidade Cornell (Nova Iorque, EUA). Ela possui suporte à classificação multi-classes, tratamento de dados não-linearmente separáveis, bem como ferramentas de validação e estimativa de erro (SIANG *et al.*, 2015). Como exemplo de sua aplicação, pode-se destacar o trabalho de (BOCK; GOUGH, 2001) o qual apresenta uma solução para o reconhecimento da interação proteína-proteína (JOACHIMS, 1999).

A biblioteca LibSVM (CHANG; LIN, 2011) é, atualmente, a ferramenta mais utilizada na comunidade científica, sendo citada em inúmeros trabalhos relacionados a classificação de dados (BEN-HUR; WESTON, 2010). A LibSVM é uma biblioteca de código aberto desenvolvida nas linguagens de programação Java e C++, possuindo suporte à classificação de multi-classes, tratamento de dados não-linearmente separáveis, ferramenta de validação cruzada, entre outros (SIANG *et al.*, 2015). Além de permitir a integração com linguagens de programação, como Python, R e MATLAB, muitas outras ferramentas incorporam esta biblioteca a fim de prover interfaces simplificadas para aplicação das SVMs. Dentre essas ferramentas, pode-se destacar o WEKA (FRANK *et al.*, 2004), RapidMiner (KOTU; DESHPANDE, 2014) e Orange (DEMŠAR *et al.*, 2013).

O principal objetivo destas interfaces é possibilitar o uso das SVMs de forma simples, permitindo ao utilizador da ferramenta focar exclusivamente na análise exploratória das informações, não sendo necessário nenhum conhecimento técnico no que diz respeito ao embasamento matemático em que a técnicas das SVMs se apoia. Além disso, são disponibilizadas técnicas que podem ser utilizadas em conjunto, em etapas anteriores e posteriores ao processo de classificação. Por exemplo, em etapas anteriores pode-se aplicar filtro com o intuito de melhorar a qualidade das amostras. Já após a classificação pode-se utilizar técnicas de validação dos modelos, bem como uma visualização dos resultados através de interfaces gráficas.

Referências

BEN-HUR, A.; WESTON, J. A user's guide to support vector machines. **Data mining techniques for the life sciences**, [S.l.]: Springer, 2010. p. 223-239.

BOCK, J. R.; GOUGH, D. A. Predicting protein-protein interactions from primary structure. **Bioinformatics**, Oxford University Press, v. 17, n. 5, p. 455-460, 2001.

BOSER, B. E.; GUYON, I. M.; VAPNIK, V. N. A training algorithm for optimal margin classifiers. *In*: ACM. **Proceedings of the fifth annual workshop on Computational learning theory**. [S.l.], 1992. p. 144-152.

- CHANG, C.-C.; LIN, C.-J. Libsvm: A library for support vector machines. **ACM transactions on intelligent systems and technology (TIST)**, Acm, v. 2, n. 3, p. 27, 2011.
- DEMŠAR, J. *et al.* Orange: data mining toolbox in python. **The Journal of Machine Learning Research**, JMLR org, v. 14, n. 1, p. 2349-2353, 2013.
- FACELI, K. *et al.* **Inteligência artificial: uma abordagem de aprendizado de máquina**. 2011.
- FRADKIN, D.; MUCHNIK, I. Support vector machines for classification. **DIMACS Series in Discrete Mathematics and Theoretical Computer Science**, Citeseer, v. 70, p. 13-20, 2006.
- FRANK, E. *et al.* Data mining in bioinformatics using weka. **Bioinformatics**, Oxford University Press, v. 20, n. 15, p. 2479-2481, 2004.
- HOFMANN, M.; KLINKENBERG, R. **RapidMiner: Data mining use cases and business analytics applications**. [S.l.]: CRC Press, 2013.
- JOACHIMS, T. Svm-light: Support vector machine. **SVM-Light Support Vector Machine, University of Dortmund**, v. 19, n. 4, 1999.
- KARUSH, W. Minima of functions of several variables with inequalities as side conditions. **Master thesis**, University of Chicago, 1939.
- KEERTHI, S. S.; LIN, C.-J. Asymptotic behaviors of support vector machines with gaussian kernel. **Neural computation**, MIT Press, v. 15, n. 7, p. 1667-1689, 2003.
- KOTU, V.; DESHPANDE, B. **Predictive analytics and data mining: concepts and practice with rapidminer**. [S.l.]: Morgan Kaufmann, 2014.
- KOWALCZYK, A. **Support Vector Machines Succinctly**. [S.l.]: Syncfusion Inc., 2017.
- KUHN, H. W.; TUCKER, A. W. Nonlinear programming. *In: Proceedings of the Second Berkeley Symposium on Mathematical Statistics and Probability*. Berkeley, Calif.: University of California Press, 1951. p. 481-492.
- LORENA, A. C.; CARVALHO, A. C. de. Uma introdução às support vector machines. **Revista de Informática Teórica e Aplicada**, v. 14, n. 2, p. 43-67, 2007.
- MEYER, D.; WIEN, F. T. Support vector machines. **The Interface to libsvm in package e1071**, p. 28, 2015.
- PLATT, J. **Sequential minimal optimization: a fast algorithm for training support vector machines**. 1998.
- RUPING, S. **MYSVM-manual**. <http://www-ai.cs.unidortmund.de/software/mysvm/>, 2000.
- SIANG, T. C. *et al.* A review of cancer classification software for gene expression data. **International Journal of Bio-Science and Bio-Technology**, v. 7, n. 4, p. 89-108, 2015.
- WEATHERS, E. A. *et al.* Reduced amino acid alphabet is sufficient to accurately recognize intrinsically disordered protein. **FEBS letters**, Wiley Online Library, v. 576, n. 3, p. 348-352, 2004.
- ZENG, Z.-Q. *et al.* Fast training support vector machines using parallel sequential minimal optimization. *In: IEEE. 2008 3rd international conference on intelligent system and knowledge engineering*. [S.l.], v. 1, p. 997-1001, 2008.

COMPUTAÇÃO PARALELA E DISTRIBUÍDA

Alex A. L. dos Santos,¹ Felipe S. Raota,² Guilherme T. Paz,³ Marcelo Brazil,⁴
André L. Martinotto⁵

Nos últimos anos, a bioinformática tem observado um crescimento acentuado e contínuo no tamanho das bases de dados, bem como o desenvolvimento de aplicações cada vez mais complexas e que possuem alta demanda computacional.

Apesar da velocidade de processamento dos computadores atuais, o crescimento das bases de dados e o aumento na complexidade das aplicações tornou-se inviável a solução desse tipo de problema, em um tempo razoável. Uma alternativa para contornar esse problema é o uso da computação paralela, que consiste em dividir o problema em partes menores, que podem ser executadas em processadores diferentes (ALMASI; GOTTLIEB, 1989).

Um sistema paralelo pode ser definido como uma coleção de unidades de processamento, que trabalham em conjunto para a solução de um determinado problema (FOSTER, 1995). Essa definição é abrangente incluindo, entre outros, supercomputadores, computadores multiprocessados, *clusters* de computadores, unidades de processamento gráfico (*Graphics Processing Unit*), etc.

Neste capítulo é feita uma breve descrição das arquiteturas paralelas mais comuns e acessíveis para a maioria dos pesquisadores. Essa descrição é introdutória e tem como objetivo situar o leitor sobre o tema. Para uma descrição mais completa sugere-se a leitura de Stallings (2018) e Tanenbaum (2013). Mais especificamente neste capítulo são abordados conceitos introdutórios sobre computadores multiprocessados, multicomputadores, *grids* computacionais e GPUs. Além disso, abordados conceitos relacionados à computação em nuvem, que passou a ser uma alternativa para grupos que não possuem condições para investir em uma infraestrutura de alto desempenho e/ou não possuem profissionais para gerenciar essa infraestrutura.

1 Multiprocessadores

Em computadores com memória compartilhada, também chamados de multiprocessadores, todas as unidades de processamento trabalham sobre uma memória comum. Desta forma, a comunicação entre os processadores pode ser feita através de

¹ Universidade de Caxias do Sul. *E-mail*: aalsant1@ucs.br

² Universidade de Caxias do Sul. *E-mail*: fsraota@ucs.br

³ Universidade de Caxias do Sul. *E-mail*: gtelespaz@gmail.com

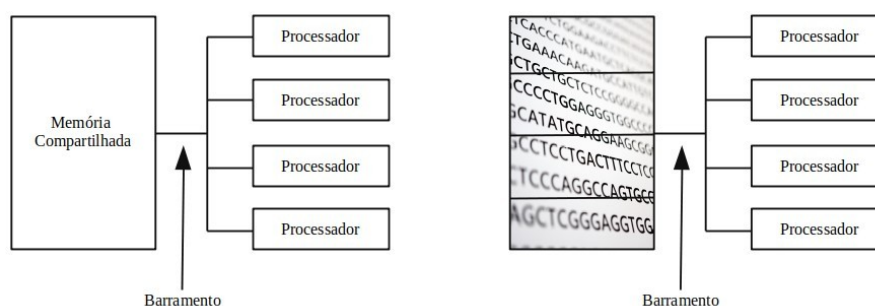
⁴ Universidade de Caxias do Sul. *E-mail*: marcelo@marcelo-brazil.com

⁵ Universidade de Caxias do Sul. *E-mail*: almartin@ucs.br

operações de escrita e leitura, nesta área de memória compartilhada (TANENBAUM, 2013). O exemplo mais comum de multiprocessadores são as arquiteturas *multicore*, nas quais múltiplos processadores, em um único *chip*, compartilham um espaço de memória comum (HENNESSY; PATTERSON, 2014). Os processadores *multicores* são encontrados atualmente em *desktops*, *laptops* e até mesmo em telefones celulares.

A Figura 1 ilustra o modo de funcionamento de um multiprocessador. Por exemplo, considere um programa que efetue algum tipo de processamento sobre uma sequência de DNA. Em um sistema de memória compartilhada, uma única cópia da sequência é mantida em memória, sendo que todos os processadores possuem acesso a essa sequência por completo. Assim, se algum processador efetuar uma alteração na sequência, essa modificação será visível a todos os demais processadores.

Figura 1 – Arquitetura de um Multiprocessador



Fonte: Elaborada pelos autores (2020).

O uso de memória compartilhada provê algumas vantagens em relação a outras arquiteturas, entre as quais uma transição mais natural de ambientes monoprocessados, menor custo de comunicação entre os processadores e a eliminação da necessidade de distribuição dos dados entre os processadores (TANENBAUM, 2013).

A principal desvantagem desse tipo de arquitetura é que, nos multiprocessadores mais comuns, todos os processadores utilizam um mesmo barramento (Figura 1) para acessar a memória. Essa característica provoca uma disputa pelo barramento, podendo fazer com que os processadores fiquem ociosos enquanto esperam para acessar a memória. Essa disputa pelo barramento aumenta com o aumento do número de processadores, limitando esse tipo de arquitetura a algumas dezenas de processadores (TANENBAUM, 2013).

A forma mais comum de exploração de paralelismo em computadores multiprocessados é o uso de múltiplas *threads* (SILBERSCHATZ, 2008). O uso de *multithreading* é uma maneira eficiente de explorar o paralelismo nesse tipo de arquitetura, uma vez que as diferentes *threads* podem ser executadas nos diferentes processadores simultaneamente (LEWIS; BERG, 1998).

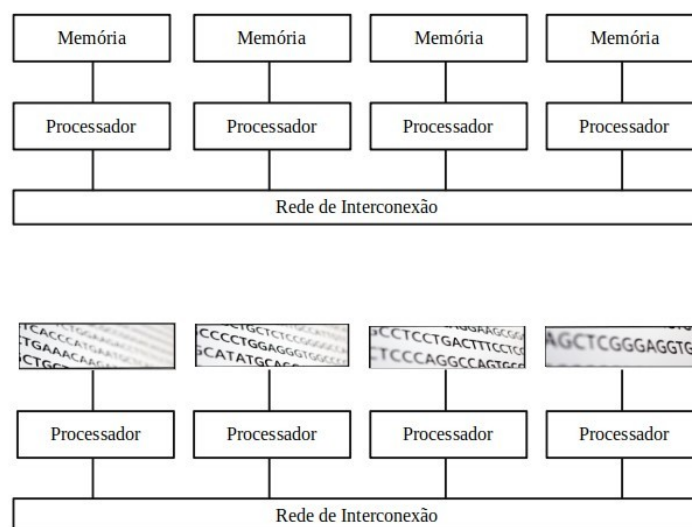
Entre as ferramentas de programação *multithreading* destaca-se a biblioteca OpenMP (CHAPMAN; JOST; PAS, 2007), sendo que essa é amplamente utilizada, inclusive, na paralelização de aplicações da área de bioinformática. Como exemplo cita-se o uso de OpenMP no alinhamento de sequências de DNA (FLOURI *et al.*, 2012; SATHE; SHRIMANKAR, 2011) e na inferência de haplótipos, usando o algoritmo adaptativo EM (*Expectation-Maximization*) (RANOK; KITTITORNKUN; TONGSIMA, 2011).

2 Multicomputadores

Em arquiteturas com memória distribuída, também chamadas de multicomputadores, cada processador possui uma memória própria, que não pode ser acessada de forma direta por outro processador. A forma mais comum para a comunicação em arquiteturas com memória distribuída é a troca de mensagens, ou seja, a comunicação é feita através do envio e recebimento de mensagens (TANENBAUM, 2013).

A Figura 2 ilustra o modo de funcionamento de um multicomputador. Considere o mesmo programa que foi apresentado na Seção 1, que efetua algum tipo de processamento sobre uma sequência de DNA. Em um sistema de memória distribuída, o programa dividiria os elementos da sequência entre as memórias de todos os processadores. No caso de um processador necessitar dos dados que estão na memória de um outro processador, ele não poderá acessar esses dados de forma direta. Neste caso, é necessário efetuar a troca de mensagens, por uma rede de interconexão, transferindo os dados de uma memória para outra (TANENBAUM, 2013).

Figura 2 – Arquitetura de um Multicomputador



Fonte: Elaborado pelos autores (2020).

A principal limitação no uso do paradigma de troca de mensagens é o tempo consumido (*overhead* de rede) para a comunicação. O alto *overhead* dificulta o desenvolvimento de aplicações de alto desempenho, podendo, até mesmo, tornar inviável o uso da troca de mensagens em aplicações com grande dependência entre as operações.

Apesar do alto custo de comunicação, o uso de multicomputadores tem crescido muito nos últimos anos. Um dos principais motivos para o crescimento desse tipo de arquitetura foi o surgimento dos *clusters* de computadores. Esses são montados a partir da união de computadores independentes, interconectados por uma rede de interconexão dedicada e rápida, formando uma plataforma de alto desempenho para a execução de aplicações paralelas (BUYA, 1999; HENNESSY; PATTERSON, 2014). A utilização de *clusters* só é uma realidade devido ao desenvolvimento e barateamento das tecnologias de redes locais e à evolução na capacidade de processamento dos computadores pessoais.

O uso de *clusters* de computadores teve um aumento significativo nos últimos anos, devido, principalmente, ao baixo custo e à escalabilidade da arquitetura. De fato, a escalabilidade de um *cluster* de computadores é, em princípio, ilimitada, pois basta acrescentar novos computadores à rede. Na Figura 3, tem-se uma imagem do *cluster* de computadores que foi montado em 2005, pelo Departamento de Informática da Universidade de Caxias do Sul.

Figura 3 – *Cluster* de computadores



Fonte: Elaborado pelos autores (2020).

Para a utilização de um *cluster*, além de uma rede de comunicação, é necessária uma camada de *software* para a troca de mensagens entre os processos. Para tanto, existem bibliotecas especializadas para a comunicação e a sincronização de processos, sendo que entre essas a mais utilizada é o MPI (*Message-Passing Interface*) (GROPP; HUSS-LEDERMAN; SNIR, 1998; GROPP *et al.*, 1999). Os *clusters* de computadores e

a biblioteca MPI são amplamente utilizados em diversas áreas de conhecimento. Na área de bioinformática, pode-se citar trabalhos que utilizam essas tecnologias para o alinhamento de sequências de DNA (LI, 2003) e na predição de estruturas de proteínas (KALEGARI; LOPES, 2013).

Embora as aplicações que utilizam troca de mensagens sejam próprias para ambientes de memória distribuída, isso não impede que sejam executadas em computadores paralelos com memória compartilhada. Porém, quando são usadas em ambientes de memória compartilhada, as aplicações que utilizam troca de mensagens não aproveitam a principal vantagem desse tipo de arquitetura, que é o uso de uma memória comum para a comunicação entre os processadores (BUYAYA, 1999).

Para alguns autores, um dos maiores problemas do paradigma de troca de mensagens é que esse é mais complexo e difícil de programar, se comparado ao uso de memória compartilhada. Uma alternativa para contornar esse problema é o uso de ferramentas que permitam simular um ambiente de memória compartilhada em ambientes com memória distribuída. Essas ferramentas são chamadas de DSM (*Distributed Shared Memory*) e adicionam custos que podem diminuir o desempenho da aplicação (PROTIC *et al.*, 1998). Apesar dessa desvantagem, a arquitetura DSM já foi utilizada com sucesso no alinhamento de sequências de DNA (MELO *et al.*, 2004).

3 GRIDS Computacionais

Um *grid* computacional é um sistema formado por um conjunto de computadores que compartilham seus recursos ociosos através de uma rede, podendo essa ser uma rede local ou uma rede de longa distância.

Os *grids* computacionais surgiram com o objetivo de criar um “supercomputador virtual” a partir da utilização de recursos ociosos de computadores independentes, sem uma preocupação com a localização física desses e sem a necessidade de investimentos em *hardware* (FOSTER; KESSELMAN, 1999; FOSTER, 2001). Esse tipo de arquitetura torna-se atrativo, uma vez que possibilita a execução de tarefas utilizando recursos computacionais que, de outra forma, estariam ociosos e evitando desta forma o desperdício de processamento.

Como exemplo de uso de um *grid* computacional, que utiliza recursos interligados por uma rede de longa distância, pode-se citar duas organizações que estão localizadas em países diferentes e que funcionam em fusos horários diferentes. Essas organizações poderiam formar um *grid*, de forma que uma organização utilizasse os recursos da outra, que estariam ociosos em determinados horários, devido à diferença de fuso.

Um outro exemplo seria um *grid* computacional formado a partir de recursos ligados por uma rede local. Em organizações de médio e grande porte, existem centenas ou milhares de computadores que são subutilizados. Esses computadores não apresentam toda sua capacidade de processamento aproveitada, ficando em grande parte do tempo em estado ocioso ou até mesmo desligados. Desta forma, poderia ser criado um *grid* computacional utilizando essa capacidade ociosa para a execução de aplicações, que demandam alto desempenho. Como exemplo, pode-se citar o GridUCS, que foi criado de forma a utilizar os recursos ociosos dos laboratórios de ensino de informática da Universidade de Caxias do Sul (MARTINOTTO *et al.*, 2008).

O uso de um *grid* computacional possui algumas complicações adicionais, em relação ao uso de um *cluster* de computadores. De fato, os *clusters* são, na sua maioria, homogêneos quanto à arquitetura (*hardware* e *software*), enquanto os *grids* podem ser constituídos por uma grande variedade de arquiteturas, incluindo *clusters*, supercomputadores e computadores pessoais. O grande desafio é garantir que essa heterogeneidade seja transparente ao usuário e que as aplicações possam ser executadas independentemente da arquitetura.

Outra dificuldade diz respeito ao número de recursos disponíveis. Os *grids* possuem uma natureza mais dinâmica, uma vez que a quantidade de computadores disponíveis é variável (DONGARRA *et al.*, 2003). Isso ocorre porque os computadores dos *grids* não são dedicados, sendo utilizados para outros propósitos, como, por exemplo, armazenamento de dados, serviços de rede ou processamento de aplicativos do usuário. Por sua vez, nos *clusters* os computadores são dedicados ao processamento, de modo que a possibilidade de mudança no número de computadores é menor. Desta forma, os sistemas utilizados no gerenciamento de um *grid*, bem como as aplicações desenvolvidas para esse tipo de infraestrutura, devem se adaptar ao número de recursos existentes, prevendo que a qualquer momento um computador pode deixar de estar disponível.

Por fim, os *clusters* interligam computadores de um mesmo domínio e geograficamente próximos, enquanto *grids* interligam recursos em uma escala muito maior, podendo estar geograficamente distantes. Desta forma, as aplicações devem ser desenvolvidas prevendo a possibilidade de problemas, devido à latência de comunicação. Esse alto *overhead* de comunicação torna os *grids* computacionais mais adequados para a execução de tarefas do tipo *bag of tasks*, ou seja, aplicações que executam de forma quase independente sem a necessidade de comunicação entre as tarefas.

Atualmente, existem vários exemplos de *grids* computacionais, cada qual com seus objetivos e dispersão geográfica. Dentre esses, alguns exemplos que merecem

destaque são o TeraGrid (WILKINS-DIEHR *et al.*, 2008), que interliga os recursos computacionais de diferentes centros de supercomputação norte-americanos, e o *Enabling Grid for E-sciencE* (EGEE) (GAGLIARDI, 2005), que interliga os recursos computacionais de centros europeus e mundiais. No Brasil pode-se citar o GridUNESP (IOPE N. LEMKE, 2010), que interliga os recursos computacionais disponíveis nos diferentes campus da Universidade Estadual Paulista (Unesp), e que pode ser utilizado pela comunidade científica em geral.

Existem diversas ferramentas para a implantação e o gerenciamento de uma estrutura de *grid*, sendo que entre essas as mais utilizadas são o Globus (FOSTER; KESSELMAN, 1997) e o HTCondor (THAIN; TANNENBAUM; LIVNY, 2005). Destaca-se que essas ferramentas já se mostraram adequadas para a execução de aplicações de diversas áreas, inclusive da área de bioinformática (SUN *et al.*, 2004; NEBRO *et al.*, 2008).

4 Unidades de processamento gráfico – GPUS

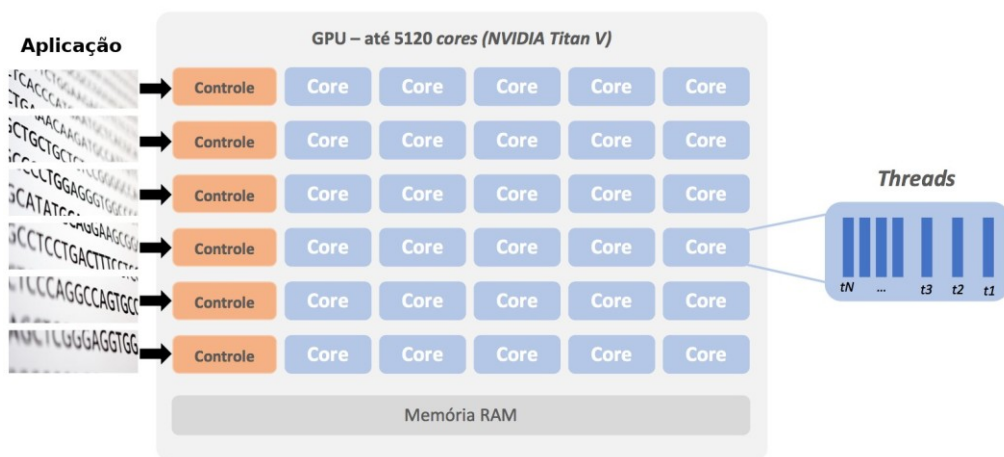
Originalmente, as GPUs tinham como propósito a aceleração e renderização de gráficos 3D em computadores. Sendo assim, as fabricantes utilizavam grande parte de seus esforços no avanço de tecnologias para uso em jogos de computador. A primeira empresa que detectou o potencial das GPUs, para além do processamento gráfico, foi a NVIDIA, que começou a desenvolver unidades para a execução de aplicações de uso geral. Essas unidades são conhecidas como Unidades de Processamento Gráfico de Propósito Geral ou GPGPUs (do inglês, *General Purpose Graphics Processing Units*) (FERREIRA *et al.*, 2013). Nos últimos anos, a comunidade científica tem utilizado as GPUs como uma alternativa de plataforma de computação de alto desempenho com baixo custo. De fato, comparando-se com sistemas tradicionais de alto desempenho, o custo de sistemas com aceleração de GPUs é baixo, em relação ao poder computacional disponível.

Embora sejam similares, as CPUs e as GPUs possuem diferenças fundamentais em sua arquitetura. O poder computacional das GPUs deve-se principalmente pela capacidade de processar trechos de código em paralelo de forma eficiente. Essa eficiência ocorre, pois o problema é dividido em diversas partes, que são processadas por centenas ou milhares de núcleos (*cores*) simples. Além disso, cada um dos núcleos conta com um grande número de *threads*, que executam um mesmo trecho de código de forma simultânea. Esse formato se opõe ao modelo tradicional de CPUs *multicore*, que possuem um número muito inferior de núcleos; porém são mais completos e independentes, permitindo a execução de instruções complexas. Na Figura 4, tem-se

uma ilustração da arquitetura de uma GPU. Como pode ser observado, uma GPU pode apresentar alguns milhares de processadores mais simples.

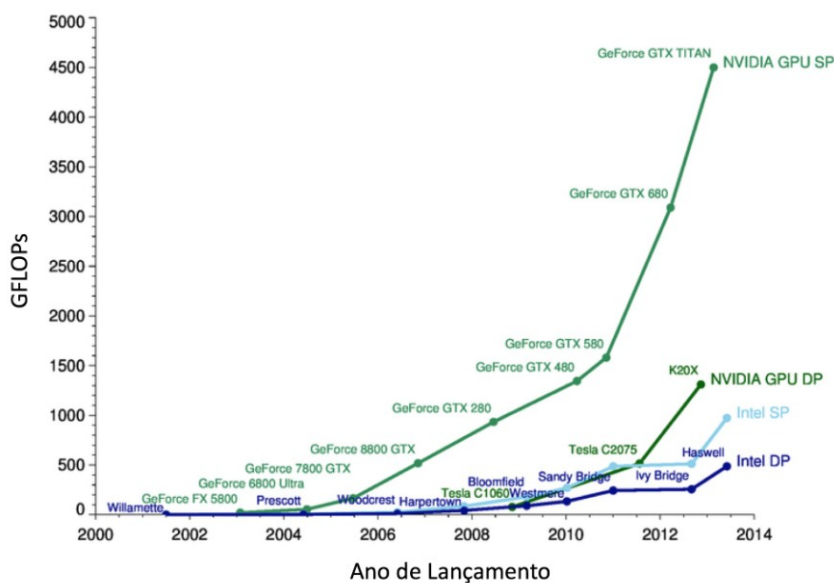
Nos últimos anos, o poder de processamento das GPUs cresceu muito mais rápido que o das CPUs. Na Figura 5, é possível verificar um comparativo da evolução do desempenho entre GPUs da marca NVIDIA com CPUs da marca Intel. Nesse comparativo, é avaliado o desempenho do processamento de operações de ponto flutuante por segundo FLOPS (do inglês, *Floatingpoint Operations per Second*). Nessa figura, é possível identificar que, em 2008, a diferença de desempenho entre elas chegava a 10 vezes, com 1000 GFLOPS para GPUs e 100 GFLOPS para CPUs (KIRK; HWU, 2010).

Figura 4 – Comparativo entre modelos de arquitetura de CPU e GPU



Fonte: Elaborado pelos autores (2020).

Figura 5 – Operações de ponto flutuante por segundo (CPU x GPU)



Fonte: Kirk e HWU (2010).

É importante destacar que o espaço físico necessário para a utilização de arquiteturas baseadas em GPUs é menor; sendo assim, uma das grandes vantagens do uso de GPUs é a economia de energia gerada com a diminuição dos sistemas de refrigeração. Por exemplo, para se obter um ambiente com aproximadamente 16,000 *cores*, utilizando CPUs seriam necessários 2,000 processadores com 8 *cores* cada. Por outro lado, utilizando somente 3 GPUs Nvidia Titan Z, chega-se a 17,280 *cores*, que fisicamente podem ser alocados em um espaço muito menor.

Atualmente, o uso de GPUs para processamento de alto desempenho possui melhor custo-benefício. Mas, mesmo tendo custo inferior aos sistemas tradicionais, ainda se tornam caros para os pesquisadores. Sendo assim, no caso da empresa NVIDIA, a comercialização de GPUs possui várias linhas, destinadas a diferentes ramos do mercado. Citam-se, por exemplo, a linha GeForce, mais utilizada no ramo de jogos em computadores pessoais; a linha Quadro focada no mercado de criação gráfica profissional, e a linha Tesla, específica para utilização em aplicações de propósito geral (SANDES, 2011). No caso da linha GeForce, encontram-se as placas que são usualmente mais baratas, e que ainda conseguem capacidade computacional na ordem de GFLOPS. Por ser considerada a linha com melhor custo/benefício, esta é amplamente utilizada em pesquisas acadêmicas.

O desenvolvimento de aplicações que utilizam GPUs, não segue um modelo tradicional de programação, devido à simplicidade das unidades de controle. Sendo assim, o desenvolvedor é forçado a assumir certos pontos de controle do processamento, utilizando bibliotecas específicas para a programação de GPUs. Dentre as bibliotecas disponíveis, destacam-se as bibliotecas OpenCL (*Open Computing Language*), desenvolvida por empresas como Apple Inc, AMD, Intel, NVIDIA e ATI (KIRK; HWU, 2010), e a biblioteca CUDA (*Compute Unified Device Architecture*), que foi desenvolvida especificamente para os produtos da empresa NVIDIA (NVIDIA, 2019).

Atualmente, as GPUs lideram a disputa de desempenho no processamento com ponto flutuante. Devido a essa tendência, pesquisadores da área de bioinformática passaram a adotar estratégias baseadas em GPUs para a paralelização de suas aplicações. Alguns exemplos de trabalhos recentes, que fazem uso dessa abordagem são: modelagem de sistemas biológicos (SONG; YANG; LEI, 2018), buscas em bases de dados de proteínas (ZHOU *et al.*, 2018) e sequenciamento de DNA (MOREANO; MELO, 2017).

5 Computação em nuvem

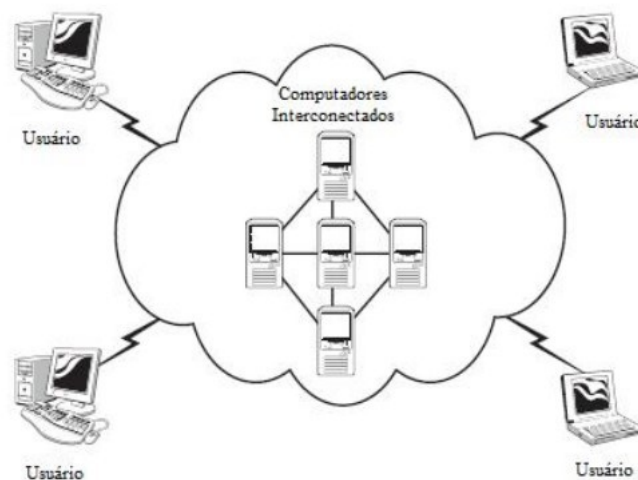
A computação em nuvem não é um novo tipo de arquitetura ou de sistema computacional. Consiste basicamente em um modelo de negócio, em que a computação (processamento, armazenamento e *softwares*) está em algum lugar da rede e é acessada remotamente, via internet (VAQUERO *et al.*, 2008).

A ideia central da computação em nuvem é similar ao que é observado em alguns tipos de serviço tais como: fornecimento de gás, energia elétrica, água e telefonia. Nestes casos, o usuário utiliza o serviço sem se preocupar com a infraestrutura existente, pagando ao provedor apenas a quantidade utilizada do recurso. Da mesma forma, uma nuvem computacional provê recursos computacionais que podem ser utilizados através da internet, sem necessariamente que o usuário tenha um conhecimento prévio da infraestrutura da nuvem. Além disso, o pagamento é realizado de acordo com o modelo *pay-per-use*, em que o usuário só paga pelos recursos que utilizar e pelo tempo que utilizar (BUYAYA *et al.*, 2009).

Na Figura 6, tem-se uma imagem em alto nível da infraestrutura de uma nuvem. Como pode ser observado, os usuários conectam-se na nuvem através de seus computadores pessoais ou dispositivos portáteis conectados à internet. Para estes usuários, a nuvem é vista como um conjunto de computadores, aplicações ou documentos. O *hardware* da nuvem, bem como o sistema operacional e os demais *softwares*, que controla a arquitetura, permanece invisível para o usuário (MILLER, 2008).

A principal vantagem da computação em nuvem é a possibilidade de utilizar esse modelo como uma ferramenta para a diminuição dos custos. De fato, as organizações podem utilizar esse modelo visando a diminuir seu investimento na aquisição e na atualização dos recursos computacionais. Por exemplo, uma organização não necessitaria investir em servidores para efetuar o armazenamento de uma grande base de dados. Essa poderia contratar um serviço de nuvem pagando apenas uma taxa mensal, que é calculada com base na quantidade de dados armazenados e na taxa de dados transferidos. Destaca-se, porém, que um serviço de nuvem pode se tornar caro no caso de aplicações que demandam alto poder de computação (processador e memória) e que possuem um longo período de execução.

Figura 6 – Infraestrutura da Nuvem Computacional



Fonte: Miller (2008).

A computação em nuvem tem atraído a atenção e os investimentos de grandes empresas como, por exemplo, Google (*Google Cloud Platform*), Microsoft (*Microsoft Azure*) e Amazon (*AWS – Amazon Web Services*). Atualmente, essas empresas fornecem diferentes tipos de recursos, que incluem desde servidores para processamento, plataformas de desenvolvimento, aplicativos, etc. De acordo com os serviços oferecidos, essas nuvens computacionais podem ser classificadas em:

- infraestrutura como serviço (*IaaS – Infrastructure as a Service*): a nuvem disponibiliza recursos computacionais fundamentais como, por exemplo, recursos para processamento e armazenamento. Nesse caso, o usuário pode escolher os recursos desejados e configurar os mesmos. Entre os recursos computacionais disponíveis, pode-se citar: servidores de alto desempenho, computadores com múltiplos processadores e servidores com GPUs. Como exemplo de nuvem do tipo IaaS, pode-se destacar a *Amazon Elastic Compute Cloud* (Amazon EC2);
- plataforma como serviço (*PaaS – Platform as a Service*): a nuvem disponibiliza uma infraestrutura completa para o desenvolvimento e a hospedagem de aplicações. Neste caso os usuários podem utilizar a infraestrutura de forma rápida, sem se preocupar em adquirir, configurar e gerenciar recursos de *hardware* e *software*. Como exemplo desse tipo de nuvem, pode-se citar a *Google App Engine* e a *Azure Cloud Services*.
- *software* como serviço (*SaaS – Software as a Service*): a nuvem prove uma série de *softwares* que podem ser utilizada através da rede pelos usuários. Estes *softwares* podem variar, desde simples editores de texto até aplicações mais

complexas, como por exemplo, um ERP (*Enterprise Resource Planning*). O exemplo mais conhecido de nuvem do tipo SaaS é o *Google Docs*.

O uso da computação em nuvem é uma das tendências mais promissoras e, em decorrência disso, tem-se observado um aumento no número de plataformas, bem como de trabalhos que utilizam esse tipo de infraestrutura. Na área de bioinformática esse tipo de plataforma tem sido utilizado, principalmente, para o armazenamento e processamento de grandes volumes de dados (DAI *et al.*, 2012; ZHAO *et al.*, 2013).

Referências

- ALMASI, G. S.; GOTTLIEB, A. **Highly parallel computing**. Redwood City, CA, USA: Benjamin-Cummings Publishing Co., Inc., 1989.
- BUYA, R. **High performance cluster computing: architectures and systems**. Upper Saddle River, NJ, USA: Prentice Hall PTR, 1999.
- BUYA, R. *et al.* Cloud computing and emerging IT platforms: vision, hype, and reality for delivering computing as the 5th utility. **Future Generation Computer Systems**, v. 25, n. 6, p. 599-616, 2009.
- CHAPMAN, B.; JOST, G.; PAS, R. v. d. **Using OpenMP: portable shared memory parallel programming (scientific and engineering computation)**. [S.l.]: The MIT Press, 2007.
- DAI, L. *et al.* Bioinformatics clouds for big data manipulation. **Biology Direct**, [S.l.: s.n.], 2012.
- DONGARRA, J. *et al.* **Sourcebook of parallel computing**. [S.l.]: W.H. Freeman and Company, 2003.
- FERREIRA, F. *et al.* Computação paralela heterogênea aplicada a problemas das ciências e engenharias. **WSCAD-SSC**, SBC, 10 2013.
- FLOURI, T. *et al.* GapMis-OMP: pairwise short-read alignment on multi-core architectures. *In: ILIADIS, L. et al.* (Ed.). **8th International Conference on Artificial Intelligence Applications and Innovations (AIAI)**. Halkidiki, Greece: Springer, 2012.
- FOSTER, I. **Designing and building parallel programs: concepts and tools for parallel software engineering**. Boston, MA, USA: Addison-Wesley Longman Publishing Co., Inc., 1995.
- FOSTER, I. The anatomy of the grid: enabling scalable virtual organizations. *In: SAKELLARIOU, R. et al.* (Ed.). **Euro-Par 2001 Parallel Processing**. Berlin, Heidelberg: Springer Berlin Heidelberg, 2001. p. 1-4.
- FOSTER, I.; KESSELMAN, C. Globus: a metacomputing infrastructure toolkit. **The International Journal of Supercomputer Applications and High Performance Computing**, v. 11, n. 2, p. 115-128, 1997.
- FOSTER, I.; KESSELMAN, C. (Ed.). **The grid: blueprint for a new computing infrastructure**. San Francisco, CA, USA: Morgan Kaufmann Publishers Inc., 1999.
- GAGLIARDI, F. The EGEE european grid infrastructure project. *In: DAYDÉ, M. et al.* (Ed.). **High performance computing for computational science - VECPAR 2004**. Berlin, Heidelberg: Springer Berlin Heidelberg, 2005. p. 194-203.
- GROPP, W. *et al.* **Using MPI: portable parallel programming with the message-passing interface**. [S.l.]: MIT Press, 1999.
- GROPP, W.; HUSS-LEDERMAN, S.; SNIR, M. **MPI: the complete reference. The MPI-2 extensions**. [S.l.]: MIT Press, 1998.
- HENNESSY, J.; PATTERSON, D. **Organização e projeto de computadores: a interface hardware/software**. [S.l.]: Elsevier Editora, 2014.

- IOPE N. LEMKE, G. A. v. W. R. L. GridUNESP: a multi-campus grid infrastructure for scientific computing. *In: 3rd Latin American Conference on High Performance Computing (CLCAR 2010)*. [S.l.: s.n.], 2010. p. 76-84.
- KALEGARI, D. H.; LOPES, H. S. An improved parallel differential evolution approach for protein structure prediction using both 2D and 3D off-lattice models. *In: 2013 IEEE Symposium on Differential Evolution (SDE)*. [S.l.: s.n.], 2013. p. 143-150.
- KIRK, D.; HWU, W. **Programming massively parallel processors: a hands-on approach**. [S.l.: s.n.], 2010.
- LEWIS, B.; BERG, D. J. **Multithreaded programming with Pthreads**. Upper Saddle River, NJ, USA: Prentice-Hall, Inc., 1998.
- LI, K.-B. ClustalW-MPI: clustalW analysis using distributed and parallel computing. **Bioinformatics**, v. 19, n. 12, p. 1585-1586, 2003.
- MARTINOTTO, A. L. *et al.* Generation of continuous random networks by simulated annealing. *In: Proceedings of the 2008 ACM Symposium on Applied Computing*. New York, NY, USA: ACM, 2008. (SAC'08), p. 48-49.
- MELO, A. C. M. A. *et al.* Local DNA sequence alignment in a cluster of workstations: algorithms and tools. **Journal of the Brazilian Computer Society**, Scielo, v. 10, p. 73-80, 11 2004.
- MILLER, M. **Cloud computing: web-based applications that change the way you work and collaborate**. [S.l.]: Que Publishing Company, 2008.
- MOREANO, N.; MELO, A. Biological sequence analysis on GPU. **Advances in GPU Research and Practice**, p. 127-162, 12 2017.
- NEBRO, A. *et al.* DNA fragment assembly using a grid-based genetic algorithm. **Computers & Operations Research**, v. 35, n. 9, p. 2776-2790, 2008.
- NVIDIA. **About CUDA**. 2019. <https://developer.nvidia.com/about-cuda>. Acesso em: 14 abr. 2019.
- PROTIC, J. *et al.* **Distributed shared memory: concepts and systems**. [S.l.]: Wiley, 1998. (Systems Series).
- RANOK, U.; KITTITORNKUN, S.; TONGSIMA, S. A multithreading methodology with OpenMP on multi-core CPUs: SNPHAP case study. *In: The 8th Electrical Engineering/ Electronics, Computer, Telecommunications and Information Technology (ECTI)*. [S.l.: s.n.], 2011. p. 459-463.
- SANDES, E. F. de O. **Comparação paralela de sequências biológicas longas utilizando unidades de processamento gráfico (GPUs)**. 2011. Tese (Doutorado) - Universidade de Brasília, 2011.
- SATHE, S. R.; SHRIMANKAR, D. D. Parallelization of DNA sequence alignment using OpenMP. *In: Proceedings of the 2011 International Conference on Communication, Computing and Security*. New York, NY, USA: ACM, 2011. (ICCCS '11), p. 200-203.
- SILBERSCHATZ, A. **Sistemas operacionais com Java**. [S.l.]: Else-vier/Campus, 2008.
- SONG, Y.; YANG, S.; LEI, J. ParaCells: a GPU architecture for cell-centered models in computational biology. **IEEE/ACM Transactions on Computational Biology and Bioinformatics**, PP, p. 1-1, 03 2018.
- STALLINGS, W. **Arquitetura e organização de computadores**. [S.l.]: Editora Pearson Education, 2018.
- SUN, C.-H. *et al.* A model of problem solving environment for integrated bioinformatics solution on grid by using condor. *In: Grid and Cooperative Computing - GCC 2004*. Berlin, Heidelberg: Springer Berlin Heidelberg, 2004. p. 935-938.
- TANENBAUM, A. **Organização estruturada de computadores**. [S.l.]: Prentice Hall Brasil, 2013.
- THAIN, D.; TANNENBAUM, T.; LIVNY, M. Distributed computing in practice: the condor experience. **Concurrency and Computation: Practice and Experience**, v. 17, n. 2- 4, p. 323-356, 2005.

VAQUERO, L. M. *et al.* A break in the clouds: towards a cloud definition. **SIGCOMM Comput. Commun. Rev.**, ACM, New York, NY, USA, v. 39, n. 1, p. 50-55, dez. 2008.

WILKINS-DIEHR, N. *et al.* Teragrid science gateways and their impact on science. **Computer**, v. 41, n. 11, p. 32-41, nov. 2008.

ZHAO, S. *et al.* Rainbow: a tool for large-scale whole-genome sequencing data analysis using cloud computing. **BMC Genomics**, v. 14, n. 1, p. 425, jun. 2013.

ZHOU, W. *et al.* A Multi-GPU protein database search model with hybrid alignment manner on distributed GPU clusters: multi-GPU hybrid search for protein database on clusters. **Concurrency and Computation: Practice and Experience**, v. 30, p. e4522, 2018.

Seção II – Aplicações

8

PORTAIS E BANCOS DE DADOS BIOLÓGICOS

Gustavo Sganzerla¹

Toda técnica/aplicação computacional surge primeiro de uma necessidade, onde uma limitação é encontrada e depois a solução do problema é transposta através de uma técnica/aplicação. Tomamos por exemplo a empresa Uber, que por sua vez, revolucionou a forma como a mobilidade urbana ocorreria. A empresa que hoje é exemplo da chamada “revolução digital” iniciou em 2009 com Garret Camp, um dos fundadores gastando cerca de 800 dólares em um motorista particular. Isso levou Camp a buscar modos de diminuir o custo de transporte direto, onde de acordo com o fundador, o custo tende a diminuir quando compartilhado entre empresa, motorista e passageiro. A limitação encontrada por Camp era o alto custo e a baixa disponibilidade oferecida pelos modais de transporte que existiam na época.

Quando uma limitação é somada com uma mente criativa e com conhecimentos de computação e programação e hoje em dia temos uma revolução de como as pessoas se locomovem diariamente em grandes centros urbanos. A lista de exemplos é enorme, mas uma grande parte dos avanços computacionais tendem a surgir de uma limitação atual, onde a própria automação do processo poderá economizar recursos para os envolvidos (CRAMER; KRUEGER, 2016).

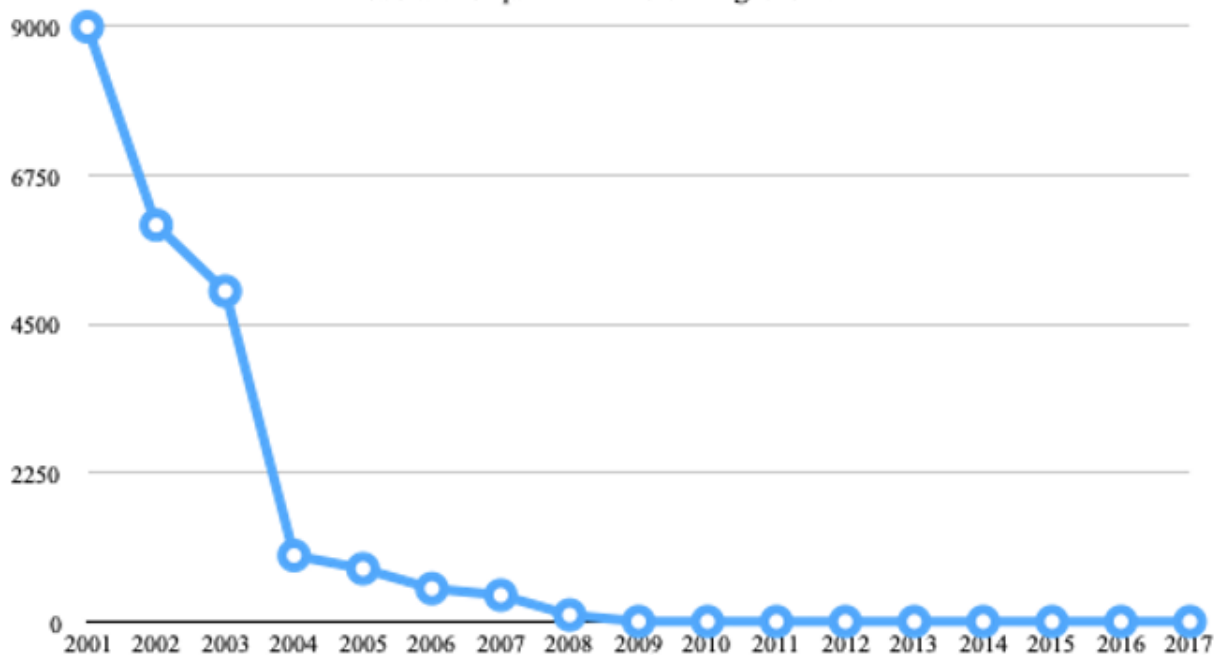
Quando a área em questão é a bioinformática e seus avanços, as diferenças não são gritantes quando comparadas ao Uber. Quando a questão é os avanços na área de bioinformática, há uma constante troca de protagonismo. Os dois protagonistas são os sensores e as técnicas computacionais. Essa constante mudança no protagonista pode ser exemplificada como: imaginemos que exista uma limitação ao extrair dados genéticos de um organismo x . Um novo sensor mais aprimorado deve então ser criado para poder coletar dados mais detalhados e de maneira mais robusta. Até então o protagonismo de nossa anedota está na mão dos engenheiros que desenvolveram o sensor, porém, após essa etapa ser bem-sucedida, indagar-se-ão sobre o que fazer com a nova grande quantidade de dados agora disponíveis. Surge outra limitação, que é seguida de um avanço. E o protagonismo muda novamente, necessitando o desenvolvimento de técnicas capazes de trabalhar com dados mais detalhados e em maior quantidade; e o ciclo continua.

Tomamos como exemplo o custo para sequenciar DNA, onde atualmente pode-se analisar uma grande diminuição no custo desse procedimento que faz o uso de técnicas de biotecnologia. A série histórica do valor pode ser observado na Figura 1. Onde em

¹ Universidade de Caxias do Sul. *E-mail*: sganzerlagustavo@gmail.com

2001, o custo era aproximadamente de 10 mil dólares por megabase de sequência de DNA, em 2017 o mesmo procedimento custava menos de 1 dólar. O baixo custo em sequenciamento levou a popularização de sensores tecnológicos capazes de extrair uma grande quantidade de informação, que por sua vez aumentou a disponibilidade de bancos de dados e técnicas para trabalhar com tais dados.

Figura 1 – Linha temporal de custo de sequenciamento
Custo de sequenciamento de megabase de DNA



Fonte: Adaptado de Wetterstrand DNA Sequencing Costs: Data from the NHGRI Genome Sequencing Program.

Essa maior disponibilidade alavancada pelo baixo custo ao produzir dados referentes às ciências da vida cria a necessidade de que existam bancos de dados biológicos, podendo assim armazenar a coleta de dados feita por outras áreas e prover acesso para que o conhecimento biológico cresça. A próxima subseção explorará os bancos de dados biológicos mais distintos e utilizados no meio científico (HIRSCHMAN *et al.*, 2012).

1 Bancos de dados biológicos

Na ciência da computação, bancos de dados tratam de coleções de arquivos que representam abstrações do mundo real. A finalidade de bancos de dados é prover uma forma eficiente de pesquisa através das coleções que o banco em questão disponibiliza. Na biologia, existem inúmeros bancos de dados com coleções de arquivos sobre as

ciências da vida, proveniente de experimentos científicos, literatura publicada e análises computacionais. É comum encontrar bancos de dados dedicados em áreas como: genômica (genoma de organismos), proteômica (proteínas), metabolômica (metabólitos produzidos por organismos) e transcriptômica (RNA). Em relatos do Nucleic Acid Research Database, em 2011 existiam mais de 1300 bancos de dados biológicos (GALPERIN; COCHRANE, 2011).

Bancos de dados biológicos existem para diversas finalidades, apresentam com dados dos mais variados organismos e são submetidos a processos diferentes de curadoria. A classificação de bancos de dados é definida por: *i*) o escopo dos dados, onde podem haver bancos de dados que abrangem diversas espécies e bancos de dados mais localizados, focando em organismos específicos. *ii*) o nível de curadoria, dividindo-se em primário, onde os dados encontram-se de maneira crua e os bancos de dados secundários, onde a informação presente passou por algum processo de curadoria e tem valor agregado. *iii*) os métodos de curadoria, onde os bancos de dados podem ter sido submetidos à avaliação de *experts* ou de uma comunidade de colaboradores; *iv*) e por fim, a última etapa de classificação de bancos de dados biológicos conta com o tipo de dado gerido, por exemplo DNA, RNA, proteínas, expressão ou doenças. A Figura 2 retrata uma representação de como bancos de dados biológicos são classificados (ZOU *et al.*, 2015).

Figura 2 – Representação de classificação de bancos de dados biológicos



Fonte: Adaptado de ZOU *et al.*, 2015.

Não existe um consenso no que diz respeito da classificação de bancos de dados biológicos, em soma ao modelo previamente apresentado, os autores Gaudet *et al.*

(2011) lançaram uma forma de classificar bancos de dados biológicos onde eles são definidos como: *i*) repositórios de arquivamento, onde há apenas dados em sua forma crua, no modelo proposto por Zou *et al.* (2015) esse tipo de banco de dados seria classificado como nível de curadoria primário; *ii*) recursos curados e; *iii*) armazém de integração, que compõem uma hierarquia mais complexa na definição de bancos de dados e possuem dados padronizados provenientes de distintas fontes (GAUDET *et al.*, 2011).

No que se diz a respeito de curadoria de bancos de dados, deve-se levar em consideração o quão confiável é o nível de anotação da sequência em questão. Existem bancos de dados que contam com a entrada de dados, e sua curadoria é apenas verificada computacionalmente, há outros, onde existe a interferência humana ao indicar que tal informação no banco de dados é comprovada através da inferência humana como verdadeiro. A curadoria de dados depende de um esforço conjunto entre pesquisadores, instituições e revistas visando criar modos para facilitar a troca de dados entre repositórios distintos. Um grande exemplo de banco de dados que conta apenas com sequências referência é o RefSeq, o qual conta com dados genéticos, de transcrição e proteômicos de mais de 80 mil organismos. Todo o repositório disponibilizado é denominado como bem anotado, ou seja, a informação presente não é redundante e apresenta um grande teor de confiabilidade (HOWE *et al.*, 2008).

Ainda sobre a curadoria de informação biológica, pesquisadores da área constantemente deparam-se com o gargalo apresentado pela diferença entre curadoria feita por máquinas e curadoria feita por humanos. Nem sempre uma análise textual feita por seres automatizados apresentará o mesmo nível de confiabilidade do que uma análise feita por humanos. Tomamos como exemplo o banco de dados RegulonDB, onde as sequências promotoras bacterianas reconhecidas pelo fator sigma 24 de *Escherichia coli* contém em grande parte dados apenas inferidos computacionalmente sem análise humana, a precisão desses elementos regulatórios tende a ser menor do que outros grupos de promotores presentes no banco de dados que passaram por uma forma de curadoria humana. E isso é visto em diversas aplicações da substituição de humanos por máquinas, onde no estágio atual da tecnologia, humanos ainda são imprescindíveis. Considerando isso, incluímos no processo de biocuradoria alguns pontos que devem ser relevados para buscar uma maneira mais eficiente de tirar proveito dos dados disponíveis (HIRSCHMAN *et al.*, 2012; CHRISTIAN, 2013; SHIMADA, 2017). São eles:

- triagem, onde pessoas encontram informação tida como relevante em artigos científicos;

- identificação e normalização de entidades biológicas, onde os dados passam por uma detecção. Onde dividem-se em genes, proteínas, moléculas;
- anotação de detecção de eventos, tais como a anotação de interação entre proteínas, caracterização de produtos genéticos em termos de sua localização molecular, função molecular, efeito fenótipo;
- associação de evidência qualificadora, onde os dados são associados com evidências testadas experimentalmente;
- gravação no banco de dados.

Desta forma, o modelo apresentado no *framework* (HIRSCHMAN *et al.*, 2012) é capaz de agir como suporte útil aos biocuradores (HIRSCHMAN *et al.*, 2012). Adicionalmente, HOWE *et al.* (2008) propuseram uma série de papéis que devem ser desempenhados por biocuradores – o termo usado pelos autores referindo-se a quem promove a curadoria de dados biológicos. Entre o que foi levantado pelos autores, podemos destacar:

- extração de conhecimento de artigos já publicados;
- realizar a conexão de informação proveniente de diferentes fontes de modo compreensivo e coerente;
- inspecionar e corrigir estruturas genéticas e proteínas;
- desenvolver e controlar índices que são cruciais para relacionar dados e recuperar grandes quantidades de dados;
- integrar bases de conhecimento, visando a representação de sistemas complexos como a rede de interação entre proteínas;
- corrigir inconsistências e erros na representação de dados;
- assistir usuários de dados, construindo sua pesquisa em um modo eficiente em questões de tempo;
- guiar o design de recursos baseados na *web*;
- interagir com pesquisadores para facilitar a submissão direta de dados em bancos de dados.

Como visto anteriormente, o crescimento de bancos de dados biológicos nos últimos anos deu-se pelo desenvolvimento de novas técnicas de coletar dados e os baixos custos ao obter esses dados. Existem várias formas de bancos de dados, não fazendo nenhuma restrição quanto a arquivos de uma única espécie ou dados de um único tipo. Coleções de arquivos podem variar muito, contendo exemplos de RNA, DNA, proteínas, elementos regulatórios.

Independente do tipo de dado presente no banco de dado biológico, nas últimas décadas foi possível analisar um crescimento exponencial na disponibilidade desses dados. Um exemplo disponibilizado na Figura 3 demonstra o alto crescimento de

número de sequências no banco de dados Uniprot Knowledgebase (UniProtKB). A curva demonstrada nesse gráfico indica um crescimento (n^x), sendo não linear. A riqueza desses dados proporciona que pesquisadores possam responder questões complexas e produzir novas descobertas científicas, e isso não se restringe apenas a proteínas, de modo que, o crescimento dos bancos de dados é visto em todas as áreas (CHRISTIAN; GRIFFITHS, 2016)

Neste ponto, já exploramos de forma mais conceitual como bancos de dados funcionam na área das ciências da vida e foi possível observar que existem bancos de dados de diversos segmentos de vários organismos. Os bancos de dados também apresentam uma variação na confiabilidade dos dados que eles armazenam, e isso deve ser relevado pelos autores e pesquisadores que estão trabalhando com os dados. Visando sempre a obtenção de resultados bastante confiáveis e condizentes com a proposta inicial dos cientistas (SHIMADA *et al.*, 2017).

A Tabela 1 representa diversos bancos de dados que são bastante difundidos no meio científico e acadêmico atual, o tipo de dados que esses repositórios contém e também a quantidade de dados disponibilizado pelos criadores, um link para acesso. A construção da Tabela 1 levou em consideração a apresentação de uma gama variada de organismos, evitando apresentar vários bancos de dados que apresentam repositórios sobre o mesmo organismo.

Figura 3 – Crescimento de sequências de proteínas no banco de dados UniProt

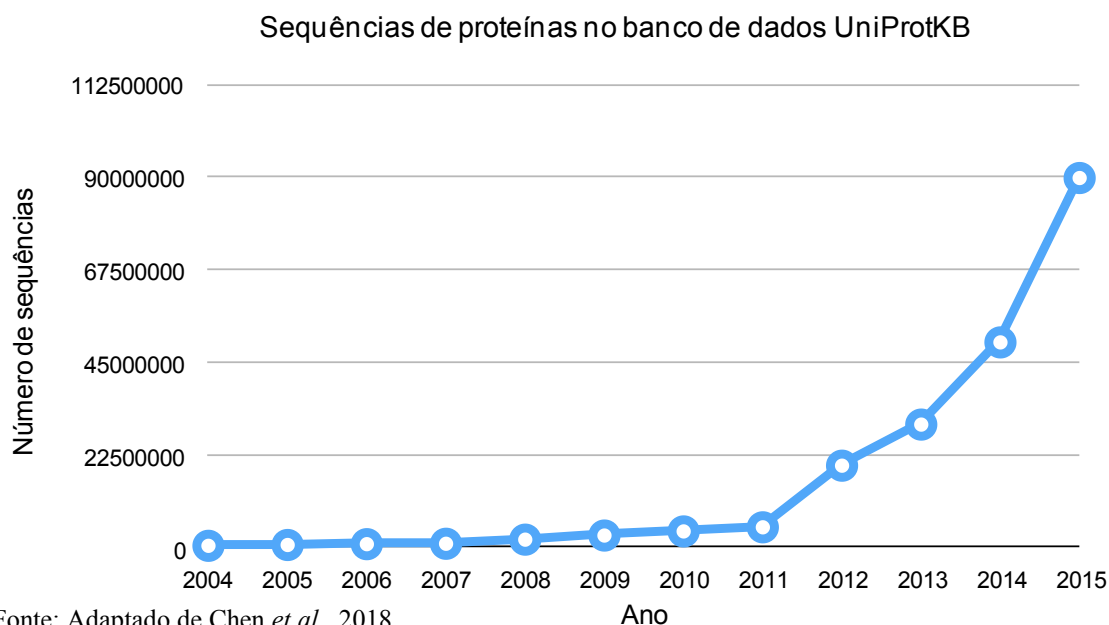


Tabela 1

Banco de dados	Tipo de dados	Quantidade	Link
GenBank¹	DNA de vários organismos	Mais de 179 milhões de sequências	https://www.ncbi.nlm.nih.gov/genbank/
UniProt	Proteínas	559.228 entradas	https://www.uniprot.org/
RegulonDB	Rede regulatória de <i>E. coli</i> K12	Trata-se apenas de <i>E. Coli</i>	http://regulondb.ccg.unam.mx/index.jsp
RNA Central	RNA de vários organismos	14.476418 entradas	https://rnacentral.org/
DDBJ¹	DNA	1.762.943 sequências	https://www.ddbj.nig.ac.jp/index-e.html
EMBL¹	Sequências de nucleotídeos	2.218.3 milhões de sequências	https://www.ebi.ac.uk/ena
Rice Wiki	Genes de arroz	86.216 genes	http://wiki.ic4r.org/index.php/Main_Page
Worm Base	Genômica de nematoides	Não divulgado	https://wormbase.org/#012-34-5
TAIR	Dados genéticos da planta <i>Arabidopsis thaliana</i>	Trata-se apenas de <i>Arabidopsis thaliana</i>	https://www.arabidopsis.org/
RFAM	RNA de diversas famílias	3016 famílias	http://rfam.xfam.org/
DISGENET	Doenças associadas com genes (DAG)	628.685 DAGs	http://www.disgenet.org/
Enzyme	Nomenclatura de enzimas	4477 entradas ativas	https://enzyme.expasy.org/
Flybase	Genes e genoma do organismo <i>Drosophila</i>	224304 referências	http://flybase.org/
GeneCards	Genes humanos	152.490 genes anotados	https://www.genecards.org/
PDBsum	Estruturas 3D de proteínas	153.222 entradas	http://www.ebi.ac.uk/thornton-srv/databases/cgi-bin/pdbsum/GetPage.pl?pdbcode=index.html
Expression Atlas	Informação sobre expressão gênica	Mais de 3000 experimentos curados	https://www.ebi.ac.uk/gxa/home
Gramene	Plantas	2.162.056 genes em 58 genomas	http://www.gramene.org/

¹ Os três bancos de dados fazem parte do International Nucleotide Sequence Database (INSD), consistindo em três bancos de dados primários.

Fonte: Elaboração do autor.

Até então, abordamos os mais diversos tipos de bancos de dados biológicos junto a sua classificação, que pode ser bastante variável. A recente explosão que pode ser vista na quantidade de dados exigiu uma forma organizada e indexada para organizá-los de forma conectada. Gaudet *et al.* (2011) afirmaram que a existência de uma grande gama de banco de dados em determinada área não indica necessariamente o aumento de conhecimento biológico. Os autores ainda afirmaram que a existência de bancos de dados biológicos ao redor do mundo deve conter alguma forma de conexão para evitar redundância e conseguir agir como um diferencial para entregar à uma extensão de

ferramentas os dados corretos a serem trabalhados. No momento em que há uma padronização e interconexão entre variados bancos de dados elimina-se custos desnecessários, acresce a interoperabilidade entre recursos, e mitiga-se o desperdício de dados e anotação quando um recurso não tem mais suporte. Entregando assim aos usuários a capacidade de ter suas necessidades atendidas, com o recurso correto localizado, combinando dados de diferentes fontes. A uniformização de um sistema descrevendo bancos de dados biológicos beneficiaria usuários e as equipes de desenvolvimento dos repositórios.

Agora que a coleta de dados já ocorreu, e os dados encontram-se disponíveis em bancos de dados, surge o segundo personagem de nossa troca de papéis constante mencionada no início deste capítulo. Cabe então aos pesquisadores desenvolver técnicas capazes de trabalhar com uma robustez maior de dados, podendo assim, entregar ao campo biológico uma pesquisa contundente e esclarecedora. O próximo subcapítulo explorará portais de biotecnologia, onde os quais empregarão diversas técnicas computacionais visando a obtenção de informação sobre os dados já obtidos e armazenados nos bancos de dados.

2 Portais biológicos

Podemos definir um portal biológico como um portal que permite o acesso às ferramentas de bioinformática, as quais geralmente são baseadas em web. Com o passar do tempo, diversas aplicações foram surgindo, promovendo a resolução de problemas biológicos. Vários dos bancos de dados explorados na última seção disponibilizam ferramentas para trabalhar com seus dados. Existem algumas demandas quanto ao que um portal de biologia deve considerar que são constantemente alteradas de acordo com o avanço tecnológico. Algumas delas compreendem:

- A possibilidade de funcionar em multiplataformas, onde atualmente, a variedade de sistemas que as pessoas usam no dia a dia são maiores do que no passado, isso exige a existência de portais que possam rodar em diferentes sistemas operacionais.
- Em grande maioria, portais biológicos devem contar com um grande poder de processamento computacional em virtude ao tamanho das bases de dados que são trabalhadas.
- O acesso a dados heterogêneos, onde o formato dos dados pode variar devido a não padronização de bancos de dados.

Muitos portais de biologia contam com algoritmos que são empregados para a resolução de um determinado problema. Um algoritmo presente e implementado por diferentes portais de biologia é o BLAST, que será explorado a seguir.

2.1 Basic Local Alignment Search Tool – BLAST (ALTSCHUL *et al.*, 1990)

O algoritmo BLAST, bastante difundido no meio de pesquisa biológico trata de encontrar similaridades entre sequências. O algoritmo compara sequências de nucleotídeos ou proteínas e estatisticamente calcula a significância das correspondências. Há outros algoritmos que são denominados de Global Alignment, onde fazem uma varredura global, a varredura realizada pelo BLAST é localizada. BLAST é comumente utilizado para identificar organismos e seu grau de relação com outros, por exemplo, ao comparar uma sequência de nucleotídeos pertencente a humanos, o algoritmo BLAST pode encontrar correspondências em genomas de chimpanzés, gatos, ratos e outras espécies onde exista similaridade entre os genomas. O BLAST também consegue captar aspectos evolucionários e funcionais de genes.

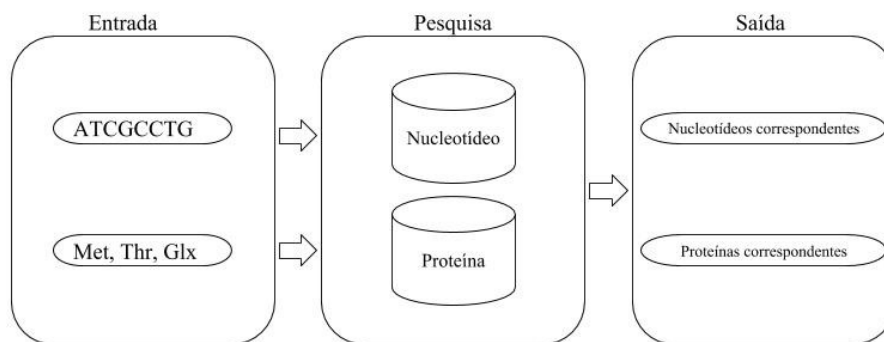
Existem diferentes implementações do algoritmo BLAST (patenteado pelo National Library of Medicine), essas variações compreendem diversos programas que fazem parte da família BLAST e, embora seu modo de operação seja similar, sua finalidade é distinta. Existem outros *websites* de bancos de dados que implementam também o algoritmo BLAST:

1. BLASTn (alinhamento nucleotídeo – nucleotídeo): realiza uma comparação de uma sequência de nucleotídeos informada pelo usuário e retorna a (s) sequência (s) de DNA mais similar (es).
2. BLASTp (alinhamento por proteínas): a entrada do usuário é uma proteína, o BLASTp faz uma varredura em banco de dados de proteína e retorna a proteína mais similar encontrada.
3. BLASTx (nucleotídeo – proteína): a partir de uma sequência de nucleotídeos, o alinhamento retorna uma proteína que é sintetizada pelo gene de entrada.
4. tBLASTn (proteína – nucleotídeo): o inverso do BLASTx, nesta versão, uma proteína é inserida e a varredura é feita para encontrar a sequência de nucleotídeos do gene que gerou a proteína de entrada.
5. tBLASTn: a comparação é feita através de uma proteína com todas as possíveis janelas de tradução da proteína em questão, tornando-se uma pesquisa mais robusta de modo que nenhuma tradução é ignorada (mais detalhes no funcionamento do algoritmo BLOSUM).
6. tBLASTx: o comparativo acontece com 6 possíveis formas de leitura no processo de entrada de dados e as 6 possíveis formas de armazenamento no banco de dados de proteína (6x6 janelas de leitura), caracterizando um

processo muito extensivo, onde pesquisas incompletas e/ou executadas erroneamente podem levar um sistema a parada.

A Figura 4 demonstra a forma com que o algoritmo BLAST trabalha, onde a partir de uma pesquisa através de sequências de nucleotídeos pode-se obter a proteína codificada pelo gene em questão, ou vice-versa, onde partindo de uma proteína, a ferramenta entrega o gene que foi traduzido.

Figura 4 – Representação de formas distintas de alinhamento de sequências no algoritmo BLAST



Fonte: Elaboração do autor.

Podemos então listar duas grandes implementações de BLAST, partindo de proteínas e de sequências de nucleotídeos. As etapas intermediárias ocorrem quando uma sequência ATCGCCGT precisa ser traduzida em uma proteína e vice-versa. Porém o alinhamento em si depende de algoritmos que serão explicados a seguir.

1. Teoria de Alinhamento de Sequências

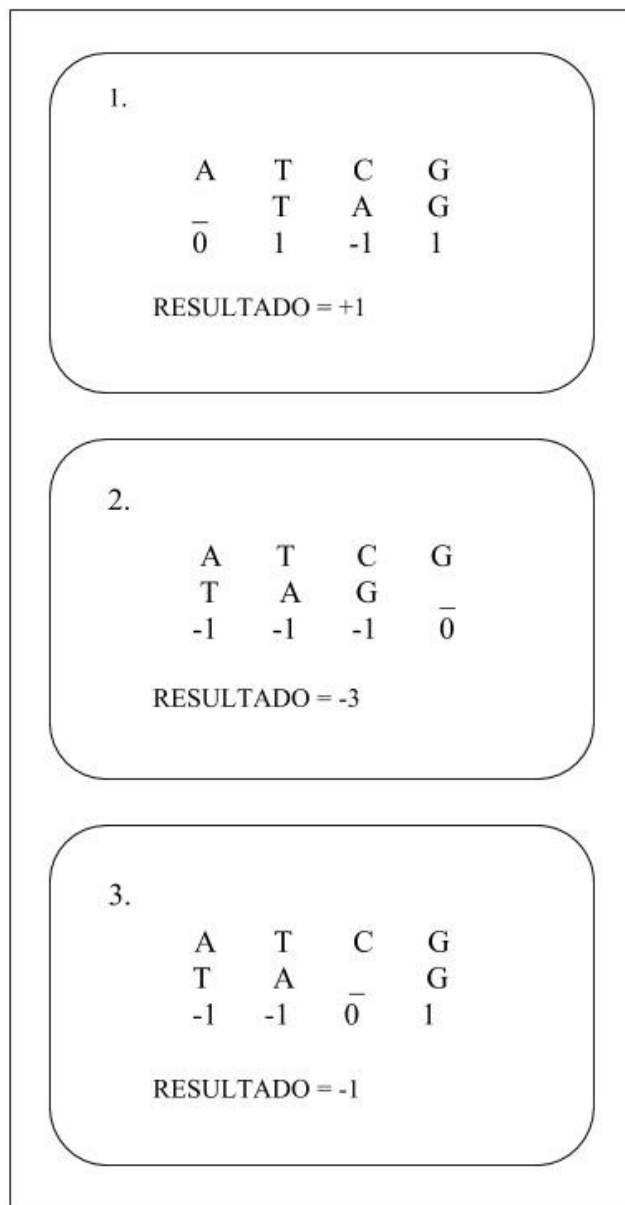
Imaginemos um cenário onde foi descoberto um gene *r* em ratos causando certa doença, os pesquisadores envolvidos nesse estudo desejam saber se o mesmo gene *r* está presente em humanos (homólogo) e pode ser estudado em uma espécie diferente. Obviamente, o genoma do rato contém diferenças quando comparado ao genoma humano, então podem haver diferenças ao buscar o gene *r* em humanos tais como: inserção ou deleção de nucleotídeos, alterações e mutações causadas pela evolução. Para sanar este problema, foi introduzida a técnica de alinhamento de sequências. Uma forma comum de trabalhar com alinhamento de sequências é através da adição de *gaps* (□), que ocorrem quando as mesmas sequências não têm o mesmo tamanho:

```

ATCGCGCGA      ATCGCGCGA
ATCCGCGA       ATC_□GCGA
  
```

Ao adicionar o *gap*, as sequências são idênticas e possuem o mesmo tamanho. A forma com que um algoritmo de alinhamento trabalha é através de atribuição de pontos para cada correspondência (*match*), *gaps* (inserção ou deleção de nucleotídeo) ou uma não correspondência (*mismatch*). Como grande parte das sequências a serem alinhadas não tem o mesmo tamanho, as possibilidades de alinhamento são variadas, o algoritmo então busca pela forma ótima de alinhar, de modo que a pontuação atribuída a *matches*, *gaps* e *mismatches* seja a mais alta possível. Tomamos um exemplo onde: *match* = 1, *gap* = 0 e *mismatch* = -1, e alhamos 3 vezes as mesmas sequências. A representação na Figura 5 mostra que dos 3 alinhamentos realizados, o primeiro é o mais favorável, que apresentou uma pontuação mais alta (LOBO, 2008).

Figura 5 – Exemplo de alinhamento de sequências com pontuação atribuída



Fonte: Elaboração do autor.

2. Block Substitution Matrix (BLOSUM)

Ao realizar o alinhamento de proteínas, temos uma estrutura matemática um pouco mais complexa do que o que fora apresentado pelo alinhamento de sequências de nucleotídeos. O alfabeto de nucleotídeos conta com apenas 4 letras (A, T, C e G). Porém, proteínas contam com combinações que derivam de 20 aminoácidos distintos, o que torna a tarefa de alinhamento carente de uma forma mais complexa para obter o melhor alinhamento possível. Aminoácidos distintos ainda apresentam propriedades químicas correspondentes, logo, a atribuição de *matches*, *mismatches* e *gaps* seria uma forma muito simplória de alinhar sequências. O algoritmo BLOSUM62 trabalha com atribuição de pontos, onde um *mismatch* pode ter uma pontuação maior do que outro *mismatch* (ALTSCHUL *et al*, 1990). A forma com que esse algoritmo funciona é através da contagem da ocorrência de um par de aminoácidos. É possível destacar os passos principais ao utilizar uma matriz de substituição BLOSUM, são eles:

- a. Contar a frequência de aminoácidos individualmente;
- b. Contar a frequência de pares de aminoácidos;
- c. Contar a frequência observada de pares de aminoácidos;
- d. Contar a frequência esperada de pares de aminoácidos;
- e. Calcular o logaritmo (base 2) da taxa de frequência esperada x observada.

2.2 Preditores

Outra forma de ferramentas biotecnológicas é os preditores. O propósito de ferramentas dessa espécie é prever algo sobre determinado dado que foi inserido como entrada na ferramenta. A grande maioria dos preditores empregam técnicas de inteligência artificial, por exemplo: redes neurais artificiais, máquinas de suporte vetorial. A predição pode ocorrer em genes, elementos regulatórios, redes de proteínas, ou seja, predições podem ser feitas em diversos campos da biologia molecular.

Um exemplo de preditor é o Bacterial Promoter Predictor (BacPP), a ferramenta faz o uso de Redes Neurais Artificiais para formular padrões sobre sequências promotoras de diferentes grupos da bactéria *Escherichia coli*. Após o treino da Rede Neural, é possível algoritmizar o aprendizado formulado para distinguir sequências promotoras de sequências não promotoras. O BacPP tem uma taxa de 90% de precisão ao identificar e prever promotores bacterianos de *E. coli*.

Há também outros preditores de regiões regulatórias, a grande maioria deles faz o uso de técnicas de inteligência artificial e contam com uma alta precisão ao prever promotores. A tabela 2 indica alguns exemplos de preditores de promotores, com o

organismo trabalhado, a forma com que os promotores são encontrados pela ferramenta e o nível de precisão que essa ferramenta apresenta ao identificar as regiões promotoras.

Tabela 2 – Relação de diferentes preditores de promotores

Preditor	Organismos	Como a ferramenta encontra promotores?	Precisão
bTSS Finder	<i>E. coli</i> Cyanobacteria	Presença e distância de elementos promotores Pontuação de oligômeros Densidade	89.22% (<i>E. coli</i>) 79.26 % (Cyanobacteria)
BacPP	<i>E. coli</i>	Presença de características físico-químicas Redes neurais são treinadas com exemplos de promotores e não promotores, então regras biológicas são extraídas	$\sigma_{24} = 86.9\%$; $\sigma_{28} = 92.8\%$; $\sigma_{32} = 91.5\%$; $\sigma_{38} = 89.3\%$; $\sigma_{54} = 97.0\%$; $\sigma_{70} = 83.6\%$
Promoter 2.0	Promotores eucariontes polll	Uma combinação de elementos similares em redes neurais e algoritmos genéticos reconhece um conjunto discreto de sub-padrões	Coefficiente de correlação = 0.63
PromH	Promotores humanos	Estatisticamente identifica regiões conservadas no genoma humano Calcula a diferença de estabilidade entre promotores e regiões codificantes através de uma divisão da sequência em janelas sobrepostas de 15 nucleotídeos	TATA boxes = 70% TSS = 80%
Prompredict	<i>E. coli</i>		Sensibilidade = 90%
CNN promoter	Humanos Ratos <i>Bacillus subtilis</i> <i>Arabidopsis thaliana</i> <i>E. coli</i>	Analisa promotores de procariontes e eucariontes. A classificação é feita via uma rede neural coevolucionária e uma abordagem de <i>deep learning</i>	Sensibilidade 90% Especificidade 96% Coefficiente de correlação 0.84

Fonte: Elaboração do autor.

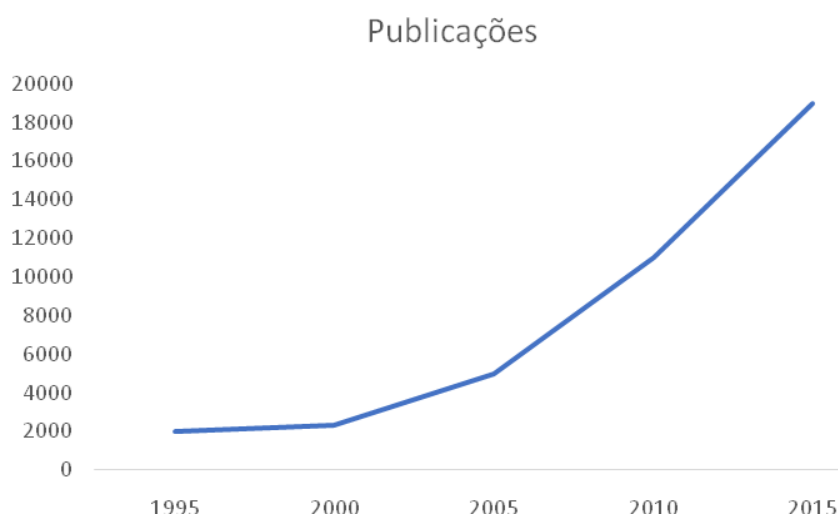
Há também os preditores de proteínas, que utilizam técnicas de inteligência artificial visando a predição de estruturas de proteínas: *i*) primárias, tratando da pura sequência de aminoácidos; *ii*) secundária, que compreende a ligação de sequências de aminoácidos conectadas por pontes de hidrogênio; *iii*) terciárias, que ocorrem quando as atrações entre proteínas está presente entre hélices alfa e folhas beta, e por fim; *iv*) as estruturas quaternárias, onde todo um conjunto de cadeias de aminoácidos formam a proteína em si. A predição pode consistir em qualquer uma dessas quatro etapas, sendo que sua complexidade cresce de acordo com a estrutura. Alguns preditores de proteínas bastante difundidos no meio da pesquisa em bioinformática são: phyre (<http://www.sbg.bio.ic.ac.uk/phyre2/html/page.cgi?id=index>), Garnier

(<http://www.bioinformatics.nl/cgi-bin/emboss/garnier>), SWISS-MODEL (swissmodel.expasy.org), I-TASSER (<https://zhanglab.ccmb.med.umich.edu/I-TASSER/>). Essas ferramentas têm diferentes formas de prever diferentes estruturas de proteínas, e contam com maneiras de modelar estruturas proteicas, por exemplo o SWISS-MODEL trabalha com um web-server onde a estrutura de proteína é automaticamente modelada.

Por fim, existem também preditores de regiões terminadoras, que semelhantemente aos promotores, atuam como sinalizadores da transcrição gênica, porém diferentemente dos promotores, as sequências terminadoras avisam a enzima RNA polimerase aonde termina o gene a ser transcrito em um RNA mensageiro. A forma com que a predição de terminadores ocorre é semelhante aos promotores, devido ao maior nível de conservação dos nucleotídeos apresentados nessa região intergênica. O *software* I2BC (<http://rssf.i2bc.paris-saclay.fr/>) é capaz de encontrar terminadores, outro exemplo é o Findterm, programa destinado a encontrar terminadores bacterianos (<http://www.softberry.com/berry.phtml?topic=findterm&group=help&subgroup=gfindb>).

A área de predição de regiões genômicas beneficia-se de grandes avanços que estão ocorrendo na inteligência artificial. A evolução da inteligência artificial pega carona no fato de que mais e mais máquinas capazes de processar grandes quantidades de dados tornando-se mais disponíveis e baratas. No decorrer do tempo, podemos ver a inteligência artificial recebendo mais investimento, recebendo uma quantia maior de publicação artigos científicos atrelados à área, como a Figura 6 demonstra. Em ambas figuras podemos ver um crescimento exponencial, relacionando o número de publicações que cresceu com maiores investimentos na área.

Figura 6 – Crescimento da publicação de artigos científicos relacionados à inteligência artificial e o crescimento de investimento na área



Fonte: Forbes, 2018).

Desta forma, podemos esperar grandes avanços futuros, onde mais ferramentas preditoras devem surgir e elas poderão ser capazes de trabalhar com grandes quantidades de dados de diferentes organismos e apresentar uma alta taxa de precisão em suas previsões.

2.3 Portais gerais

Alguns dos portais que serão explorados nesta seção representam uma coleção de recursos de bioinformática, fazendo com que alguns deles apresentem o banco de dados do organismo em si e ferramentas utilizadas para trabalhar com os dados disponibilizados.

O Kyoto Encyclopedia of Genes and Genomes ou KEGG é um banco de dados com recursos voltados a compreensão e funcionalidade de sistemas biológicos. O KEGG conta com web services voltados a manipulação de dados biológicos com diversas finalidades, primeiramente, iremos explorar as diferentes coleções de dados apresentados pelo KEGG, são elas:

- KEGG Pathway, onde o banco de dados apresenta uma coleção manualmente desenhada com a interação de moléculas em diversas redes, como: metabolismo, informação genética, processos celulares, desenvolvimento de drogas, doenças humanas.
- KEGG Brite, onde o website apresenta uma coleção manualmente criada de arquivos demonstrando hierarquias de vários objetos biológicos, diferentemente do KEGG Pathway, onde a coleção apresentada é apenas entre relações e interações moleculares. As relações entre objetos biológicos compreendidas pelo KEGG Brite contam com genes, proteínas, compostos, drogas, doenças, organismos e células.
- KEGG Orthology que é um banco de dados de funções moleculares representadas em genes ortólogos, onde espécies diferentes apresentam um gene diferente em termos de código genético, porém a proteína produzida a partir dos dois genes distintos terá função similar mesmo em diferentes espécies. Através de interações moleculares feitas através do KEGG Pathway, é possível encontrar genes e proteínas em diferentes organismos através do KEGG Orthology.
- KEGG Enzyme conta com uma implementação de um sistema de implementação de nomenclatura de enzimas provenientes do comitê biomédico de nomenclatura IUBMB/IUPAC.

Existem outros repositórios mais específicos relacionados a doenças, drogas, reações, moléculas pequenas, relação entre doenças. Atuando ao mesmo tempo como um banco de dados biológico e um portal biológico, o KEGG conta com ferramentas de análise, são estas:

- KEGG Mapper, que compreende ferramentas de mapeamento baseadas nas coleções KEGG Pathway e KEGG Brite.
- BlastKOALA trata-se de uma implementação de BLAST, porém somente sobre genes presentes no banco de dados KEGG. A limitação dessa ferramenta, em oposição ao GhostKOALA que será explorado a seguir é o tamanho do dataset aceito como entrada, sendo mais favorável para anotação genômica de alta qualidade.
- GhostKOALA implementa uma ferramenta que é capaz de aceitar um dataset de entrada maior do que o BlastKOALA, sendo utilizado em situação de anotação de metagenomas.
- KofamKOALA, onde a busca de similaridades em objetos biológicos presente no banco de dados KEGG é dada através de genes ortólogos.
- As próprias implementações do KEGG nos algoritmos de busca de similaridades BLAST e FASTA entre sequências presentes no KEGG.
- SIMCOMP, onde a busca de similaridades é através da estrutura química de objetos KEGG.

Diferentemente do KEGG, onde banco de dados/portal apresenta dados e ferramentas referente a genes e genomas, o Universal Protein Resource (UniProt) é um portal biológico direcionado a informação sequencial e funcional de proteínas. Seguindo a mesma linhagem dos outros exemplos vistos anteriormente, a quantidade de dados disponibilizados pelo UniProt teve uma grande explosão entre os anos de 2005 e 2010, onde os dados partiram de aproximadamente 200,000 para beiras os 500,000 apenas em 5 anos.

Com toda essa coleção de proteínas disponível, o que torna o UniProt uma referência na comunidade científica quando o assunto é coleção de proteínas e ferramentas para trabalhar com elas, o UniProt conta com as ferramentas: BLAST, o mesmo algoritmo visto anteriormente, utilizado para encontrar similaridades entre proteínas; Align, uma ferramenta de alinhamento de duas ou mais sequências de proteínas, buscando uma visão global das características das proteínas umas ao lado das outras. A Figura 7 é um exemplo do alinhamento da proteína TAP de humanos e porcos. Pode-se perceber que a ferramenta de alinhamento separou as cadeias de aminoácidos em janelas de 60, para facilitar a visualização, e ao comparar um aminoácido com outro,

Como demonstrado pela Tabela 3, existe uma grande gama de variedades em ferramentas que o portal biológico NCBI disponibiliza. Da mesma forma com que o KEGG e UniProt foram explorados, onde essas ferramentas não se limitam apenas a bancos de dados generalizados ou especializados, os portais vão mais além e apresentam ferramentas de diversas finalidades para trabalhar com os dados que elas mesmas disponibilizam, movimentando um conhecimento mais abrangente e possibilitando análises em diversas áreas de estudo.

Tabela 3 – Seleção de ferramentas do NCBI

Ferramenta	Área de atuação	Descrição
Amino Acid Explorer	Proteínas	A ferramenta explora propriedades, funções e substituições de aminoácidos
BLAST	Genômica e proteínas	Encontra regiões com certo nível de similaridade em sequências biológicas
CDTree	Proteínas	Classifica sequências de proteínas e investiga seu relacionamento evolucionário
Cn3D	Proteínas	Mostra e manipula estruturas de proteínas 3D e alinhamentos de um banco de dados de estrutura de proteínas
COBALT	<i>E. coli</i>	Calcula a diferença de estabilidade entre promotores e regiões codificantes através de uma divisão da sequência em janelas sobrepostas de 15 nucleotídeos
Digital Differential Display (DDD)	Genes	Identifica genes com uma expressão significativamente diferente através de uma comparação de perfis EST
Electronic PCR	Genomas	Identifica sítios marcados em uma sequência de DNA
Genome Workbench	Genomas	Aplicação integrada para visualizar e analisar dados sequenciados
Open Reading Frame Finder	Genes	Sugere a possível localização de ORFs em genomas
Primer-BLAST	Genes	Utiliza o algoritmo Primer3 para montar <i>primers</i> de PCR para uma sequência modelo
Related Structures	Proteínas	Busca estruturas 3D que são similares em questão de sequência para buscar uma proteína
Sequence Viewer	Genes	Demonstração gráfica de uma sequência biológica
Viral Genotyping Tool	Genomas	Identifica o genótipo de uma sequência de vírus

Fonte: Elaboração do autor.

A área de bioinformática vem fazendo até então um bom uso de todas as ferramentas que tem em sua disposição, de acordo com o que fora visto até aqui, podemos esperar um futuro muito promissor no que diz a respeito de portais e bancos de dados biológicos. Nesse capítulo buscamos realizar uma integração de como portais e bancos de dados biológicos trabalham em sinergia. Isso demonstra a multidisciplinariedade que permeia a área e a forma de se fazer ciência atualmente, onde a rapidez com que a tecnologia tem evoluído requer que cientistas e profissionais da área tenham maior integração com os demais campos de estudo, que até então eram apenas tidos como complementares. Nossa jornada através de exemplos práticos dessas duas modalidades de ferramentas indicou que, na maioria dos casos, a sua existência aparece casada. Conseguimos notar que o avanço da ciência e tecnologia vem cada vez mais criando espaço para o surgimento mais acelerado de novas formas de extrair informações biológicas de organismos em diversos níveis, como sequenciamento de genomas, identificação de regiões regulatórias, caracterização e predição de proteínas e todo esse conhecimento necessita ser armazenado em bancos de dados. A partir do momento em que o banco de dados existe, já surge a necessidade de obter ferramentas robustas que sejam capazes de trabalhar e extrair conhecimento sobre essa grande quantidade de dados. A tecnologia, que vem avançando em escala não linear, vem fazendo com que a obtenção, o armazenamento e o uso de recursos biológicos aconteça de maneira muito rápida. O que nos leva a esperar grandes avanços para o futuro na área da biotecnologia.

Referências

ALTSCHUL, S. F. *et al.* Basic Local Alignment Search Tool. **Journal of Molecular Biology**, n. 215, p. 403-410, 1990.

CHEN, C.; HUANG, H.; WU, C. H. Protein Bioinformatics Databases and Resources. **Methods in molecular biology** (Clifton, N.J.), n. 1558, p. 3-39, 2017.

CHRISTIAN, B. **O humano mais humano: o que a inteligência artificial nos ensina sobre a vida.** São Paulo: Companhia das Letras, 2013.

CHRISTIAN, B.; GRIFFITH, T. **Algoritmos para viver.** São Paulo: Companhia das Letras, 2017.

GALPERIN, M.Y.; COCHRANE, G.R. The 2011 Nucleic Acids Research Database Issue and the online Molecular Biology Database Collection. **Nucleic Acids Res.** 39(Database issue):D1–D6. 2010. doi:10.1093/nar/gkq1243

GAUDET, P.; BAIROCH, A.; FIELD, D.; SANSONE, S. A.; TAYLOR, C.; ATTWOOD, T. K.; BATEMAN, A.; BLAKE, J. A.; BULT, C. J.; CHERRY, J. M.; CHISHOLM, R. L.; COCHRANE, G.; COOK, C. E.; EPPIG, J. T.; GALPERIN, M. Y.; GENTLEMAN, R.; GOBLE, C. A.; GOJOBORI, T.; HANCOCK, J. M.; HOWE, D. G.; IMANISHI, T.; KELSO, J.; LANDSMAN, D.; LEWIS, S. E.; MIZRACHI, I. K.; ORCHARD, S.; OUELLETTE, B. F.; RANGANATHAN, S.; RICHARDSON, L.; ROCCA-SERRA, P.; SCHOFIELD, P. N.; SMEDLEY, D.; SOUTHAN, C.; TAN, T. W.; TATUSOVA, T.; WHETZEL, P. L.; WHITE, O.; YAMASAKI, C. BioDBCore Working Group. Towards BioDBCore: a community-defined information specification for biological databases. **Nucleic acids research**, 39(Database issue), D7-10, 2010.

- HIRSCHMAN, L.; BURNS, G.A.P.C.; KRALLINGER, M.; ARIGHI, C.; COHEN, K.B.; VALENCIA, A.; WU, C.H.; CHATR-ARYAMONTRY, A.; DOWELL, K.G.; HUALA, E.; LOURENÇO, A.; NASH, R.; VEUTHEY, A.; WIEGERS, T.; WINTER A.G. **Text mining for the biocuration workflow**. Database(Oxford). 2012. DOI:10.1093/database/bas020
- HOWE, D.; CONSTANZO, M.; FEY, P.; GOJOBORY, T.; HANNICK, L.; HIDE, W.; HILL, D. P.; KANIA, R.; SCHAEFFER, M.; ST PIERRE, S.; TWIGGER, S.; WHITE, O.; RHEE, S. Y. The future of biocuration. **Nature**, n. 455, p. 47-50, 2008.
- LOBO, I. Basic Local Alignment Search Tool (BLAST). **Nature Education**, v. 1, n. 1, p. 215, 2008.
- ZOU, D.; MA, L.; YU, J.; ZHANG, Z. Biological databases for human research. **Genomics Proteomics Bioinformatics**, v. 13, n. 1, p. 55-63, 2015.
- COLLUMBUS, L. 10 Charts That Will Change Your Perspective on Artificial Intelligence's Growth. Forbes. 12 Jan. 2018. Disponível em: <https://www.forbes.com/sites/louiscolumbus/2018/01/12/10-charts-that-will-change-your-perspective-on-artificial-intelligences-growth/#7d6058e24758>. Acesso em: 15 mar. 2019.

FERRAMENTAS DE ANÁLISE E PROCESSAMENTO DE METAGENOMAS

Tahila Andrighetti¹

Os micro-organismos são os seres vivos mais abundantes da Terra. Bactérias, arqueias, vírus e microeucariotos (fungos e protozoários) fazem parte de todos os ecossistemas terrestres que têm condições de suportar vida, desde os mais amenos – como solo, tecidos animais e vegetais e oceanos – até ambientes extremos, como fumarolas, minas ácidas e geleiras, onde muitas vezes são os únicos habitantes. Comunidades microbianas cumprem papéis cruciais na dinâmica dos ecossistemas, decompondo matéria morta e disponibilizando novamente nutrientes como enxofre, carbono, nitrogênio e oxigênio, para serem adquiridos por outros organismos (COUNCIL, 2007; WOOLEY; GODZIK; FRIEDBERG, 2010).

A habilidade de reciclagem de nutrientes torna os micro-organismos indispensáveis para a vida na Terra e atrai o interesse humano para aplicações que podem ser úteis em diversas áreas. Comunidades microbianas associadas a outros organismos influenciam na fisiologia do hospedeiro e contribuem para sua saúde e crescimento. A microbiota no intestino de bovinos produz enzimas para a digestão de celulose; o entendimento sobre a relação entre a digestão e as enzimas produzidas pelos micro-organismos fornecem informações que podem servir de embasamento para a melhoria da produção de leite e carne e também para a diminuição do impacto ambiental causado pela criação de gado (MORGAVI *et al.*, 2013).

Micro-organismos também habitam tecidos de seres humanos. O número de células de bactérias presentes no corpo humano excede 100 trilhões, dez vezes mais do que a quantidade das células do próprio corpo (BELLA *et al.*, 2013). Alguns micro-organismos que hospedam-se em seres humanos podem ser patógenos, mas a maioria é indispensável para sua vida. Como exemplos, temos a comunidade microbiana presente na pele, que garante imunidade e proteção contra agentes externos, e a microbiota do trato gastrointestinal, cuja composição influencia na aquisição de nutrientes, no rendimento de energia e em diversas vias metabólicas; seu desequilíbrio pode facilitar a indução de doenças como diabetes tipo 2 e obesidade. Informações obtidas dos estudos de microbiomas do corpo humano têm o potencial de auxiliar no desenvolvimento de métodos alternativos de tratamento e de prevenção de diversas doenças (DEVARAJ; HEMARAJATA; VERSALOVIC, 2013).

¹ Universidade Estadual Paulista Júlio de Mesquita Filho. *E-mail*: tahilaandrighetti@gmail.com

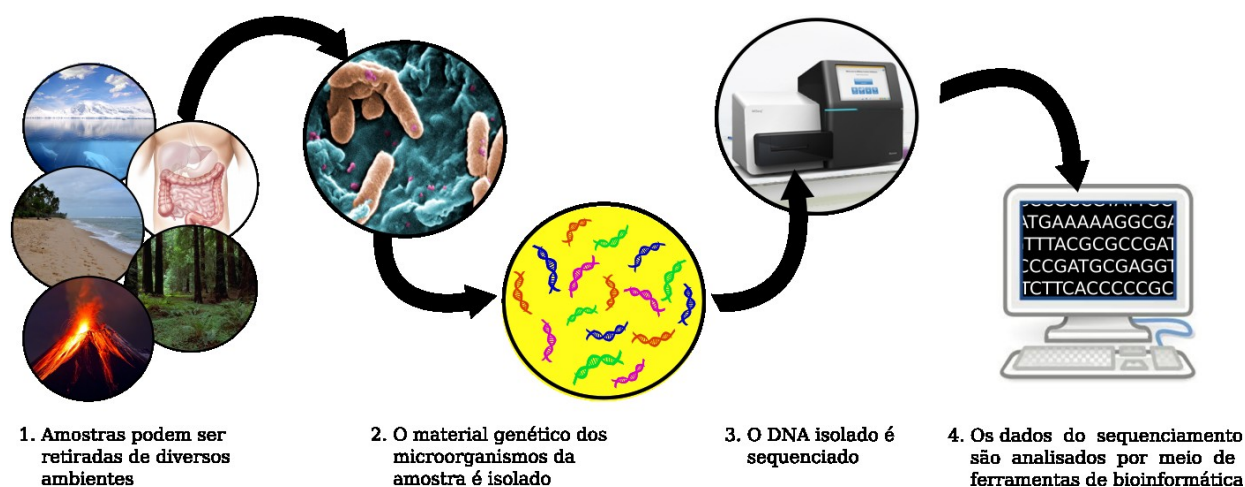
Em solos, a associação da composição microbiana com plantas é indispensável, pois desempenha papéis na qualidade do solo e produtividade e saúde das plantas hospedeiras, através de mecanismos diretos ou indiretos, como na mineralização da matéria orgânica do solo, ativação dos mecanismos de defesa de plantas e produção de antibióticos contra patógenos; o melhoramento de microbiomas do solo pode auxiliar para maior rendimento agrícola e controle de pestes, bem como aprimoramento de alimentos como vinhos e queijos (ZARRAONAINDIA *et al.*, 2015). Em oceanos, podem ser observadas diferenças significativas nas comunidades microbianas em diferentes profundidades, influenciadas por características ambientais como oxigenação, salinidade e temperatura; em ambientes marinhos poluídos observou-se a presença de genes de resistência a arsênico e a metais pesados e de redução de sulfato, refletindo a alta capacidade de adaptação dos micro-organismos (BIK, 2014).

Entretanto, apenas 1% dos micro-organismos podem ser cultivados em laboratório (HANDELSMAN, 2004), limitando consideravelmente a extensão a que estudos de microbiomas podem ser conduzidos, a partir de meios de cultura. Essa dificuldade foi superada com o advento das tecnologias de sequenciamento de DNA que possibilitaram o estabelecimento de um novo campo de estudo inserido na genômica: a metagenômica. O termo, cunhado por Handelsman em 1998 (HANDELSMAN *et al.*, 1998), define o estudo dos genomas de comunidades microbianas presentes em um determinado *habitat* a partir do DNA extraído desse ambiente, sem a necessidade de cultivo dos micro-organismos. Deste modo, permitiu a revelação da diversidade microbiana e genética de diversos sistemas biológicos, relações genômicas entre função e filogenia de organismos não cultiváveis e perfis evolucionários de comunidades, além de outras interações biomoleculares (MARCO, 2011; THOMAS; GILBERT; MEYER, 2012).

Como exemplos de grandes iniciativas baseadas nessa tecnologia, temos o Projeto Microbioma Humano (*Human Microbiome Project*), financiado pelo NIH (*National Institutes of Health*), e o consórcio europeu MicroWine. O primeiro tem como objetivo sequenciar o metagenoma de partes do corpo humano, como cavidade gastrointestinal, olhos, pele, vias aéreas, trato urogenital e sangue, para esclarecer o papel do microbioma na saúde e desenvolver novas ferramentas que possam ser utilizadas posteriormente em prol de outras pesquisas (PETTERSSON; LUNDEBERG; AHMADIAN, 2009). Já o MicroWine explora comunidades de micro-organismos que desempenham papéis importantes em todos os estágios da viticultura – auxiliando o acesso das plantas a nutrientes do solo e na sua imunidade contra patógenos –, até os processos de vinificação, que influenciam nos sabores e aromas característicos de cada vinho (MICROWINE, 2016).

A primeira etapa de qualquer estudo metagenômico envolve a retirada das amostras do ambiente de estudo e posterior isolamento, fragmentação e sequenciamento do material genético dos micro-organismos relacionados àquele meio (Figura 1). Há três gerações de métodos de sequenciamento. Os métodos de primeira e segunda geração fragmentam o DNA em segmentos (*reads*) cujos comprimentos variam entre 35 e 700 pares de base. Devido à sua natureza, a análise dos dados metagenômicos resultantes dessas técnicas, através de ferramentas computacionais torna-se bastante complexa. O método de primeira geração, também chamado de sequenciamento de Sanger, ainda é utilizado devido à sua baixa taxa de erros e *reads* relativamente longos, com mais de 700 pb, facilitando a análise pós-sequenciamento. Entretanto, seu custo é mais elevado do que das plataformas de nova geração – U\$ 400 mil por gigabase – e limita-se a até 96 Kb de informação por sequenciamento. Em contrapartida, as plataformas de segunda geração podem chegar a custar U\$ 50,00 por gigabase e retornam mais de 1 GB por sequenciamento. Conseqüentemente, essas tecnologias vêm substituindo o sequenciamento de Sanger, através de plataformas como Illumina/Solexa, 454/Roche e Applied Biosystems SOLiD. Por sua vez, o sequenciamento de terceira geração, também conhecido como sequenciamento de molécula única, propõe o rendimento de mais dados a menores custos e *reads* de tamanho maior do que 10 mil pb; as duas tecnologias de sequenciamento de molécula única mais utilizadas são *Pacific Biosciences* e *Oxford Nanopore*. Apesar de suas vantagens, sequenciamentos de terceira geração ainda são pouco utilizados na metagenômica, devido a sua alta taxa de erros (Tabela 1) (MOROZOVA; MARRA, 2008; THOMAS; GILBERT; MEYER, 2012; LAND *et al.*, 2015; OULAS *et al.*, 2015; LEE *et al.*, 2016).

Figura 1 – Etapas detalhadas da realização da metagenômica



Fonte: Elaboração da autora.

A diminuição de preço das tecnologias de sequenciamento de segunda e terceira gerações (NGS, do inglês *New Generation Sequencing* – sequenciamento de nova geração) permitiu a popularização da metagenômica entre os pesquisadores e, conseqüentemente, o aumento na quantidade de dados disponibilizada em bancos de dados. Entretanto, o poder computacional e o desenvolvimento de algoritmos de análise de metagenomas não acompanha o crescimento na quantidade de dados produzidos. O primeiro problema está relacionado com a disponibilidade dos metagenomas: os sistemas de armazenamento de seqüências não suportam quantidades de dados tão massivas e o formato dos dados não é padronizado. Outro obstáculo está relacionado às características dos dados produzidos: *reads* muito curtos e grande quantidade de erros gerados pelas plataformas de nova geração fazem com que a análise de metagenomas demande algoritmos mais complexos e alto custo computacional. Deste modo, é evidente a necessidade de novas ferramentas de análise de metagenomas para a maioria das etapas de processamento dos dados obtidos através de sequenciamento (KIM *et al.*, 2013; KUMAR *et al.*, 2015).

Tabela 1 – Lista de plataformas de sequenciamento, o tamanho de seus *reads* e seu custo por GB

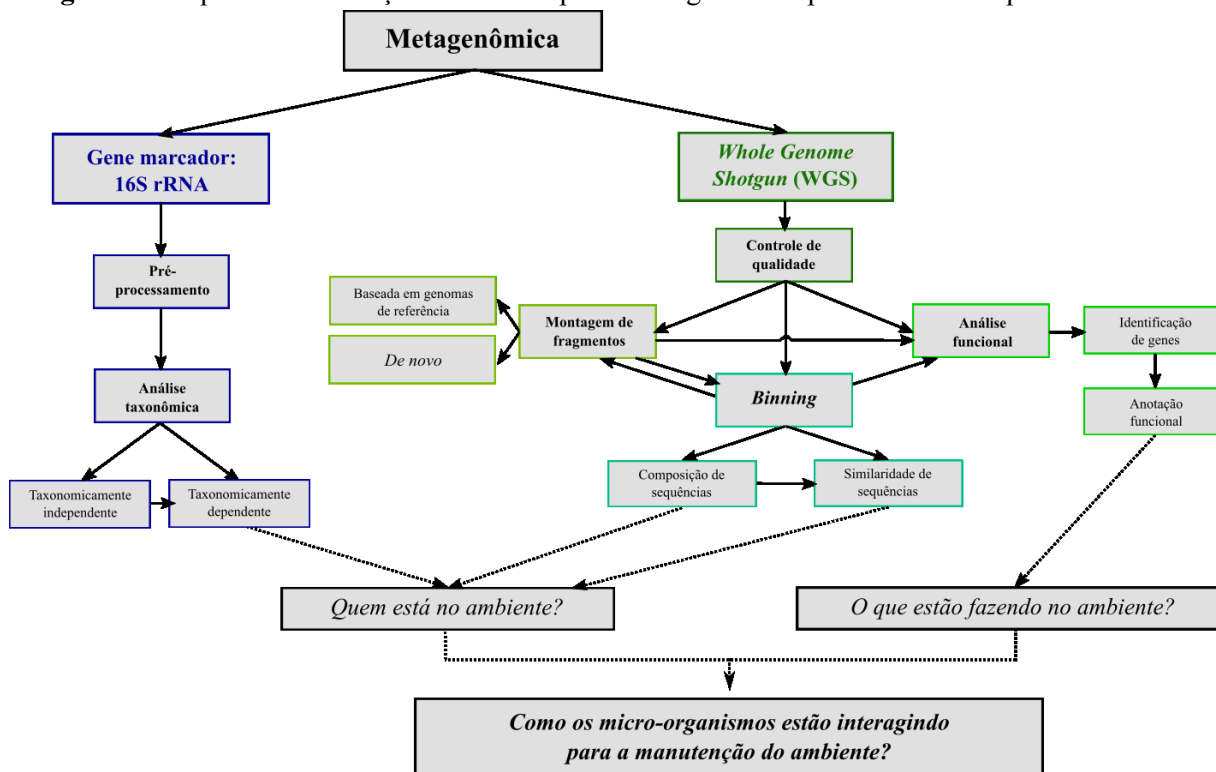
Geração	Tecnologia de sequenciamento	Tamanho dos <i>reads</i>	Custo por GB (aprox.)	Rendimento por sequenciamento
1 ^a	Sanger	> 700 pb	US\$ 400 000	96 kb
2 ^a	454 / Roche	400 – 700 pb	US\$ 20 000	80 – 120 Mb
2 ^a	Illumina	100 – 150 pb	US\$ 50	1 Gb
2 ^a	Life Technologies / SOLiD	35 – 75 pb	US\$ 130	1 – 3 Gb
3 ^a	Pacific Biosciences	10 – 15 kpb	US\$ 500	5 Gb
3 ^a	Oxford Nanopore Technologies	5 – 10 kpb	US\$ 1000	> 40 Gb
3 ^a	Ion Torrent (318 chip)	200 pb	US\$ 1000	1 Gb

Fonte: MOROZOVA; MARRA, 2008; THOMAS; GILBERT; MEYER, 2012; BAHASSI; STAMBROOK, 2014; LEE *et al.*, 2016.

Existem duas categorias para o sequenciamento de metagenomas. A primeira, metagenômica a partir de genes marcadores, um gene específico é isolado através de PCR e sequenciado; o gene mais utilizado para a identificação de bactérias e archaeas é o 16S rRNA, portanto, focaremos na metagenômica a partir desse gene em nossa revisão. O segundo tipo de metagenômica, metagenômica por WGS (do inglês, *whole genome shotgun*), todo o DNA dos micro-organismos presentes na amostra é sequenciado (OULAS *et al.*, 2015). A escolha da técnica mais adequada depende do objetivo de análise dos dados. Há métodos de análise e ferramentas específicas para estudar os dados de cada abordagem. A Figura 2 apresenta um esquema ilustrando as

etapas de cada categoria de metagenômica. Entraremos em detalhes sobre as etapas e ferramentas utilizadas nas seções a seguir.

Figura 2 –Esquema de execução de cada etapa da metagenômica por 16S rRNA e por WGS



Fonte: Elaboração da autora.

1 Metagenômica a partir do gene 16S rRNA

Embora os estudos a partir de WGS estejam sendo desenvolvidos com frequência cada vez mais alta, a análise dos 16S rRNA ainda é amplamente aceita e é uma ferramenta poderosa para o estudo das comunidades microbianas em alta resolução (SUN *et al.*, 2011). A utilização do gene 16S rRNA, como marcador taxonômico, possibilitou o desenvolvimento de um método de identificação de micro-organismos de uma amostra sem a necessidade de cultivo dos micro-organismos. A primeira execução bem sucedida ocorreu em 1991, quando foram registradas novas espécies a partir da análise dos genes 16S rRNA de amostras de oceano (SCHMIDT; DELONG; PACE, 1991; RIESENFELD; SCHLOSS; HANDELSMAN, 2004).

A inclusão da análise taxonômica de microbiomas, a partir do gene 16S rRNA no conceito de *metagenômica*, ainda é uma controvérsia entre os pesquisadores. Muitos deles sugerem que esse tipo de análise seja denominada *metagenética*, por utilizar apenas um gene e não todo o genoma (ESPOSITO; KIRSCHBERG, 2014). Entretanto,

para fins didáticos, neste capítulo consideraremos que a análise taxonômica a partir do gene 16S rRNA também faz parte do campo da metagenômica.

O gene 16S rRNA codifica a subunidade pequena do RNA ribossômico de Archaeas e Bactérias e mostrou-se adequado por apresentar regiões hiperconservadas intercaladas com regiões variáveis ao longo de sua sequência. As regiões conservadas são quase idênticas dentre os micro-organismos, portanto são utilizadas para desenvolver primers universais para o isolamento dos genes da amostra. As outras regiões variam proporcionalmente à proximidade filogenética entre os táxons, sendo, portanto utilizadas como parâmetros de comparação para a identificação dos micro-organismos (Figura 3) (KIM *et al.*, 2013; NIKOLAKI; TSIAMIS, 2013).

Figura 3 – Figura ilustrando a estrutura do gene 16S rRNA. Em azul, estão representadas as regiões conservadas e em cinza, as regiões variáveis

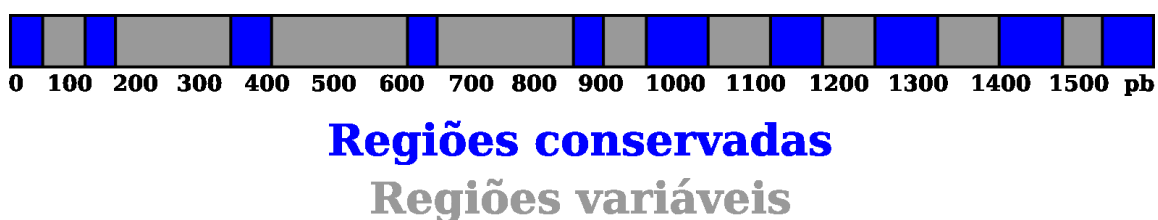
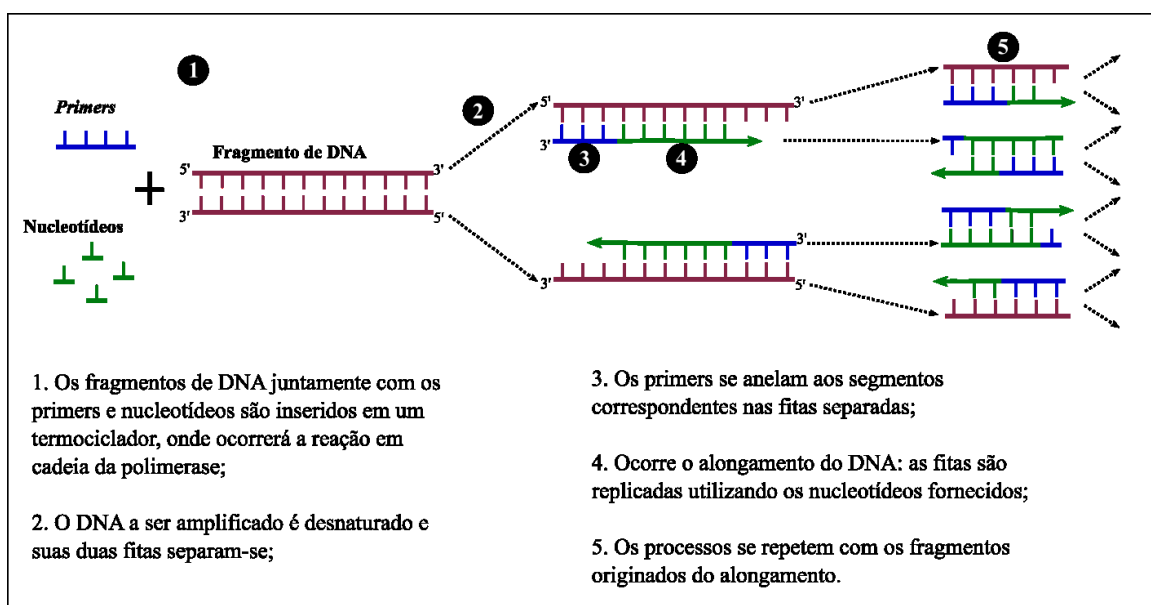


Figura 4 – Ilustração da amplificação de genes no processo de PCR



Fonte: Elaboração da autora.

Entretanto, as regiões variáveis do gene 16S rRNA apresentam baixa resolução entre as espécies, permitindo a classificação eficiente somente até o nível de gênero.

Outro obstáculo da técnica 16S rRNA é a alta susceptibilidade a vieses, pois os *primers* podem apresentar mais afinidade por sequências de determinadas espécies. Essa tendência pode favorecer a seleção dos genes de alguns organismos em detrimento dos outros presentes na amostra, e impedir que novos táxons sejam representados (NIKOLAKI; TSIAMIS, 2013; PORETSKY *et al.*, 2014).

Mesmo com as limitações apresentadas, a metagenômica por 16S rRNA ainda tem sido mais utilizada para análises taxonômicas e filogenéticas do que a análise por WGS. Isso porque o advento das tecnologias de sequenciamento de nova geração não facilitou somente o sequenciamento de genomas inteiros, mas também permitiu o desenvolvimento de novas tecnologias específicas para as análises com o 16S rRNA, tornando esse método mais rápido, fácil e barato. Conseqüentemente, a quantidade dos dados de referência se ampliou significativamente, facilitando suas análises (KUNIN *et al.*, 2008; TRINGE; HUGENHOLTZ, 2008).

O primeiro passo para a execução da metagenômica a partir do gene 16S rRNA é a obtenção do DNA dos micro-organismos de um meio ambiente. Depois de isolado, esse DNA passa pelo processo de PCR, em que ocorre a amplificação dos genes a partir de *primers*, que identificam a localização do gene 16S rRNA e o replicam diversas vezes (Figura 4). O produto dessa amplificação é submetido à técnica de eletroforese, em que o gene pode ser identificado em um gel por meio de bandas. As bandas que correspondem ao gene 16S rRNA são selecionadas e o DNA nelas contido é purificado e sequenciado (SANSCHAGRIN; YERGEAU, 2014; ZHOU *et al.*, 2015).

Depois do sequenciamento, os dados resultantes (*output*) são armazenados em um computador e devem ser processados e analisados, a partir de ferramentas de bioinformática.

1.1 Pré-processamento dos dados brutos

O *output* do sequenciamento de DNA é um conjunto de dados brutos que, além das sequências de interesse, contém erros de replicação e sequências de baixa qualidade, que prejudicam a análise dos genes sequenciados. Para a filtragem das sequências e análise mais precisa dos metagenomas, é necessária a realização de uma etapa denominada, em inglês, *denoising*, que consiste no pré-processamento dos dados brutos, para que reste somente as sequências que representam a comunidade microbiana com qualidade (KIM *et al.*, 2013; OULAS *et al.*, 2015). Abaixo estão citadas algumas ferramentas de *denoising* mais utilizadas:

- **Pyronoise:** ferramenta de *denoising* dos dados obtidos pelo pirosequenciamento 454, da Roche, uma das plataformas mais utilizadas para a metagenômica, a partir de 16S rRNA. Os dados brutos gerados por essa

plataforma são fluxogramas que representam os *reads* sequenciados. O PyroNoise realiza agrupamentos desse fluxograma, de acordo com características específicas e utiliza uma medida de distância que modela o ruído do sequenciamento. Esse método permite a identificação das sequências verdadeiras em meio aos fragmentos sequenciados (QUINCE *et al.*, 2009; GASPAR; THOMAS, 2013);

- **Amplicon-Noise:** essa ferramenta utiliza primeiramente o algoritmo do PyroNoise para remover os erros a partir do agrupamento de fluxogramas, mas sem realizar alinhamento de sequências, como é o caso do *software* original do PyroNoise (QUINCE *et al.*, 2011; GASPAR; THOMAS, 2013);
- **QHIME:** é uma versão mais rápida do PyroNoise original. O algoritmo alinha e agrupa os fluxogramas em um único passo, levando em conta tanto os erros do pirosequenciamento quanto os do PCR. Aceita dados em formato fastq, portanto pode ser utilizado para outras plataformas além da 454 Roche (CAPORASO *et al.*, 2010; GASPAR; THOMAS, 2013);
- **DADA, Divisive Amplicon Denoising Algorithm:** realiza o *denoising*, executando um modelo estatístico paramétrico de substituição de erros, juntamente com um algoritmo de agrupamento hierárquico divisivo (ROSEN *et al.*, 2012).

Inclusas, na categoria de sequências de baixa qualidade, estão as quimeras. Quimeras são recombinantes artificiais que se formam entre duas ou mais sequências durante a amplificação do DNA pela técnica de *Polymerase Chain Reaction* (PCR). Normalmente formam-se quando fragmentos de DNA que terminam prematuramente a amplificação anelam-se a outros. Essas moléculas artificiais dificultam a diferenciação das sequências originais das recombinantes, resultando na superestimação do nível de diversidade microbiana presente na amostra (KIM *et al.*, 2013).

Entretanto, a detecção das quimeras na amostra não é um processo trivial, uma vez que a união das moléculas ocorre em posições aleatórias, a maioria das plataformas de NGS geram *reads* curtos, dificultando a localização das sequências originais que possuam informação taxonômica suficiente (KIM *et al.*, 2013).

Há algumas ferramentas específicas para a remoção de quimeras dos dados obtidos por NGS. O *software* ChimeraSlayer realiza o alinhamento das sequências a serem filtradas com sequências de referência presentes em bancos de dados para identificar possíveis quimeras em meio aos dados e retirá-las (HAAS *et al.*, 2011). A maioria das outras ferramentas, como Perseus (QUINCE *et al.*, 2011), Decipher (WRIGHT; YILMAZ; NOGUERA, 2012) e UCHIME (EDGAR, 2010), utiliza

informações de frequências de sequências para detectar as quimeras, assumindo que sequências quiméricas são menos representadas do que as amplificadas normalmente.

1.2 Caracterização taxonômica do microbioma

A etapa seguinte à filtragem dos dados é a classificação taxonômica do gene 16S. Os pesquisadores que optam por realizar seus estudos metagenômicos, a partir do 16S rRNA, normalmente visam à obtenção de um perfil taxonômico ou filogenético da comunidade microbiana em questão, para responder a pergunta: *Quem está no meio?*. A partir das informações obtidas, é possível realizar estudos relacionados à evolução da comunidade microbiana, associação entre micro-organismos, comparação da composição microbiótica de diferentes meios, entre outros.

A classificação dos genes pode ocorrer de duas maneiras: de forma taxonomicamente dependente ou independente. Nas análises taxonomicamente dependentes, as sequências desconhecidas são comparadas com outras já classificadas presentes em bancos de dados e então atribuídas a táxons cujas sequências apresentaram maior similaridade. Em análises taxonomicamente independentes, as sequências são agrupadas de acordo com índices de similaridade obtidos com a comparação de umas com as outras, sem utilizar bases de dados como referência (SUN *et al.*, 2011). Embora muitas ferramentas apliquem os métodos separadamente, essas abordagens podem ser utilizadas concomitantemente para uma análise mais prática e aprofundada.

Os métodos de análise taxonômica de 16S rRNA e suas ferramentas estão a seguir.

1.2.1 Análises taxonomicamente dependentes

Na realização das análises taxonomicamente dependentes, as sequências de 16S rRNA desconhecidas são comparadas com sequências conhecidas disponíveis em bancos de dados e atribuídas aos táxons, com os quais apresentam maior similaridade (SUN *et al.*, 2011).

Uma das ferramentas mais utilizadas para as análises taxonomicamente dependentes é o BLAST (ALTSCHUL *et al.*, 1990). Esse *software* realiza o alinhamento de sequências desconhecidas com sequências de bancos de dados que podem ser fornecidos pelo usuário. Existem bancos de dados específicos com informações de RNA ribossomal como SILVA (QUAST *et al.*, 2013), EzTaxon-e (KIM *et al.*, 2012), RDP (COLE *et al.*, 2009) e Greengenes (DESANTIS *et al.*, 2006), que podem ser importados no BLAST para análises que utilizam sequências de referência.

Os dados resultantes do alinhamento obtido por algoritmos, como BLAST, precisam ser submetidos a outras ferramentas que processam esses *outputs* e apresentam

as atribuições taxonômicas ao pesquisador, para que ele possa interpretar os dados. Exemplos de *softwares* utilizados para esse fim são MEGAN (HUSON *et al.*, 2007), que utiliza algoritmo do menor ancestral em comum (LCA), e TANGO, que atribui os *reads* a um nó da taxonomia de referência, minimizando um *score* de penalidade que generaliza o mapeamento baseado no valor de medida F (CLEMENTE; JANSSEN; VALIENTE, 2011; KIM *et al.*, 2013).

Outra alternativa para a abordagem taxonomicamente dependente, é a comparação da similaridade das sequências por sua composição. RDP utiliza a frequência de oligonucleotídeos de 8 bases das sequências, para treinar as redes uma ferramenta redes Bayesianas, e atribuir táxons às sequências desconhecidas (WANG *et al.*, 2007; KIM *et al.*, 2013).

Existe ainda outro método de análise taxonomicamente dependente, que classifica as sequências de acordo com sua alocação em uma árvore filogenética guia baseada em modelos evolucionários. É uma alternativa útil para casos em que não há sequências de referência de micro-organismos de táxons próximos às sequências desconhecidas. Algoritmos que utilizam essa estratégia incluem SEPP (MIRARAB; NGUYEN; WARNOW, 2012), EPA (BERGER; KROMPASS; STAMATAKIS, 2011), pplacer (MATSEN; KODNER; ARMBRUST, 2010), QIIME (CAPORASO *et al.*, 2010) e AMPHORA2 (WU; SCOTT, 2012; KIM *et al.*, 2013).

1.2.2 Análises taxonomicamente independentes

A realização de análises taxonomicamente independentes normalmente baseia-se no agrupamento de OTUs. OTU, sigla em inglês para *operational taxonomic unit*, que significa “unidades taxonômicas operacionais” em português. OTUs são os agrupamentos dos genes 16S rRNA de acordo com sua similaridade. Níveis de similaridade maiores do que 97% para bactérias e archaeas correspondem à mesma espécie (PATIN *et al.*, 2013; OULAS *et al.*, 2015).

Algoritmos para agrupamento de sequências de 16S rRNA utilizam basicamente duas estratégias: a partir de alinhamento de sequências, no qual as sequências desconhecidas de 16S rRNA são alinhadas entre elas, podendo haver ou não a utilização de sequências de referência; e estratégias independentes de alinhamento. Abaixo, estão citadas algumas ferramentas para agrupamento de OTUs (KIM *et al.*, 2013):

- **NAST**: compara as sequências de 16S rRNA alinhando-as com sequências de referência não quiméricas (DESANTIS *et al.*, 2006);
- **SINA aligner**: realiza o alinhamento das sequências com um algoritmo baseado em ordem parcial (PRUESSE; PEPLIES; GLÖCKNER, 2012);

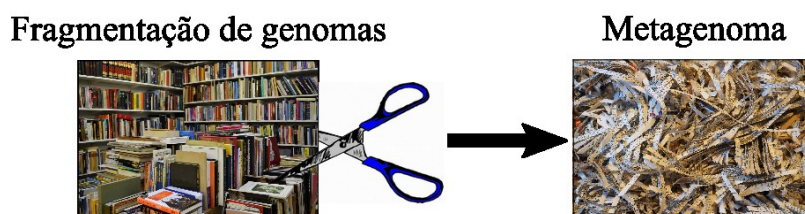
- **Infernal**: baseia-se nos alinhamentos de perfis de estruturas secundárias de RNA, para realizar o agrupamento de OTUs (NAWROCKI; KOLBE; EDDY, 2009);
- **UCLUST**: organiza as sequências para diminuir o número de oligonucleotídeos em comum, explorando o fato de que sequências similares tendem a ter pequenos oligonucleotídeos em comum (EDGAR, 2010);
- **CD-HIT**: primeiramente, utiliza a maior sequência de entrada como primeira representante do agrupamento. Em seguida, compara as sequências restantes em ordem decrescente de tamanho. A partir da comparação realizada, o algoritmo classifica as sequências como redundantes ou representativas em comparação com as que já foram consideradas como representativas previamente. As similaridades são calculadas pela contagem dos oligonucleotídeos em comum (FU *et al.*, 2012);
- **ESPIRIT-Tree**: emprega uma técnica de particionamento de espaço para organizar os objetos hierarquicamente em células. Deste modo, o algoritmo encontra o vizinho mais próximo de cada objeto em células adjacentes, utilizando uma estratégia de “dividir e conquistar” (CAI; SUN, 2011).

2 Metagenômica por *whole genome shotgun* (WGS)

A metagenômica a partir do gene 16S rRNA é um modo eficaz para estudar a diversidade microbiana das comunidades e seu impacto nos ambientes em que estão presentes. Entretanto, a partir dessa abordagem é possível obter somente informações sobre a composição dos micro-organismos dos meios analisados. A partir de metagenômica por WGS, é possível adquirir as informações de biodiversidade relacionadas à composição funcional do meio, permitindo responder às questões: *Quem está presente no ambiente?*, *O que esses micro-organismos estão fazendo?* e *Como eles interagem?* (FORDE; O'TOOLE, 2013; OULAS *et al.*, 2015).

No procedimento para metagenômica por WGS está incluído o isolamento do material genético da amostra. Nesse caso, todo o DNA é fragmentado (Figura 5) e submetido ao sequenciamento. Como o genoma dos micro-organismos da amostra é aleatoriamente fragmentado, os *reads* do metagenoma pertencem a qualquer micro-organismo e a qualquer parte do genoma: desde sequências repetitivas até elementos taxonomicamente informativos, como o gene 16S rRNA, ou sequências codificadoras que fornecem informações sobre funções que os micro-organismos podem realizar no ambiente (SHARPTON, 2014).

Figura 5 – Representação lúdica da fragmentação de genomas para o sequenciamento de metagenomas. Para o sequenciamento do metagenoma, é necessário fragmentar os genomas dos micro-organismos da amostra. Os metagenomas são conjuntos de fragmentos de genomas de diversos micro-organismos misturados. Na figura, os genomas são representados como livros que, se fragmentados, originam um monte de pedaços de papéis, representando os metagenomas. É muito difícil identificar a que livro pertence cada fragmento de papel, assim como é difícil determinar a origem dos fragmentos de metagenômica.



Fonte: Adaptado de Sharpton (2014).

O desenvolvimento de ferramentas para análise de metagenomas por WGS é mais desafiador do que o de ferramentas para metagenomas de 16S rRNA, dada sua maior complexidade. Primeiramente, porque é mais difícil determinar a origem taxonômica dos *reads* dada sua procedência aleatória de dentro do genoma. Depois, na maioria das vezes não é possível obter a representação de todos os micro-organismos do ambiente devido à sua grande diversidade. Além disso, a etapa de filtragem é mais complicada do que a do 16S rRNA: a identificação e remoção de sequências contaminantes é problemática devido à dificuldade de diferenciar as sequências dos micro-organismos das sequências de organismos indesejados. Também pode haver a necessidade de montar os genomas, etapa desafiadora para os bioinformatas, devido ao alto custo computacional e à demanda por grandes quantidades de informações (SHARPTON, 2014).

Essas limitações têm esmaecido com o avanço das tecnologias de informática, que vêm apresentando computadores mais potentes ao longo do tempo, e com as tecnologias de sequenciamento, que prezam por *reads* mais longos. Abaixo estão detalhadas as etapas de análise de bioinformática para metagenômica por WGS e ferramentas utilizadas para cada finalidade.

2.1 Controle de qualidade das sequências

O primeiro passo a ser realizado, depois que o metagenoma é sequenciado, é o controle de qualidade, ou filtragem dos dados retornados para retirar sequências de baixa qualidade do metagenoma. Essa é uma importante etapa a ser realizada, pois erros e sequências de organismos não desejados dificultam a montagem dos *reads* e sua

análise, especialmente quando o contaminante é altamente abundante ou tem um grande genoma (BRAGG; TYSON, 2014; SHARPTON, 2014). Alguns dos programas utilizados para o controle de qualidade são específicos para determinadas plataformas de sequenciamento. Os mais utilizados estão citados abaixo:

- **FASTX toolkit**: conjunto de linhas de comando para pré-processamento de *reads* curtos de dados FASTA/FASTQ (FASTX-TOOLKIT, 2016);
- **FASTQC**: ferramenta de controle de qualidade para dados retornados de sequenciamento pela plataforma Illumina. FASTQC fornece uma interface gráfica para visualização e simplificação da filtragem desses dados (BABRAHAM BIOINFORMATICS, 2016);
- **ngs_backbone**: aplicável a dados de sequenciamento de Sanger, 454, Illumina e SOLiD. Além da limpeza de dados, executa funções como: montagem e anotação de transcriptomas, leitura de mapeamento e busca e seleção por polimorfismos de nucleotídeo único (SNP, do inglês, *single nucleotide polymorphism*) (BLANCA *et al.*, 2011);
- **Pyrobayes**: *software* de controle de qualidade de dados retornados por pirosequenciamento 454. Pyrobayes permite busca por SNPs em aplicações de resequenciamento, produzindo buscas mais confiáveis do que o programa nativo da plataforma (QUINLAN *et al.*, 2008);
- **Shore**: realiza busca por polimorfismos em dados retornados por sequenciamento na plataforma Illumina (OSSOWSKI *et al.*, 2008).

2.2 Montagem dos genomas

Depois de pré-processados, os *reads* dos metagenomas podem ser montados. Nessa etapa, os fragmentos são unidos com outros originados do mesmo genoma para formar sequências maiores e, assim, facilitar a análise. É muito difícil realizar a montagem de genomas inteiros a partir de dados de metagenômica, pois o genoma da maioria dos micro-organismos representados na amostra não é inteiramente sequenciado e é difícil atribuir exatamente à qual espécie cada *read* pertence. Em alguns casos, é possível montar grande parte dos genomas para realizar estudos que requerem a estrutura do genoma, como, por exemplo, em análises funcionais que busquem por regiões codificantes (WOOLEY; GODZIK; FRIEDBERG, 2010; KIM *et al.*, 2013; SHARPTON, 2014).

A maioria dos *softwares* de montagem de genomas é desenvolvida para a junção de fragmentos obtidos por sequenciamento de genomas inteiros, no qual os *reads* procedem de somente um organismo. Para a montagem de fragmentos de metagenômica, essas ferramentas são pouco eficazes, pois nesse caso as sequências

apresentam diferentes origens. Além disso, a falta de genomas de referência de micro-organismos não cultiváveis e o pequeno tamanho dos fragmentos gerados pelo sequenciamento tornam a tarefa ainda mais desafiadora (KUMAR *et al.*, 2015).

Existem dois tipos de montagem de genomas que podem ser executados: a montagem baseada em genomas-referência e a montagem *de novo* (KIM *et al.*, 2013; SHARPTON, 2014; OULAS *et al.*, 2015).

2.2.1 Montagem baseada em genomas de referência

Para a realização de montagem baseada em genomas de referência, um ou mais genomas disponíveis em bancos de dados são utilizados como “mapas”, em que os *reads* podem ser posicionados em regiões de similaridade. Quando dois ou mais *reads* dispõem-se um ao lado do outro em relação ao genoma de referência, significa que são sequenciais, portanto, são unidos para formar fragmentos maiores (Figura 6 – a).

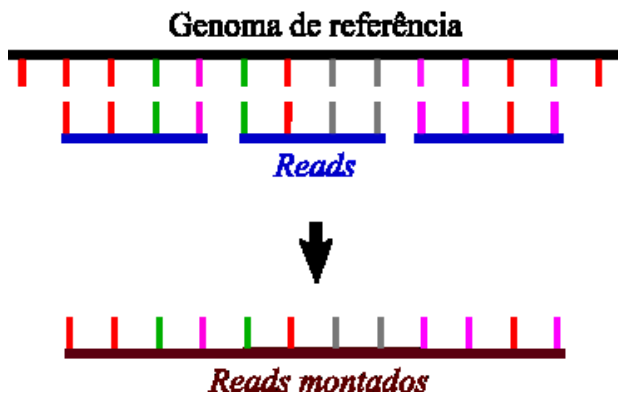
Ferramentas utilizadas para a montagem a partir de genomas de referência incluem MetaAMOS (TREANGEN *et al.*, 2013), Newbler (montagem de *reads* da 454-Roche) e MIRA4 (CHEVREUX *et al.*, 2004). Essas ferramentas têm baixo custo computacional, mas são adequadas para a aplicação em metagenomas de ambientes bem explorados, cuja composição microbiótica já é conhecida. Nesses casos, é mais provável que genomas de micro-organismos próximos aos presentes nesses ambientes já estejam disponíveis nos bancos de dados, podendo assim serem utilizados como referência.

2.2.2 Montagem de novo

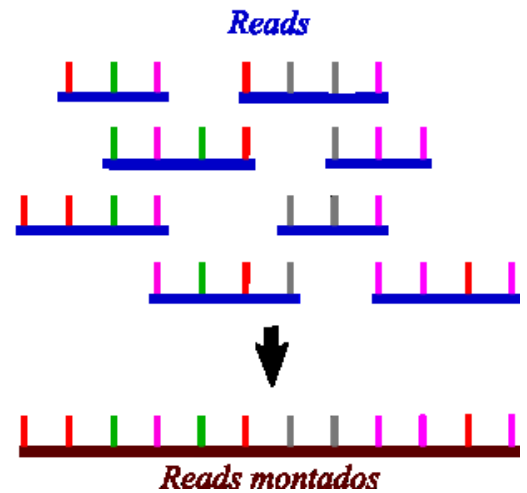
A montagem *de novo* de fragmentos de DNA consiste na junção de fragmentos sem a utilização de genomas de referência (Figura 6 – b). Esses *softwares* contêm algoritmos mais complexos do que os que utilizam genomas de referência, como, por exemplo, grafos de-Bruijin (COMPEAU; PEVZNER; TESLER, 2011). Programas para montagem *de novo* também são computacionalmente mais caros e demandam computadores com grandes quantidades de memória para longos processos computacionais. Abyss (SIMPSON *et al.*, 2009), Velvet (ZERBINO; BIRNEY, 2008), SOAP (LI *et al.*, 2008) e EULER (PEVZNER; TANG; WATERMAN, 2001) são exemplos de ferramentas de montagem *de novo* de fragmentos de DNA.

Figura 6 – Esquema ilustrando os métodos de montagem de *reads* de metagenômica

a) MONTAGEM A PARTIR DE GENOMAS DE REFERÊNCIA



b) MONTAGEM DE NOVO



Fonte: Elaboração da autora.

2.2.3 Próxima geração de ferramentas de montagem

As sequências retornadas pelo sequenciamento de metagenomas são originadas de organismos diferentes, mas a maioria dos algoritmos citados acima foi desenvolvida para a montagem de *reads* sequenciados a partir de genomas únicos. Deste modo, ocorrem algumas dificuldades na aplicação dessas ferramentas em dados de metagenômica. Uma das maiores dificuldades está relacionada à classificação taxonômica dos fragmentos metagenômicos, pois espécies próximas podem apresentar grandes variações em suas sequências, assim como espécies mais distantes podem apresentar sequências muito similares. Além disso, a diferença na quantidade de DNA de cada espécie na amostra também interfere na montagem do genoma. Outros algoritmos têm sido desenvolvidos para superar essas dificuldades.

Dois exemplos de *softwares* da nova geração são Meta-Velvet-SL (NAMIKI *et al.*, 2012; AFIAHAYATI; MULYANA, 2015) e Meta-IDBA (PENG *et al.*, 2011), que combinam ferramentas de *binning* (mais detalhes sobre *binning* estão apresentados abaixo) e de montagem de *reads* para unir os fragmentos de metagenômica com mais acurácia. Esses programas utilizam valores de frequências de oligonucleotídeos (*k-mers*) para detectar torções nos grafos de-Bruijjn e limiares de *k-mers*, para decompor os grafos em subgrafos. Os fragmentos são conectados baseados nas informações obtidas da decomposição dos subgrafos e no agrupamento das sequências em diferentes espécies.

2.3 Análise taxonômica e *binning*

O processo de *binning* pode ser realizado para analisar a diversidade taxonômica de uma amostra de metagenômica (MANDE; MOHAMMED; GHOSH, 2012). A partir desse procedimento, os *reads*, que até então não têm sua origem taxonômica determinada, são agrupados em táxons de acordo com diferentes características da sequência. A realização do *binning* possibilita a quantificação dos micro-organismos de diferentes táxons que estão presentes no ambiente e a redução da complexidade do conjunto de dados para facilitar posteriores fases do estudo, como a montagem de fragmentos (que pode ser feita antes ou depois do *binning*) ou análise funcional (SHARPTON, 2014).

A partir de análises taxonômicas é possível estudar o papel da composição das comunidades microbianas nos ecossistemas em que fazem parte. Zarraindia *et al.* mostraram que a diversidade de espécies associadas aos órgãos das parreiras (folhas, flores, frutos e raízes) e ao solo onde estão plantadas é importante para definir o sabor final dos vinhos produzidos (ZARRAINDIA *et al.*, 2015). Outro estudo demonstrou que a microbiota intestinal pode influenciar no desenvolvimento da obesidade (RIDAURA *et al.*, 2013). Estes resultados, dentre outros, evidenciam a importância da composição microbiana em diversos ecossistemas.

Apesar da importância das análises taxonômicas, esse procedimento representa grandes desafios para os pesquisadores e desenvolvedores de *software*, principalmente por causa do comprimento curto dos *reads* obtidos pelas NGS. Por serem muito pequenos, muitas vezes os fragmentos não apresentam informações suficientes para que seja possível classificá-los. Conseqüentemente, muitos *reads* acabam sendo excluídos do agrupamento. Para minimizar esses problemas, pode ser realizada a pré-montagem dos fragmentos e deve ser utilizada a ferramenta de análise mais adequada para o tipo de dados que se está pesquisando (KIM *et al.*, 2013).

Existem basicamente duas categorias de ferramentas que utilizam diferentes abordagens para a classificação taxonômica dos *reads* de metagenômica: i) similaridade de sequências e ii) composição de sequências.

2.3.1 Ferramentas de similaridade de sequências

As ferramentas de similaridade de sequências classificam os *reads* desconhecidos de acordo com sua similaridade com sequências conhecidas armazenadas em bancos de dados. Primeiramente, os dados são alinhados com ferramentas de alinhamento como BLAST. Em seguida, os dados retornados são submetidos a *softwares* de análise de metagenômica que utilizam as informações de similaridade para realizar inferências taxonômicas e filogenéticas (KIM *et al.*, 2013).

O método de similaridade de sequências para a classificação taxonômica dos metagenomas fornece maior resolução e acurácia da análise do que o *binning*, a partir da composição de sequências (descrito na seção 3.3.2). Entretanto, seu custo computacional é maior e aumenta exponencialmente com a diminuição do comprimento dos *reads* (LIU *et al.*, 2013; SHARPTON, 2014).

Abaixo, estão citadas ferramentas de análise taxonômica por similaridade popularmente utilizadas (SHARPTON, 2014):

- **MEGAN**: utiliza BLAST para comparar os *reads* de metagenômica com bancos de dados de sequências identificadas com a taxonomia do NCBI. Em seguida, o *software* atribui à sequência o táxon do “último ancestral comum” (LCA, do inglês, *last common ancestor*), dentre os que contêm homologia com o *read* (HUSON *et al.*, 2007);
- **MG-RAST**: classifica cada *read* taxonomicamente a partir da reconstrução filogenética das sequências de referência às quais ele apresente similaridade (MEYER *et al.*, 2008);
- **CARMA**: utiliza modelos-índice de evolução gene-família atribuídos aos melhores alinhamentos para classificar os *reads* taxonomicamente (GERLACH; STOYE, 2011).

2.3.2 Ferramentas de composição de sequências

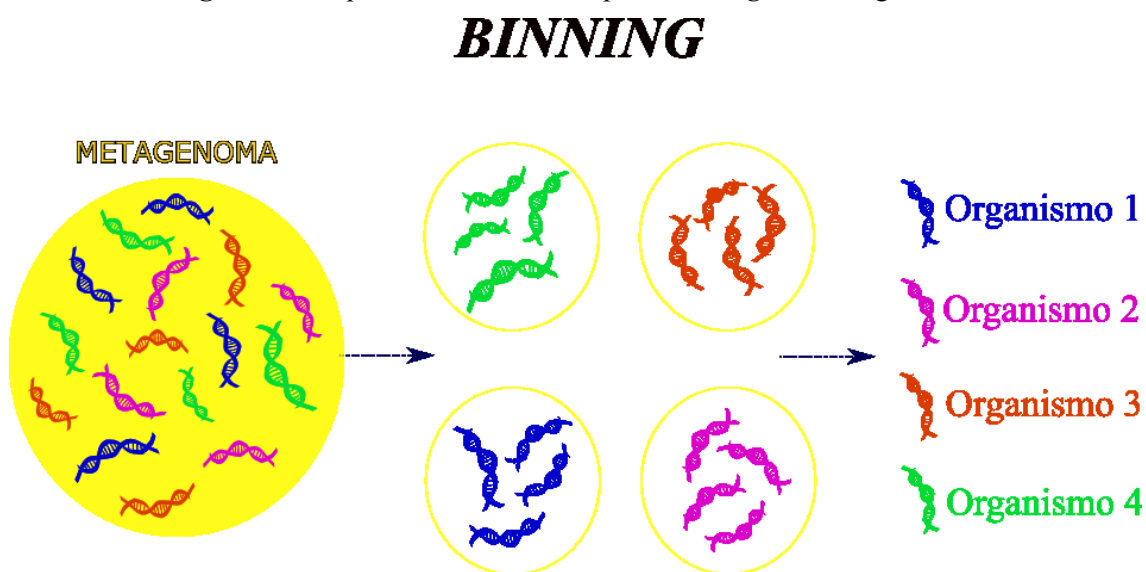
As ferramentas de composição de sequências utilizam características intrínsecas das sequências (*e.g.* conteúdo GC, frequência de oligonucleotídeos, índice de utilização de códons, assinaturas periódicas) para classificá-las taxonomicamente. Essas características, também chamadas de **assinaturas genômicas**, são moldadas em cada grupo taxonômico ao longo da evolução, de acordo com as pressões evolutivas às quais os micro-organismos estão sujeitos. Levando em conta que organismos filogeneticamente mais próximos apresentarão maior similaridade na composição de suas sequências, é possível agrupar ou classificar os *reads* de metagenômica, de acordo com essas características (CAMPBELL; RUAN; WEI, 1999; THOMAS; GILBERT; MEYER, 2012).

As ferramentas que utilizam assinaturas genômicas são mais rápidas e custam menos computacionalmente do que as ferramentas de similaridade. Entretanto, apresentam dificuldade na identificação de fragmentos muito pequenos (*i.e.*, 150 pb), pois eles não contêm informação suficiente para uma classificação eficiente (MANDE; MOHAMMED; GHOSH, 2012).

Os programas de composição de sequências costumam utilizar algoritmos de aprendizagem de máquina supervisionados ou não supervisionados, para agrupar e

classificar os *reads*. Os supervisionados utilizam genomas conhecidos para que o algoritmo reconheça os padrões de cada grupo taxonômico; os *reads* são atribuídos a um táxon, de acordo com as características reconhecidas pela máquina. Algoritmos não supervisionados não utilizam sequências de referência, eles comparam um fragmento com outro reconhecendo padrões comuns entre eles e agrupando-os em conjuntos, de acordo com suas características compartilhadas. *Binning* a partir de algoritmos não supervisionados reúne os *reads* em grupos taxonomicamente distintos, mas é necessária a utilização de ferramentas adicionais para atribuir táxons a cada agrupamento (BRAGG; TYSON, 2014).

Figura 7 – Esquema ilustrando a etapa de *binning* dos metagenomas



Fonte: Elaborada pela autora.

Os *softwares* mais popularmente utilizados para o *binning* por composição de sequências são os de Kim *et al.* (2013) e Sharpton (2014):

- **PhyloPithia** e **PhyloPithiaS**: primeiramente, treinam o algoritmo de aprendizagem supervisionada de máquina *support vector machine* (SVM) com dados de frequência de oligonucleotídeos (*k-mers*) de sequências conhecidas. Em seguida, os *reads* são classificados a partir de suas características em comparação com o modelo obtido (MCHARDY *et al.*, 2007; PATIL *et al.*, 2011);
- **Phymm**: ferramenta supervisionada que utiliza modelos Markovianos interpolados para classificar os *reads* de metagenômica. Adequado para sequências pequenas originadas de NGS (BRADY; SALZBERG, 2009; BRADY; SALZBERG, 2011);

- **NBC**: utiliza classificador supervisionado de Naive Bayes treinado com perfis de frequência de *k-mers* de cada grupo taxonômico. Adequado para sequências pequenas originadas de NGS (ROSEN; REICHENBERGER; ROSENFELD, 2011);
- **TACOA**: ferramenta não supervisionada que utiliza “regra do vizinho mais próximo” (*k-nearest neighbor*), para agrupar os *reads* (DIAZ *et al.*, 2009).

2.3.3 Ferramentas híbridas

Há ferramentas que utilizam tanto a abordagem de composição de sequências quanto o alinhamento para a classificação taxonômica dos *reads* de metagenômica. Essas ferramentas, também conhecidas como híbridas, permitem compensar as vantagens e desvantagens de ambas as abordagens. Alguns exemplos estão descritos abaixo:

- **PhymmBL**: combina a ferramenta Phymm, descrita anteriormente, com alinhamentos em BLAST para aumentar a precisão da classificação (BRADY; SALZBERG, 2009);
- **RITA**: combina BLAST com a ferramenta NBC (descrita anteriormente), mas atribui mais peso aos resultados do BLAST (MACDONALD; PARKS; BEIKO, 2012);
- **SPHINX**: primeiramente, SPHINX compara a composição de tetranucleotídeos (4-mers) do *read* com a dos genomas de referência. Esse procedimento possibilita uma pré-filtragem, mantendo apenas os grupos taxonômicos a que ele possa pertencer. Depois, utiliza algoritmos de alinhamento de sequência, para atribuir ao fragmento táxons mais restritos (MOHAMMED *et al.*, 2011).

2.4. Análise funcional

A análise funcional dos metagenomas fornece informações sobre as funções codificadas nos genomas dos micro-organismos da comunidade, respondendo à pergunta: *O que os micro-organismos estão fazendo no ambiente estudado?*

A partir da caracterização dos genes do metagenoma, é possível traçar um perfil funcional da comunidade microbiana que pode ser utilizado para comparar metagenomas de diferentes ambientes, revelar a presença de novos genes ou fornecer informações do mesmo ambiente em diferentes condições (SHARPTON, 2014).

A primeira etapa da análise funcional do metagenoma consiste na identificação de sequências codificadas dentre os fragmentos. Em seguida, sua função é identificada a partir de sua comparação com dados de referência de função conhecida, como genes, proteínas e vias metabólicas disponíveis em bancos de dados.

2.4.1 Identificação de genes

A busca por genes em meio ao metagenoma pode ser realizada com ou sem a montagem dos *reads*. Se os fragmentos estiverem montados e as sequências codificadoras estiverem completas, a predição de genes pode muitas vezes ser realizada com os mesmos programas de busca de genes em genomas completos, desde que não requeiram parâmetros espécie-específicos, pois os *reads* de metagenomas são originados de diversas linhagens. A análise funcional de metagenomas não montados é mais desafiadora, pois envolve a predição de sequências codificadoras incompletas (SHARPTON, 2014).

A maioria dos programas de predição de genes utiliza informações de códon (*i.e.*, códon de iniciação – AUG), para identificar quadros de leitura abertos (ORFs, do inglês, *open reading frames*) e classificar as sequências como codificadoras ou não codificadoras (OULAS *et al.*, 2015).

Os *softwares* de identificação de genes em metagenomas utilizam diferentes modelos de predição de genes, como aprendizagem de máquina (HAYES; BORODOVSKY, 1998), modelos ocultos de Markov (HMM, do inglês, *hidden Markov models*) (YADA *et al.*, 1999) e valores de tendência de utilização de di-códons (NGUYEN *et al.*, 2009). Abaixo, estão citados alguns algoritmos de busca de genes em metagenomas (KIM *et al.*, 2013).

- **MetaGeneMark**: utiliza valores de tendências de uso de códon incorporados a HMMs (ZHU; LOMSADZE; BORODOVSKY, 2010);
- **Prodigal**: utiliza algoritmos de aprendizagem de máquina para a predição de genes em metagenomas (HYATT *et al.*, 2010);
- **MetaGene**: utiliza valores de tendência de utilização de di-códons (TANENBAUM *et al.*, 2010);
- **FragGeneScan**: utiliza modelos de erro de sequenciamento e HMM incorporando valores de tendência de utilização de códon (RHO; TANG; YE, 2010);
- **MetaGeneAnnotator**: utiliza algoritmos de aprendizagem de máquina com informações de tendência de utilização de di-códons (NOGUCHI; TANIGUCHI; ITOH, 2008);
- **Glimmer-MG**: utiliza HMM para a predição de genes (SALZBERG *et al.*, 1998);
- **Orphelia**: utiliza algoritmos de aprendizagem de máquina com informações de tendência de utilização de di-códons (HOFF *et al.*, 2009).

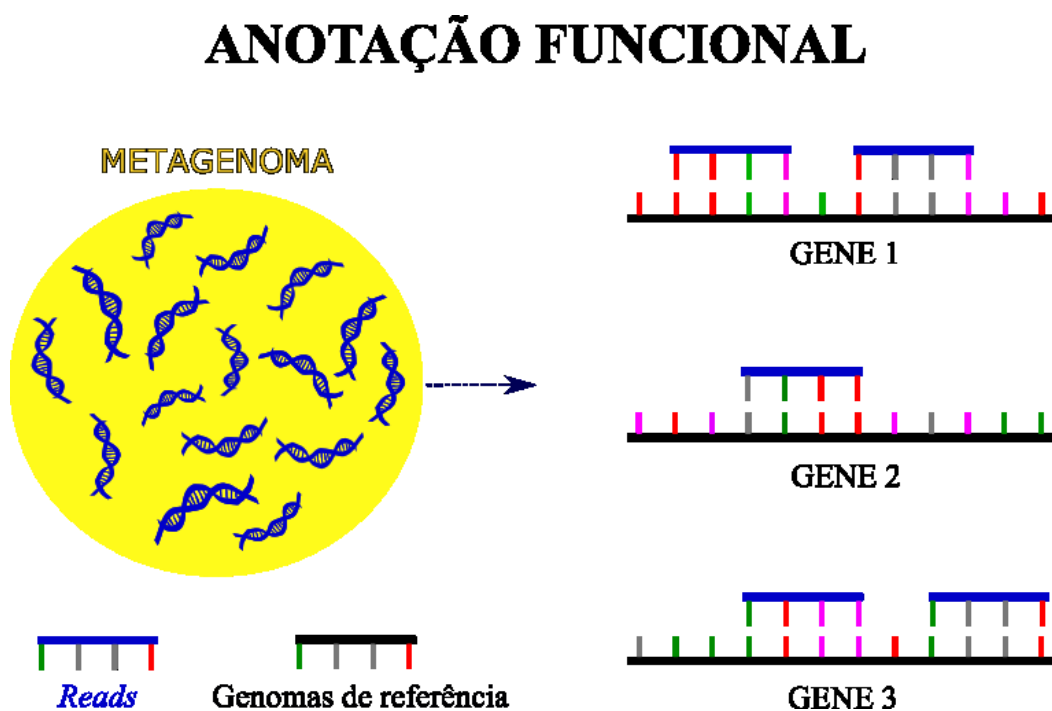
2.4.2 Anotação funcional dos genes

Depois que os genes são identificados no metagenoma, a próxima etapa é encontrar a função desempenhada por eles no microbioma. Essa etapa tem maior custo computacional devido ao grande tamanho dos metagenomas e, muitas vezes, o comprimento muito pequeno dos *reads*.

Os genes encontrados no metagenoma são comparados com genes já caracterizados disponíveis nos bancos de dados, a partir de ferramentas de alinhamento de sequência, como BLAST. A função que desempenham no microbioma é inferida de acordo com a similaridade dos genes desconhecidos com os genes de referência (OULAS *et al.*, 2015).

As comparações dos genes desconhecidos com os de referência possibilitam a determinação de quais funções e vias estão presentes no metagenoma e sua quantidade, pois os bancos de dados fornecem informações sobre domínios e classificação de proteínas por suas funções (BELLA *et al.*, 2013). Os bancos mais utilizados para a busca de informações de genes conhecidos incluem: KEGG (*Kyoto Encyclopedia of Genes and Genomes*) (KANEHISA; GOTO, 2000), COG (*Clusters of Orthologous Groups System*) (TATUSOV *et al.*, 2003), Pfam (BATEMAN *et al.*, 2004), CDD (Conserved Domains Database) (MARCHLER-BAUER *et al.*, 2005), SEED (OVERBEEK *et al.*, 2005), TIGRFAM (SELENGUT *et al.*, 2007) e eggNOG (MULLER *et al.*, 2010).

Figura 8 – Esquema ilustrando a etapa de anotação funcional dos metagenomas



Fonte: Elaborada da autora.

2.4.3 Ferramentas genéricas

Há ferramentas que foram desenvolvidas para executar a análise funcional dos metagenomas de forma mais simplificada, promovendo a interação de algoritmos de identificação de genes, alinhamento de sequências e bancos de dados. Muitas delas também processam outras etapas como filtragem dos dados e comparação de metagenomas e fornecem visualização acessível dos resultados. Esses programas estão descritos abaixo (THOMAS; GILBERT; MEYER, 2012; BELLA *et al.*, 2013; OULAS *et al.*, 2015):

- **MG-RAST**: ferramenta que possui banco de dados próprio e executa controle de qualidade, predição de genes, anotação funcional e ambiente de comparação de metagenomas, retornando ao usuário dados em forma de perfil de abundância e informações taxonômicas.

Primeiramente, o *software* MG-RAST prediz os genes dos metagenomas. Em seguida, executa o alinhamento com a ferramenta BLAT (*BLAST-like alignment tool*) (KENT, 2002) e seleciona os genes dos bancos de dados com melhores homologias (acima de 70% de identidade), em comparação aos *reads*. A partir desse ponto da análise, são utilizados os genes homólogos identificados pelo alinhamento, e não mais os genes encontrados nos metagenomas.

Embora a utilização dos genes homólogos e não dos originais ocasione uma série de limitações, fornece maior rapidez ao método, pois os dados dos homólogos já foram pré-processados. Desse modo, a única etapa computacionalmente intensa é a do alinhamento dos genes do metagenoma com os de bancos de dados (MEYER *et al.*, 2008; GLASS *et al.*, 2010);

- **IMG/MER 4**: *software* que executa controle de qualidade, predição de genes e anotação funcional.

Inicia o processamento dos metagenomas com a predição de todos os genes do metagenoma. Depois, os genes originais do metagenoma são submetidos à identificação de proteínas correspondentes no banco de dados PFAM.

Os genes desconhecidos são associados a PFAM a partir de perfis de HMM, e então suas funções são identificadas com COGs. Para a anotação de sequências de proteínas são utilizados bancos de dados com matrizes de corte posição-específica (PSSMs, do inglês, *position-specific scoring matrix*) para COGs, que são obtidos do NCBI. Além disso, os genes são caracterizados usando KEGG e números EC e sua filogenia é atribuída utilizando buscas de

similaridade. IMG/MER pode utilizar seus próprios dados, uma vez que apresenta um grande repositório público de genomas.

Em comparação com o MG-RAST, a vantagem do IMG/MER é que o último utiliza o banco de dados PFAM, que não é aceito pelo MG-RAST. PFAM fornece informações muito mais detalhadas do que o COGs, único banco de dados de proteínas utilizado pelo MG-RAST. Além disso, PFAM fornece uma análise com maior cobertura do que COGs, pois o número de metagenomas inserido no sistema do PFAM é maior do que o do COGs. Entretanto, a maior limitação do IMG/MER é o crescimento exponencial do número de genes, característica não vinculada ao MG-RAST, pois ele não mantém os metagenomas para análise (MARKOWITZ *et al.*, 2013);

- **EBI Metagenomics service:** utiliza estrutura de metadata e formatos que obedecem aos padrões do GSC (Consórcio de Padrões Genômicos, sigla em inglês, *Genomic Standards Consortium*). Além disso, está adotando um novo esquema de dados que atualmente está sendo hospedado pelo EBI-EMBL: o ENA (Arquivo Europeu de Nucleotídeos. Sigla em inglês para *European Nucleotide Archive*). O ENA tem o objetivo de integrar dados derivados das tecnologias de sequenciamento em um padrão mutualmente aceito.

O EBI Metagenomics oferece um serviço de análise de dados genômicos obtidos por *shotgun* e também por genes marcadores. Isso permite a extração dos dados de rRNA de metagenomas obtidos por WGS utilizando ferramentas como rRNASelector (LEE; YI; CHUN, 2011), que analisa metagenômica por genes marcadores. Também é compatível com ferramentas de análise 16S rRNA, como Qiime (citado no item sobre “Metagenômica a partir do gene 16S rRNA”) para atribuição taxonômica correta dessas sequências.

Para busca de sequências codificadoras nos metagenomas, EBI Metagenomics utiliza FragGeneScan para identificar as sequências codificadoras de proteínas. Para a anotação funcional, utiliza bancos de dados como Interpro, que é um sistema cumulativo e composto de múltiplos bancos de dados de famílias de proteínas e permite predição de domínios de proteínas e atribuição funcional aos genes.

EBI Metagenomics fornece arquivamento de dados via ENA e números de acesso únicos para cada conjunto de dados submetido. As políticas de arquivamento requerem que os dados sejam públicos, entretanto, há um período de dois anos, a partir da submissão, durante o qual os dados são

mantidos em modo privativo, até que o usuário publique os resultados analisados (HUNTER *et al.*, 2014);

- **CAMERA**: serviço de nuvem *online* que fornece ferramentas de *software* hospedadas e infraestrutura computacional de alto desempenho para a análise de dados de metagenômica. Permite a publicação do conjunto de dados e comparação entre os metagenomas.

É uma ferramenta flexível, que permite que o usuário interfira no processo de análise. Entretanto, essa característica exige experiência e conhecimento do usuário para que a análise possa ser executada de maneira correta e os resultados possam ser corretamente interpretados (SESHADRI *et al.*, 2007);

- **MEGAN**: ferramenta para visualização de resultados de análises taxonômicas ou funcionais derivados do BLAST. Apresenta diversas opções de visualização, como dendrogramas, gráficos de barras e outros tipos de gráficos que permitem que dados hierárquicos sejam explorados e torna a análise mais visualmente acessível (HUSON *et al.*, 2007).

3 Conclusão e perspectivas futuras

Nas últimas duas décadas, avanços importantes nas subáreas metagenômicas foram promovidos. A disponibilidade de métodos de extração de DNA de quase todo tipo de amostra ambiental; a diminuição brusca nos preços do sequenciamento; a evolução das tecnologias NGS e progressos no campo da computação, como poder de processamento e armazenamento e desenvolvimento de algoritmos mais complexos, possibilitaram a obtenção de análises de comunidades microbióticas cada vez mais complexas e completas (KUMAR *et al.*, 2015).

Os progressos no campo da metagenômica ainda apresentam potencial de progressão: novas tecnologias de sequenciamento, como PacBio ou sequenciamento por nanoporo, prometem maiores facilidades para os protocolos de análise desde a montagem até o processo de anotação funcional. Além das plataformas, as próprias ferramentas computacionais vêm se aprimorando no decorrer do tempo: a crescente quantidade de genomas de referência de micro-organismos cultiváveis e não cultiváveis fornece um aumento progressivo na quantidade de informações disponíveis, consequentemente aumentando a precisão dos algoritmos de análises taxonômica e funcional (OULAS *et al.*, 2015).

A falta de padronização dos dados é outra dificuldade que está sendo superada: o GSC vem desenvolvendo protocolos como o MARMS (em inglês, *Minimum Analysis Requirements of Metagenome Sequences*, ou “requisitos mínimos para a análise de sequências metagenômicas” em português). MARMS será composta de metodologias padronizadas e consensos na escolha de *softwares*, etapas de análise, valores de limite e parâmetros aplicados nas análises de metagenômica. Esse projeto tem o objetivo de minimizar os vieses que podem ser gerados pela análise de múltiplas metodologias. O GSC também pretende facilitar a troca de dados padronizando seus formatos e estruturas mutualmente aceitáveis que auxiliarão a troca de informações no campo da microbiologia ambiental (OULAS *et al.*, 2015).

Entretanto, o desenvolvimento de ferramentas e bancos de dados para estudos metagenômicos ainda é incipiente, e há muitas limitações para serem contornadas (KIM *et al.*, 2013). Primeiramente, a precisão das ferramentas de análise taxonômica e funcional necessita ser melhorada; o aumento dos genomas de referência está auxiliando para a melhora da precisão, mas ainda há o que aprimorar nos próprios algoritmos. Segundo, é necessária uma melhora na infraestrutura computacional para gerenciamento, disponibilidade e armazenamento de dados, pois o rápido aumento do tamanho e da quantidade de dados está superando a capacidade computacional. Terceiro, é necessário o aperfeiçoamento de métodos estatísticos, especialmente para metagenomas originados de comunidades complexas, nas quais os dados de táxons e genes podem ser esparsos. Por último, é necessário o aprimoramento de sistemas experimentais para a manipulação de comunidades microbianas; é possível estabelecer a relação dos micro-organismos com a composição do ambiente, modificando a composição dos meios de cultura (*e.g.*, administração de antibióticos, suplementação probiótica, transplante de comunidades, mudanças físicas como de pH, temperatura, pressão) e observando a resposta do microbioma (SHARPTON, 2014).

Os estudos de metagenômica apresentam um potencial para a exploração de comunidades microbianas que nenhuma outra abordagem conseguira antes. Com a superação das dificuldades descritas anteriormente, as comunidades microbianas poderão ser estudadas cada vez com mais rapidez e precisão, proporcionando o aumento do conhecimento sobre a dinâmica das comunidades microbianas e impulsionando o avanço da ciência em diversas áreas.

Referências

- AFIAHAYATI, A.; MULYANA, S. Multiple sequence alignment menggunakan hidden markov model. **Seminar Nasional Informatika (SEMNASIF)**, [S.l.: s.n.], v. 1, n. 1, 2015.
- ALTSCHUL, S. F. et al. Basic local alignment search tool. **Journal of molecular biology**, Elsevier, v. 215, n. 3, p. 403-410, 1990.

- BABRAHAM BIOINFORMATICS. **FASTQC, A quality control tool for high throughput sequence data**. 2016. Disponível em: <http://www.bioinformatics.bbsrc.ac.uk/projects/fastqc/>. Acesso em: 10 jun. 2016.
- BAHASSI, E. M.; STAMBROOK, P. J. Next-generation sequencing technologies: breaking the sound barrier of human genetics. **Mutagenesis**, Oxford University Press, v. 29, n. 5, p. 303-310, 2014.
- BATEMAN, A. *et al.* The pfam protein families database. **Nucleic acids research**, Oxford Univ Press, v. 32, n. suppl 1, p. D138-D141, 2004.
- BELLA, J. M. D. *et al.* High throughput sequencing methods and analysis for microbiome research. **Journal of microbiological methods**, Elsevier, v. 95, n. 3, p. 401-414, 2013.
- BERGER, S. A.; KROMPASS, D.; STAMATAKIS, A. Performance, accuracy, and web server for evolutionary placement of short sequence reads under maximum likelihood. **Systematic biology**, Oxford University Press, p. syr010, 2011.
- BIK, H. M. Deciphering diversity and ecological function from marine metagenomes. **The Biological Bulletin**, MBL, v. 227, n. 2, p. 107-116, 2014.
- BLANCA, J. M. *et al.* ngs_backbone: a pipeline for read cleaning, mapping and snp calling using next generation sequence. **BMC genomics**, BioMed Central, v. 12, n. 1, p. 1, 2011.
- BRADY, A.; SALZBERG, S. Phymmbl expanded: confidence scores, custom databases, parallelization and more. **Nature methods**, Nature Publishing Group, v. 8, n. 5, p. 367-367, 2011.
- BRADY, A.; SALZBERG, S. L. Phymm and phymmbl: metagenomic phylogenetic classification with interpolated markov models. **Nature methods**, Nature Publishing Group, v. 6, n. 9, p. 673-676, 2009.
- BRAGG, L.; TYSON, G. W. Metagenomics using next-generation sequencing. **Environmental Microbiology: Methods and Protocols**, Springer, p. 183-201, 2014.
- CAI, Y.; SUN, Y. Esprit-tree: hierarchical clustering analysis of millions of 16s rRNA pyrosequences in quasilinear computational time. **Nucleic acids research**, Oxford Univ Press, p. gkr349, 2011.
- CAMPBELL, S. A.; RUAN, S.; WEI, J. Qualitative analysis of a neural network model with multiple time delays. **International Journal of Bifurcation and Chaos**, World Scientific, v. 9, n. 08, p. 1585-1595, 1999.
- CAPORASO, J. G. *et al.* Qiime allows analysis of high-throughput community sequencing data. **Nature methods**, Nature Publishing Group, v. 7, n. 5, p. 335-336, 2010.
- CHEVREUX, B. *et al.* Using the miraest assembler for reliable and automated mRNA transcript assembly and snp detection in sequenced ests. **Genome research**, Cold Spring Harbor Lab, v. 14, n. 6, p. 1147-1159, 2004.
- CLEMENTE, J. C.; JANSSON, J.; VALIENTE, G. Flexible taxonomic assignment of ambiguous sequencing reads. **BMC bioinformatics**, BioMed Central, v. 12, n. 1, p. 1, 2011.
- COLE, J. R. *et al.* The ribosomal database project: improved alignments and new tools for rRNA analysis. **Nucleic acids research**, Oxford Univ Press, v. 37, n. suppl 1, p. D141-D145, 2009.
- COMPEAU, P. E.; PEVZNER, P. A.; TESLER, G. How to apply de bruijn graphs to genome assembly. **Nature biotechnology**, Nature Publishing Group, v. 29, n. 11, p. 987-991, 2011.
- COUNCIL, N. R. **The New Science of Metagenomics: Revealing the Secrets of Our Microbial Planet**. The National Academies Press, 2007. ISBN 978-0-309-10676-4. Disponível em: <http://www.nap.edu/catalog/11902/the-new-science-of-metagenomics-revealing-the-secrets-of-our>. Acesso em: jun. 2016.
- DESANTIS, T. Z. *et al.* Greengenes, a chimera-checked 16s rRNA gene database and workbench compatible with arb. Applied and environmental microbiology. **Am Soc Microbiol**, v. 72, n. 7, p. 5069-5072, 2006.
- DEVARAJ, S.; HEMARAJATA, P.; VERSALOVIC, J. The human gut microbiome and body metabolism: implications for obesity and diabetes. Clinical chemistry. **Am Assoc Clin Chem**, v. 59, n. 4, p. 617-628, 2013.

- DIAZ, N. N. *et al.* Tacoa—taxonomic classification of environmental genomic fragments using a kernelized nearest neighbor approach. **BMC bioinformatics**, BioMed Central Ltd, v. 10, n. 1, p. 56, 2009.
- EDGAR, R. C. Search and clustering orders of magnitude faster than blast. **Bioinformatics**, Oxford Univ Press, v. 26, n. 19, p. 2460-2461, 2010.
- ESPOSITO, A.; KIRSCHBERG, M. How many 16s-based studies should be included in a metagenomic conference? it may be a matter of etymology. **FEMS Microbiology Letters**, v. 351, p. 145-146, 2014.
- FASTX-TOOLKIT. **A short-reads pre-processing tools**. 2016. Disponível em: http://hannonlab.cshl.edu/fastx_toolkit/index.html. Acesso em: 10 jun. 2016.
- FORDE, B. M.; O'TOOLE, P. W. Next-generation sequencing technologies and their impact on microbial genomics. **Briefings in functional genomics**, Oxford University Press, v. 12, n. 5, p. 440-453, 2013.
- FU, L. *et al.* Cd-hit: accelerated for clustering the next-generation sequencing data. **Bioinformatics**, Oxford Univ Press, v. 28, n. 23, p. 3150-3152, 2012.
- GASPAR, J. M.; THOMAS, W. K. Assessing the consequences of denoising marker-based metagenomic data. **PLoS One**, Public Library of Science, v. 8, n. 3, p. e60458, 2013.
- GERLACH, W.; STOYE, J. Taxonomic classification of metagenomic shotgun sequences with carma3. **Nucleic acids research**, Oxford Univ Press, p. gkr225, 2011.
- GLASS, E. M. *et al.* Using the metagenomics rast server (mg-rast) for analyzing shotgun metagenomes. **Cold Spring Harbor Protocols**, Cold Spring Harbor Laboratory Press, v. 2010, n. 1, p. pdb-prot5368, 2010.
- HAAS, B. J. *et al.* Chimeric 16s rRNA sequence formation and detection in sanger and 454-pyrosequenced PCR amplicons. **Genome research**, Cold Spring Harbor Lab, v. 21, n. 3, p. 494-504, 2011.
- HANDELSMAN, J. Metagenomics: application of genomics to uncultured microorganisms. **Microbiology and molecular biology reviews**, Am Soc Microbiol, v. 68, n. 4, p. 669-685, 2004.
- HANDELSMAN, J. *et al.* Molecular biological access to the chemistry of unknown soil microbes: a new frontier for natural products. **Chemistry & biology**, Elsevier, v. 5, n. 10, p. R245-R249, 1998.
- HAYES, W. S.; BORODOVSKY, M. How to interpret an anonymous bacterial genome: machine learning approach to gene identification. **Genome research**, Cold Spring Harbor Lab, v. 8, n. 11, p. 1154-1171, 1998.
- HOFF, K. J. *et al.* Orphelia: predicting genes in metagenomic sequencing reads. **Nucleic acids research**, Oxford Univ Press, v. 37, n. suppl 2, p. W101-W105, 2009.
- HUNTER, S. *et al.* Ebi metagenomics – a new resource for the analysis and archiving of metagenomic data. **Nucleic acids research**, Oxford Univ Press, v. 42, n. D1, p. D600-D606, 2014.
- HUSON, D. H. *et al.* Megan analysis of metagenomic data. **Genome research**, Cold Spring Harbor Lab, v. 17, n. 3, p. 377-386, 2007.
- HYATT, D. *et al.* Prodigal: prokaryotic gene recognition and translation initiation site identification. **BMC bioinformatics**, BioMed Central, v. 11, n. 1, p. 1, 2010.
- KANEHISA, M.; GOTO, S. Kegg: kyoto encyclopedia of genes and genomes. **Nucleic acids research**, Oxford Univ Press, v. 28, n. 1, p. 27-30, 2000.
- KENT, W. J. Blat-the blast-like alignment tool. **Genome research**, Cold Spring Harbor Lab, v. 12, n. 4, p. 656-664, 2002.
- KIM, M. *et al.* Analytical tools and databases for metagenomics in the next generation sequencing era. **Genomics & informatics**, v. 11, n. 3, p. 102-113, 2013.
- KIM, O.-S. *et al.* Introducing ezTaxon-e: a prokaryotic 16s rRNA gene sequence database with phylotypes that represent uncultured species. **International journal of systematic and evolutionary microbiology**, Microbiology Society, v. 62, n. 3, p. 716-721, 2012.

- KUMAR, S. *et al.* Metagenomics: Retrospect and prospects in high throughput age. **Biotechnology research international**, Hindawi Publishing Corporation, 2015.
- KUNIN, V. *et al.* A bioinformatician's guide to metagenomics. **Microbiology and molecular biology reviews**, Am Soc Microbiol, v. 72, n. 4, p. 557-578, 2008.
- LAND, M. *et al.* Insights from 20 years of bacterial genome sequencing. **Functional & integrative genomics**, Springer, v. 15, n. 2, p. 141-161, 2015.
- LEE, H. *et al.* Third-generation sequencing and the future of genomics. **bioRxiv, Cold Spring Harbor Labs Journals**, p. 048603, 2016.
- LEE, J.-H.; YI, H.; CHUN, J. rrnselector: a computer program for selecting ribosomal rna encoding sequences from metagenomic and metatranscriptomic shotgun libraries. **The Journal of Microbiology**, Springer, v. 49, n. 4, p. 689-691, 2011.
- LI, R. *et al.* Soap: short oligonucleotide alignment program. **Bioinformatics**, Oxford Univ Press, v. 24, n. 5, p. 713-714, 2008.
- LIU, Y. *et al.* Gene prediction in metagenomic fragments based on the svm algorithm. **BMC bioinformatics**, BioMed Central Ltd, v. 14, n. Suppl 5, p. S12, 2013.
- MACDONALD, N. J.; PARKS, D. H.; BEIKO, R. G. Rapid identification of high-confidence taxonomic assignments for metagenomic data. **Nucleic acids research**, Oxford Univ Press, p. gks335, 2012.
- MANDE, S. S.; MOHAMMED, M. H.; GHOSH, T. S. Classification of metagenomic sequences: methods and challenges. **Briefings in bioinformatics**, Oxford Univ Press, p. bbs054, 2012.
- MARCHLER-BAUER, A. *et al.* Cdd: a conserved domain database for protein classification. **Nucleic acids research**, Oxford Univ Press, v. 33, n. suppl 1, p. D192-D196, 2005
- MARCO, D. **Metagenomics: current innovations and future trends**. [S.l.]: Horizon Scientific Press, 2011.
- MARKOWITZ, V. M. *et al.* Img 4 version of the integrated microbial genomes comparative analysis system. **Nucleic acids research**, Oxford Univ Press, p. gkt963, 2013.
- MATSEN, F. A.; KODNER, R. B.; ARMBRUST, E. V. pplacer: linear time maximum-likelihood and bayesian phylogenetic placement of sequences onto a fixed reference tree. **BMC bioinformatics**, BioMed Central, v. 11, n. 1, p. 1, 2010.
- MCHARDY, A. C. *et al.* Accurate phylogenetic classification of variable-length dna fragments. **Nature methods**, Nature Publishing Group, v. 4, n. 1, p. 63-72, 2007.
- MEYER, F. *et al.* The metagenomics rast server – a public resource for the automatic phylogenetic and functional analysis of metagenomes. **BMC bioinformatics**, BioMed Central Ltd., v. 9, n. 1, p. 386, 2008.
- MICROWINE, A MARIE CURIE INITIAL TRAINING NETWORK. **MicroWine, A Marie Curie Initial Training Network**. 2016. Disponível em: <http://www.microwine.eu/>. Acesso em: 10 jun. 2016.
- MIRARAB, S.; NGUYEN, N.; WARNOW, T. Sepp: Saté-enabled phylogenetic placement. *In*: CITESEER. **Pac Symp Biocomput**. [S.l.], 2012. v. 17, p. 247-258.
- MOHAMMED, M. H. *et al.* Sphinx-an algorithm for taxonomic binning of metagenomic sequences. **Bioinformatics**, Oxford Univ Press, v. 27, n. 1, p. 22-30, 2011.
- MORGAVI, D. P. *et al.* Rumen microbial (meta) genomics and its application to ruminant production. **Animal**, Cambridge Univ Press, v. 7, n. s1, p. 184-201, 2013.
- MOROZOVA, O.; MARRA, M. A. Applications of next-generation sequencing technologies in functional genomics. **Genomics**, Elsevier, v. 92, n. 5, p. 255-264, 2008.
- MULLER, J. *et al.* eggNOG v2. 0: extending the evolutionary genealogy of genes with enhanced non-supervised orthologous groups, species and functional annotations. **Nucleic acids research**, Oxford Univ Press, v. 38, n. suppl 1, p. D190-D195, 2010.
- NAMIKI, T. *et al.* Metavelvet: an extension of velvet assembler to de novo metagenome assembly from short sequence reads. **Nucleic acids research**, Oxford Univ Press, v. 40, n. 20, p. e155-e155, 2012.

- NAWROCKI, E. P.; KOLBE, D. L.; EDDY, S. R. Infernal 1.0: inference of rna alignments. **Bioinformatics**, Oxford Univ Press, v. 25, n. 10, p. 1335-1337, 2009.
- NGUYEN, M. N. *et al.* Di-codon usage for gene classification. *In*: NGUYEN, M. N. **Pattern Recognition in Bioinformatics**. Springer, 2009. p. 211-221.
- NIKOLAKI, S.; TSIAMIS, G. Microbial diversity in the era of omic technologies. **BioMed research international**, Hindawi Publishing Corporation, 2013.
- NOGUUCHI, H.; TANIGUCHI, T.; ITOH, T. Metagene annotator: detecting species-specific patterns of ribosomal binding site for precise gene prediction in anonymous prokaryotic and phage genomes. **DNA research**, Kazusa DNA Resh Ins, v. 15, n. 6, p. 387-396, 2008.
- OSSOWSKI, S. *et al.* Sequencing of natural strains of *Arabidopsis thaliana* with short reads. **Genome research**, Cold Spring Harbor Lab, v. 18, n. 12, p. 2024-2033, 2008.
- OULAS, A. *et al.* Metagenomics: tools and insights for analyzing next-generation sequencing data derived from biodiversity studies. **Bioinformatics and biology insights**, Libertas Academica, v. 9, p. 75, 2015.
- OVERBEEK, R. *et al.* The subsystems approach to genome annotation and its use in the project to annotate 1000 genomes. **Nucleic acids research**, Oxford Univ Press, v. 33, n. 17, p. 5691-5702, 2005.
- PATIL, K. R. *et al.* Taxonomic metagenome sequence assignment with structured output models. **Nature methods**, Nature Publishing Group, v. 8, n. 3, p. 191-192, 2011.
- PATIN, N. V. *et al.* Effects of otu clustering and PCR artifacts on microbial diversity estimates. **Microbial ecology**, Springer, v. 65, n. 3, p. 709-719, 2013.
- PENG, Y. *et al.* Meta-IdBA: a de novo assembler for metagenomic data. **Bioinformatics**, Oxford Univ Press, v. 27, n. 13, p. i94-i101, 2011.
- PETTERSSON, E.; LUNDEBERG, J.; AHMADIAN, A. Generations of sequencing technologies. **Genomics**, v. 93, n. 2, p. 105-111, Feb 2009.
- PEVZNER, P. A.; TANG, H.; WATERMAN, M. S. An Eulerian path approach to DNA fragment assembly. **Proceedings of the National Academy of Sciences**, National Acad Sciences, v. 98, n. 17, p. 9748-9753, 2001.
- PORETSKY, R. *et al.* Strengths and limitations of 16S rRNA gene amplicon sequencing in revealing temporal microbial community dynamics. **PLoS one**, Public Library of Science, v. 9, n. 4, p. e93827, 2014.
- PRUESSE, E.; PEPLIES, J.; GLÖCKNER, F. O. SINA: accurate high-throughput multiple sequence alignment of ribosomal rRNA genes. **Bioinformatics**, Oxford Univ Press, v. 28, n. 14, p. 1823-1829, 2012.
- QUAST, C. *et al.* The SILVA ribosomal rRNA gene database project: improved data processing and web-based tools. **Nucleic acids research**, Oxford Univ Press, v. 41, n. D1, p. D590-D596, 2013.
- QUINCE, C. *et al.* Accurate determination of microbial diversity from 454 pyrosequencing data. **Nature methods**, v. 6, n. 9, p. 639, 2009.
- QUINCE, C. *et al.* Removing noise from pyrosequenced amplicons. **BMC bioinformatics**, BioMed Central, v. 12, n. 1, p. 1, 2011.
- QUINLAN, A. R. *et al.* PyroBayes: an improved base caller for SNP discovery in pyrosequences. **Nature methods**, Nature Publishing Group, v. 5, n. 2, p. 179-181, 2008.
- RHO, M.; TANG, H.; YE, Y. FragGenScan: predicting genes in short and error-prone reads. **Nucleic acids research**, Oxford Univ Press, v. 38, n. 20, p. e191-e191, 2010.
- RIDAURA, V. K. *et al.* Gut microbiota from twins discordant for obesity modulate metabolism in mice. **Science**, American Association for the Advancement of Science, v. 341, n. 6150, p. 1241214, 2013.
- RIESENFELD, C. S.; SCHLOSS, P. D.; HANDELSMAN, J. Metagenomics: genomic analysis of microbial communities. **Annu. Rev. Genet.**, Annual Reviews, v. 38, p. 525-552, 2004.

- ROSEN, G. L.; REICHENBERGER, E. R.; ROSENFELD, A. M. Nbc: the naïve bayes classification tool webserver for taxonomic classification of metagenomic reads. **Bioinformatics**, Oxford Univ Press, v. 27, n. 1, p. 127-129, 2011.
- ROSEN, M. J. *et al.* Denoising per-amplified metagenome data. **BMC bioinformatics**, BioMed Central Ltd., v. 13, n. 1, p. 283, 2012.
- SALZBERG, S. L. *et al.* Microbial gene identification using interpolated markov models. **Nucleic acids research**, Oxford Univ Press, v. 26, n. 2, p. 544-548, 1998.
- SANSCHAGRIN, S.; YERGEAU, E. Next-generation sequencing of 16s ribosomal rna gene amplicons. **JoVE (Journal of Visualized Experiments)**, n. 90, p. e51709-e51709, 2014.
- SCHMIDT, T. M.; DELONG, E.; PACE, N. Analysis of a marine picoplankton community by 16s rna gene cloning and sequencing. **Journal of bacteriology**, Am Soc Microbiol, v. 173, n. 14, p. 4371-4378, 1991.
- SELENGUT, J. D. *et al.* **Tigrfams and genome properties**: tools for the assignment of molecular function and -D264, 2007.
- SESHADRI, R. *et al.* Camera: a community resource for metagenomics. **PLoS biology**, v. 5, n. 3, 2007.
- SHARPTON, T. J. **An introduction to the analysis of shotgun metagenomic data**. Frontiers Research Foundation, 2014.
- SIMPSON, J. T. *et al.* Abyss: a parallel assembler for short read sequence data. **Genome research**, Cold Spring Harbor Lab, v. 19, n. 6, p. 1117-1123, 2009.
- SUN, Y. *et al.* A large-scale benchmark study of existing algorithms for taxonomy-independent microbial community analysis. **Briefings in bioinformatics**, Oxford Univ Press, p. bbr009, 2011.
- TANENBAUM, D. M. *et al.* The jvarkit standard operating procedure for annotating prokaryotic metagenomic shotgun sequencing data. **Standards in genomic sciences**, Springer, v. 2, n. 2, p. 229-237, 2010.
- TATUSOV, R. L. *et al.* The cog database: an updated version includes eukaryotes. **BMC bioinformatics**, BioMed Central, v. 4, n. 1, p. 1, 2003.
- THOMAS, T.; GILBERT, J.; MEYER, F. Metagenomics-a guide from sampling to data analysis. **Microb Inform Exp**, v. 2, n. 3, 2012.
- TREANGEN, T. J. *et al.* Metamos: a modular and open source metagenomic assembly and analysis pipeline. **Genome Biol**, v. 14, n. 1, p. R2, 2013.
- TRINGE, S. G.; HUGENHOLTZ, P. A renaissance for the pioneering 16s rna gene. **Current opinion in microbiology**, Elsevier, v. 11, n. 5, p. 442-446, 2008.
- WANG, Q. *et al.* Naive bayesian classifier for rapid assignment of rna sequences into the new bacterial taxonomy. **Applied and environmental microbiology**, Am Soc Microbiol, v. 73, n. 16, p. 5261-5267, 2007.
- WOOLEY, J. C.; GODZIK, A.; FRIEDBERG, I. A primer on metagenomics. **PLoS Comput Biol**, v. 6, n. 2, p. e1000667, Feb 2010. Disponível em: <http://dx.doi.org/10.1371/journal.pcbi.1000667>. Acesso em: jun. 2016.
- WRIGHT, E. S.; YILMAZ, L. S.; NOGUERA, D. R. Decipher, a searchbased approach to chimera identification for 16s rna sequences. **Applied and environmental microbiology**, Am Soc Microbiol, v. 78, n. 3, p. 717-725, 2012.
- WU, M.; SCOTT, A. J. Phylogenomic analysis of bacterial and archaeal sequences with amphora2. **Bioinformatics**, Oxford Univ Press, v. 28, n. 7, p. 1033-1034, 2012.
- YADA, T. *et al.* Modeling and predicting transcriptional units of Escherichia coli genes using hidden markov models. **Bioinformatics**, Oxford Univ Press, v. 15, n. 12, p. 987-993, 1999.
- ZARRAONAINDIA, I. *et al.* The soil microbiome influences grapevine-associated microbiota. **mBio**, American Society for Microbiology, v. 6, n. 2, p. e02527-14, mar. 2015.

ZERBINO, D. R.; BIRNEY, E. Velvet: algorithms for de novo short read assembly using de bruijn graphs. *Genome research*, Cold Spring Harbor Lab, v. 18, n. 5, p. 821-829, 2008.

ZHOU, J. *et al.* High-throughput metagenomic technologies for complex microbial community analysis: open and closed formats. *MBio*, Am Soc Microbiol, v. 6, n. 1, p. e02288-14, 2015.

ZHU, W.; LOMSADZE, A.; BORODOVSKY, M. Ab initio gene identification in metagenomic sequences. *Nucleic acids research*, Oxford Univ Press, v. 38, n. 12, p. e132-e132, 2010.

1 Introdução

A análise em larga escala de sistemas biológicos possibilitou entender como ocorrem as interações entre os diferentes tipos de moléculas de origem biológica, assim para compreender de forma mais ampla, a função exercida por genes, proteínas e outras moléculas em um determinado contexto biológico (BARABÁSI; OLTVAI, 2004). Um grande desafio da biologia é compreender como populações de organismos são estruturadas a partir das suas interações com o meio ambiente e como sistemas complexos têm evoluído (KARSENTI, 2012). Sistemas complexos e organismos vivos, em particular, surgem a partir da ocorrência de processos dinâmicos simultâneos em diferentes escalas de tempo (KARSENTI, 2012).

Uma das ferramentas usadas para o entendimento da complexidade biológica é a Biologia de Sistemas, uma área interdisciplinar que visa a compreender como funcionam as interações entre populações de moléculas, células e organismos, devido ao aumento da complexidade de processos biológicos (KARSENTI, 2012). O estudo das interações entre os componentes de um sistema biológico envolve a teoria dos grafos, matemática discreta e processamento computacional (BARABÁSI; OLTVAI, 2004; O'MALLEY; DUPRÉ, 2005; SIEGAL *et al.*, 2007). Assim, o uso destas ferramentas e técnicas, aliadas à relação entre a topologia de uma rede biológica e suas propriedades funcionais ou evolucionárias foram utilizadas para descrever as interações de sistemas celulares por meio das redes de livre-escala (ROSVALL; SNEPPEN, 2003; BARABÁSI; OLTVAI, 2004; O'MALLEY; DUPRÉ, 2005; SIEGAL *et al.*, 2007).

Este capítulo aborda os conceitos das redes biológicas e ferramentas de biologia de sistemas.

2 Redes biológicas

Uma rede complexa descreve uma grande variedade de sistemas na natureza e na sociedade fortemente conectados (BARABÁSI; ALBERT, 2002), tais como redes sociais, redes celulares, redes de computadores, redes de interação de proteínas e a internet, entre outros exemplos. O comportamento de sistemas complexos, desde a célula até a internet, caracteriza-se por atividades orquestradas nas quais muitos

¹ Universidade de Caxias do Sul. *E-mail*: dlnotari@ucs.br

² Universidade Federal do Rio Grande do Sul. *E-mail*: diegobonatto@gmail.com

componentes interagem um com o outro, através de interações de um par de componentes (BARABÁSI; ALBERT, 2002; BARABÁSI; OLTVAI, 2004). Estes componentes representam um conjunto de nós que são conectados com outros componentes por uma ligação; cada ligação representa interações entre os dois componentes, sendo que, desta forma, o conjunto de todos os nós e todas as ligações entre os nós forma uma rede (BARABÁSI; ALBERT, 2002; BARABÁSI; OLTVAI, 2004).

Desta forma, uma rede pode representar um sistema biológico completo, no qual os nós e suas ligações podem ser diferentes entidades biológicas, como moléculas, células, entre outras (WESTON; HOOD, 2004; SIEGAL *et al.*, 2007). No contexto da análise de dados proteômicos, os nós são representados pelas proteínas e os conectores pelas reações químicas (BARABÁSI; ALBERT, 2002; ROSVALL; SNEPPEN, 2003; BARABÁSI; OLTVAI, 2004; O'MALLEY; DUPRÉ, 2005; SIEGAL *et al.*, 2007).

Pesquisas realizadas demonstram que redes reais, independentemente de sua idade, função ou escopo, convergem para uma arquitetura homogênea, uma característica que permite a pesquisadores de diferentes áreas usarem a teoria de redes como um paradigma comum (BARABÁSI, 2009). As redes de livre-escala ajudaram a proliferar o estudo sobre as redes em diversos campos da ciência, um campo recente de pesquisa com seus desafios e habilidades (BARABÁSI, 2009).

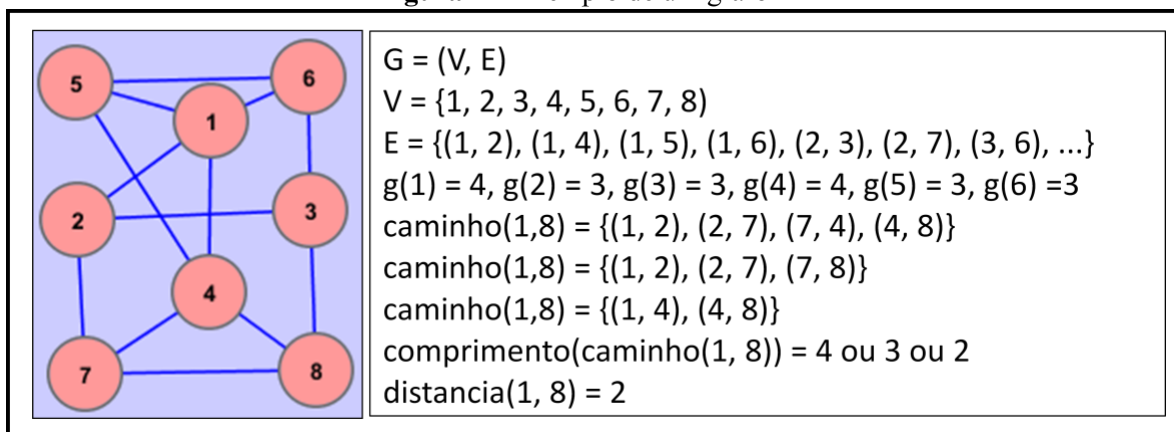
Uma rede biológica pode ser representada por um grafo computacional. Um grafo é uma forma de representar relacionamentos existentes entre pares de objetos (GOODRICH; TAMASSIA, 2007). Um grafo é definido geometricamente como um conjunto de pontos no espaço, que são interconectados por um conjunto de linhas (GIBBONS, 1994). Formalmente, um grafo $G(V, E)$ consiste de um conjunto de vértices³ $V = \{v_1, v_2, \dots\}$ e um conjunto de arestas $E = \{e_1, e_2, \dots\}$, em que uma aresta⁴ representa um conjunto de pares não dirigidos de elementos de V (GIBBONS, 1994; GOODRICH; TAMASSIA, 2007; Shaffer, 2011). Os vértices representam os objetos; as arestas representam relações entre os objetos, e cada aresta define uma relação simétrica (grafos não dirigidos; GIBBONS, 1994; GOODRICH; TAMASSIA, 2007; SHAFFER, 2011).

³ Um vértice também pode ser chamado ponto, nodo ou nó. Para a explicação da teoria dos grafos, adotaremos o termo *vértice* e, a partir da explicação do seu uso na biologia de sistemas, utilizaremos o termo *nó*.

⁴ Uma aresta também é chamada de elo, ligação ou conector. Para a explicação da teoria dos grafos adotaremos o termo *aresta* e, a partir da explicação do seu uso na biologia de sistemas, utilizaremos o termo *conector* ou *ligação*.

Um grafo pode ser representado visualmente usando pequenos círculos para vértices e retas ou curvas para arestas. A Figura 1 apresenta exemplo de um grafo $G = (V, E)$, em que V representa os vértices e E representa as arestas entre os vértices.

Figura 1 – Exemplo de um grafo



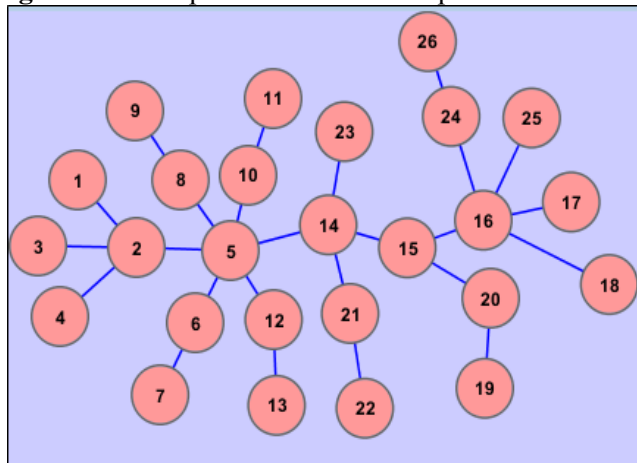
Fonte: Elaborada pelos autores.

Paul Erdős e Alfréd Rényi (1960) inicialmente estabeleceram os conceitos matemáticos da topologia de redes de livre-escala através da equação $P(k) \sim k^{-\gamma}$. A equação representa a probabilidade de um nó apresentar k conexões é aproximadamente o número de graus elevado à potência negativa (BARABÁSI; ALBERT, 1999). Os autores discutiram que as redes reais de livre-escala são baseadas em dois mecanismos genéricos de muitas redes reais (BARABÁSI; ALBERT, 1999; BARABÁSI; ALBERT, 2002): i) uma rede começa com um número fixo n de nós que, então, são aleatoriamente conectados sem modificar n . Muitas redes reais descrevem sistemas abertos que crescem com a inclusão contínua de novos nós. Assim, uma rede é criada inicialmente com um núcleo pequeno de nós, ocorrendo um aumento gradual de nós, durante o ciclo de vida da rede; e, ii) os modelos de rede assumem a probabilidade de que dois nós são conectados independentemente do grau dos nós, ou seja, novas ligações são feitas aleatoriamente.

A análise das redes de livre-escala envolve o estudo do fenômeno de interação molecular, através da integração multinível de dados e modelos (BARABÁSI; OLTVAI, 2004; O'MALLEY; DUPRÉ, 2005; SIEGAL *et al.*, 2007). A rede livre de escala mais conhecida é a rede Barabási-Albert (BA) que possui dois mecanismos básicos (BARABÁSI; ALBERT, 1999; BARABÁSI; ALBERT, 2002; BARABÁSI; OLTVAI, 2004): i) o crescimento da rede significa que a rede emerge através da ligação de nós subsequentes; e, ii) novos nós ligam-se, preferencialmente, aos nós mais conectados, ou seja, os nós com maior grau de distribuição. Um modelo baseado nestes

dois mecanismos indica que o desenvolvimento de grandes redes é governado por um fenômeno de auto-organização, que está além de características específicas de sistemas individuais (BARABÁSI; ALBERT, 1999).

Figura 2 –Exemplo de uma rede do tipo Barabási-Albert



Fonte: Elaboração dos autores.

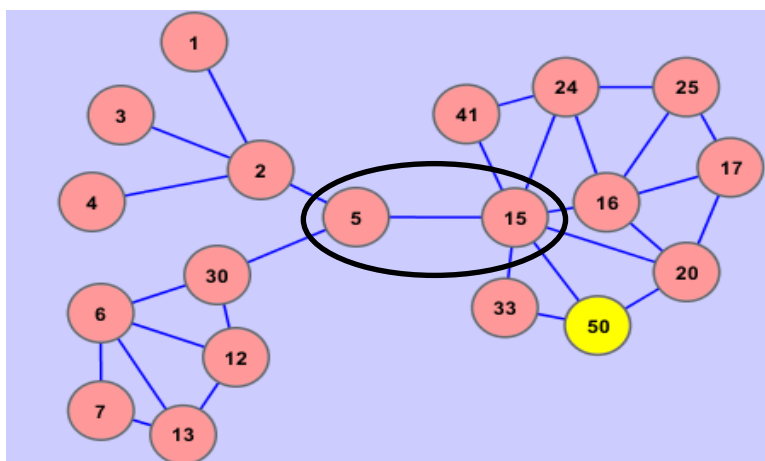
A Figura 2 apresenta um exemplo da rede BA, em que os nós cinco e quinze são os mais conectados da rede e, por outro lado, os nós terminais⁵ são os menos conectados. Os nós altamente conectados têm maior probabilidade de ter um efeito na funcionalidade da rede do que nós esparsos, como, por exemplo, proteínas altamente conectadas tendem a ser mais letais, quando forem eliminadas (*knocked out*) (SIEGAL *et al.*, 2007).

As redes de livre-escala predizem que os nós que aparecem antes na história são os nós mais conectados da rede (BARABÁSI; OLTVAI, 2004). Assim, as proteínas evolutivamente mais antigas têm mais conexões quando comparadas com proteínas recentes (BARABÁSI; OLTVAI, 2004). Essa descoberta empírica demonstra uma preferência por formação de novas conexões com proteínas evolutivamente antigas (BARABÁSI; OLTVAI, 2004), enfatizando que estes nós, com um número maior de conexões, podem desempenhar funções bioquímicas importantes na célula (SIEGAL *et al.*, 2007.). Em grafos dirigidos, as interações entre dois nós têm uma direção bem definida, como, por exemplo, a direção do fluxo de material de um substrato para um produto, em uma reação metabólica (BARABÁSI; OLTVAI, 2004). Em grafos não dirigidos, as ligações não têm uma direção assinalada; por exemplo, em uma rede de interação de proteínas, uma ligação representa uma relação mútua de amarração bilateral: se a proteína A se liga à proteína B, então a proteína B também se amarra à proteína A (BARABÁSI; OLTVAI, 2004).

⁵ Os nós terminais são os nós nas extremidades do grafo.

As redes de livre-escala representam as redes de interação entre proteínas (PPI) (BARABÁSI; OLTVAI, 2004; SIEGAL *et al.*, 2007), sendo que a análise de uma rede pode levar em consideração as informações a respeito da sua topologia, que pode ser classificada em local ou global (AITTOKALLIO; SCHWIKOWSKI, 2006).

Figura 3 – Exemplo de uma rede biológica com um ponto de gargalo na rede



Fonte: Elaboração dos autores.

A Figura 3 apresenta exemplo de uma rede biológica, em que o nó 15 representa um *hub*. (Caracteriza-se por ter um grande número de ligações e por manter os nós unidos de acordo com Barabási e Albert, 2002; Barabási e Oltvai, 2004). O círculo preto indica um ponto de gargalo na rede.

O grau de distribuição é usado para descobrir padrões de interação baseados na análise de centralidade⁶ (AITTOKALLIO; SCHWIKOWSKI, 2006). O grau de distribuição também fornece uma medida de conectividade da rede, em que uma rede com pouca conectividade de nós possui poucos nós com um grau alto de conectividade na rede. Por outro lado, redes com uma média alta de conexões entre os nós possuem poucos nós com um grau baixo de conectividade, sendo esta uma característica das redes aleatórias, também conhecidas como redes randômicas (SIEGAL *et al.*, 2007). A consequência da presença desta propriedade em uma rede é que poucos *hubs* altamente conectados mantêm o controle de toda a rede juntos (BARABÁSI *et al.*, 2011).

O coeficiente de agrupamento é calculado pela equação $coeficiente_{agrupamento}(v) = \frac{grau(v)}{\sum_{i=1}^n arestas(v)}$, em que o número de ligações dos vizinhos de um nó (calculado pelo grau do nó) é dividido pelo número máximo de ligações destes nós (número de arestas de um nó) (AITTOKALLIO; SCHWIKOWSKI, 2006). O

⁶ A ser descrito na próxima seção.

coeficiente de agrupamento visa a descobrir padrões de *pathways* através da decomposição da rede em grupos, sendo que, para analisar um padrão de *pathways*, é necessário lembrar que um caminho é um conjunto de nós únicos (sem repetição) conectados através de um grafo dirigido sem ramificações ou círculos (AITTOKALLIO; SCHWIKOWSKI, 2006). Desta forma, um *pathway* em uma rede celular representa a transformação de um caminho de um nutriente em um produto na rede metabólica, ou uma série de modificações pós-traducional do sentido de um sinal para o destino pretendido de um sinal de tradução da rede (ALBERT, 2005).

3 Ferramentas de biologia de sistemas

Esta seção descreve as ferramentas que podem ser utilizadas para análise de biologia de sistemas: análise de modularidade, ontologia gênica e análise de centralidade.

3.1 Análise de modularidade

Das inúmeras propriedades topológicas observadas para as redes de interações entre moléculas, esta é aquela que possibilita separar as redes em inúmeras sub-redes distintas ou módulos. Um módulo é uma estrutura de natureza funcional e padronizada, e que pode ser conectada a outros módulos que possuam características semelhantes ou mesmo distintas para a construção de um objeto mais complexo. Do ponto de vista biológico, os genes, as proteínas ou os metabólitos formam módulos específicos; a conexão entre estes promove a emergência de processos biológicos essenciais para a funcionalidade celular. Estes processos biológicos, que juntamente com a localização intracelular dos componentes do módulo e a sua funcionalidade bioquímica, constituem as chamadas ontologias gênicas, necessárias para definirem o contexto celular de atuação de uma rede de interações biológicas (BARABÁSI; OLTVAI, 2004; BONATTO; NAKAYA, 2015).

Quando olhamos para as Ciências Biológicas e suas mais variadas formas de redes de interações, percebemos que todas estas redes estão repletas de módulos. Por exemplo, os complexos existentes entre proteínas ou proteínas e RNA que, por definição são módulos, compõem as funções moleculares básicas de uma célula. De modo similar, os grupos de moléculas, que são corregulados temporalmente, são conhecidos por atuarem em vários passos do ciclo celular, na transdução de sinais externos em bactérias para a quimiotaxia ou nas vias de resposta a hormônios em células de organismos vertebrados (BONATTO; NAKAYA, 2015).

Considerando a complexidade de um módulo, assim como sua relação em potencial com uma função biológica, é necessário aplicar os conceitos fornecidos pela Teoria dos Grafos. Assim, um módulo ou agrupamento pode ser descrito como um grupo de nós que possuem um grau de conectividade maior do que aquele observado para a rede como um todo. Em termos matemáticos, um módulo é definido como um conjunto de triângulos, cuja densidade em uma rede pode ser calculada pelo coeficiente de agrupamento. Dessa maneira, um alto coeficiente de agrupamento influencia no coeficiente de agrupamento médio de uma determinada rede ($\langle C \rangle$), indicando a presença em potencial de módulos. Praticamente todas as redes de interações entre moléculas biológicas possuem alto valor de $\langle C \rangle$, sendo esta uma característica fundamental das redes biológicas (BONATTO; NAKAYA, 2015).

De forma complementar ao conceito de coeficiente de agrupamento, os módulos observados em uma rede biológica podem ser definidos como um conjunto de inúmeros elementos unitários denominados de subgrafos. Cada subgrafo define, em sua maioria, um mecanismo biológico específico e, por isso, recebendo a denominação de motivos. Os motivos biológicos constituem processos ou funções biológicas que se apresentam super-representados em uma rede e, caracteristicamente, são conservados em termos evolutivos. Vários exemplos de motivos cujas funções são conservadas evolutivamente podem ser encontrados em praticamente todos os organismos, como é o caso da replicação de DNA e a transcrição de RNA.

Uma questão importante relacionada à modularidade e à formação de motivos em redes biológicas diz respeito aos componentes moleculares de um motivo específico, interagem com os nós que estão fora do motivo. As observações empíricas, especialmente aquelas realizadas com as redes de regulação transcricional de *Escherichia coli*, indicam que tipos específicos de motivos podem se agregar com estes nós de forma espontânea para formar grandes módulos, sendo esta uma propriedade geral das redes (BARABÁSI; OLTVAI, 2004; BONATTO; NAKAYA, 2015).

Considerando a Teoria dos Grafos e a definição matemática dos módulos, estes, caracteristicamente, seriam grupos de nós quase isolados de um sistema, o que não ocorre em sistemas complexos reais. De fato, a presença de nós com alto número de conexões ou *hubs* torna a presença de módulos isolados praticamente inexistente, especialmente em sistemas biológicos. Assim, pela sua presença em sistemas complexos e pela sua interpolação com outros nós que não são parte de módulos, a análise de modularidade em Biologia de Sistemas requer o uso de ferramentas matemáticas apropriadas, como as análises de agrupamentos ou mesmo a avaliação da topologia global de uma rede, considerando a presença dos chamados cliques (nos quais todos os

nós, dentro de um módulo, podem ser atingidos pelo menor número de caminhos possíveis) ou pelas definições de redes pequeno mundo (BONATTO; NAKAYA, 2015).

3.2 Ontologia gênica

A análise de redes biológicas permite determinar as categorias presentes em um conjunto de genes ou em um subgrafo biológico, através do uso de ontologia gênica (GO). Isto é feito através da identificação de *hubs* em uma rede, com o intuito de permitir a análise de suas funcionalidades e conexões usando a classificação determinada pelo Consórcio de Ontologia Gênica⁷ (ASHBURNER *et al.*, 2000; The Gene Ontology Consortium *et al.*, 2019).

O objetivo deste consórcio é produzir (e manter atualizado) um vocabulário controlado para ser usado em eucariotos, mesmo sabendo-se que o conhecimento sobre os genes e papéis das proteínas em células cresce e muda constantemente (The Gene Ontology Consortium *et al.*, 2019). As ontologias servem para classificar as redes conforme suas funções biológicas, que podem ser componente celular, processo biológico ou função molecular (ASHBURNER *et al.*, 2000).

O GO é uma ferramenta disponível na *web* que fornece conhecimento computacional a respeito das funções dos genes e dos seus produtos. As ontologias das funções moleculares têm sido revisadas para melhor representar todas as atividades dos produtos de genes, com foco na transcrição de atividades regulatórias (The Gene Ontology Consortium *et al.*, 2019).

3.3 Análise de centralidade

As análises de centralidades são também chamadas de métricas locais de grafos. Estas métricas possuem inúmeras definições matemáticas, que descrevem as diferentes propriedades para os nós de uma rede, considerando sua posição topologia na rede, o número de caminhos mais curtos que trespasam o nó e como este nó influencia a topologia dos nós vizinhos. Das diferentes métricas aplicadas para determinar quais nós são os mais ou os menos importantes em uma determinada rede, o grau de nó é o mais simples e um dos mais informativos. Em termos matemáticos, o grau de um nó (k) indica quantas conexões o mesmo possui com os outros nós (GIBBONS, 1994; GOODRICH; TAMASSIA, 2007; SHAFFER, 2011). As conexões podem ser tanto não direcionadas quanto direcionadas; os conectores não apresentam uma ordem específica de entrada ou saída para as conexões não direcionadas, enquanto os nós podem ter conectores que entram e que saem deste, caracterizando as conexões direcionadas. Para

⁷ GO. Disponível em: <http://geneontology.org>. Acesso em: 30 abr. 2019.

as conexões direcionadas, o número de conectores que entram no nó é definido como *k-entrada*, enquanto o número de conectores que saem do nó é definido como *k-saída*.

Para as redes que possuem conexões não direcionadas, pode-se calcular o grau médio de conectividade por nó em uma determinada rede ou $\langle k \rangle$. A equação $\langle k \rangle = 2L/N$ é aplicada no cálculo da conectividade média por nó em uma dada rede: L representa o número total de conexões dividido pelo número total de nós (N). Além dos graus de conectividade de um nó, a distribuição probabilística desses graus em um grafo ou $P(k)$ é outra propriedade importante de uma rede. Os valores de $P(k)$ são obtidos por meio da contagem do número de nós, ou $N(k)$, que possuam um valor de conectividade k igual a 1, 2, 3,... seguido pela divisão do número total de nós N. Em sistemas complexos, comumente observa-se que a maioria dos nós possui um valor de k igual ou menor ao valor $\langle k \rangle$. Por outro lado, um pequeno número de nós possui valores de k superior ao de $\langle k \rangle$ e recebem a denominação de *hubs*. Os *hubs* são, usando a sua definição computacional, os centros de controle ou de distribuição de dados, em uma rede qualquer. Em sistemas biológicos, os *hubs* são representados por biomoléculas que atuam em várias rotas bioquímicas simultaneamente, mas não necessariamente são elementos essenciais naquele sistema (BARABÁSI; ALBERT, 2002; BARABÁSI; OLTVAI, 2004). Além do valor de k , outra propriedade matemática importante das redes é o chamado valor de expoente do grau ou valor gama. Este conceito considera que toda a distribuição de graus $P(k)$ de uma rede é inversamente proporcional ao valor de conectividade k elevado ao valor de gama, cuja definição matemática foi demonstrada previamente neste capítulo. Dessa maneira, todas as redes que representam sistemas complexos, tais como os sistemas biológicos, são formadas por um pequeno conjunto de nós com alto grau de conectividade (os nós do tipo *hubs*, como visto anteriormente), enquanto que a maioria dos nós dessas redes possui um grau de conectividade igual ou inferior ao grau médio de conectividade (BARABÁSI; ALBERT, 2002; BARABÁSI; OLTVAI, 2004, BONATTO; NAKAYA, 2015).

Além da conectividade e das suas propriedades, o número de passos ou vias necessários, que conectam dois ou mais nós quaisquer (também chamado de distância em uma rede), é outra característica importante. Dos inúmeros passos potenciais existentes entre dois ou mais nós, o caminho com o menor número de passos entre dois nós é um dos mais importantes. Para as redes com conectores direcionados, a distância entre um nó A e um nó B (ou ℓ_{AB}) pode ser diferente da distância entre o nó B e o A (ℓ_{BA}) e, muitas vezes, não há um caminho direto entre dois nós. Dessa maneira, o caminho médio ($\langle \ell \rangle$) representa a média de todos os caminhos mais curtos entre todos os pares de nós, indicando uma medida da “navegabilidade” para a rede. Por fim, o coeficiente de agrupamento de uma rede indica a probabilidade de um nó estar

conectado a outros nós. Esta probabilidade é estimada usando a equação $CI=2nI/k(k-1)$, em que nI é igual ao número de conectores que ligam um número kI de nós vizinhos ao nó I . Assim, CI representa o número de triângulos que passam pelo nó I e $kI(kI-1)/2$ é o número total de triângulos que podem passar pelo nó I , considerando que todos os nós vizinhos de I estejam conectados. Sabendo o CI de cada nó, pode-se obter o coeficiente de agrupamento médio ou $\langle C \rangle$, que caracteriza o potencial dos nós em formar módulos. A função $C(k)$ indica a média do coeficiente de agrupamento de todos os nós com um número k de conectores e, caracteristicamente, se esses nós formam ou não módulos que, em redes biológicas, estão ligados a determinadas funções (BONATTO; NAKAYA, 2015).

4 Vertentes da biologia de sistemas

Sabendo-se da diversidade de redes biológicas existentes e da multiplicidade de técnicas experimentais aplicadas para a compreensão de um modelo biológico, a Biologia de Sistemas possui diferentes vertentes usadas para a geração de modelos que possibilitam a visualização e compreensão desses sistemas. Das diferentes variações ou tipos de biologia de sistemas conhecidos, duas são rotineiramente usadas. A primeira e a mais comum é a chamada Biologia de Sistemas *top-down* (do “mais complexo ao menos complexo” ou “de cima-para-baixo”), que propõe gerar modelos generalistas de um sistema biológico, a partir de seus componentes e das suas interações. A Biologia de Sistemas *top-down* é aplicada no estudo de mecanismos moleculares pouco conhecidos e como esses integram-se com outros processos em determinadas condições celulares (BONATTO; NAKAYA, 2015).

Por sua vez, a Biologia de Sistemas *bottom-up* (do “menos complexo ao mais complexo” ou “de baixo-para-cima”) busca caracterizar um sistema biológico utilizando uma abordagem reducionista e quantitativa para cada componente sistema, especialmente por meio da simulação das cinéticas enzimáticas e estequiométricas de cada processo bioquímico, integrando essas informações para a geração de modelos preditivos do comportamento do sistema frente a alguma perturbação, como a administração de fármacos para o tratamento de uma patologia, por exemplo. Apesar de sua importância, deve ser ressaltado que a determinação de parâmetros quantitativos em sistemas moleculares possui uma série de problemas experimentais que, devido aos erros inerentes das técnicas e, muitas vezes, da simplificação da análise, os “verdadeiros” parâmetros cinéticos não são conhecidos. Em vários sistemas moleculares é impossível, pela tecnologia atual, quantificar os parâmetros cinéticos *in vitro*, como é

o caso das rotas de sinalização celular em organismos eucarióticos (BONATTO; NAKAYA, 2015).

5 Considerações finais

A Biologia de Sistemas é uma área interdisciplinar nova e que está crescendo em importância na medida que dados em larga escala estão sendo gerados cada vez mais em quantidade e qualidade. O aumento visto na obtenção deste tipo de dados tem respaldo nas novas tecnologias de sequenciamento de nova geração e de quantificação e determinação proteica, como espectrometria de massas, que tem se tornado cada vez mais acessível aos pesquisadores das áreas biológicas. Assim, a geração massiva de dados precisa dos diferentes tipos de Biologia de Sistemas e de suas ferramentas para dar sentido e compreensão à informação biológica que, de outra forma, torna-se apenas um ruído inteligível para o pesquisador. Com base em uma pergunta biológica e tendo acesso aos dados gerados pelas tecnologias de análises em larga escala, o pesquisador pode extrair a conexão entre os diferentes componentes do seu sistema ou modelo biológico e avaliar quais são os componentes mais importantes ou mecanismos representativos neste modelo, a fim de criar novas hipóteses experimentais e, por fim, alimentar os dados com novas informações. Desta maneira, a Biologia de Sistemas cria um círculo virtuoso de geração de hipóteses e de dados experimentais, possibilitando que os sistemas biológicos sejam compreendidos no seu âmbito.

Referências

- AITTOKALLIO, T.; SCHWIKOWSKI, B. Graph-based methods for analyzing networks in cell biology. **Briefings in Bioinformatics**, v. 7, n. 3, p. 243-255, 2006.
- ALBERT, R. Scale-free networks in cell biology. **Journal of Cell Science**, v. 118, p. 4947-57, 2005.
- ASHBURNER, M. *et al.* Gene ontology: tool for the unification of biology. **Nat Genet**, v. 25, n. 1, p. 25-29, 2000.
- BARABÁSI, A. L.; ALBERT, R. Emergence of Scaling in Random Networks. **Science**, v. 286, p. 509, 1999.
- BARABÁSI, A. L.; ALBERT, R. Statistical physics of complex networks. **Reviews of Modern Physics**, v. 74, n. 47, 2002.
- BARABÁSI, A. L.; OLTVAI, Z. N. Network biology: understanding the cell's functional organization. **Nature Reviews. Genetics**, v. 5, p. 101-113, 2004.
- BARABÁSI, A. L. Scale-Free Networks: A Decade and Beyond. **Science**, v. 325 (5939), p. 412-413, 2009.
- BARABÁSI, A. L., Gulbahce, N. and Loscalzo, J. Network medicine: a network-based approach to human disease. **Nature Reviews, Genetics**, v. 12, p. 56-68, 2011.
- BONATTO, D.; NAKAYA, HTI. Genômica e biologia de sistemas. *In*: MOREIRA, L.M. **Ciências genômicas: fundamentos e aplicações**. Ribeirão Preto: Sociedade Brasileira de Genética, 2015. p. 277-302,

- ERDŐS, P.; RÉNYI, A. On the evolution of random graphs. *Publ. Math. Inst. Hung. Acad. Sci., Ser. v. A* 5, p. 17-61, 1960.
- GIBBONS, A. **Algorithm graph theory**. Cambridge University Press, 1994.
- GOODRICH, M. T. E Tamassia, R. **Estrutura de dados e algoritmos em Java**. 4. ed. Porto Alegre: Bookman, 2007.
- KARSENTI, E. Towards an ‘Oceans Systems Biology’. **Molecular Systems Biology**, v. 8, p. 575, 2012.
- O’MALLEY, M. A.; DUPRE, J. Fundamental issues in system biology. **Bioessays**, v. 27, p. 1270-1276, 2005.
- RAHN, J. J.; ADAIR, G. M.; NAIRN, R. S. Multiple roles of ERCC1-XPF in mammalian interstrand crosslink repair. **Environmental and Molecular Mutagenesis**, v. 51, n. 6, p. 567-81, 2010.
- ROSVALL, M.; SNEPPEN, K. Modeling Dynamics of Information Networks. **Physical Review Letters**, v. 91, 2003.
- SHAFFER, C. A. **Data Structures and Algorithm Analysis**. Department of Computer Science, Virginia Tech, Blacksburg. Dover Publications, 2011.
- SIEGAL, M. L.; PROMISLOW, D. E. L. and Bergman, A. Functional and evolutionary inference in gene networks: does topology matter? **Genetica**, 10, v. 129, n.1, p. 83103, 2007.
- WESTON, A. D.; HOOD, L. Systems Biology, Proteomics, and the Future of Health Care: Toward Predictive, Preventative, and Personalized Medicine. **Journal of Proteome Research**, v. 3, n. 2, p. 179-196, 2004.
- The Gene Ontology Consortium. The gene ontology resource: 20 years and still Going strong. **Nucleic Acids Res**, v. 47(D1), p. D330-D338, 2019.

1 Introdução

“Forma é função”. Esta sentença é repetida inúmeras vezes nas disciplinas de bioinformática estrutural; o conceito de que a forma de uma proteína determina a sua função pode ser considerada o dogma da biologia estrutural. Esta frase encerra em si um reducionismo, que em última análise, engendra a espinha dorsal da área.

Quando Max Perutz decidiu utilizar a técnica de difração de raios X, para resolver a estrutura de hemoglobina de cavalo, como seu projeto de doutorado, descobriu um universo antes desconhecido, ali nascia a biologia estrutural, posteriormente a utilização do método de substituição isomorfa. Para a resolução do problema de fase definiu a técnica como uma das mais importantes, até hoje, para a resolução da estrutura tridimensional de macromoléculas de sua vida. Por óbvio, tantos outros cientistas brilhantes antecederam e precederam Max Perutz nesta incrível jornada, como Wilhelm Conrad Röntgen, William Henry Bragg, William Lawrence Bragg, Dorothy Hodgkin e tantos outros, mas Perutz, com a sua tenacidade, independentemente das incertezas, investiu 22 anos de sua vida para resolver a sua primeira estrutura.

A biologia estrutural é um ramo da biologia molecular, que se ocupa em explicar o mundo, pelo menos o mundo das macromoléculas, principalmente das proteínas. A natureza parece não gostar de dogmas, visto que ela sempre encontra uma maneira de nos mostrar como somos insignificantes frente à grandiosidade e beleza do universo, a afirmação de que a forma define a sua função não parece fazer sentido, quando o assunto são príons. Príons (*PR*oteínaceous *I*nfectious *ON*ly *P*article) são proteínas infecciosas relacionadas a um grupo de doenças neurodegenerativas invariavelmente fatais que afetam seres humanos e animais, sendo comumente conhecida como doença da vaca louca. Uma das hipóteses acerca dos príons é que a proteína príon celular, que na sua estrutura possui uma preponderância de hélices alfa (Figura 1), sofra uma mudança conformacional que resulta na proteína príônica patológica, cuja estrutura tridimensional ainda não foi experimentalmente resolvida, mas que é rica em folhas beta. O mais intrigante é que estas isoformas, príon celular e príon patológico, apresentam a mesma sequência de aminoácidos, com evidentes conteúdos de estruturas

¹ Universidade Federal de Ciências da Saúde de Porto Alegre. *E-mail*: bruna.schuck4@gmail.com

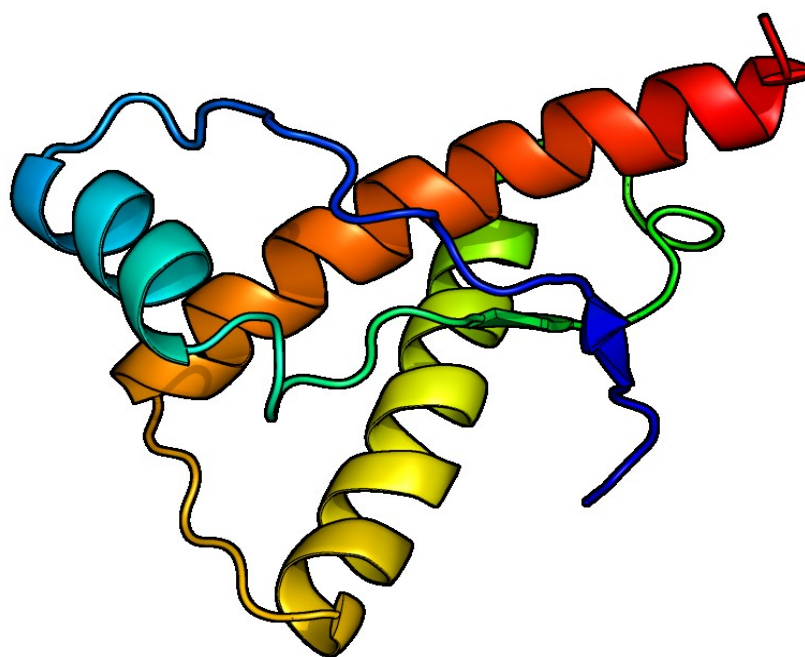
² Universidade do Vale do Taquari. *E-mail*: lmpossamai1@gmail.com

³ Universidade do Vale do Taquari. *E-mail*: luis.timmers@univates.br

⁴ Universidade Federal de Ciências da Saúde de Porto Alegre. *E-mail*: rafaelca@ufcspa.edu.br

secundárias e, conseqüentemente, distintas estruturas terciárias. Diante destes fatos, o dogma central da biologia estrutural nos escorre entre os dedos tal qual areia, tal qual um retrovírus diante do dogma central da biologia molecular. Com a determinação de Perutz, devemos seguir, passo a passo, e enfrentar os novos desafios e desenvolver novas técnicas, como a microscopia eletrônica criogênica (cryo-EM), que vem crescendo em importância de resolução de grandes estruturas e máquinas moleculares. Neste capítulo, trazemos uma pequeníssima parcela do que a biologia estrutural pode contribuir para o desenvolvimento e a prospecção de moléculas que possuam potenciais terapêuticos, ou mesmo, no estudo do comportamento que macromoléculas podem adotar em ambientes fisiológicos, questão esta fundamental para o desenvolvimento de fármacos e muitas vezes negligenciada nos processos de desenho racional de fármacos assistidos por computador. Esperamos que gostem. Uma boa leitura!

Figura 1 – Estrutura terciária da proteína príon celular de hamster sírio. Representação do tipo *Cartoon* da cadeia principal da estrutura cristalográfica (código de acesso ao PDB: 1B10)



Fonte: Imagem gerada com o PyMOL (SCHRÖDINGER, 2010).

2 Triagem virtual reversa: em busca de alvos moleculares

Ao explorar o meio ambiente não é incomum depararmos com novas moléculas que aparentam ocasionar certa resposta biológica, porém, seus mecanismos de ação e alvos moleculares para tal efeito nos são desconhecidos. Um dos meios para

se descobrir possíveis alvos moleculares e melhor elucidar mecanismos de ação de compostos é a triagem virtual reversa (*inverse* ou *reverse virtual screening*, *in silico* ou *computational target fishing*, *reverse pharmacognosy*, em inglês).

Enquanto a triagem virtual parte de um alvo molecular conhecido que, normalmente, se quer inibir e procura em bibliotecas de compostos por moléculas com maior probabilidade de se ligar a este alvo específico, a triagem virtual reversa busca encontrar os alvos mais prováveis de uma determinada molécula (CERETO-MASSAGUÉ *et al.*, 2014). Na triagem virtual reversa, certas características dessas moléculas, como seus farmacóforos (parte da molécula responsável pela resposta biológica) ou estrutura molecular, são utilizadas para encontrar alvos moleculares com características eletrônicas correspondentes.

A triagem virtual reversa permite a predição de certas características de uma molécula, tais como:

- bioatividade, ou seja, o(s) alvo(s) molecular(es) com que interage(m);
- mecanismo de ação, ou seja, o que é preciso ocorrer em nível molecular para que o efeito esperado seja gerado;
- efeitos adversos, ou seja, efeitos prejudiciais ou indesejáveis ocorridos durante ou após o uso de um medicamento, em que há a possibilidade razoável de relação causal entre o tratamento e o efeito;
- polifarmacologia, ou seja, sua capacidade de interagir com variados alvos e gerar efeitos sinérgicos ou não;
- reposicionamento ou reaproveitamento de fármacos, ou seja, a busca por novas indicações terapêuticas para fármacos já conhecidos.

A polifarmacologia, efeitos adversos e reposicionamento ou reaproveitamento de fármacos têm como alicerce a promiscuidade das moléculas. Esta promiscuidade é definida como a habilidade de pequenas moléculas interagirem especificamente com diversos alvos moleculares, podendo exibir efeitos farmacológicos similares ou diferentes (JASIAL; HU; BAJORATH, 2016; MEI; YANG, 2018). Esta interação com diferentes alvos pode possibilitar o uso de determinada molécula no tratamento de mais de uma condição ou doença, assim como a ocorrência de efeitos adversos indesejáveis (JASIAL; HU; BAJORATH, 2016; MEI; YANG, 2018).

Inicialmente vista de forma negativa, a promiscuidade das moléculas tem sido mais explorada desde o início do século. Fármacos multialvos, nomeados assim por interagirem com múltiplos alvos e resultarem em efeitos farmacológicos benéficos similares ou idênticos, têm ocasionado melhores efeitos no tratamento de doenças complexas (como doenças cardiovasculares, do sistema nervoso central e cânceres), que

envolvem múltiplos genes e fatores, frente a fármacos com um único alvo molecular (MEI; YANG, 2018).

Ao mesmo tempo em que um fármaco promíscuo pode possuir maior eficácia, como é o caso da ziprasidona, um fármaco antipsicótico atípico, que detém atividade antagonística contra os receptores serotoninérgicos e dopaminérgicos, ele também pode gerar efeitos adversos indesejáveis e sérios, como ocorreu com o antialérgico astemizol, antagonista do receptor histamínico H₁ que foi retirado do mercado por ocasionar arritmia, devido à inibição de canais de potássio hERG no coração (MEI; YANG, 2018).

A predição *in silico* de alvos moleculares para novas moléculas e conhecidos fármacos possibilita a otimização do processo de descoberta e desenvolvimento de fármacos, visto que pode prevenir o retrabalho exigido, quando se descobrem efeitos *off-target* em ensaios pré-clínicos e clínicos; sugerir novas indicações terapêuticas que podem ser mais vantajosas que as propostas ou já existentes; permitir a otimização de compostos-líder para direcioná-los a um alvo molecular mais específico, dentro do *pool* descoberto, etc.

2.1 Formicamicinas e seu potencial antibiótico

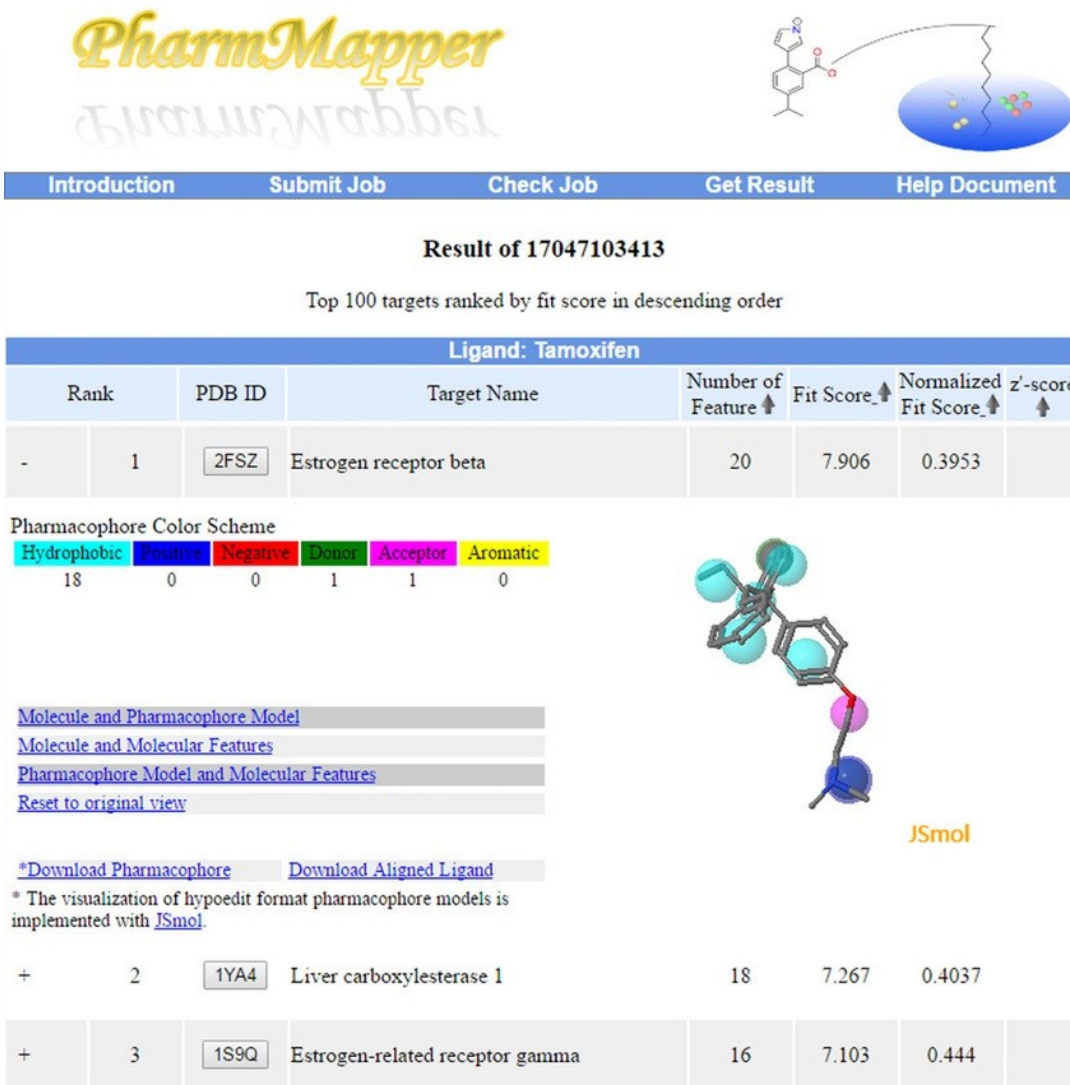
Ao estudar a ecologia química de mutualismos formados entre bactérias e insetos cultivadores de fungos, Qin e colaboradores (2017) isolaram uma nova espécie de *Streptomyces*, o *S. formicae*. A partir de uma das cepas desta espécie, foi possível demonstrar *in vitro* uma atividade antagonística contra fungos e bactérias resistentes a fármacos, incluindo a bactéria *Staphylococcus aureus* resistente à meticilina (MRSA), *Enterococcus* resistente à vancomicina (VRE) e o fungo *Lomentospora prolificans* multirresistente. Este efeito foi atribuído a novos policetídeos pentacíclicos naturais nomeados pelos autores de formicamicinas.

As formicamicinas tiveram suas fórmulas moleculares, estruturas moleculares, via biossintética, etc., evidenciadas. Porém seu mecanismo de ação e alvos moleculares para tal efeito antimicrobiano não foram estudados (QIN *et al.*, 2017). Cientes desta lacuna, investigamos os potenciais alvos moleculares para a ação antimicrobiana da família formicamicina, através de uma triagem virtual reversa realizada pelo servidor *web* de livre acesso PharmMapper.

O PharmMapper utiliza o mapeamento de farmacóforos para procurar por potenciais candidatos a alvos moleculares para uma dada molécula (Figura 2). Na molécula submetida no servidor são sugeridos farmacóforos que serão relacionados com um banco de dados de modelos farmacofóricos tridimensionais, o PharmTargetDB, anotado de todas as informações de alvos do BindingBD, TargetBank, DrugBank e PDTD (WANG *et al.*, 2017). A sugestão de farmacóforos se baseia em estudos prévios

da interação de determinadas moléculas com seus alvos biológicos, e os resultados obtidos tendem a gerar uma biblioteca quimicamente diversa, pois, por exemplo, se o farmacóforo é uma hidroxila, procura-se por um doador de ligação de hidrogênio em determinada coordenada XYZ, permitindo maior diversidade de moléculas resultantes (WANG *et al.*, 2017). Como resultado tem-se uma lista com os 300 melhores possíveis receptores ligantes ordenados por sua pontuação de ajuste normalizada em ordem decrescente.

Figura 2 – Interface do PharmMapper. Lista das moléculas que satisfazem o modelo farmacofórico 3D do Tamoxifeno



PharmMapper

Introduction Submit Job Check Job Get Result Help Document

Result of 17047103413

Top 100 targets ranked by fit score in descending order

Ligand: Tamoxifen						
Rank	PDB ID	Target Name	Number of Feature ↑	Fit Score ↑	Normalized Fit Score ↑	z'-score ↑
-	1	2FSZ Estrogen receptor beta	20	7.906	0.3953	
Pharmacophore Color Scheme						
Hydrophobic	Positive	Negative	Donor	Acceptor	Aromatic	
18	0	0	1	1	0	
Molecule and Pharmacophore Model Molecule and Molecular Features Pharmacophore Model and Molecular Features Reset to original view						
*Download Pharmacophore Download Aligned Ligand * The visualization of hypoedit format pharmacophore models is implemented with JSmol .						
+	2	1YA4 Liver carboxylesterase 1	18	7.267	0.4037	
+	3	1S9Q Estrogen-related receptor gamma	16	7.103	0.444	

Fonte: Wang *et al.* (2017).

Esta metodologia permitiu a identificação de alvos moleculares que interagiriam com as formicamicinas. Acompanhada de uma revisão bibliográfica para verificar a

essencialidade de tais alvos para o ciclo de vida bacteriano e docagens e dinâmicas moleculares para melhor analisar as interações entre alvos e formicamicinas, foi possível evidenciar três potenciais alvos moleculares para as referidas moléculas: glicerol-3-fosfato desidrogenase [NAD^+], β -lactamase e anidrase carbônica 2.

2.2 Romã e seu potencial anticâncer

Infrutescência da romãzeira, a romã (*Punica granatum* L.) tem sido investigada devido às suas propriedades fitoterápicas relatadas popularmente. A sua casca possui propriedades antioxidantes e compostos ativos, tais como taninos, flavonoides e alcaloides (USHA *et al.*, 2015). Tais propriedades levaram ao seu estudo como tratamento alternativo e na prevenção de certas doenças, incluindo o câncer (BASSIRI-JAHROMI, 2018).

De modo a melhor esclarecer o mecanismo responsável pelo potencial terapêutico anticâncer da romã, Usha e colaboradores (2015) realizaram duas triagens virtuais reversas. Foram triados alvos moleculares nos servidores *online* PharmMapper e ReverseScreen 3D para três compostos ativos da casca de romã: quercetina (flavanoide), corilagina (tanino) e pseudopeletierina (alcaloide) (USHA *et al.*, 2015).

O ReverseScreen3D é um servidor *web* para triagem virtual reversa baseada na estrutura tridimensional (3D) do ligante, método similar ao utilizado pelo LigMatch (KINNINGS; JACKSON, 2009; KINNINGS; JACKSON, 2011). A estrutura 3D de confôrmeros, gerados a partir do ligante submetido, é comparada com um subconjunto automaticamente atualizado de ligantes biologicamente relevantes extraídos do *Protein Data Bank* (PDB) (BERMAN *et al.*, 2000). Nesta comparação são calculados o coeficiente de Tanimoto bi e tridimensional (KINNINGS; JACKSON, 2011).

Tendo obtido uma lista com potenciais alvos moleculares do ReverseScreen3D (baseado na estrutura do ligante) e do PharmMapper (baseado no mapeamento de farmacóforos), Usha e colegas (2015) identificaram quais destes alvos poderiam estar relacionados à atividade anticâncer da romã em dois bancos de dados: *NPACT* (*Naturally occurring Plant-based Anticancerous Compound-activity-Target database*) e *Herbal Ingredients*. Com os potenciais alvos moleculares anticâncer selecionados, seguiram-se predições ADME-Tox, docagem molecular e validação das posições de docagem (USHA *et al.*, 2015).

Como resultados, foram triados 24 potenciais alvos no PharmMapper e ReverseScreen3D para a quercetina, 11 alvos para a corilagina e dois alvos para a pseudopeletierina. Destes alvos, apenas os da quercetina foram submetidos à docagem molecular, resultando em 21 simulações com ligações favoráveis energeticamente entre ligante e alvo molecular. Tendo sido o receptor de vitamina D aquele com menor

energia de ligação, foi realizada a validação da posição da quercetina na estrutura 3D e um sítio ativo predito no receptor foi tido como o mais provável para a ação da molécula (USHA *et al.*, 2015).

2.3 Dolastatina 16

Inicialmente isolada do molusco marinho *Dolabella auricularia*, a família de peptídeos dolastatina apresentou citotoxicidade contra várias linhagens celulares cancerosas (NIEDERMEYER; BRÖNSTRUP, 2012). Liang e colaboradores (2018) estudaram os potenciais alvos moleculares para a dolastatina 16. A predição dos alvos foi feita no PharmMapper e conferida em um interatoma químico-proteico (CPI) utilizado para sugerir alvos moleculares, assim como ligantes que poderiam interagir com a dolastatina 16 (através dos servidores *web* DRAR-CPI e DDI-CPI, respectivamente) (YANG *et al.*, 2011; LUO *et al.*, 2011; LUO *et al.*, 2014).

O DRAR-CPI evidencia *off-targets*, alvos moleculares inesperados que não estão relacionados ao efeito terapêutico de um fármaco por exemplo. Estes *off-targets* podem auxiliar na descoberta de novos alvos moleculares para determinada molécula e possibilitar o reposicionamento de fármacos ou evidenciar interações moleculares responsáveis por efeitos adversos apresentados ao administrar um medicamento (LUO *et al.*, 2011).

Duas bibliotecas, uma contendo moléculas com descrições, indicações e efeitos adversos conhecidos e outra com proteínas com funções conhecidas, são utilizadas para gerar uma terceira biblioteca de interatomas entre ligantes, proteínas e a(s) molécula(s) submetida(s), utilizando o programa DOCK. Um escore de docagem de todos os compostos, em relação a todas as proteínas, é utilizado para criar um escore z' que varia de -1 (alvo favorável) a 1 (alvo não favorável) (LUO *et al.*, 2011).

Já o DDI-CPI foca em interações entre moléculas que também podem causar efeitos adversos no tratamento medicamentoso. Partindo das mesmas duas bibliotecas, uma de ligantes e outra de proteínas, utiliza-se o AutoDock Vina, para realizar a docagem molecular das referidas proteínas e ligantes e a(s) molécula(s) submetida(s). O menor escore de energia obtido e a pose correspondente dos ligantes são selecionados para construir o perfil CPI e prever as probabilidades de determinada interação entre a molécula submetida e os ligantes da biblioteca ocorrer, assim como proteínas que possam estar envolvidas nessa interação (LUO *et al.*, 2014).

O principal resultado da triagem no PharmMapper foi a enzima FKBP1A (PDB ID: 1BL4), alvo do imunossupressor tacrolimus (FK506). Tanto tacrolimus como dolastatina 16 foram submetidos nos servidores DRAR-CPI e DDI-CPI para corroborar os achados do PharmMapper e uma docagem molecular foi realizada com as moléculas

individualmente e a isomerase FKBP1A, para validar o alvo molecular selecionado, além de explorar o modo de ligação entre alvo e cada ligante (LIANG *et al.*, 2018).

2.4 Quercetina

Os flavonoides possuem ação antioxidante bastante caracterizada e efeitos neuroprotetivos, anti-inflamatórios e anticâncer emergentes. Neste grupo, está contida a quercetina. Alguns alvos moleculares inibidos pela quercetina e outros flavonoides já foram evidenciados, porém supõe-se que a modulação de múltiplos alvos suporte o potencial terapêutico da quercetina (CARVALHO *et al.*, 2017). Visando a descobrir potenciais alvos moleculares para a quercetina, Carvalho e colaboradores (2017) propuseram uma triagem hierárquica desses alvos, na procura por similaridade entre ligantes, seguida de comparação do sítio de ligação e docagem reversa.

Primeiramente, foi criada uma biblioteca de ligantes, a partir dos contidos no PDB. O *software Molecular Operating Environment* (MOE) foi utilizado para aplicar filtros e incrementar a biblioteca por meio da correção da hidrogenação, remoção de moléculas com baixa massa molecular (tais como sais e moléculas de água), etc. Com a biblioteca pronta, o algoritmo SHAFTS (*SHApe-Feature Similarity*) foi empregado para selecionar ligantes semelhantes à quercetina (CARVALHO *et al.*, 2017).

O SHAFTS é um algoritmo para o cálculo de similaridade estrutural tridimensional e triagem virtual baseada no ligante. A similaridade calculada é baseada tanto na similaridade entre os formatos moleculares como nos modelos farmacofóricos de diferentes conformações da molécula submetida com os ligantes conhecidos (LIU; JIANG; LI, 2011).

Após a aferição da similaridade entre ligantes, os alvos moleculares obtidos foram submetidos a uma comparação de sítio de ligação contra proteínas contidas no PDB ou da biblioteca própria dos pesquisadores. A ferramenta LIBRA (*Ligand Binding site Recognition Application*) foi utilizada para buscar similaridades estruturais locais entre os já mencionados alvos. LIBRA emprega um banco de dados de sítios de ligação gerados pela extração de resíduos cercando o ligante de estruturas 3D do PDB (HUNG *et al.*, 2015).

O servidor idTarget permitiu a docagem reversa das proteínas selecionadas nas etapas anteriores. Através de um algoritmo de procura, ele prediz sítios de ligação na superfície proteica e, por meio de um escore de função otimizado, estima a energia de ligação. Como resultado, uma proteína é sugerida para cada ligante, considerando-se o perfil de afinidade pelo ligante. Por fim, dinâmicas moleculares foram realizadas, e os potenciais alvos moleculares foram anotados conforme suas funções. Partindo de moléculas com estrutura semelhante à dos flavonoides, 74 potenciais-alvo foram

obtidos e submetidos ao idTarget e 4 alvos moleculares foram simulados por dinâmica molecular com a quercetina.

3 Dinâmica molecular clássica

A simulação por dinâmica molecular (DM) clássica é uma das técnicas mais versáteis para o estudo de macromoléculas biológicas no âmbito das técnicas *in silico*, principalmente para compreender os processos envolvidos na flexibilidade. Conceitualmente, a DM clássica é uma abordagem computacional, na qual são empregadas equações newtonianas para a resolução de representações atômicas de um sistema molecular, com base na Mecânica Clássica, com o intuito de obter informações a respeito de suas propriedades, em função do tempo (KARPLUS; PETSKO, 1990; van GUNSTEREN; MARK, 1998; KARPLUS; McCAMMON, 2002). Os algoritmos utilizados nestes programas consistem na solução numérica destas equações do movimento, fornecendo uma trajetória (coordenadas e momentos conjugados, em função do tempo) do sistema em estudo. Podemos caracterizar a flexibilidade de uma proteína por sua capacidade de responder a estímulos ambientais, como interação com ligantes, temperatura, pH, entre outros. Diversas funções de uma proteína podem ser relacionadas à sua flexibilidade, por exemplo, a capacidade de uma enzima em acomodar o seu substrato e catalisar a reação, ou a capacidade das proteínas de organismos termófilos em resistir a altas temperaturas (TEILUM *et al.*, 2011). Portanto, a DM clássica apresenta-se como uma importante ferramenta para extrair informações relacionadas com os tipos e a estabilidade das interações mediadas entre diferentes sistemas como, proteína-ligante ou proteína-proteína, o que permite assim, sua utilização em projetos de desenvolvimento racional de novos fármacos.

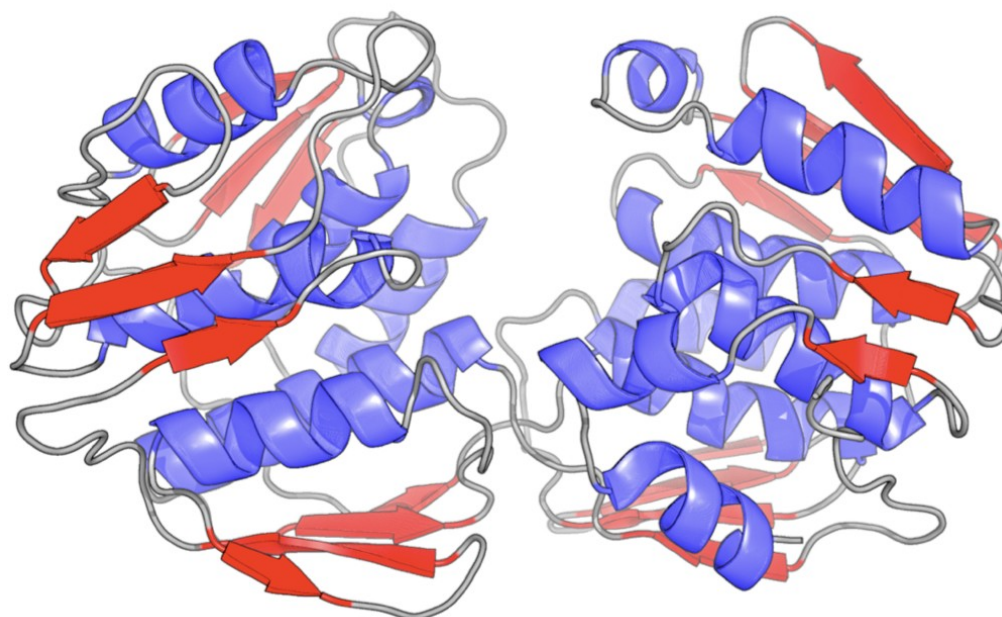
4 Aplicação de dinâmica molecular para o estudo de flexibilidade em enzimas

4.1 Estudo de caso: a enzima EPSP sintase de *Mycobacterium tuberculosis*

Podemos encontrar na literatura um vasto número de artigos científicos que utilizam a técnica de DM clássica para descrever a dinâmica de uma proteína em um ambiente aquoso. Dentre estes trabalhos, Timmers e colaboradores (2017) utilizaram como modelo de estudo a enzima 5-enolpiruvilchiquimato-3-fosfato (EPSP) sintase de *Mycobacterium tuberculosis*. Esta enzima é codificada pelo gene *aroA*, o qual faz parte da via do ácido chiquímico, catalisando a sexta reação. Esta via é a responsável pela produção de precursores de aminoácidos aromáticos. Além disso, está presente em

plantas, algas, fungos, bactérias e parasitas do filo apicomplexa. Até o presente momento, ela não foi encontrada em mamíferos. Estruturalmente, a EPSP sintase apresenta em média 450 resíduos de aminoácidos distribuídos em dois domínios globulares interligados por um pentapeptídeo (Figura 3).

Figura 3 – Estrutura terciária da enzima EPSP sintase. Representação do tipo *Cartoon* da cadeia principal da estrutura cristalográfica (código de acesso ao PDB: 2BJB). As estruturas secundárias de ambas proteínas estão coloridas com as mesmas cores, com exceção da alça modelada que está colorida em rosa



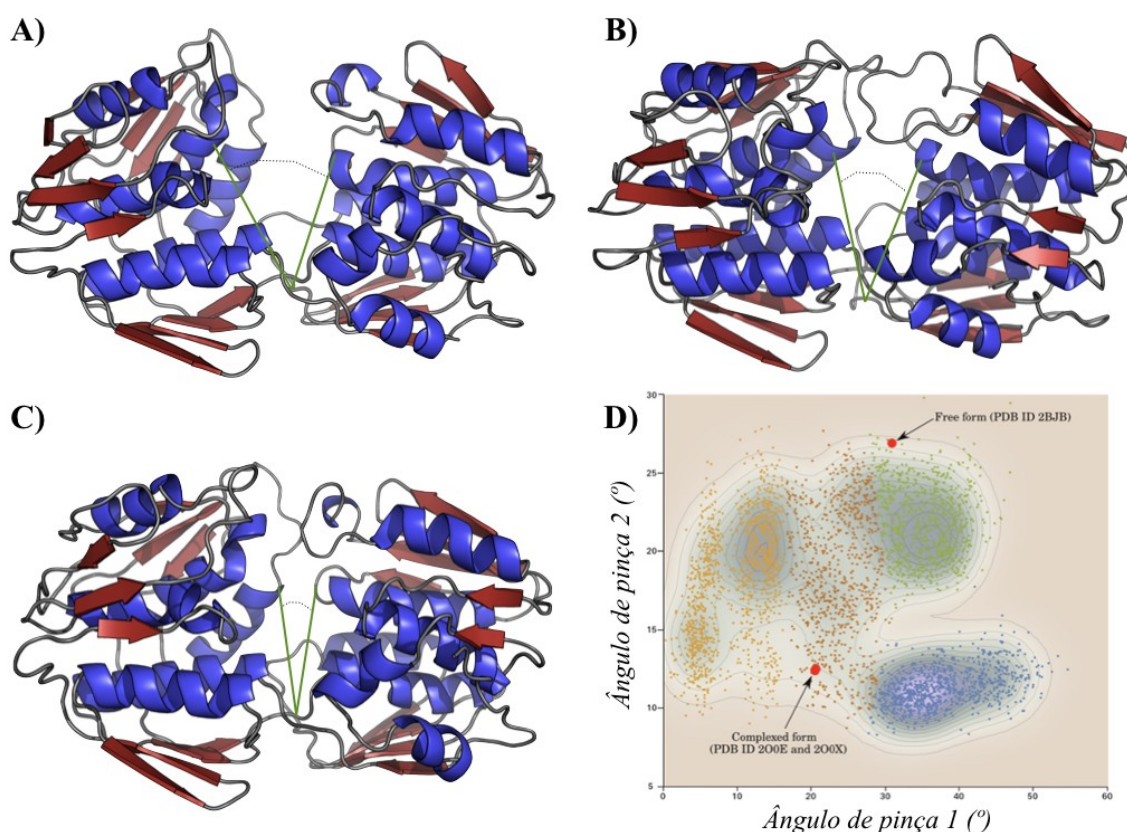
Fonte: Imagem gerada com o PyMOL (SCHRÖDINGER, LLC, 2010).

Esta topologia confere à enzima a possibilidade de assumir conformações bem distintas, representadas por estados abertos e fechados. Este fenômeno é muito importante, porque é na interface entre esses domínios que se encontra o sítio ativo. Para descrever os processos de aproximação e distanciamento entre os domínios globulares da enzima, os autores determinaram dois ângulos de pinça, um na parte anterior e outro na parte posterior da enzima, onde suas variações foram analisadas ao longo da simulação.

De acordo com as estruturas cristalográficas, a enzima EPSP sintase apresenta seu estado aberto na ausência de ligantes no seu sítio ativo e o estado fechado na presença de ligantes. Porém, foi observado que essa enzima pode assumir os dois tipos de conformação (aberto ou fechado), mesmo na ausência de ligantes no sítio ativo, demonstrando a importância da dinâmica da enzima EPSP sintase (Figura 4A, 4B e 4C). A partir destes resultados, foram propostos cinco estados conformacionais mais

prováveis da enzima em solução (Figura 4D), os quais poderiam ser utilizados para uma busca racional de pequenas moléculas com capacidade de atuar como moduladoras da atividade enzimática.

Figura 4 – Estados conformacionais da EPSP sintase. (A) EPSP sintase na sua conformação aberta. (B) EPSP sintase na conformação intermediária. (C) EPSP sintase na conformação fechada. (D) Disposição dos estados conformacionais observados ao longo da simulação por meio de dinâmica molecular. A estrutura da MtEPSP sintase está representada por *cartoon* e colorida por estrutura secundária (hélices: azul, fitas: vermelho e alças e voltas: cinza).



Fonte: Imagem gerada com PyMOL (SCHRÖDINGER, 2010). Figura 2D foi adaptada de TIMMERS *et al.*, 2017.

Além da importância para o reconhecimento de diferentes ligantes pelo sítio ativo, a flexibilidade das enzimas pode auxiliar a melhor compreensão sobre o impacto de mutações. Apesar de não haver estudos de mutações pontuais em *Mycobacterium tuberculosis*, Mizyed e colaboradores (2003) publicaram um extenso estudo da influência de mutações pontuais na atividade do EPSP sintase em *Escherichia coli* (MIZYED *et al.*, 2003). Neste estudo foram investigadas diversas mutações, sendo que quatro eram extremamente conservadas no gene *aroA* (Lys22, Asp49, Arg100 e Arg386), em diferentes organismos, as quais causavam uma diminuição na atividade

enzimática. A partir dos resultados, foram propostas duas hipóteses para explicar a diminuição da atividade, um relacionado à posição das moléculas de água no sítio ativo, e outro que estaria relacionado com a estabilidade da proteína, ou seja, que estas mutações afetariam a dinâmica da enzima. Como estes resíduos de aminoácidos são conservados em diferentes organismos, Timmers e colaboradores avaliaram a segunda hipótese na EPSP sintase de *M. tuberculosis* por meio de simulações computacionais. Para esta etapa, foi utilizada uma técnica mais elaborada de dinâmica molecular denominada de metadinâmica. Os resultados das simulações permitiram demonstrar que o Asp54, que é equivalente à mutação Asp49 em *E. coli*, favorece o posicionamento da cadeira lateral da Lys23 (Lys22 em *E. coli*) na superfície da cavidade central do sítio ativo. E esta Lys23 em EPSP sintase de *M. tuberculosis* está diretamente envolvida na catálise da reação e na acomodação do substrato. Assim, qualquer alteração na posição do Asp54 na EPSP sintase por um resíduo de características hidrofóbica, teria um impacto significativo na atividade enzimática da enzima. Com isso, foi proposto que o Asp54 apresenta como função principal a coordenação da cadeia lateral da Lys22, favorecendo que sua conformação permaneça voltada para o centro reativo da enzima.

Referências

- BASSIRI-JAHROMI, Shahindokht. Punica granatum (Pomegranate) activity in health promotion and cancer prevention. **Oncology Reviews**, Thran, v. 12, n. 345, p.1-7, 30 jan. 2018.
- BERMAN, Helen M. *et al.* The protein data bank. **Nucleic Acids Research**, Piscataway, v. 28, n. 1, p. 235-242, 1º jan. 2000.
- CARVALHO, Diego *et al.* Structural evidence of quercetin multi-target bioactivity: a reverse virtual screening strategy. **European Journal of Pharmaceutical Sciences**, Montevideo, v. 106, p. 393-403, ago. 2017.
- CERETO-MASSAGUÉ, Adrià *et al.* Tools for in silico target fishing. **Methods**, Tarragona, v. 71, p. 98-103, set. 2014.
- HUNG, Le Viet *et al.* LIBRA: LIgand Binding site Recognition Application. **Bioinformatics**, Rome, p.4020-4022, 26 ago. 2015.
- KARPLUS, M., McCAMMON, J. Molecular dynamics simulations of biomolecules. **Nature Structural Biology**, v. 9, 646-652, 2002.
- KARPLUS, M.; PETSKO, G. A. Molecular dynamics simulations in biology. **Nature**, v. 347, p. 631-639.
- KINNINGS, Sarah L.; JACKSON, Richard M.. LigMatch: A Multiple Structure-Based Ligand Matching Method for 3D Virtual Screening. **Journal of Chemical Information and Modeling**, Leeds, v. 49, n. 9, p. 2056-2066, 17 ago. 2009.
- KINNINGS, Sarah L.; JACKSON, Richard M. ReverseScreen3D: A Structure-Based Ligand Matching Method To Identify Protein Targets. **Journal of Chemical Information and Modeling**, Leeds, v. 51, n. 3, p. 624-634, 28 fev. 2011.
- JASIAL, Swarit; HU, Y; BAJORATH, Jürgen. Determining the degree of promiscuity of extensively assayed compounds. **Plos One**, Bonn, v. 11, n. 4, p.1-15, 15 abr. 2016.
- LIANG, Ting-ting *et al.* Modeling Analysis of Potential Target of Dolastatin 16 by Computational Virtual Screening. **Chemical and Pharmaceutical Bulletin**, Shanghai, v. 66, n. 6, p.602-607, 1º jun. 2018.

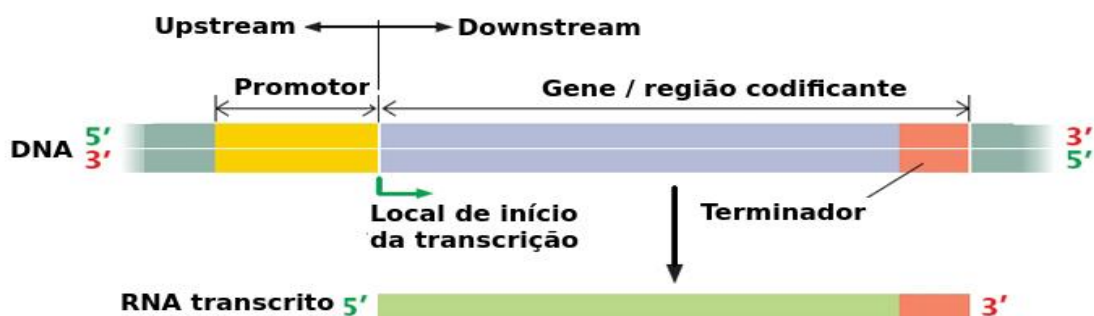
- LUO, Heng *et al.* DRAR-CPI: a server for identifying drug repositioning potential and adverse drug reactions via the chemical-protein interactome. **Nucleic Acids Research**, Shanghai, v. 39, n. 2, p.W492-W498, 10 maio 2011.
- LUO, Heng *et al.* DDI-CPI, a server that predicts drug–drug interactions through implementing the chemical–protein interactome. **Nucleic Acids Research**, Shanghai, v. 42, n. 1, p.W46-W52, 29 maio 2014.
- LIU, Xiaofeng; JIANG, Hualiang; LI, Honglin. SHAFTS: a hybrid approach for 3D molecular similarity calculation. 1. Method and Assessment of Virtual Screening. **Journal of Chemical Information and Modeling**, Shanghai, v. 51, n. 9, p. 2372-2385, 25 ago. 2011.
- MEI, Yicheng; YANG, Baowei. Rational application of drug promiscuity in medicinal chemistry. **Future Medicinal Chemistry**, Hualan, v. 10, n. 15, p.1835-1851, ago. 2018.
- MIZYED, S., WRIGHT, J. E. I., BYCZYNSKI, B., BERTI, P. J. Identification of the Catalytic Residues of AroA (Enol pyruvylshikimate 3-Phosphate Synthase) Using Partitioning Analysis. **Biochemistry**. v. 42, p. 6986-6995, 2003.
- NIEDERMEYER, Timo; BRÖNSTRUP, Mark. Natural product drug discovery from microalgae. *In*: POSTEN, Clemens; WALTER, Christian (Ed.). **Microalgal Biotechnology: Integration and Economy**. Boston: De Gruyter, 2012. p. 169-189.
- QIN, Zhiwei *et al.* Formicamycins, antibacterial polyketides produced by *Streptomyces formicae* isolated from African *Tetraponera* plant-ants. **Chemical Science**, Norwich, v. 8, n. 4, p. 3218-3227, fev. 2017.
- TEILUM, K., OLSEN, J. G., KRAGELUND, B. B. Protein stability, flexibility and function. **Biochimica et Biophysica Acta – Proteins and Proteomics**, v. 1814, p. 969-976, 2011.
- TIMMERS, L. F. S. M.; NETO, A. M. S.; MONTALVÃO, R. W.; BASSO, L. A.; SANTOS, D. S.; NORBERTO de SOUZA, O. EPSP synthase flexibility is determinant to its function: computational molecular dynamics and metadynamics studies. **Journal of Molecular Modeling**, v. 23, p. 197-204, jul. 2017.
- USHA, Talambedu *et al.* Identification of anti-cancer targets of eco-friendly waste punica granatum peel by dual reverse virtual screening and binding analysis. **Asian Pacific Journal of Cancer Prevention**, Bangalore, v. 15, n. 23, p. 10345-10350, 6 jan. 2015.
- van GUNSTEREN, W., MARK, A. Validation of molecular dynamics simulation. **The Journal of Chemical Physics**, v. 108, p. 6109-6116, 1998.
- WANG, Xia *et al.* PharmMapper 2017 update: a web server for potential drug target identification with a comprehensive target pharmacophore database. **Nucleic Acids Research**, Shanghai, v. 45, n. 1, p.W356-W360, 3 maio 2017.
- YANG, Lun *et al.* Chemical-protein interactome and its application in off-target identification. **Interdisciplinary Sciences: Computational Life Sciences**, Shanghai, v. 3, n. 1, p. 22-30, mar. 2011.

1 Conceitos básicos da área

Os procariotos são organismos unicelulares que apresentam seu DNA livre na célula. Mesmo não possuindo membrana celular, a regulação em nível de transcrição ainda é complexa. Eles podem ser diferenciados com base na estrutura das suas paredes celulares nas e suas relações filogenéticas. Esta separação foi definida por Hans Christian Gram em 1884 por meio da diferença observada na constituição da parede celular (técnica de coloração diferencial de Gram). Desta forma, pôde-se separar em dois grupos filogenéticos bacterianos com diferenças morfológicas, genéticas e metabólicas, incluindo as sequências promotoras: (A) bactérias Gram-negativas e (B) bactérias Gram-positivas (RUFF; RECORD; ARTSIMOVITCH, 2015).

A unidade de transcrição em procariotos representa todos os elementos que determinam a expressão de um gene. Ela é determinada pela presença da própria sequência da região codificante e por mais duas regiões que sinalizam o início e o fim da unidade de transcrição: a região promotora e a região terminadora da transcrição (vide Figura 1).

Figura 1 – Organização da unidade de transcrição de um gene bacteriano



Fonte: Adaptada de Reys *et al.* (2011).

Um promotor bacteriano típico de *E. coli* (exemplo-base de bactéria Gram-negativa) localiza-se aproximadamente 70pb antes do ponto de início da transcrição

¹ Instituto Federal do Rio Grande do Sul. *E-mail*: rafael.coelho@farroupilha.ifrs.edu.br

² Universidade de Caxias do Sul. *E-mail*: gdalba@alumni.ubc.ca

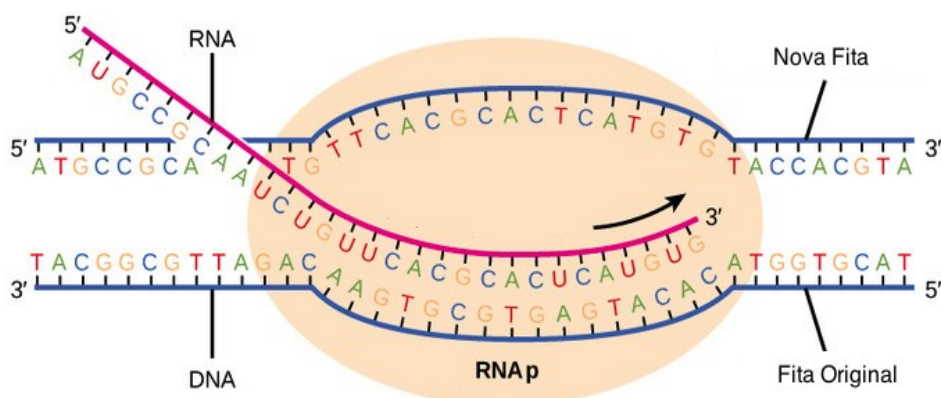
³ Universidade de Caxias do Sul. *E-mail*: sasilva6@ucs.br

gênica. Através da análise comparativa de vários promotores bacterianos, foram definidas duas sequências consenso: uma localizada a $-10pb$ ($5'$ -TATAAT- $3'$) do ponto de início da transcrição e uma localizada a $-35pb$ ($5'$ -TTGAC- $3'$) (NELSON; COX, 2011).

A regulação da expressão gênica em procariotos ocorre essencialmente no nível de transcrição. Embora proteínas de ligação ao DNA possam afetar a eficiência da transcrição, ela depende primordialmente das interações entre a enzima de transcrição RNA polimerase (RNAP) e as regiões promotoras (locais específicos de DNA). RNAP bacterianas podem ser isoladas de duas formas: (1) o núcleo de RNAP catalisa a polimerização de ribonucleotídeos para o complemento de RNA de uma sequência de DNA; e (2) holoenzima RNAP, contém as subunidades da molécula do núcleo mais um fator protéico (Sigma), que permite que a holoenzima reconheça elementos promotores e inicie a transcrição nesses locais.

Os fatores sigma (σ) são reguladores do processo de transcrição e compõem uma classe de proteínas que é composta pelas subunidades α 1, α 2, β , β' e ω . A holoenzima RNAP não apresenta afinidade específica pela sequência de DNA a não ser que ela seja associado à subunidade σ , capaz de reconhecer promotores gênicos específicos. No entanto, esta associação com a RNAP é transitória. Sendo assim, para iniciar a transcrição de uma região codificante, a enzima RNAP deve reconhecer a região promotora, a qual é constituída por uma sequência de nucleotídeos específicos que sinalizam e direcionam a transcrição do gene adjacente. À medida que a RNA polimerase avança (vide Figura 2), uma molécula de RNA transcrito é gerada (REYS *et al.*, 2011).

Figura 2 – Ligação da RNA Polimerase ao DNA



Fonte: Adaptada de Reys *et al.* (2011).

Promotores de *E. coli* e *B. subtilis* possuem tanto suas similaridades quanto diferenças. Encontram-se as seguintes similaridades entre promotores transcritos pelos fatores $E\sigma^A$ ou $E\sigma^{70}$: (a) sequências conservadas nos hexâmeros 35 e 10; (b) distância entre os dois hexâmeros; e (c) posição do sítio de início de transcrição. No entanto, genes estruturais ligados a alguns promotores de *E. coli* (por exemplo, o gene *lacUV5*) não conseguem ser transcritos pela RNA polimerase de *B. subtilis*. Além disso, estudos bioquímicos demonstram que *B. subtilis* é menos tolerante ao desvio em relação ao consenso de 12pb, diferentemente do que *E. coli* (YAMADA *et al.*, 1991; O'NEILL, 1991).

Por fim, existem sequências cis-regulatórias localizadas imediatamente ao montante da região -35 (elementos UP, *upstream elements*), que também podem afetar o reconhecimento e a atividade dos promotores bacterianos. Este tipo de sequência apresenta 20pb de comprimento e o consenso é $^{-59}\text{NNAAAWWTWT TTTTNNNAAANN}^{-38}$, em que W pode ser substituído por A ou T e N pode representar qualquer base. Os elementos UP podem ser divididos em duas sub-regiões distintas: distal (centralizada na posição -52pb) e proximal (centralizada na posição -42pb). Estas são reconhecidas pelo domínio C-terminal de uma das duas subunidades α da holoenzima RNAP, estimulando assim a transcrição (HOOK-BARNARD; HINTON, 2007).

Os promotores possuem características estruturais próprias (conformação), diferentes das regiões não promotoras, que também podem ser incorporadas nos estudos destes elementos, tais como a curvatura e a estabilidade dos promotores. A curvatura do DNA está envolvida em processos biológicos, como a transcrição do DNA. Em bactérias, a curvatura é mais acentuada na região antecedente ao promotor. Para realizar o cálculo de curvatura do DNA, analisam-se os ângulos de enovelamento, torção e inclinação dos nucleotídeos. As principais metodologias existentes para a obtenção destes valores são: (1) eletroforese em gel de agarose; (2) análise do ângulo dos dinucleotídeos AA; e (3) cristalografia de raio-X (KOZOBAY-AVRAHAM *et al.*, 2008). Já a estabilidade é uma propriedade que depende primordialmente da soma de interações entre os dinucleotídeos da sequência. A estabilidade geral para um oligonucleotídeo pode ser descoberta a partir da contribuição relativa de cada interação vizinha mais próxima no DNA (KANHERE *et al.*, 2005).

Percebe-se que o reconhecimento de um promotor bacteriano é um ponto de regulação da expressão gênica, já que este reconhecimento determina o início do gene, o momento e a quantidade que será expressa pela célula (RUFF; RECORD; ARTSIMOVITCH, 2015).

2 Trabalhos relacionados

A literatura apresenta diversos métodos computacionais, que têm como objetivo a predição de regiões promotoras em procariotos: Análise Probabilística, Reconhecimento de Sinais (RS) e Aprendizado de Máquina (AM).

Os trabalhos encontrados na literatura de reconhecimento de sinais são os seguintes: Sequência Consenso (SC) (LISSER; MARGALIT, 1993) e Matriz de Posições Ponderadas (MPP) (HERTZ; STORMO, 1999; XING *et al.*, 2005). O método Sequência Consenso consiste em alinhar um conjunto de sequências identificadas previamente como promotora e, posteriormente, pesquisar por regiões conservadas em seu conteúdo. Cada coluna no alinhamento fornece a variação encontrada nesta posição do promotor. Já no caso das Matrizes de Posições Ponderadas, assume-se que cada linha da matriz corresponde a um dos quatro nucleotídeos, e cada coluna, a um alinhamento. Os elementos da matriz são os pesos utilizados para pontuar uma sequência teste, conforme uma medida que quantifica a aderência ao modelo. Calcula-se então a pontuação pela soma dos pesos de cada letra (nucleotídeo) alinhada em cada posição.

Segundo Song *et al.* (2007), todas as técnicas baseadas em reconhecimento de sinais apresentam limitações similares: (1) a variação dos nucleotídeos é grande; (2) assumem a independência entre bases adjacentes; (3) não permitem a presença de múltiplos elementos dos promotores, inserções, deleções ou espaço variável entre os elementos; e (4) o resultado pode variar de acordo com o método de alinhamento.

Enquanto isto, os trabalhos encontrados de aprendizado de máquina podem ser divididos em: Modelos Ocultos de Markov (MOM) (HAYKIN, 1998), Redes Neurais Artificiais (DEMELER; ZHOU, 1991; O'NEILL, 1991; MAHADEVAN; GHOSH, 1994; PEDERSEN; ENGELBRECHT, 1995; OPPON, 2000; COTIK *et al.*, 2005; BURDEN *et al.*, 2005; SANTOS *et al.*, 2008; DE AVILA E SILVA *et al.*, 2011; MENG *et al.*, 2013; UMAROV e SOLOVYEV, 2017; KUMAR; BANSAL, 2017) e Máquinas de Vetor de Suporte (GORDON *et al.*, 2003; KIRYU *et al.*, 2005; GORDON *et al.*, 2007; POLAT; GUNES, 2007; ZANATY, 2012; DE JONG *et al.*, 2012; MEYSMAN *et al.*, 2014; SIWO *et al.*, 2016).

Um Modelo Oculto de Markov é um modelo estatístico, cujo sistema é um processo de Markov com parâmetros desconhecidos, em que o desafio é determinar os parâmetros ocultos a partir dos parâmetros observáveis. Os parâmetros extraídos do modelo podem então ser usados para realizar novas análises, realimentando assim o sistema. A vantagem desta metodologia é a capacidade de capturar regularidades em sequências de caracteres, considerando a variação nos símbolos observados em cada estado. No entanto, segundo Reis (2005), uma das restrições deste tipo de solução é o tamanho do conjunto de treinamento, visto a grande quantidade de parâmetros que

precisam ser estimados. Além disso, há uma dificuldade na incorporação de outras características dos promotores, em seu algoritmo, como a composição dos dinucleotídeos ou trinucleotídeos da sequência e informações sobre a estabilidade.

Já as Redes Neurais Artificiais são sistemas de aprendizado de máquina, inspirados no funcionamento de redes neurais biológicas. Segundo Wu e Mclarty (2000), elas aprendem a partir dos exemplos e apresentam alguma capacidade de generalização do conjunto de treinamento. A vantagem desta solução é o fato de elas aprendeendem a reconhecer padrões degenerados, imprecisos e incompletos. Estas são características dos promotores, o que é essencial para o presente trabalho. Além disso, segundo Cotik *et al.* (2005), apresentam ótimo desempenho em grandes sequências genômicas. Como desvantagem, destaca-se a necessidade de sincronismo nos dados de entrada (alinhamento de sequências).

Por fim, as Máquinas de Vetor de Suporte também podem ser utilizadas para classificações de padrões. Neste modelo, procura-se construir um hiperplano como superfície de decisão de tal forma que a margem de separação entre exemplos positivos e negativos seja máxima. As SVMs podem fornecer um bom desempenho de generalização em problemas de classificação de padrões, apesar de não incorporarem conhecimento do domínio do problema (HAYKIN, 1998). No Quadro 1, apresenta-se uma comparação entre as vantagens e desvantagens de usar cada uma das abordagens computacionais citadas previamente.

Quadro 1 – Comparação entre as abordagens computacionais para reconhecimento de regiões promotoras

Metodologia	Técnica	Vantagens	Desvantagens
Reconhecimento de Sinais (RS)	Sequencia Consenso (SC)	– simplicidade	– alta variação dos nucleotídeos. – assume a independência entre bases adjacentes
	Matrizes de Posições Ponderadas (MPP)	– eficiente em sequências pequenas	– não permite múltiplos elementos dos promotores – varia de acordo com o método de alinhamento
Aprendizado de Máquina (AM)	Modelo Oculto de Markov (MOM)	– captura regularidades em sequências de caracteres	– conjunto de treinamentos grandes – incorporação de características dos promotores
	Máquinas de Vetor de Suporte (SVM)	– desempenho de generalização na classificação	– não incorpora conhecimento do domínio do problema
	Rede Neural Artificial (RN)	– reconhecer padrões degenerados – desempenho em sequências genômicas.	– sincronismo nos dados de entrada (alinhamento de sequências)

Fonte: Elaboração dos autores.

3 Ferramentas computacionais relacionadas

As primeiras aplicações de Redes Neurais Artificiais na predição de promotores apresentados nos trabalhos de Demeler e Zhou (1991) e O'Neill (1991), obtiveram alta acurácia na predição, mas o número de falsos-positivos foi igualmente alto. Outra abordagem foi apresentada por Mahadevan e Ghosh (1994), através de uma combinação entre duas RNs para a identificação de promotores de *E. coli*. Todos os promotores deste trabalho tinham espaçamento entre 15 e 21 nucleotídeos entre os hexâmeros característicos. A primeira RN predizia os hexâmeros consensuais, enquanto a segunda foi designada para o reconhecimento da sequência (65 nucleotídeos), sendo o espaço entre os hexâmeros variável. Uma vez usada a informação da sequência inteira, ocorreram dependências entre as bases em várias posições. Isto refletiu-se em um treinamento pobre e uma predição realizada por duas redes sem neurônios, na camada oculta.

Pedersen e Engelbrecht (1995) predisseram o local de início da transcrição (TSS), a medida do conteúdo da informação e identificaram novos sinais característicos correlacionados com o local de início. Para isto, foram usados dois diferentes esquemas de codificação, com janelas de 51 e 65 nucleotídeos. Uma ideia interessante, neste trabalho, foi a medida do conteúdo de informação relativa dos dados de entrada, pelo uso da habilidade da RN, para aprender corretamente, como avaliado pelo coeficiente de correlação do teste máximo.

Outra ferramenta baseada em RNs é o *Nureka Artificial Neural Systems* (NANS). Oppon (2000) executou um teste neste sistema, a partir do conjunto composto de 31 sequências de 75 bases, sendo cinco regiões promotoras e 26 regiões codificantes de *Escherichia coli*. Com um limiar de corte de 6, o NANS acerta a classificação de uma sequência como promotora em 60% das vezes e em 50% das vezes em afirmar que não é promotora. Uma provável explicação para este baixo desempenho, além do baixo número de amostras, é a especificidade do sistema, que foi desenvolvido especificamente para um organismo. Em 2005, Burden *et al.* melhoraram este sistema incorporando nele a informação sobre a distância entre o sítio de início de transcrição (TSS) e o sítio de início da tradução TLS (primeiro nucleotídeo do gene). Com um conjunto de dados de 771 promotores, eles melhoraram a predição em 60%, quando comparado ao trabalho de Oppon (2000), e reduziram o número de falsos-positivos.

Cotik *et al.* (2005) criaram uma metodologia híbrida, denominada HPAM (*Hybrid Promoter Analysis Methodology*) que combina redes neurais, lógica *Fuzzy* e algoritmos genéticos. Seu funcionamento pode ser descrito pelos seguintes passos: (1) decomposição através da rede neural em módulos dos motivos das regiões de ligação

referentes a sequências não específicas de DNA; (2) inferências de lógica *Fuzzy* para associação entre os módulos identificados pela rede; e (3) utilização do método de reconhecimento de padrões que usa algoritmos genéticos chamados MOSS (*Multi-objective Scatter Search*), para encontrar os motivos mais representativos (BAJESTANI *et al.*, 2009).

Na sequência, Santos *et al.* (2008) propuseram um método de aprendizado para um classificador *Bayesiano* para reconhecimento de promotores procarióticos. Para isto, implementaram em Java e Matlab um modelo de classificador, tendo como base o método *Naive Bayes* para identificação de promotores reconhecidos pelo fator Sigma70. A bactéria utilizada foi a *Escherichia coli* e o método apresentou acurácia de 90%, mas a quantidade de sequências utilizadas foi 99.

Bland *et al.* (2010) utilizaram redes neurais artificiais que usavam perfis de dados SIDD (*Stress-Induced Duplex Destabilization*). Ou seja, redes que levam em consideração propriedades estruturais para a predição de uma região promotora. Segundo os autores, houve um ganho significativo no desempenho da predição, quando utilizado SIDD juntamente com redes neurais. Eles utilizaram o genoma de *E. coli* e locais de início de transcrição do banco de dados Regulon, totalizando 1.648 promotores. A melhor acurácia obtida ficou na faixa de 70% a 80%.

De Avila e Silva *et al.* (2011) desenvolveram um *software* chamado BacPP (*Bacterial Promoter Prediction*), cujo objetivo é a utilização de Redes Neurais Artificiais na predição, caracterização e no reconhecimento de promotores de bactérias Gram-negativas, a partir de um determinado sigma. Ele está disponível em versão Web (implementada em linguagem PHP e Python) e versão *desktop* (implementada em linguagem Python e linguagem R). Para utilizar este sistema, o usuário deve inserir a sequência de nucleotídeos ou enviar um arquivo no formato FASTA. Após, ele deve selecionar fatores σ e escolher o formato de saída (monitor ou arquivo). Os resultados obtidos com o BacPP foram satisfatórios e comparáveis com a literatura. Através da análise da melhor arquitetura para cada sigma, eles obtiveram acurácia média de 71.67%, especificidade de 71.08% e sensibilidade de 72.98%, com baixa variação entre os fatores sigma ($\sigma 70$ obteve o melhor desempenho, 77% de acurácia). Através de suas simulações, foi possível perceber que o aumento do número de neurônios na camada oculta não necessariamente melhora o desempenho da rede neural.

Meng *et al.* (2013) desenvolveram uma biblioteca para melhorar a eficiência do treinamento da rede neural, através da análise quantitativa dos possíveis pontos de mutação e de sequências conservadas. Eles utilizaram a ferramenta *Neural Network Toolbox* disponibilizada no Matlab. Eles configuraram a rede com três camadas, sendo uma oculta. Foram utilizados 896 neurônios na camada de entrada e um neurônio na

camada de saída. No entanto, sua validação foi feita apenas com 100 sequências, o que não permite avaliar de maneira satisfatória o seu desempenho, devido à pequena amostragem.

Por fim, os trabalhos mais recentes encontrados sobre predição de promotores, através do uso de redes neurais artificiais, foram realizados por Umarov e Solovyev (2017) e por Kumar e Bansal (2017).

Umarov e Solovyev implementaram um *software* em linguagem de programação Python (biblioteca Keras) chamado CNNProm que usa rede neural convolucional e obteve alta acurácia (*E. coli* 84% e *B. Subtilis* 86%), porém apenas utilizaram $\sigma 70$ totalizando 746 sequências promotoras com 81 nucleotídeos cada. Seus dados foram coletados do banco de dados DBTBS (SIERRO *et al.*, 2008). Este tipo de rede se baseia na organização do *cortex* visual dos animais para usar como padrão de conectividade entre os neurônios, permitindo a redução de conexões desnecessárias e o compartilhamento do peso das arestas.

Kumar e Bansal focaram seus esforços na análise de características estruturais (baixa estabilidade, baixa flexibilidade e alta curvatura) de promotores de *E. coli*. Eles afirmam que regiões promotoras associadas com alta expressão genética têm estruturas de baixa estabilidade, mais rígidas e maior curvatura, se comparadas com outros genes. Eles utilizaram sequências de 1001 nucleotídeos divididos em *5-Fold*, selecionando 80% dos dados para treinamento.

A Tabela 2 apresenta uma comparação entre os trabalhos descritos anteriormente, que utilizam Redes Neurais Artificiais, como abordagem computacional para o problema de predição de regiões promotoras. Como pode ser observado, em ambas as tabelas a maioria dos trabalhos visa a tratar da predição de promotores em bactérias Gram-negativas, tendo em vista a maior quantidade de dados disponíveis.

Quadro 2 – Comparação entre as soluções que utilizam a técnica de Redes Neurais Artificiais

Ferramenta	Bactéria	Características	Desempenho	Autores
NeuralWare	<i>E. coli</i>	- arquitetura simplificada - número de neurônios (1 a 10, 24) - <i>software Neuralware II Professional</i> (Demeler e Zhou) - linguagem FORTRAN em um SUN 386 (O'Neill) - linguagem C em um sistema UNIX (Mahadevan e Ghosh)	- número de amostras: 80, 39 e 126	Demeler e Zhou (1991); O'Neill (1991); Mahadevan e Ghosh (1994)
KullbackDist	<i>E. coli</i>	- coeficiente de correlação do teste máximo - neurônios na camada oculta (2 a 3) - linguagem C em um sistema UNIX	- número de amostras: 167	Pedersen e Engelbrecht (1995)
NANS	<i>E. coli</i> ; <i>B. subtilis</i> ; <i>Mycobacterium tuberculosis</i>	- análise de distribuição de frequência - <i>Hidden Markov Model</i> - <i>software Nureka Artificial Neural Systems</i>	- acurácia (50 a 60%) - número de amostras: 10 a 50 - sequências não normalizadas (40 a 75pb) - não foi feita análise de falsos negativos	Oppon (2000)
NNPP2.2	<i>E. coli</i>	- distância entre o local de início da transcrição e o local de início da tradução	- número de amostras: 272 - acurácia (64%)	Burden <i>et al.</i> (2005)
Naive Bayes	<i>E. coli</i>	- classificador <i>Bayesiano (método de Naive Bayes)</i> - <i>softwares</i> de alinhamento <i>WSONSENSUS</i> e <i>PATSER</i> - linguagem <i>Java</i> e <i>software</i> GNU Octave	- número de amostras: 99	Santos <i>et al.</i> (2008)
SIDD	<i>E. coli</i>	- perfis de dados SIDD (dados estruturais dos promotores)	- acurácia (70 a 80%) - número de amostras: 1648	Bland <i>et al.</i> (2010)
BacPP	<i>E. coli</i>	- análise de diferentes fatores de transcrição - validação cruzada (<i>k-cross validation</i>) - análise de desempenho através de três métricas: sensibilidade, especificidade e acurácia. - linguagens R e Python	- acurácia (s24, 86.9%; s28, 92.8%; s32, 91.5%; s38, 89.3%, s54, 97.0%; e s70, 83.6%). No geral, 76%. - número de amostras (s24, 69; s28; 21; s32, 71; s38, 99; s54, 38; s70, 740)	De Avila e Silva <i>et al.</i> (2011)
<i>Neural Network Toolbox</i>	<i>E. coli</i>	- análise de pontos de mutação - análise de sequências conservadas - coeficiente de correlação - neurônios na camada oculta (19) - linguagem Matlab em um Microsoft Windows 7 64-bit	- número de amostras: 100 - alto tempo de treinamento devido ao número de neurônios na camada oculta.	Meng <i>et al.</i> (2013)
CNNProm	<i>E. coli</i> ; <i>B. subtilis</i>	- página web (<i>webserver</i>) - não utiliza nenhuma informação sobre características de promotores específicos	- acurácia (<i>E. coli</i> 84% e <i>B. subtilis</i> 86%) - apenas σ_{70} e σ_A	Umarov e Solovyev (2017)
TSS RN	<i>E. coli</i>	- análise de características estruturais (baixa estabilidade, baixa flexibilidade e alta curvatura) de promotores - seis diferentes transcriptogramas procarióticos	- o perfil de curvatura obtido apresenta maior curva em regiões promotoras	Kumar e Bansal (2017)

Fonte: Elaborado pelos autores.

4 Potencialidades e limitações

Sequências promotoras são, via de regra, curtas, pouco conservadas em nível de posição relativa ao esperado e em nível de composição das regiões -10 e -35. A heterogeneidade tolerada biologicamente (ou seja, que não impede a sobrevivência de um micro-organismo) de sequências promotoras é descrita como um dos principais desafios na aplicação de técnicas *in silico* sobre estes dados. A implicação mais relevante da estrutura dos promotores é a constante necessidade de compreender as distintas características que os compõem, sendo bastante variada entre organismos de um mesmo ou de distintos grupos.

Além da quantidade de dados, a heterogeneidade dos promotores leva a dois grandes problemas de cunho computacional: quantidades elevadas de falsos-positivos – podendo esta ser explicada tanto pela heterogeneidade quanto por dificuldades no processo de manejo (extração e preparação) de dados para aplicação em ferramentas *in silico* – e a compreensão incompleta, embora constantemente investigada, de parâmetros que atuam como discriminantes entre sequências promotoras (DALL’ALBA, 2017).

Abbas *et al.* (2015) definem alguns critérios para discutir quais parâmetros são mais eficientes na predição de promotores. Os autores argumentam que abordagens que fazem a combinação de sequência de nucleotídeos e de características estruturais apresentam parâmetros de classificação melhores que abordagens individuais. Ao integrar distintos elementos, removem-se características redundantes das sequências e, como consequência, reduz-se a geração de falsos-positivos. Portanto, integrar dados sobre estabilidade de promotores, energia de empilhamento, curvatura, *Stress Induced Duplex Destabilization* (SIDDD), entropia e entalpia além da composição de nucleotídeos pode refinar a busca *in silico* por promotores (TOWSEY *et al.*, 2007). Estas abordagens integrativas estão descritas na literatura, como trabalhos incluindo codificação de dados em estabilidade (ASKARY *et al.*, 2009; de AVILA E SILVA *et al.*, 2011; de AVILA E SILVA *et al.*, 2014; RANGANNAN; BANSAL, 2007); curvatura (DU e YU, 2008) e SIDDD (TOWSEY *et al.*, 2007).

Mesmo assim, não há garantia de que certas características estruturais dos promotores serão discriminantes entre sequências. Towsey *et al.* (2007), por exemplo, concluem que a codificação de dados, de acordo com valores de SIDDD, não fornece modelos discriminantes o suficiente para serem confiáveis na predição de TSS. Os autores fornecem, como explicação, que regiões SIDDD são amplas o suficiente para conter diversos candidatos fortes (ou seja, diversas regiões possuem as características esperadas de um TSS). O mesmo ocorre para a elevada curvatura em regiões intergênicas. Indica-se, assim, que a heterogeneidade esperada em um conjunto de

dados de promotores – como demonstrado por Dall’Alba *et al.* (2019) – torna improvável que um modelo ou um conjunto de características consiga servir totalmente aos requisitos de uma ferramenta preditora.

Outro desafio relevante, como foi exposto anteriormente, é proveniente das limitações similares entre técnicas empregadas para predição de promotores. Alguns dos desafios provêm de elementos como a variação de nucleotídeos, a não consideração de elementos distintos dos promotores, inserções, deleções ou espaço variável entre os elementos e a variação no resultado conforme o método de alinhamento. Enquanto abordagens mais complexas passam a responder adequadamente a alguns dos desafios, elas passam a depender de um elevado custo de processamento computacional.

Predizer promotores é uma tarefa de relevante auxílio para pesquisas laboratoriais, uma vez que estas podem ser caras, demoradas e demandarem bastante tempo e cuidado com amostras, equipamentos, entre outros. Portanto, buscar por promotores em um momento antecedente à pesquisa experimental pode auxiliar na mitigação do seu tempo e custo (JACQUES *et al.*, 2006).

Ainda, é necessário expandir as pesquisas destinadas a aplicações de predição em genomas inteiros. Ainda há muito a ser investigado quanto a técnicas de predição em um genoma completo, uma vez que grandes quantidades de dados resultam em elevadas taxas de falsos-positivos e demandam maior capacidade de processamento computacional (DE AVILA; SILVA; ECHEVERRIGARAY, 2012). Enquanto existem técnicas precisas de predição de Sítios de Início de Transcrição e de sequências promotoras, o número elevado de falsos-positivos permanece um desafio a ser resolvido.

É notável que a Bioinformática passa a estar encarregada, cada vez mais, por análises de dados pangenômicos, metagenômicos, proteômicos e metabolômicos – que, em sua essência, são compostos por grandes quantidades de informação a ser processada. Reforça-se, com isso, a importância de haver melhorias constantes no desempenho de ferramentas computacionais, o que inclui ferramentas de predição de elementos promotores. Contudo, não limita-se a necessidade apenas ao lado computacional da pesquisa: a qualidade dos dados (genomas bem sequenciados, livres de *gaps*, mais genomas representativos, descrição de poliploidias, etc.) facilita o desenvolvimento de abordagens mais eficazes, corretas (*e.g.*, com elevada precisão) e computacionalmente econômicas (ABDURAKHMONOV, 2016).

Referências

- ABBAS, M. M., MOHIE-ELDIN, M. M., EL-MANZALAWY, Y. Assessing the Effects of Data Selection and Representation on the Development of Reliable E.coli Sigma70 Promoter Region Predictors. **PLoS ONE**, v. 10, n. 3, p. 1-18, 2015.
- ABDURAKHMONOV, I. Bioinformatics: Basics, Development, and Future. *In*: ABDURAKHMONOV, Ibrokhim Y. (Ed.). **Bioinformatics**: updated features and applications. Croacia: Intech. p. 1-27. Cap. 1.
- ASKARY, A. *et al.*. N4: a precise and highly sensitive promoter predictor using neural network fed by nearest neighbors. **Genes. Genet. Syst.**, v. 84, n. 6, p. 425e430, 2009.
- BURDEN, S., LIN, Y.-X., ZHANG, R. Improving promoter prediction for the NNPP2.2 algorithm: a case study using *Escherichia coli* DNA sequences. **Bioinformatics**, v.21, n. 5, p. 601-607, 2005.
- COTIK, V., ZALIZ, R., ZWIR, I. A hybrid promoter analysis methodology for prokaryotic genomes. **Fuzzy Sets Syst.**, v. 152, n. 1, p. 83-102, 2005.
- DALL'ALBA, Gabriel. **Caracterização de seqüências promotoras de *Escherichia coli* para aprimoramento da ferramenta BacPP**. 2017. 20f. Trabalho de Conclusão de Curso – Universidade de Caxias do Sul, Caxias do Sul, 2017.
- DALL'ALBA, G., CASA, P. L., NOTARI, D. L., ADAMI, A. G., ECHEVERRIGARAY, S., DE AVILA E SILVA, S. Analysis of the nucleotide content of *Escherichia coli* promoter sequences related to the alternative sigma factors. **Journal of Molecular Recognition**, v. 32, n. 5, p. e2770, 2019.
- DE AVILA E SILVA, S.; ECHEVERRIGARAY, S.; GERHARDT, G. J. L. BacPP: bacterial promoter prediction a tool for Accurate Sigma Factor Specific Assignment in *Enterobacteria*. **Journal of Theoretical Biology**, v. 287, n. 0, p. 92-99, 2011.
- DE AVILA E SILVA, S.; FORTE, F., SARTOR, I. T. S.; ANDRIGHETTI, T.; GERHARDT, G. J. L.; DELAMARE, A. P. L.; ECHEVERRIGARAY, S. DNA duplex stability as discriminative characteristic for *Escherichia coli* s54- and s28-dependent promoter sequences. **Biologicals**, v. 42, p. 22-28, 2014.
- DE AVILA E SILVA, S.; ECHEVERRIGARAY, S. Bacterial Promoter Features Description and Their Application on *E. coli* in silico Prediction and Recognition Approaches. *In*: PÉREZ-SÁNCHEZ, Horacio (Ed.). **Bioinformatics**. Croacia: Intech. Cap. 10. p. 241-260, 2012.
- DE JONG, A.; PIETERSMA, H.; CORDES, M.; KUIPERS, O.; JAN, K. PePPER: a webserver for prediction of prokaryote promoter elements and regulons. **BMC Genomics**, v. 13, n. 1, p. 1-10, 2012.
- DEMELER, B.; ZHOU, G. Neural network optimization for *E. coli* promoter prediction. **Nucleic Acids Research**, v. 19, n. 7, p. 1593-1599, 1991.
- DU, Y., WU, T. A novel method of prokaryotic promoter regions prediction with feature selection: quadratic discriminant analysis approach. *In*: **Asian-Pacific Conference on Medical and Biological Engineering**, 7., Springer. p. 608-614, 2008.
- GORDON, L.; CHERVONENKIS, A. Y.; GAMMERMAN, A. J.; SHAHMURADOV, I. A.; SOLOVYEV, V. V. Sequence Alignment for Recognition of Promoter Regions. **Bioinformatics**, v. 19, p. 1964-1971, 2003.
- GORDON, J. J.; TOWSEY, M.; HOGAN, J.; MATHEWS, S. A.; TIMMS, P. Improved prediction of bacterial transcription start sites. **Bioinformatics**, v. 22, p. 142-148, 2006.
- HAYKIN, S. **Neural networks**: a comprehensive foundation. 2. ed. New Jersey: Prentice-Hall, 1999.
- HERTZ, Gerald Z.; STORMO, Gary D. Identifying DNA and protein patterns with statistically significant alignments of multiple sequences. **Bioinformatics**, v.15, n. 7/8, p. 563-577, 1999.
- HOOK-BARNARD, I.; JOHNSON, X. B.; HINTON, D. M. *Escherichia coli* RNA Polymerase Recognition of a σ_{70} – Dependent Promoter Requiring a -35 DNA Element and an Extended -10 TGN Motif. **Journal of Bacteriology**, v. 188, p. 8352-8359, 2006.
- JACQUES, P.-E.; RODRIGUE, S.; GAUDREAU, L.; GOULET, J.; BRZEZINSKI, R. Detection of prokaryotic promoters from the genomic distribution of hexanucleotides pairs. **BMC Bioinformatics**, v. 7, p. 423, 2006.

- KANHERE, A.; BANSAL, M. Structural properties of promoters: similarities and differences between prokaryotes and eukaryotes. **Nucleic Acids Research**, v. 33, n. 10, p. 3165-3175, 2005a.
- KIRYU, H.; OSHIMA, T.; KIYOSHI, A. Extracting relations between promoter sequences and their strengths from microarray data. **Bioinformatics**, v. 21, n. 7, p. 1062-1068, 2005.
- KUMAR, A.; BANSAL, M. Unveiling DNA structural features of promoters associated with various types of TSSs in prokaryotic transcriptomes and their role in gene expression. **DNA Research**, v. 24, n. 1, p. 25-35, 2016.
- LISSER, S.; MARGALIT, H. Compilation of *E. coli* mRNA promoter sequences. **Nucleic Acids Research**, v. 21, n. 7, p. 1507-1516, 1993.
- MAHADEVAN, I.; GHOSH, I. Analysis of *E. coli* promoter structures using neural networks. **Nucleic Acids Research**, v. 22, n. 11, p. 2158-2165, 1994.
- MENG, H., WANG, J., XIONG, Z., XU, F., ZHAO, G., WANG, Y. Quantitative design of regulatory elements based on high-precision strength prediction using artificial neural network. **PLoS ONE**, v. 8, n. 4, p. 1-9, 2013.
- MEYSMAN, P.; COLLADO-VIDES, J.; MORETT, E.; VIOLA, R.; ENGELEN, K.; LAUKENS, K. Structural properties of prokaryotic promoter regions correlate with functional features. **PloS one**, v. 9, n. 2, 2014.
- NELSON, D.; COX, M. **Princípios de Bioquímica de Lehninger**. 6. ed. Porto Alegre: Artmed, 2011.
- KOZOBAY-AVRAHAM, L., HOSID, S., VOLKOVICH, Z., BOLSHOY, A. Prokaryote clustering based on DNA curvature distributions. **Discrete Applied Mathematics**, v. 11, p. 2378-2387, 2008.
- O'NEILL, M. C. Training back-propagation neural networks to define and detect DNA-binding sites. **Nucleic Acids Research**, v. 19, n. 2, p. 313-318, 1991.
- OPPON, Ekow CruickShank. **Synergistic use of promoter prediction algorithms: a choice for a small training dataset?**. 2000. 238 f. Tese (Doutorado em Ciência da Computação) – South African National Bioinformatics Institute (SANBI), 2000.
- PEDERSEN, A. G.; ENGELBRECHT, J. Investigations of *Escherichia coli* promoter sequences with artificial neural networks: New signals discovered upstream of the transcriptional start point. **Proceedings of the Third International Conference on Intelligent Systems for Molecular Biology (ISMB-95)**, v. 3, p. 292-299, 1995.
- POLAT, K.; GÜNES, S. A novel approach to estimation of *E. coli* promoter gene sequences: Combining feature selection and least square support vector machine (FS_LSSVM). **Applied Mathematics and Computation**, v. 190, p. 1574-1582, 2007.
- RANGANNAN, V.; BANSAL, M., Identification and annotation of promoter regions in microbial genome sequences on the basis of DNA stability. **Journal of Biosciences**, v. 32, n. 5, p. 851-862, 2007.
- REIS, Adriana Neves dos. **Reconhecimento e predição de promotores procarióticos: investigação de uma metodologia in silico baseada em HMMs**. 2005. 114 f. Dissertação (Mestrado em Computação Aplicada) – Programa Interdisciplinar de Pós-Graduação em Computação Aplicada (PIPICA), 2005.
- REYS, L. **Dogma Central da Biologia Molecular e Introdução à Bioinformática**. Brasília-DF: W Educacional Editora e Cursos Ltda., 2011.
- RUFF, E. F.; RECORD, M. T. A. Initial events in bacterial transcription initiation. **Biomolecules**, v. 5, n. 2, p. 1035-106., 2015.
- SANTOS, I.; ALVES, R.; BLAHA, C. Estudo para Identificação de Promotores Procarióticos Através de Classificação *Bayesiana*. In: ENCONTRO REGIONAL DE MATEMÁTICA APLICADA E COMPUTACIONAL – VIII ERMAC, 8., 2008, Natal. **Anais [...]**, Natal, 2008.
- SIWO, G.; RIDER, A.; TAN, A.; PINAPATI, R.; EMRICH, S.; CHAWLA, N.; FERDIG, M. Prediction of fine-tuned promoter activity from DNA Sequence. **F1000Research**, v. 5, 2016.
- SONG, W.; MAISTE, P. J.; NAIMAN, D. Q.; WARD, M. J. Sigma 28 Promoter Prediction in Members of the Gammaproteobacteria. **FEMS Microbiology Letters**, v. 271, p. 222-229, 2007.

TOWSEY, M., HOGAN, J. M., MATHEUS, S., TIMMS, P. The in silico prediction of promoters in bacterial genomes. **International Conference on Genome Informatics**, v. 19, 178-189, 2007.

WU, C. H.; MCLARTY, J. W. **Neural networks and genome informatics**. New York: Elsevier, 2000.

UMAROV, R. K.; SOLOVYEV, V. V. Recognition of prokaryotic and eukaryotic promoters using convolutional deep learning neural networks. **PloS one**, v. 12, n. 2, 2017.

ZANATY, E. Support Vector Machines (SVMs) versus Multilayer Perceptron (MLP) in Data Classification. **Egyptian Informatics Journal**, p. 177-183, 2012.

1 Conceitos básicos da área

A transcrição é o processo no qual o RNA é sintetizado a partir de um molde de DNA. Esse processo ocorre tanto em procariotos quanto em eucariotos, quando a enzima RNA polimerase liga-se à região promotora de um gene e inicia a síntese de RNA. Enquanto que os procariotos possuem apenas uma RNA polimerase, que gera diferentes tipos de RNA (mensageiro: mRNA, ribossomal: rRNA, transportador: tRNA), os organismos eucariotos possuem três enzimas RNA polimerases (I, II, e III) que sintetizam, cada uma delas, os diferentes tipos de RNA (GLISOVIC *et al.*, 2008; LIEBERMAN; MARKS, 2009).

De maneira geral, o fluxo da informação genética ocorre da seguinte maneira: a informação gênica contida no DNA é transcrita em mRNA e, posteriormente, traduzida em proteínas, as quais desenvolvem diversas funções no organismo, como o controle da temperatura corporal, por exemplo. Neste capítulo iremos abordar apenas estudos envolvendo o mRNA, uma vez que este serve de molde para a produção de proteínas e, portanto, o estudo dos transcritos (mRNA) pode indicar situações de diferenciação celular ou adaptação a determinadas condições celulares e, ainda, pode ser utilizado como um indicador do *status* celular.

Quanto maior a complexidade dos organismos, maior é a regulação do sistema transcricional. Uma das principais diferenças entre os eucariotos e procariotos é que os eucariotos possuem mecanismos mais elaborados para o processamento dos transcritos: a primeira forma de RNA gerada é o RNA nuclear heterogêneo (hnRNA ou pré-mRNA), o qual contém sequências denominadas éxons (regiões codificadoras de proteínas) e íntrons (regiões não codificadoras de proteínas), este pré-mRNA é modificado em suas extremidades, a fim de evitar a degradação e, em seguida os íntrons são removidos através de um processo denominado *splicing*, o qual gera o mRNA maduro que é exportado para o citoplasma e servirá de molde para a síntese de proteínas. O processamento conhecido como *splicing* alternativo permite que diferentes partes codificantes (éxons) de um mesmo pré-mRNA possam ser utilizadas para produzir diferentes mRNAs maduros, e assim gerar proteínas distintas a partir de um único gene. Dessa forma, o *splicing* alternativo favorece a produção de transcritos maduros

¹ Universidade Federal do Rio Grande do Sul. *E-mail*: ivaine.sauthier@gmail.com

² Universidade de Caxias do Sul. *E-mail*: mvrossetto@ucs.br

distintos, conferindo maior plasticidade à expressão gênica e ao aumento na diversidade de proteínas (LIEBERMAN; MARKS, 2009).

Em condições celulares normais, apenas um pequeno número do total de genes de cada célula é expresso (isto é, é ativado para sintetizar mRNA e gerar uma proteína), enquanto que o restante dos genes estão inativos. Uma vez que os processos de transcrição de RNA e síntese de proteínas consomem uma quantidade considerável de energia, a ativação desses processos é requerida em determinadas situações (*e.g.*: para a manutenção das funções celulares e em resposta a estímulos do ambiente celular). Quando ocorrem mudanças no ambiente em que as células estão inseridas, ambos os organismos procariotos e eucariotos respondem a essas alterações, ativando ou reprimindo a expressão de um conjunto específico de genes (LIEBERMAN; MARKS, 2009).

Atualmente existem técnicas de biologia molecular capazes de quantificar o mRNA produzido por células, uma delas é a reação em cadeia da polimerase (PCR, *polimerase chain reaction*), a qual se baseia no processo de amplificação de fragmentos de DNA. Uma vez que estamos interessados em quantificar o mRNA, realiza-se um passo adicional a esta reação, no qual o mRNA da amostra de interesse é convertido em DNA complementar (cDNA), através da ação da enzima transcriptase reversa (RT). Essa enzima, proveniente de enzimas de vírus de genoma de RNA, utiliza o RNA como molde para criar cópias de cDNA, que são amplificadas e em seguida quantificadas. Um avanço importante dessa técnica foi o acompanhamento em tempo real da detecção e quantificação dos produtos gerados durante cada ciclo de amplificação, os quais são proporcionais à quantidade de cDNA disponível no início do processo; essa técnica é denominada PCR em tempo real quantitativa (qPCR). Ainda, aperfeiçoamentos como a multiplexagem, *i.e.*, a possibilidade de analisar diferentes fragmentos de cDNA de uma só vez permite que seja realizada a quantificação de até quatro fragmentos diferentes em uma mesma reação através da utilização de diferentes fluoróforos (ZAHA *et al.*, 2014; DOBNIK *et al.*, 2016).

Outra técnica muito utilizada para quantificação de cDNA é o microarranjo; essa técnica de alto rendimento permite avaliar uma enorme quantidade de transcritos de uma só vez, possibilitando a identificação de genes que contribuem para diferentes processos celulares como: diferenciação celular, resposta a estímulos específicos (*e.g.*: mudança de temperatura, resposta a infecções por patógenos).

O microarranjo de DNA (também conhecido como *chip* de DNA) é uma coleção de ordenada de *spots* (pontos de DNA), inseridos em uma superfície sólida. Cada *spot* contém sequências de DNA que correspondem a um gene específico, denominadas sondas que, por complementaridade de bases, são capazes de hibridizar (*i.e.*: ligar-se) à

sequência-alvo de cDNA que se quer analisar. Quanto maior a complementaridade de bases entre a sonda-cDNA maior será a ligação não covalente entre essas sequências. Após a hibridização, é realizada uma lavagem do *chip* para que as sequências com ligação não específicas sejam eliminadas. A hibridização sonda-cDNA nos *spots* é quantificada através da detecção de cDNA marcados no *chip* (em especial com fluoróforos, podendo também ser utilizada prata ou quimioluminescência para a marcação das amostras). A força do sinal de fluorescência no *spot* depende da quantidade das ligações entre o cDNA e as sondas presentes. Essa quantificação é realizada por um feixe de luz, e os dados captados são armazenados em um *software* que registra o padrão de emissão de fluorescência de cada *spot*. Essa técnica baseia-se na quantificação relativa, na qual a intensidade de uma característica é comparada à intensidade da mesma característica, sob uma condição diferente, *i.e.*: comparação entre amostras de tecido hepático normal e de tecido hepático doente, sendo que o tecido normal é marcado com um fluoróforo e o tecido doente com outro, a fim de determinar abundância relativa dos genes de interesse.

As mudanças no padrão de expressão gênica têm grande implicação no funcionamento das células, dos tecidos e organismos. Essas mudanças podem estar relacionadas ao desenvolvimento celular, à manifestação de doenças e à morte celular. A tecnologia de microarranjos de DNA permite a análise de variações no padrão de expressão de forma rápida e confiável (ZAHA *et al.*, 2014). As plataformas de microarranjo mais utilizadas em estudos de expressão gênica são das marcas Affymetrix e Agilent.

Outra técnica de alto rendimento, desenvolvida recentemente para a quantificação da expressão gênica, é o sequenciamento de RNA (RNA-seq), também conhecido como transcriptoma *shotgun*³ de sequenciamento completo, que utiliza a tecnologia de sequenciamento de nova geração (sistema automatizado para identificação da sequência de nucleotídeos da amostra analisada). Essa técnica baseia-se na conversão do mRNA em cDNA, o qual é fragmentado em pequenas sequências de cerca de 2 a 5 kb, de maneira aleatória, gerando uma biblioteca de cDNA, a fim de facilitar o sequenciamento. Os fragmentos sequenciados são sobrepostos para a realização da montagem e, em seguida, é realizada a anotação dos mesmos, *i.e.*: a identificação dos genes que codificam proteínas e, por fim, é realizada a quantificação desses fragmentos.

O emprego da técnica de RNA-seq apresenta grande contribuição em estudos transcriptômicos, uma vez que são gerados dados altamente reprodutíveis, informativos e com precisão na quantificação de transcritos. Dentre as plataformas de RNA-seq

³ *Shotgun*: bombardear aleatoriamente o cDNA a ser sequenciado com partículas que promovem sua fragmentação.

utilizadas, podemos citar a *HiSeq* da marca Illumina, a 454 da Roche e a SOLiD (*Sequencing by Oligonucleotide Ligation and Detection*) da Thermo Fisher. Ao contrário do microarranjo, o RNA-seq dispensa o conhecimento da sequência dos genes de interesse, portanto é muito utilizado para a identificação de genes e transcritos desconhecidos, além da identificação de isoformas geradas através do *splicing* alternativo. Ainda, o aperfeiçoamento dos métodos de detecção dos fragmentos permite a identificação de transcritos pouco abundantes (HRDLICKOVA *et al.*, 2017).

É cada vez maior o volume de dados produzido em estudos de biologia molecular e celular através de técnicas como microarranjo e RNA-seq, uma vez que as mesmas são capazes de produzir uma enorme quantidade de dados de maneira rápida a custos cada vez mais acessíveis (BROOKS, 2012; NALEJSKA *et al.*, 2014). A crescente demanda na busca de maior capacidade de armazenamento de dados deu origem a repositórios públicos como o *Gene Expression Omnibus* (GEO), no ano de 2000 (BARRETT *et al.*, 2011) e o *ArrayExpress*, em 2003 (KOLESNIKOV *et al.*, 2014). Os bancos de dados públicos surgem com o intuito de, além de armazenar dados moleculares de alto rendimento (genômicos, transcriptômicos e proteômicos), fornecer mecanismos de fácil utilização que permita que o usuário pesquise, localize, revise e faça *download* dos dados de estudos moleculares de interesse. Além dos dados armazenados nos repositórios, estão disponibilizadas informações sobre a classificação de cada amostra (*e.g.*: espécie, tecido do organismo estudado), o tipo de tratamento que cada amostra foi submetida (*e.g.*: utilização de fármaco) e ainda pode conter outras informações relacionadas (*e.g.*: dados clínicos de pacientes). Da mesma forma, outros bancos de dados públicos surgiram como o *The Cancer Genome Atlas* (TCGA) (TOMCZAK *et al.*, 2015), que engloba informações multidimensionais sobre alterações em 33 tipos tumorais humanos. As informações disponibilizadas nos diferentes bancos de dados podem ser utilizadas em estudos a fim de auxiliar a comunidade científica na busca do aprimoramento do conhecimento biológico.

O grande desafio da era ômica baseia-se na busca de informações biologicamente relevantes, o que requer análises robustas que consigam processar as interações complexas, geradas através das técnicas de alto rendimento. Neste sentido, o uso de ferramentas de biologia computacional tem encontrado grande utilidade nas análises ômicas, em especial na transcriptômica.

2 Ferramentas computacionais relacionadas

A fim de que se possa realizar comparações entre as amostras submetidas à quantificação de transcritos, os dados brutos resultantes das técnicas de microarranjo e

RNA-seq devem passar por um controle de qualidade, serem pré-processados e normalizados. Esses passos são necessários para identificar e filtrar possíveis *outliers*, que possam interferir nas análises posteriores. *Outliers* são classificados como observações que se diferenciam de maneira aberrante das outras observações, e podem introduzir viés às análises estatísticas e interpretação dos resultados.

O primeiro passo para a análise de dados de microarranjo é realizar o *download* dos arquivos de dados brutos no formato .CEL e, em seguida, realizar o controle de qualidade dos dados brutos através da análise de componentes principais (PCA) e da análise de correlação de intensidade (*array array intensity correlation*, AAIC), para a identificação dos *outliers*. Após a avaliação inicial de qualidade, o próximo passo no processamento de dados de microarranjo é a correção do *background* (*i.e.*: ruído de fundo). Esse passo é essencial, pois uma parte das medições de intensidade é devida à hibridização não específica e também ao ruído do sistema de detecção óptica. Portanto, as intensidades observadas precisam ser corrigidas para fornecer medições precisas de hibridizações específicas. O próximo passo inclui a normalização dos dados, e um método muito utilizado é o *robust multi-array average expression measure* (RMA) (IRIZARRY *et al.*, 2003).

Após é realizada a sumarização, uma vez que nas plataformas da marca Affymetrix, por exemplo, diversas sondas correspondem a um gene, ou seja, vários *spots* no *chip* correspondem a um gene. Para cada transcrito, a correção de *background* e a normalização das intensidades das sondas são sumarizadas em um número que representa a quantidade de cada transcrito na amostra analisada. Por fim, os dados sumarizados podem ser anotados com informações relativas aos identificadores gênicos (*e.g.*: *gene symbol*, *entrez gene*, *ensembl*).

Os dados provenientes de RNA-seq também passam pelo controle de qualidade, pré-processamento e normalização. O controle de qualidade e o pré-processamento dos dados brutos de RNA-seq é realizado através da PCA e da análise de correlação de intensidade para a identificação dos *outliers*. Diferentemente do microarranjo, que é baseado na correção do *background*, a normalização dos dados de RNA-seq leva em conta a quantificação dos transcritos (contagens discretas) e a extensão dos mesmos. Genes mais longos terão maior contagem de *reads* que genes mais curtos quando a expressão é a mesma. Portanto, a normalização é realizada dividindo-se a contagem (por milhão de *reads* mapeadas) pelo comprimento do gene.

Criado em 1993, o *software* de fonte aberta R (TEAM R, 2013) é uma ferramenta muito utilizada para análise de dados moleculares com plexos utilizando técnicas estatísticas básicas e avançadas. O R pode ser estendido através do uso de pacotes de funções, desenvolvidos pela comunidade científica e amplamente disponibilizados, a

fim de auxiliar análises complexas, o que acrescenta grandes potencialidades à sua versão-base.

O *Bioconductor* é um repositório que agrega diversos pacotes implementados no R que permitem a análise de dados de alto rendimento (GENTLEMAN *et al.*, 2004). Nele podemos encontrar algoritmos que realizam o controle de qualidade, pré-processamento, normalização e anotação dos dados de microarranjos de maneira simples, como o pacote *affy* (GAUTIER *et al.*, 2004) e, como o pacote *oligo* (CARVALHO; IRIZARRY, 2010), que pode ser aplicado em uma única etapa; ambos são utilizados para análise de plataformas da Affymetrix, por exemplo. Nos dados brutos de RNA-seq, o controle de qualidade, pré-processamento, normalização e anotação dos dados podem ser analisados com os pacotes *DESeq2* (LOVE *et al.*, 2014) ou *edgeR* (MCCARTHY *et al.*, 2012), e ainda os dados de RNA-seq provenientes da base de dados do TCGA podem ser analisados utilizando o pacote *TCGAbiolinks* (COLAPRICO *et al.*, 2015).

Outros pacotes de funções incluem: o *limma* (*linear models for microarray data*) (RITCHIE *et al.*, 2015), o qual é muito utilizado para a identificação de genes diferencialmente expressos, tanto em dados de microarranjo quanto em dados de RNA-seq. O desenvolvimento de projetos como o *Gene Ontology* (GO) (GENE ONTOLOGY CONSORTIUM, 2018) e o projeto *Kyoto Encyclopedia of Genes and Genomes* (KEGG) (KANEHISA *et al.*, 2000) originaram a criação do pacote *clusterProfiler* (YU *et al.*, 2012). O GO descreve produtos gênicos em vocabulários estruturados, denominados ontologias, em três níveis: (1) processos biológicos; (2) componentes celulares; e (3) funções moleculares para várias espécies. O KEGG é uma base de dados que disponibiliza informações curadas sobre vias de sinalização e interação molecular quanto ao: metabolismo celular, processamento de informação gênica e ambiental e sistemas organizacionais. Os pacotes *clusterProfiler* e o *topGO* (YU *et al.*, 2012; ALEXA; RAHNENFUHRER, 2010) são utilizados para realizar análises de enriquecimento funcional e interação molecular, apontando processos biológicos e vias de sinalização ativadas, por exemplo.

Devido à complexidade dos processos envolvidos na análise de dados de expressão gênica, foram desenvolvidas ferramentas *web user friendly* as quais proporcionam ao usuário maior facilidade na busca de biomarcadores e análise de dados. Uma delas é o GEO2R (BARRETT *et al.*, 2012), alimentada pelo banco de dados GEO; permite realizar a comparação de grupos amostrais para identificar genes diferencialmente expressos em condições experimentais. O cBioPortal (CERAMI *et al.*, 2012; GAO *et al.*, 2013) é um portal *web*, alimentado pela base de dados de tumores humanos TCGA, que permite a análise de dados moleculares multidimensionais,

incluindo dados transcriptômicos, nos quais os pesquisadores podem explorar padrões de alterações moleculares em múltiplos estudos. O *Expression Atlas* (PETRYSZAK *et al.*, 2015) é um banco de dados que fornece informações referentes à expressão gênica e de proteínas em diferentes espécies e contextos. O usuário pode consultar genes de interesse e explorar sua expressão; esse processo pode ser realizado através ou dentro de espécies, tecidos, estágios de desenvolvimento, em um contexto constitutivo ou diferencial, representando efeitos de doenças, condições ou intervenções experimentais. Outra ferramenta recentemente desenvolvida é o C-Gemis (www.cgemis.com), que é utilizado para a realização de análises de expressão gênica de tumores gástricos em humanos de maneira detalhada, utilizando subclassificações histológicas.

3 Trabalhos relacionados

Diversos biomarcadores relacionados a mRNA têm sido descritos em estudos ômicos recentes e, como resultado, há um incremento de dados biológicos em repositórios públicos que são disponibilizados para a comunidade científica, a fim de que se possa realizar análises que visam a contribuir para a identificação de soluções de problema biológicos.

Um estudo publicado na revista *BMC Plant Biology* utilizou a técnica de RNA-seq (Illumina Hiseq), para caracterizar o transcriptoma de *Amaranthus tuberculatus* em resposta à resistência desenvolvida a herbicidas inibidores da 4-hidroxifenilpiruvato dioxigenase (HPPD). Essa espécie invasora pode ocasionar perdas no rendimento agrícola de até 74% em milho e 56% em soja. Esse estudo transcriptômico identificou genes diferencialmente expressos em amarantos resistentes e suscetíveis a herbicidas inibidores de HPPD, fornecendo dados que poderão permitir o desenvolvimento de novas abordagens de controle (KOHLHASE *et al.*, 2019).

Em estudo envolvendo a esquistossomíase, uma das doenças parasitárias mais prevalentes nos países em desenvolvimento, pesquisadores utilizaram a plataforma de RNA-seq Illumina Iix para caracterizar o perfil transcricional de quatro momentos do ciclo de vida do parasita *Schistosoma mansoni*. Ao longo dos estágios do ciclo de vida estudados, foram identificadas alterações em 9.535 genes, sendo que alguns processos celulares centrais foram consistentemente superexpressos, incluindo a superexpressão de genes que codificam enzimas glicolíticas e tradução de proteínas. As cercárias de vida livre utilizam estoques internos de glicogênio e, conseqüentemente, genes envolvidos na glicólise e no ciclo do ácido tricarboxílico (TCA) são altamente expressos. Depois de penetrar na pele e transformar-se em endoparasitas obrigatórios, os esquistossômulos alternam para o metabolismo anaeróbico, antes que o metabolismo

aeróbico seja parcialmente retomado no adulto. No esquistossômulo, há uma mudança para a alta expressão de L-lactato desidrogenase, enquanto que a transcrição do ciclo do TCA diminui acentuadamente. Os dados resultantes dessa análise foram depositados nos bancos de dados GeneDB (www.genedb.org) e SchistoDB (www.schistodb.net). Os autores também destacam como vantagem dessa técnica a capacidade de melhorar a anotação gênica, quantificando com precisão as mudanças na expressão gênica (PROTASIO *et al.*, 2012).

A heterogeneidade celular e molecular dos tumores de mama e o grande número de genes potencialmente envolvidos no controle do crescimento, morte e diferenciação celular reforçam a importância de se estudar, em conjunto, as múltiplas alterações transcriptômicas. A investigação sistemática do padrão de expressão de milhares de genes em tumores, usando microarranjos e sua correlação com características específicas pode fornecer a base para uma melhor classificação dos subtipos de câncer de mama. Em estudo prévio, pesquisadores analisaram 40 amostras de tumores de mama através do uso de microarranjos e identificaram quatro subgrupos que apresentavam variações no padrão de expressão gênica, relacionado a diferentes características moleculares do epitélio mamário. O mesmo grupo de pesquisadores ampliou o tamanho amostral e realizou novas análises de microarranjo, a fim de refinar as classificações anteriores, explorando o valor clínico dos subtipos, buscando correlações entre padrões de expressão gênica e parâmetros clinicamente relevantes. Além da classificação em cinco subgrupos, foi observado que a classificação de tumores, com base nos padrões de expressão gênica, pode ser usada como um marcador prognóstico em relação à sobrevida global e livre de doença. Os subgrupos são classificados em: (1) Luminal A: que apresentam receptor de estrogênio positivo, e/ou receptor de progesterona positivo, e amplificação e/ou superexpressão de HER2 negativa (ER+ / PR+ / HER2-), geralmente são classificados como tumores de baixo grau; (2) Luminal B: (ER + / PR -/+ / HER2 +/-), geralmente são classificados como tumores de alto grau com taxas de proliferação elevadas; (3) subtipo normal que se assemelha ao tecido mamário normal e está associado a um bom prognóstico; (4) câncer de mama triplo-negativo (ER- / PR- / HER2-); e (5) subtipo enriquecido com HER2 (ER- / PR- / HER2+). (SORLIE *et al.*, 2001; PEROU *et al.*, 2000). Estudos envolvendo análises e classificações baseadas em mRNA, aliadas às informações clínicas, podem conferir importantes fontes de informações para o correto diagnóstico, prognóstico ou a predição à resposta de tratamentos.

O *splicing* alternativo é um mecanismo essencial para gerar diversidade funcional, pois permite que genes individuais expressem múltiplos mRNA e codifiquem várias proteínas, através do rearranjo de domínios existentes. Estima-se que cerca de 95% dos

genes de mamíferos sofram *splicing* alternativo, com forte impacto em processos regulatórios essenciais, como a modificação da cromatina e a transdução de sinal celular. Por permitir aumentar a diversidade de proteínas a um custo mínimo para o organismo, o *splicing* alternativo sofreu uma rápida evolução nos vertebrados e acredita-se que tenha um papel fundamental na adaptação dos primatas, incluindo os humanos. Uma função em que a inovação é essencial é a imunidade, pois a corrida constante contra patógenos invasores requer a habilidade do hospedeiro de se adaptar rapidamente a novos mecanismos patogênicos, enquanto mantém a homeostase. Assim, para entender a regulação da função imune em humanos, faz-se necessário estudar o grau de variação em nível populacional do *splicing* alternativo. Em estudo recente publicado na *Nature Communications*, os pesquisadores utilizaram dados de RNA-seq de monócitos primários humanos tanto no estado basal quanto após estimulação com diferentes ligantes, em 200 indivíduos saudáveis de ascendência africana e europeia. Foi possível caracterizar o cenário do *splicing* alternativo quanto à resposta imune inata e ainda, explorar a evolução recente e a longo prazo desse mecanismo. Foi demonstrado que a ativação imunológica provoca uma remodelação no repertório de isoformas, enquanto aumenta os níveis de *splicing* errôneo. Além disso, foi observada uma plasticidade aumentada (e de longa data) no *splicing* de genes envolvidos com a resposta imune, apontando que a seleção positiva e a introgressão Neanderthal contribuíram para diversificar o *splicing* alternativo nas populações humanas. Esses achados sugerem que o uso diferencial de isoformas tem sido importante para a inovação das respostas imunes a longo prazo e, na evolução recente, para a adaptação local da população (ROTIVAL *et al.*, 2019).

Referências

- ALEXA, Adrian; RAHNENFUHRER, Jorg. topGO: enrichment analysis for gene ontology. **R package version**, v. 2, n. 0, p. 2010, 2010.
- BARRETT, Tanya *et al.* NCBI GEO: archive for functional genomics data sets – 10 years on. **Nucleic acids research**, v. 39, n. suppl_1, p. D1005-D1010, 2010.
- BARRETT, Tanya *et al.* NCBI GEO: archive for functional genomics data sets – update. **Nucleic acids research**, v. 41, n. D1, p. D991-D995, 2012.
- BROOKS, James D. Translational genomics: the challenge of developing cancer biomarkers. **Genome research**, v. 22, n. 2, p. 183-187, 2012.
- CARVALHO, Benilton S.; IRIZARRY, Rafael A. A framework for oligonucleotide microarray preprocessing. **Bioinformatics**, v. 26, n. 19, p. 2363-2367, 2010.
- CERAMI, Ethan *et al.* **The cBio cancer genomics portal**: an open platform for exploring multidimensional cancer genomics data. 2012.
- COLAPRICO, Antonio *et al.* TCGAbiolinks: an R/Bioconductor package for integrative analysis of TCGA data. **Nucleic acids research**, v. 44, n. 8, p. e71-e71, 2015.

- DOBNIK, David *et al.* Multiplex quantification of four DNA targets in one reaction with Bio-Rad droplet digital PCR system for GMO detection. **Scientific reports**, v. 6, p. 3.5451, 2016.
- GAO, Jianjiong *et al.* Integrative analysis of complex cancer genomics and clinical profiles using the cBioPortal. **Sci. Signal.**, v. 6, n. 269, p. 11, 2013.
- GAUTIER, Laurent *et al.* affy – analysis of affymetrix genechip data at the probe level. **Bioinformatics**, v. 20, n. 3, p. 307-315, 2004.
- GENE ONTOLOGY CONSORTIUM. The Gene Ontology resource: 20 years and still GOing strong. **Nucleic acids research**, v. 47, n. D1, p. D330-D338, 2018.
- GENTLEMAN, Robert C. *et al.* Bioconductor: open software development for computational biology and bioinformatics. **Genome biology**, v. 5, n. 10, p. R80, 2004.
- GLISOVIC, Tina *et al.* RNA- binding proteins and post- transcriptional gene regulation. **FEBS letters**, v. 582, n. 14, p. 1977-1986, 2008.
- HRDLICKOVA, Radmila; TOLOUE, Masoud; TIAN, Bin. RNA- Seq methods for transcriptome analysis. **Wiley Interdisciplinary Reviews: RNA**, v. 8, n. 1, p. e1364, 2017.
- IRIZARRY, Rafael A. *et al.* Exploration, normalization, and summaries of high density oligonucleotide array probe level data. **Biostatistics**, v. 4, n. 2, p. 249-264, 2003.
- KANEHISA, Minoru; GOTO, Susumu. KEGG: kyoto encyclopedia of genes and genomes. **Nucleic acids research**, v. 28, n. 1, p. 27-30, 2000.
- KOHLHASE, Daniel R. *et al.* Using RNA-seq to characterize responses to 4-hydroxyphenylpyruvate dioxygenase (HPPD) inhibitor herbicide resistance in waterhemp (*Amaranthus tuberculatus*). **BMC plant biology**, v. 19, n. 1, p. 182, 2019.
- KOLESNIKOV, Nikolay *et al.* ArrayExpress update – simplifying data submissions. **Nucleic acids research**, v. 43, n. D1, p. D1113-D1116, 2014.
- LIEBERMAN, Michael; MARKS, Allan D. **Marks' basic medical biochemistry: a clinical approach.** Lippincott Williams & Wilkins, 2009.
- LOVE, Michael I.; HUBER, Wolfgang; ANDERS, Simon. Moderated estimation of fold change and dispersion for RNA-seq data with DESeq2. **Genome biology**, v. 15, n. 12, p. 550, 2014.
- MCCARTHY, Davis J.; CHEN, Yunshun; SMYTH, Gordon K. Differential expression analysis of multifactor RNA-Seq experiments with respect to biological variation. **Nucleic acids research**, v. 40, n. 10, p. 4288-4297, 2012.
- NALEJSKA, Ewelina; MAĆZYŃSKA, Ewa; LEWANDOWSKA, Marzena Anna. Prognostic and predictive biomarkers: tools in personalized oncology. **Molecular diagnosis & therapy**, v. 18, n. 3, p. 273-284, 2014.
- PAPATHEODOROU, Irene *et al.* Expression Atlas: gene and protein expression across multiple studies and organisms. **Nucleic acids research**, v. 46, n. D1, p. D246-D251, 2017.
- PEROU, Charles M. *et al.* Molecular portraits of human breast tumours. **Nature**, v. 406, n. 6797, p. 747, 2000.
- PETRYSZAK, Robert *et al.* Expression Atlas update – an integrated database of gene and protein expression in humans, animals and plants. **Nucleic acids research**, v. 44, n. D1, p. D746-D752, 2015.
- PROTASIO, Anna V. *et al.* A systematically improved high quality genome and transcriptome of the human blood fluke *Schistosoma mansoni*. **PLoS neglected tropical diseases**, v. 6, n. 1, p. e1455, 2012.
- RITCHIE, Matthew E. *et al.* limma powers differential expression analyses for RNA-sequencing and microarray studies. **Nucleic acids research**, v. 43, n. 7, p. e47-e47, 2015.
- ROTIVAL, Maxime; QUACH, Hélène; QUINTANA-MURCI, Lluís. Defining the genetic and evolutionary architecture of alternative splicing in response to infection. **Nature communications**, v. 10, n. 1, p. 1671, 2019.

SORLIE, Therese *et al.* Gene expression patterns of breast carcinomas distinguish tumor subclasses with clinical implications. **Proceedings of the National Academy of Sciences**, v. 98, n. 19, p. 10.869-10.874, 2001.

TEAM, R. Core *et al.* R: A language and environment for statistical computing. 2013.

TOMCZAK, Katarzyna; CZERWIŃSKA, Patrycja; WIZNEROWICZ, Maciej. The Cancer Genome Atlas (TCGA): an immeasurable source of knowledge. **Contemporary oncology**, v. 19, n. 1A, p. A68, 2015.

YU, Guangchuang *et al.* clusterProfiler: an R package for comparing biological themes among gene clusters. **Omics: a journal of integrative biology**, v. 16, n. 5, p. 284-287, 2012.

ZAHA, Arnaldo; FERREIRA, Henrique Bunselmeyer; PASSAGLIA, Luciane MP. **Biologia Molecular Básica-5**. Porto Alegre: Artmed Editora, 2014.

Inúmeras informações biológicas podem ser obtidas através da análise de imagens. À medida que esse conjunto de imagens cresce e, potencialmente, contém um número maior de informações, há uma crescente necessidade de se substituir a inspeção visual e manual pelo processamento computacional de imagens (MEIJERING; CAPPELLEN, 2007). Em sua forma mais simples, a análise computadorizada de imagens supera as limitações e o viés de um observador humano, podendo atingir um nível de sensibilidade, acurácia e objetividade superiores ao alcançado através de uma análise manual (SHARIFF *et al.*, 2010).

A visão computacional é um conceito que busca reproduzir, por meio de um modelo computacional, as funções relacionadas ao sistema visual humano. Para isso, ferramentas de *hardware* e *software* são utilizadas em conjunto para criar um sistema, com o objetivo de analisar e compreender as informações relevantes contidas em imagens (BACKES; JUNIOR, 2019).

De modo geral, o processo de análise de imagens é dividido em quatro etapas: pré-processamento, segmentação, extração de características e classificação. Neste contexto, o presente capítulo apresenta uma breve fundamentação sobre essas etapas, destacando os métodos mais conhecidos e as ferramentas computacionais que implementam esses métodos. Para um estudo mais completo, sugere-se a leitura de Nixon e Aguado (2012) e Sonka, Hlavac; Boyle (2014).

1 Pré-processamento

A etapa de pré-processamento ou filtragem consiste na aplicação de técnicas de transformação na imagem, com o objetivo de corrigir, suavizar ou realçar características que podem ser relevantes para as etapas posteriores. Por exemplo, a aplicação de um filtro pode realçar os contornos de uma imagem, facilitando a etapa de segmentação.

De modo geral, a escolha do filtro de pré-processamento depende do objetivo desejado. Caso o propósito seja a redução de ruídos, filtros do tipo passa-baixa mostram-se mais adequados. Estes filtros são assim denominados, pois preservam os

¹ Universidade de Caxias do Sul. E-mail: ipassos@ucs.br / iago.dpassos@gmail.com

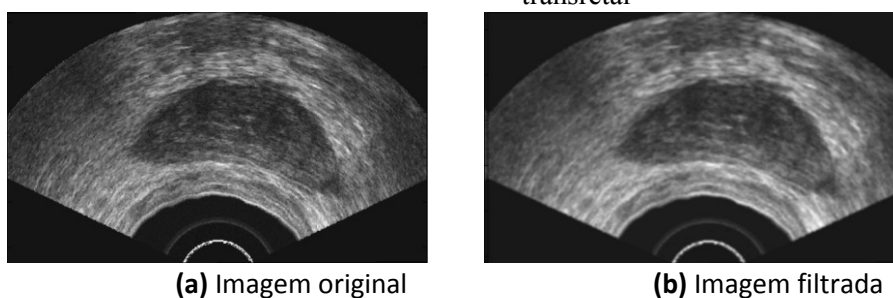
² Universidade de Caxias do Sul. E-mail: lpdutra@ucs.br / ld.lucasdutra@gmail.com

³ Universidade de Caxias do Sul. E-mail: almartin@ucs.br

sinais de baixa frequência, atenuando os sinais de altas frequências.⁴ Os valores de frequência atenuados pelos filtros passa-baixa variam de acordo com o *design* específico do filtro, sendo que os filtros mais utilizados são os filtros Ideal, Gaussiano e *Butterworth* (MAKANDAR; HALALLI, 2015).

No trabalho de Chouie, Fieguth e Rahnamayan (2006), é realizado o pré-processamento de imagens de ultrassom transretal, com o objetivo de auxiliar na localização da próstata. Na Figura 1 tem-se uma imagem que ilustra a aplicação de um filtro Gaussiano passa-baixa para a redução dos ruídos.

Figura 1 –Exemplo de aplicação de filtro Gaussiano passa-baixa em imagens de ultrassom transretal



Fonte: Kachouie, Fieguth e Rahnamayan (2006).

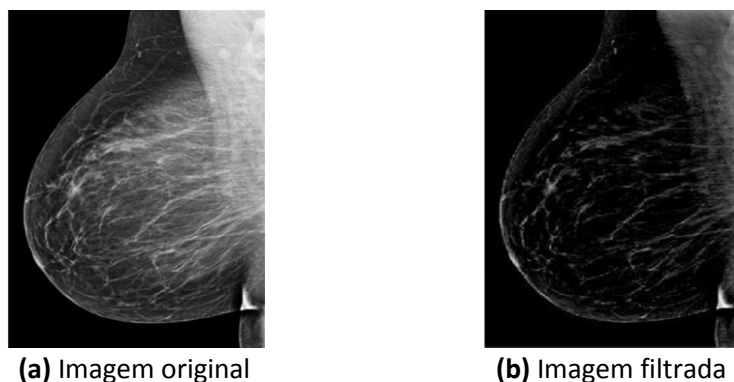
Os filtros do tipo passa-alta mostram-se mais adequados para a identificação de mudanças abruptas de coloração, como bordas, linhas ou manchas. Ao contrário dos filtros passa-baixa, os filtros passa-alta preservam os sinais de alta frequência, atenuando os sinais de baixas frequências (CRÓSTA, 1999). Os valores de frequência atenuados também dependem do *design* do filtro utilizado, sendo que os filtros passa-alta mais utilizados são os filtros Ideal, Gaussiano e *Butterworth*⁵ (MAKANDAR; HALALLI, 2015).

No trabalho de Pillai e Kwartowitz (2014), um filtro *Butterworth* passa-alta é utilizado para realçar os contornos em imagens de mamografias. O realce dos contornos facilita os processos posteriores de segmentação e de identificação de nódulos mamários. Na Figura 2, tem-se um exemplo da aplicação do filtro *Butterworth* passa-alta para o realce dos contornos na imagem.

⁴ A frequência em uma imagem denota a taxa de variação da intensidade entre os *pixels*. Desta forma, mudanças significativas entre *pixels* vizinhos caracterizam áreas de alta frequência. Áreas homogêneas, isto é, onde *pixels* próximos apresentam pouca variação, caracterizam áreas de baixa frequência.

⁵ Apesar dos filtros passa-alta e passa-baixa citados possuírem a mesma nomenclatura, suas implementações são diferentes para cada modalidade de filtragem. Maiores detalhes sobre esses filtros podem ser encontradas em Makandar e Halalli (2015) e Dogra e Bhalla (2014).

Figura 2 – Exemplo de aplicação de filtro *Butterworth* passa-alta em imagens de mamografias



Fonte: adaptado de Pillai e Kwartowitz (2014).

Diferentes ferramentas e bibliotecas implementam os principais filtros existentes. Dentre essas ferramentas, destacam-se o ImageJ (ABRÀMOFF; MAGALHÃES; RAM, 2004) e o BioImageXD (KANKAANPÄÄ *et al.*, 2012). Com relação às bibliotecas de programação, destacam-se o OpenCV (BRADSKI; KAEHLER, 2008) e o *scikit-image* (WALT *et al.*, 2014). Todas as ferramentas e as bibliotecas citadas possuem suporte para a utilização dos filtros Ideal, Gaussiano e *Butterworth*, nas modalidades passa-alta e passa-baixa.

2 Segmentação

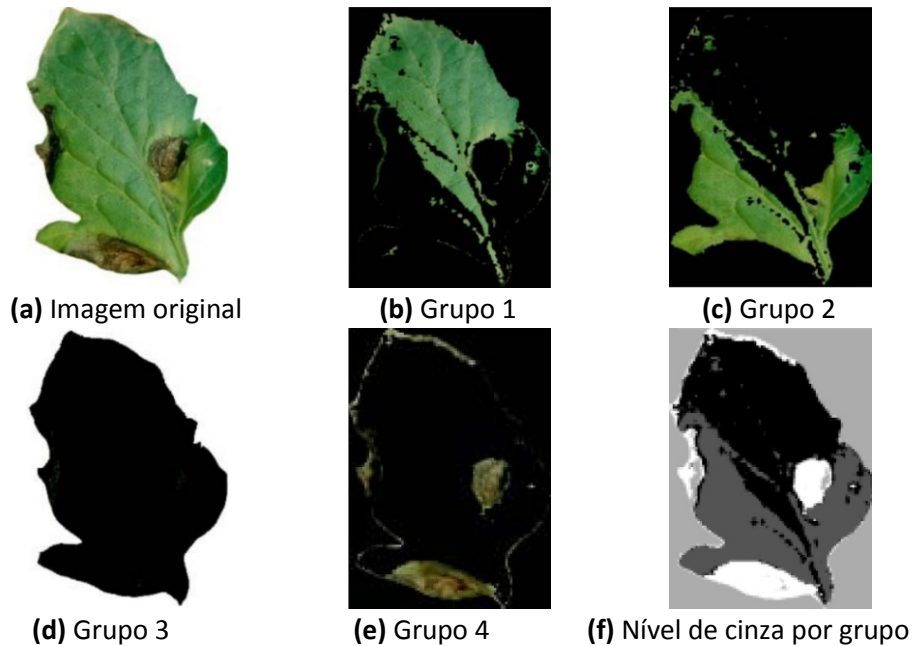
A etapa de segmentação consiste em dividir a imagem em regiões, permitindo que seja considerada na análise somente a parte da imagem na qual se tem interesse (ALBUQUERQUE; ALBUQUERQUE, 2000). Para delimitar as regiões da imagem, é necessário que os *pixels* sejam associados a diferentes regiões. Esta associação pode ser efetuada baseando-se em duas propriedades: similaridade e descontinuidade (GONZALEZ; WOODS, 2011).

Na segmentação por similaridade, os *pixels* são associados a uma região a partir do grau de similaridade entre eles, isto é, a segmentação busca agrupar *pixels* com características semelhantes (SOLOMON; BRECKON, 2000). As técnicas de segmentação por similaridade mais utilizadas são: *K-means Clustering* (MACQUEEN *et al.*, 1967) e Método de Otsu (OTSU, 1979).

A técnica *K-means Clustering* tem como objetivo dividir os *pixels* da imagem em grupos (*clusters*), na qual os *pixels* de cada *cluster* possuem maior similaridade entre si e menor similaridade com os *pixels* dos outros grupos (TAKAHASHI; BEDREGAL; LYRA, 2005). Na Figura 3, tem-se um exemplo da aplicação desta técnica para a

segmentação de regiões defeituosas em imagens de folhas. A imagem original é segmentada em quatro grupos para, no final do processo, compor uma única imagem incorporando todos os grupos. Para destacar os diferentes grupos na imagem, são utilizados diferentes tons de cinza (Figura 3f).

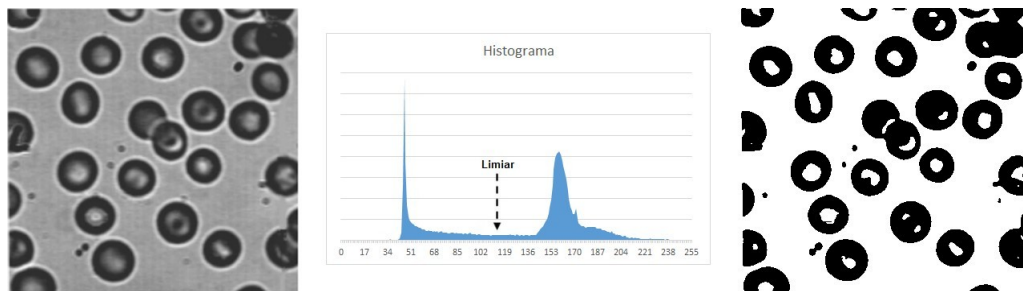
Figura 3 – Exemplo de segmentação por similaridade utilizando a técnica *K-means Clustering*



Fonte: Bashish; Braik; Bani-Ahmad (2011).

O Método de Otsu é um algoritmo de limiarização que tem como objetivo determinar os valores mais adequados (limiares), que separem os elementos de uma imagem em tons de cinza. Por exemplo, o Método de Otsu pode ser utilizado para a separação de objetos do fundo da imagem. Neste contexto, na Figura 4, tem-se um exemplo da aplicação deste método para a separação de células sanguíneas do fundo da imagem.

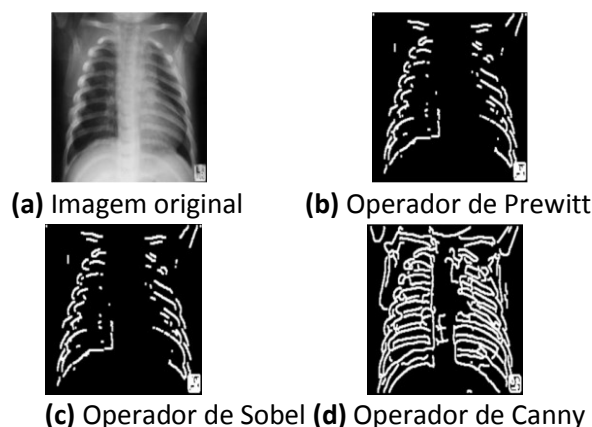
Figura 4 – Método de Otsu



Fonte: Elaboração dos Autores.

Na segmentação por descontinuidade, a separação é feita através de métodos de contorno/fronteira. Esses métodos apresentam como objetivo a identificação de descontinuidades em uma imagem (SOLOMON; BRECKON, 2000). As descontinuidades encontradas em uma imagem podem ser pontos, linhas ou limites (bordas) de um objeto. Essas descontinuidades destacam-se por possuírem tons distintos da região na qual estão inseridas (caso de pontos e linhas) ou por apresentarem mudanças bruscas de tons entre regiões (caso de bordas e linhas) (SALDANHA; FREITAS, 2009). As técnicas de segmentação por descontinuidade mais utilizadas são os operadores de Prewitt (PREWITT, 1970), Sobel (MONSON; WIRTHLIN; HUTCHINGS, 2013) e Canny (CANNY, 1986). Os operadores de Sobel e Prewitt baseiam-se no gradiente de intensidade⁶ dos *pixels* da imagem para a detecção das bordas (ZUNIGA; HARALICK, 1987). Já o operador de Canny é considerado um algoritmo multipassos que combina um filtro Gaussiano passa-baixa para suavização de ruídos e um operador de gradiente de intensidade (Sobel ou Prewitt) para a detecção das bordas (GREEN, 2002). A Figura 5 demonstra a aplicação dos operadores de Prewitt, Sobel e Canny, para a detecção de bordas em imagens de radiografias torácicas.

Figura 5 – Exemplo de aplicação dos operadores de Prewitt, Sobel e Canny



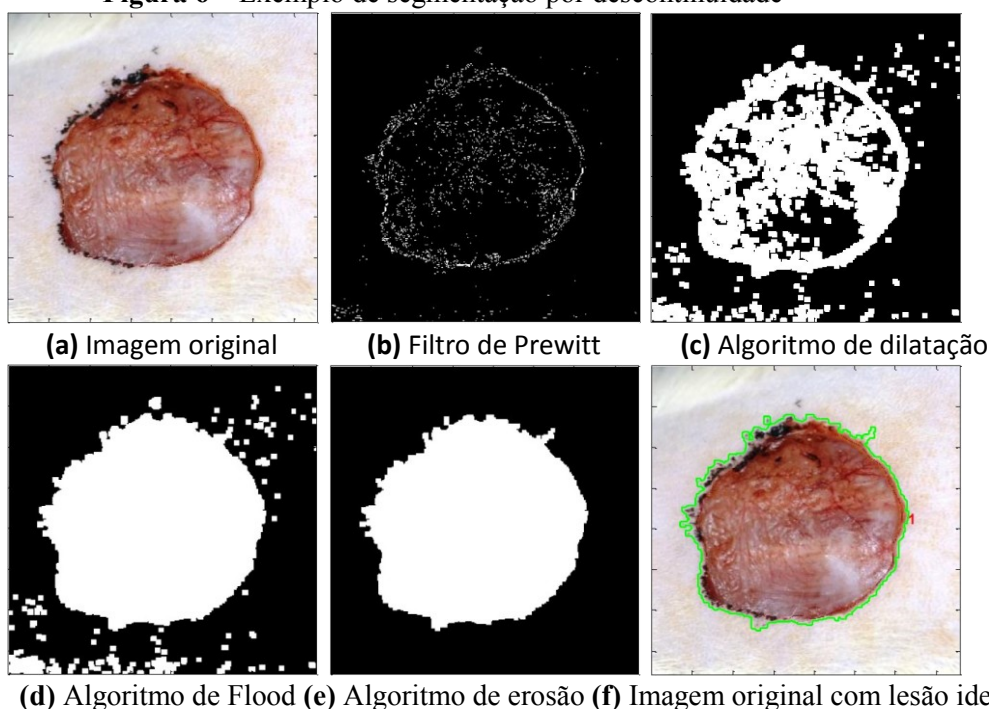
Fonte: Elaboração dos autores.

Na utilização de técnicas de segmentação por descontinuidade, é comum a aplicação conjunta de filtros passa-alta e algoritmos para manipulação de imagens. No trabalho de Albarello (2014), imagens de lesões em ratos são segmentadas com o objetivo de detectar o tamanho da lesão para um acompanhamento da cicatrização. A metodologia consiste, inicialmente, na aplicação de um filtro de Prewitt (PREWITT,

⁶ O gradiente de intensidade representa a direção e a taxa de mudança de colorações claras para colorações escuras.

1970) para realçar contornos (Figura 6b). Em seguida, aplica-se um algoritmo de dilatação (CHEN; HARALICK, 1995) para estender a continuidade dos contornos (Figura 6c). Após, o algoritmo de *Flood Fill* (LAW, 2013) é utilizado para o preenchimento de lacunas (Figura 6d) e, posteriormente, é empregado um algoritmo de erosão (CHEN; HARALICK, 1995) a fim de remover as imperfeições restantes na imagem (Figura 6e). Por fim, a partir do contorno obtido, constrói-se a imagem original com a lesão identificada (Figura 6f).

Figura 6 – Exemplo de segmentação por descontinuidade



Fonte: Albarello (2014).

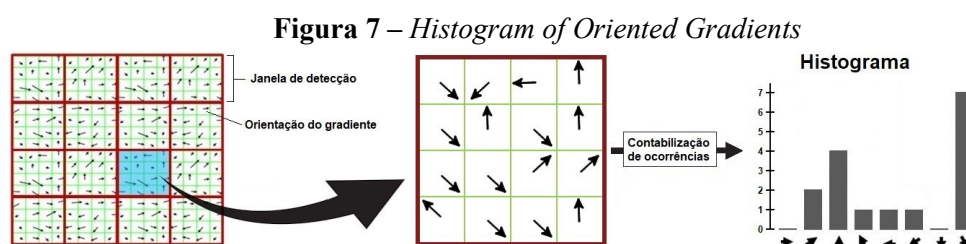
Os principais algoritmos de segmentação por similaridade e por descontinuidade encontram-se implementados em diversas ferramentas e bibliotecas. Dentre as ferramentas disponíveis, destacam-se o CellProfiler (LAMPRECHT; SABATINI; CARPENTER, 2007), Icy (CHAUMONT *et al.*, 2012) e o Imaris (BITPLANE, 1992), sendo que o Método de Ostu e os operadores de Prewitt, Sobel e Canny encontram-se implementados em todas essas ferramentas através do uso de *plugins*. Já o método *K-means Clustering* encontra-se disponível apenas nas ferramentas Icy e Imaris. Com relação às bibliotecas de programação, destacam-se o OpenCV (BRADSKI; KAEHLER, 2008) e o *scikit-image* (WALT *et al.*, 2014), que possuem suporte para todos os métodos de segmentação mencionados.

3 Extração de características

A etapa de extração de características consiste em representar as informações relevantes da imagem em um novo formato, que facilite as etapas posteriores de análise e de classificação das imagens. Esse novo formato é frequentemente obtido através da transformação das informações da imagem em um espaço de menor dimensão (vetor de características), sendo que esse novo formato procura preservar as características originais da imagem (KUMAR; BHATIA, 2014).

Diversos métodos para extração de características são apresentados na literatura, sendo denominados de descritores de imagem. Esses métodos procuram gerar um vetor de características que representa as informações elementares da imagem, como forma, cor ou textura (MANJUNATH; SALEMBIER; SIKORA, 2002). Dentre os descritores existentes, os mais utilizados são o LBP (*Local Binary Pattern*) e o HOG (*Histogram of Oriented Gradients*) (LIN *et al.*, 2011). No trabalho de (YANG *et al.*, 2015), por exemplo, o método LBP é empregado para extração de características em imagens de ressonância magnética de tumores cerebrais. Já no trabalho de Sugiarto *et al.* (2017), o método HOG é aplicado para a extração de características de imagens de madeira.

O método do Histograma de Gradientes Orientados (do inglês, *Histogram of Oriented Gradients*) é um descritor de imagem que tem como principal objetivo a detecção de objetos. De modo geral, o método HOG constrói um histograma contabilizando as ocorrências de orientação dos gradientes em porções localizadas da imagem. Essas porções são chamadas de janelas de detecção ou regiões de interesse (SILVA, 2017). Na Figura 7, tem-se um histograma com a ocorrência de cada direção de gradiente na região em destaque.

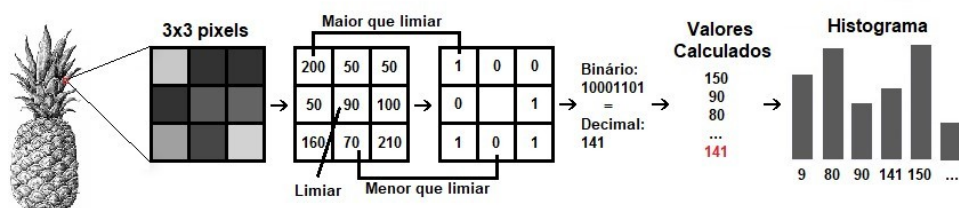


Fonte: Elaboração dos autores.

O método *Local Binary Pattern* é um descritor de textura em tons de cinza, que também é utilizado para a detecção de objetos. O principal objetivo do LBP é rotular os *pixels* da imagem através de uma análise dos *pixels* da sua vizinhança. Para cada *pixel*, a vizinhança é binarizada utilizando-se como limiar o valor do *pixel* central, ou seja, os *pixels* vizinhos com valores acima do valor do *pixel* central recebem o valor 1, caso

contrário atribui-se o valor 0 a estes. Após a binarização, tem-se, para cada *pixel*, uma matriz contendo valores 0 e 1. A união destes valores representa um número em base binária,⁷ sendo que este número deve ser convertido para a base decimal. Por fim, os valores decimais calculados para todos os *pixels* são contabilizados em um histograma (SILVA, 2017) (Figura 8).

Figura 8 – Local Binary Pattern



Fonte: Elaboração dos autores.

O uso de descritores de imagem é realizado principalmente através de bibliotecas de programação. Dentre as bibliotecas existentes, destacam-se o OpenCV (BRADSKI; KAEHLER, 2008) e o *scikit-image* (WALT *et al.*, 2014), que implementam os métodos HOG e LBP.

4 Classificação

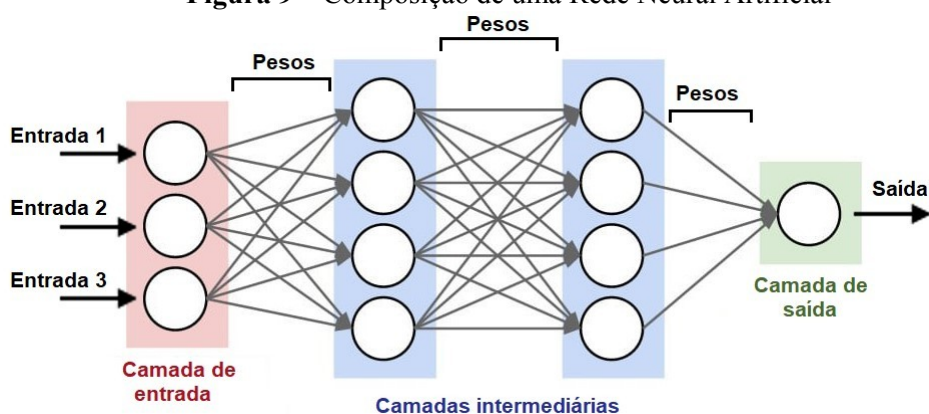
A classificação é o processo utilizado para a separação de um conjunto de imagens em classes distintas. Isto é, a etapa de classificação tem como objetivo atribuir classes às imagens analisadas de acordo com seus elementos (CHEN, 2015). Por exemplo, dado um conjunto de imagens de animais, pode ser realizado um processo de classificação objetivando separá-las de acordo com a espécie do animal. Os métodos mais empregados na etapa de classificação são os algoritmos de aprendizagem de máquina supervisionada.

Os algoritmos de aprendizagem de máquina supervisionada buscam construir um modelo de distribuição de classes baseado em conhecimentos adquiridos em uma fase de treinamento. A fase de treinamento tem como objetivo a inclusão de informações em uma base de conhecimento, a qual, no contexto de análise de imagens, é constituída a partir de imagens previamente classificadas. Essa base será posteriormente utilizada na tomada de decisão para a classificação de novas imagens (MITCHELL, 1997). Dentre os algoritmos de aprendizagem supervisionada, os mais utilizados são as Redes Neurais Artificiais (RNAs) e as Máquinas de Vetores de Suporte (SVMs).

⁷ A base binária é um sistema de numeração posicional em que os valores são representados pelos algarismos 0 e 1.

As Redes Neurais Artificiais (RNAs) são técnicas computacionais inspiradas no funcionamento do cérebro humano (HAYKIN, 1998). Uma RNA é composta por um conjunto de neurônios conectados entre si, sendo atribuído um peso a cada conexão entre esses neurônios. Um único neurônio é capaz de realizar apenas tarefas simples de classificação. O poder das RNAs se dá através das conexão de vários neurônios entre si, normalmente dispostos em camadas. Cada camada de neurônio se comunica com a próxima camada através de seus canais de entrada e saída, sendo que a saída de um neurônio compõe a entrada de outro (Figura 9). Durante a fase de treinamento, os pesos entre as conexões são ajustados, a fim de armazenar o conhecimento da rede. Esse conhecimento resulta no comportamento inteligente para o reconhecimento de padrões e características (CERA, 2005).

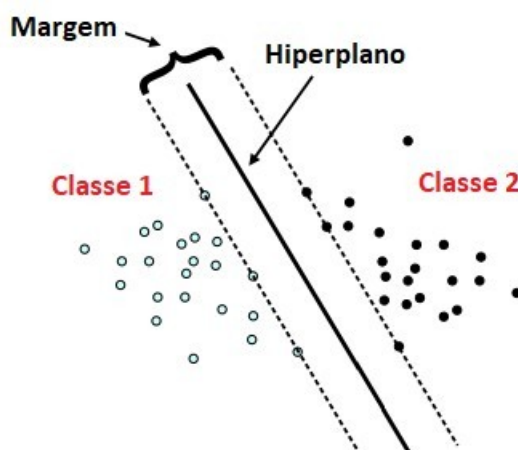
Figura 9 – Composição de uma Rede Neural Artificial



Fonte: Elaboração dos autores.

As Máquinas de Vetores de Suporte (SVMs, do inglês *Support Vector Machines*) são uma técnica computacional que busca construir uma regra para separação das classes analisadas. Essa regra é gerada durante a fase de treinamento, através da definição de um hiperplano para a separação das amostras já classificadas. Esse hiperplano é definido obtendo-se a maior margem entre os pontos mais próximos das duas classes distintas (LORENA; CARVALHO, 2007). Desta forma, a classificação de novas amostras ocorre com a verificação da posição na qual esta amostra se encontra em relação ao hiperplano. Por exemplo, na Figura 10 tem-se as amostras da Classe 1, em um lado do hiperplano e as amostras da Classe 2, no lado oposto.

Figura 10 – Separação das classes pela técnica das SVMs através de um hiperplano



Fonte: Elaboração dos autores.

Como exemplo da aplicação de um método de aprendizagem supervisionada, pode-se citar o trabalho de Rocha *et al.* (2010), que empregam SVMs para a classificação automatizada de imagens contendo frutas e vegetais. De forma semelhante, o trabalho de Zhang *et al.* (2014) utiliza RNAs para a classificação de imagens contendo diferentes tipos de frutas.

Dentre as ferramentas que implementam as RNAs e as SVMs, destacam-se o Weka (FRANK *et al.*, 2004) e o *CellProfiler Analyst* (DAO *et al.*, 2016). Também são encontradas na literatura bibliotecas de programação que implementam estas técnicas. Dentre estas bibliotecas, a LibSVM (CHANG; LIN, 2011) é a mais utilizada para o emprego de SVMs e as bibliotecas Tensorflow (ABADI *et al.*, 2016) e Encog (HEATON, 2015), para a utilização de RNAs.

Referências

- ABADI, M. *et al.* Tensorflow: A system for large-scale machine learning. *In: USENIX SYMPOSIUM ON OPERATING SYSTEMS DESIGN, 12., IMPLEMENTATION (OSDI 16)*. [S.l.: s.n.], Anais, Sawannagh, Georgia (LA) EUA, 2016. p. 265-283.
- ABRÀMOFF, M. D.; MAGALHÃES, P. J.; RAM, S. J. Image processing with imagej. **Biophotonics international**, Laurin Publishing, v. 11, n. 7, p. 36-42, 2004.
- ALBARELLO, J. d. R. **Processamento de imagens digitais para modelagem e controle do tratamento de feridas cutâneas**. Ijuí: Unijuí, 2014.
- ALBUQUERQUE, M. P. de; ALBUQUERQUE, M. P. de. **Processamento de imagens: métodos e análises**. Rio de Janeiro: Centro Brasileiro de Pesquisas Físicas (CBPF), 2000. v. 12.
- BACKES, A. R.; JUNIOR, J. J. d. M. S. **Introdução à visão computacional usando Matlab**. [S.l.]: Alta Books Editora, 2019.
- BASHISH, D. A.; BRAIK, M.; BANI-AHMAD, S. Detection and classification of leaf diseases using k-means-based segmentation and. **Information Technology Journal**, v. 10, n. 2, p. 267-275, 2011.
- BITPLANE, A. Imaris. 1992. *In: BRADSKI, G.; KAEHLER, A. Learning OpenCV: computer vision with the OpenCV library*. [S.l.]: “O’Reilly Media, Inc.”, 2008.

- CANNY, J. A computational approach to edge detection. **IEEE Transactions on pattern analysis and machine intelligence**, Ieee, n. 6, p. 679-698, 1986.
- CERA, M. C. Uso de redes neurais para o reconhecimento de padrões. **UFRGS. Trabalho da disciplina Arquiteturas Especiais de Computadores**. Porto Alegre, 2005.
- CHANG, C.-C.; LIN, C.-J. LIBSVM: A library for support vector machines. **ACM Transactions on Intelligent Systems and Technology**, v. 2, p. 27, 2011.
- CHAUMONT, F. D. *et al.* Icy: an open bioimage informatics platform for extended reproducible research. **Nature methods**, Nature Publishing Group, v. 9, n. 7, p. 690, 2012.
- CHEN, C.-h. **Handbook of pattern recognition and computer vision**. [S.l.]: World Scientific, 2015.
- CHEN, S.; HARALICK, R. M. Recursive erosion, dilation, opening, and closing transforms. **IEEE Transactions on image processing**, IEEE, v. 4, n. 3, p. 335-345, 1995.
- CRÓSTA, A. P. **Processamento digital de imagens de sensoriamento remoto**. [S.l.]: Unicamp/Instituto de Geociências, 1999.
- DAO, D. *et al.* Cellprofiler analyst: interactive data exploration, analysis and classification of large biological image sets. **Bioinformatics**, Oxford University Press, v. 32, n. 20, p. 3210-3212, 2016.
- DOGRA, A.; BHALLA, P. Image sharpening by gaussian and butterworth high pass filter. **Biomedical and Pharmacology Journal**, v. 7, n. 2, p. 707-713, 2014.
- FRANK, E. *et al.* Data mining in bioinformatics using weka. **Bioinformatics**, Oxford University Press, v. 20, n. 15, p. 2479-2481, 2004.
- GONZALEZ, R.; WOODS, R. **Digital Image Processing**. [S.l.]: Pearson Education, 2011.
- GREEN, B. Canny edge detection tutorial. **Retrieved: March**, v. 6, p. 2005, 2002.
- HAYKIN, S. **Neural networks: a comprehensive foundation**. 2. ed. Upper Saddle River, NJ, USA: Prentice Hall PTR, 1998.
- HEATON, J. Encog: Library of interchangeable machine learning models for java and c#. **Journal of Machine Learning Research**, v. 16, p. 1243-1247, 2015.
- KACHOUIE, N. N.; FIEGUTH, P.; RAHNAMAYAN, S. An elliptical level set method for automatic trus prostate image segmentation. *In: IEEE. 2006 IEEE International Symposium on Signal Processing and Information Technology*. [S.l.], 2006. p. 191-196.
- KANKAANPÄÄ, P. *et al.* Bioimagexd: an open, general-purpose and high-throughput image-processing platform. **Nature methods**, Nature Publishing Group, v. 9, n. 7, p. 683, 2012.
- KUMAR, G.; BHATIA, P. K. A detailed review of feature extraction in image processing systems. *In: IEEE. Fourth international conference on advanced computing & communication technologies*. [S.l.], 2014. p. 5-12.
- LAMPRECHT, M. R.; SABATINI, D. M.; CARPENTER, A. E. CellprofilerTM: free, versatile software for automated biological image analysis. **Biotechniques**, Future Science, v. 42, n. 1, p. 71-75, 2007.
- LAW, G. Quantitative comparison of flood fill and modified flood fill algorithms. **International Journal of Computer Theory and Engineering**, IACSIT Press, v. 5, n. 3, p. 503-508, 2013.
- LIN, Y. *et al.* Large-scale image classification: fast feature extraction and svm training. *In: IEEE. CVPR 2011*. [S.l.], 2011. p. 1689-1696.
- LORENA, A. C.; CARVALHO, A. C. de. Uma introdução às support vector machines. **Revista de Informática Teórica e Aplicada**, v. 14, n. 2, p. 43-67, 2007.
- MACQUEEN, J. *et al.* Some methods for classification and analysis of multivariate observations. *In: OAKLAND, CA, USA. Proceedings of the fifth Berkeley symposium on mathematical statistics and probability*. [S.l.], v. 1, n. 14, p. 281-297, 1967.
- MAKANDAR, A.; HALALLI, B. Image enhancement techniques using highpass and lowpass filters. **International Journal of Computer Applications**, Citeseer, v. 109, n. 14, p. 12-15, 2015.

- MANJUNATH, B. S.; SALEMBIER, P.; SIKORA, T. **Introduction to MPEG-7: multimedia content description interface**. [S.l.]: John Wiley & Sons, 2002.
- MEIJERING, E.; CAPPELLEN, G. van. Quantitative biological image analysis. *In: Imaging cellular and molecular biological functions*. [S.l.]: Springer, 2007. p. 45-70.
- MITCHELL, T. M. **Machine Learning**. New York, NY, USA: McGraw-Hill, Inc., 1997.
- MONSON, J.; WIRTHLIN, M.; HUTCHINGS, B. L. Optimization techniques for a high level synthesis implementation of the sobel filter. *In: IEEE. 2013 International Conference on Reconfigurable Computing and FPGAs (ReConFig)*. [S.l.], 2013. p. 1-6.
- NIXON, M.; AGUADO, A. S. **Feature extraction and image processing for computer vision**. [S.l.]: Academic Press, 2012.
- OTSU, N. A threshold selection method from gray-level histograms. **IEEE transactions on systems, man, and cybernetics**, IEEE, v. 9, n. 1, p. 62-66, 1979.
- PILLAI, A.; KWARTOWITZ, D. Ameliorating mammograms by using novel image processing algorithms. *In: INTERNATIONAL SOCIETY FOR OPTICS AND PHOTONICS. Medical Imaging 2014: Biomedical Applications in Molecular, Structural, and Functional Imaging*. [S.l.], 2014.
- PREWITT, J. M. Object enhancement and extraction. **Picture processing and Psychopictorics**, Academic Press New York, v. 10, n. 1, p. 15-19, 1970.
- ROCHA, A. *et al.* Automatic fruit and vegetable classification from images. **Computers and Electronics in Agriculture**, Elsevier, v. 70, n. 1, p. 96-104, 2010.
- SALDANHA, M. F.; FREITAS, C. Segmentação de imagens digitais: Uma revisão. **Divisão de Processamento de Imagens-Instituto Nacional de Pesquisas Espaciais (INPE)**. São Paulo, 2009.
- SHARIFF, A. *et al.* Automated image analysis for high-content screening and analysis. **Journal of biomolecular screening**, SAGE Publications Sage CA: Los Angeles, CA, v. 15, n. 7, p. 726-734, 2010.
- SILVA, J. A. da. **Deteção de Imagens Manipuladas utilizando Descritores Locais**. Tese (Doutorado) – Universidade Federal de Pernambuco, 2017.
- SOLOMON, C.; BRECKON, T. **Fundamentos de processamento digital de imagens: uma abordagem prática com exemplos em Matlab**. [S.l.]: Grupo Gen-LTC, 2000.
- SONKA, M.; HLAVAC, V.; BOYLE, R. **Image processing, analysis, and machine vision**. [S.l.]: Cengage Learning, 2014.
- SUGIARTO, B. *et al.* Wood identification based on histogram of oriented gradient (hog) feature and support vector machine (svm) classifier. *In: IEEE. 2017 2nd International conferences on Information Technology, Information Systems and Electrical Engineering (ICITISEE)*. [S.l.], 2017. p. 337-341.
- TAKAHASHI, A.; BEDREGAL, B. R. C.; LYRA, A. Uma versão intervalar do método de segmentação de imagens utilizando o k-means. **Trends in Applied and Computational Mathematics**, v. 6, n. 2, p. 315-324, 2005.
- WALT, S. Van der *et al.* scikit-image: image processing in python. **PeerJ**, PeerJ Inc., v. 2, p. e453, 2014.
- YANG, D. *et al.* Evaluation of tumor-derived mri-texture features for discrimination of molecular subtypes and prediction of 12-month survival status in glioblastoma. **Medical physics**, Wiley Online Library, v. 42, n. 11, p. 6725-6735, 2015.
- ZHANG, Y. *et al.* Fruit classification using computer vision and feedforward neural network. **Journal of Food Engineering**, Elsevier, v. 143, p. 167-177, 2014.
- ZUNIGA, O. A.; HARALICK, R. M. Integrated directional derivative gradient operator. **IEEE Transactions on systems, man, and cybernetics**, IEEE, v. 17, n. 3, p. 508-517, 1987.

15
ANOTAÇÃO GENÔMICA
Alexandre Rafael Lenz¹

1 Introdução

A genômica compreende a análise *in silico* da sequência completa de nucleotídeos de um dado organismo, ou simplesmente genoma. Essa ciência pode se dedicar a determinar a sequência completa do DNA de organismos ou, em menor escala, pode limitar-se a uma porção do genoma que seja de interesse. Sua relevância tem se expandido desde o primeiro mapeamento dos genes relacionados ao fenótipo da doença de Huntington no cromossomo 4 em humanos (GUSELLA *et al.*, 1983). Desde então, o estudo da genômica tem crescido, rompendo fronteiras e desafiando os cientistas, devido a sua aplicabilidade nas mais diversas áreas.

Os projetos de sequenciamento genômico foram por muito tempo confinados a organismos modelo e de interesse biomédico e exigiram o esforço conjunto de grandes consórcios. No entanto, o rápido progresso na tecnologia de sequenciamento de alto desempenho e o desenvolvimento simultâneo de ferramentas bioinformáticas democratizaram esse campo. Atualmente, o sequenciamento de genomas completos encontra-se acessível a grupos de pesquisa individuais, permitindo o sequenciamento do genoma de qualquer organismo de interesse.

Os projetos de sequenciamento de genomas completos, particularmente o Projeto Genoma Humano, expandiram a aplicabilidade das informações genômicas (HOOD; ROWEN, 2013). A genômica pode ser empregada na pesquisa forense para ajudar a resolver crimes (ARENAS *et al.*, 2017), para traçar os caminhos percorridos pelo *Homo sapiens* desde a África até a América (NIELSEN *et al.*, 2017) ou, ainda, para conservação de espécies ameaçadas de extinção, como a onça-pintada (FIGUEIRÓ *et al.*, 2017). No entanto, os primeiros passos da genômica se deram na medicina e a área médica continua sendo um dos principais alvos de pesquisa (BERG *et al.*, 2017).

Apesar da redução considerável dos custos para a realização de um sequenciamento de genoma, o custo e o esforço envolvidos ainda são consideráveis. Assim, o primeiro passo importante é considerar exaustivamente se o sequenciamento completo do genoma é necessário para abordar a demanda biológica em questão. Uma vez tomada a decisão de realizar o sequenciamento, um projeto de sequenciamento de genoma requer um planejamento cuidadoso com relação ao organismo envolvido e à qualidade pretendida do sequenciamento.

¹ Universidade de Caxias do Sul. *E-mail*: arlenz@ucs.br / alenz@uneb.br / arlenz@gmail.com

Os bioinformatas desempenham papel-chave, estabelecendo a conexão entre os biólogos e os especialistas em sistemas computacionais (LAMPA *et al.*, 2013). A colaboração entre biólogos, bioinformatas e especialistas em sistemas computacionais deve ser estabelecida já na fase de planejamento de qualquer projeto de sequenciamento de genoma (EKBLÖM; WOLF, 2014).

Um projeto de sequenciamento de genoma completo pode ser dividido em três fases: sequenciamento, montagem e anotação. Este capítulo revisa brevemente o estado da arte e fornece uma introdução ao fluxo de trabalho envolvido na fase de anotação genômica, com referência particular a genomas de organismos eucariotos.

2 Anotação genômica

A expressão *anotação genômica* inclui a identificação de sequências que codificam proteínas e sequências não codificadoras (*e.g.*, sequências repetitivas, rDNA, e ncRNA) em genomas, atribuindo informações biológicas (metadados) a esses elementos genômicos identificados (HARIDAS; SALAMOV; GRIGORIEV, 2018).

Embora a anotação do genoma envolva a caracterização de uma infinidade de elementos biologicamente significativos em uma sequência genômica, na prática no esforço despendido para anotação genômica concentra-se na predição correta de sequências codificadoras de proteínas (CDSs) e na atribuição de nomes e funções com significado biológico para esses genes (EKBLÖM; WOLF, 2014).

Contudo, isso não diminui o papel essencial desempenhado por sequências não codificantes na regulação transcricional, mas principalmente porque as abordagens para caracterizá-las são razoavelmente diretas (*e.g.*, detecção de ncRNA), ou são o foco de análises muito especializadas (*e.g.*, sítios de ligação de fatores de transcrição e elementos promotores) (DOMINGUEZ DEL ANGEL *et al.*, 2018).

Esse processo de anotação de sequências de DNA consiste em várias etapas sucessivas, sendo tipicamente complicado por envolver uma grande quantidade de ferramentas computacionais e muitos arquivos de entrada e saída. Uma anotação completa de genoma constitui um esforço considerável e requer proficiência em bioinformática. Assim, torna-se fundamental a utilização de um conjunto de ferramentas adequadas para facilitar essa análise complexa e ter fluxos de trabalho reprodutíveis (EKBLÖM; WOLF, 2014).

Esses conjuntos de ferramentas de anotação são geralmente chamados de pipelines de anotação. A qualidade da anotação genômica é fortemente dependente da qualidade da montagem e da disponibilidade de dados associados, tais como sequências de RNA e proteínas do organismo em questão ou de algum organismo intimamente

relacionado (DOMINGUEZ DEL ANGEL *et al.*, 2018; HARIDAS; SALAMOV; GRIGORIEV, 2018).

Embora os pipelines de anotação genômica costumam ter detalhes diferentes, eles compartilham um conjunto principal de recursos. Geralmente, a anotação de estruturas gênicas é dividida em duas fases distintas. Na primeira fase, a fase computacional, ocorre a identificação de sequências que codificam proteínas, no genoma também conhecida como predição de genes. Na segunda fase, a fase de anotação, ocorre a atribuição de informações biológicas aos elementos genômicos, sendo comumente chamada de anotação funcional (YANDELL; ENCE, 2012).

2.1 Predição de genes

A predição de genes é o processo de determinar corretamente a localização e a estrutura dos genes codificadores de proteínas em um genoma. Esse processo está bem estabelecido e conta com o suporte de muitos algoritmos de sucesso desenvolvidos nas últimas décadas. Esta fase inicia-se com a identificação de sequências repetitivas, etapa fundamental para garantir o funcionamento adequado dos algoritmos de predição.

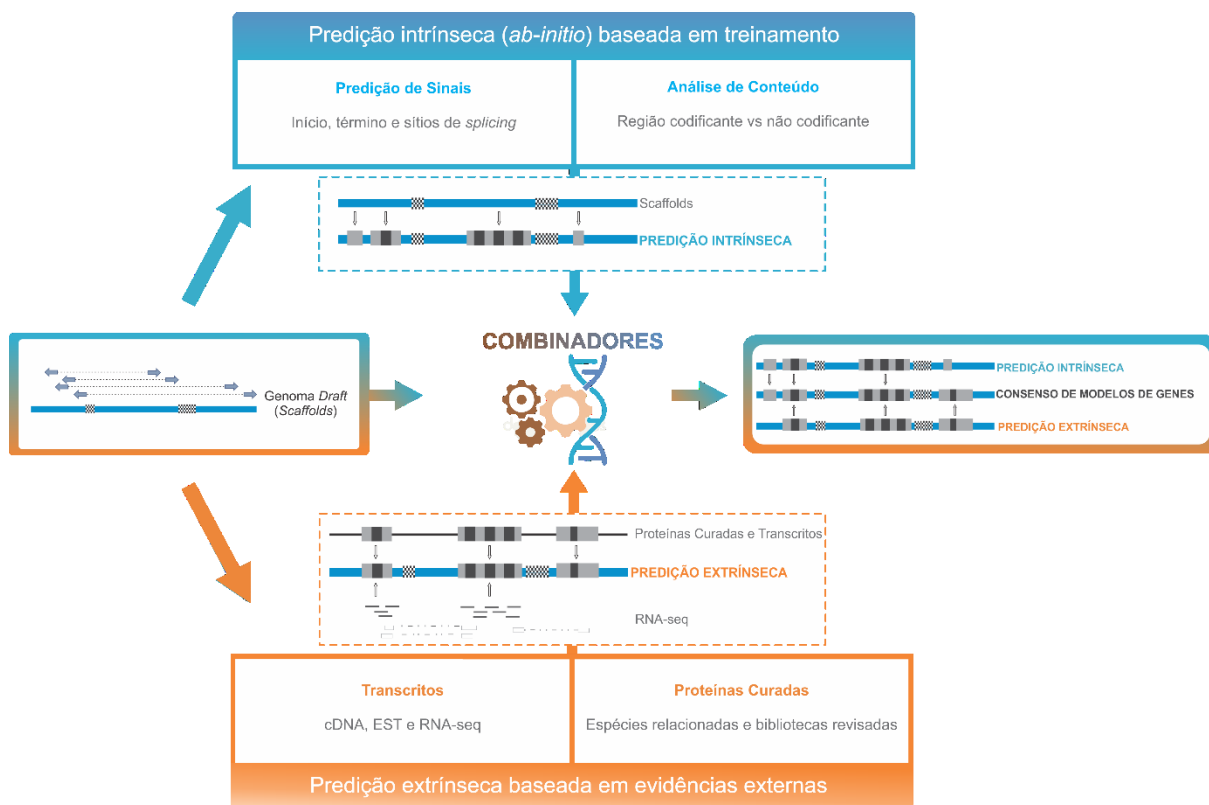
A maioria dos genomas de organismos mais complexos abriga uma abundância de sequências repetitivas que precisam ser excluídas dos passos subsequentes. Existem vários tipos de repetições, desde casos simples que compreendem dezenas a várias centenas de repetições de um mesmo motivo longo de nucleotídeos a genes móveis, elementos transponíveis ou fragmentos de genes de origem viral, que confundem os genomas hospedeiros (BAO; KOJIMA; KOHANY, 2015).

As sequências repetitivas podem compreender até bem mais da metade da sequência total de um genoma e podem levar a resultados espúrios de anotação, se não forem levadas em consideração. Antes da predição de genes em eucariotos, é importante mascarar sequências repetitivas, incluindo regiões de baixa complexidade e elementos transponíveis. Portanto, este é o primeiro passo da anotação automática de genes, sendo realizado por ferramentas computacionais especializadas. Após o mascaramento de repetições, o próximo passo envolve a predição de genes propriamente dita (YANDELL; ENCE, 2012; EKBLUM; WOLF, 2014).

Em geral, como pode ser observado na Figura 1, existem três abordagens principais para predição de genes em um genoma: intrínseca (ou *ab-initio*), extrínseca e combinada. Enquanto a abordagem intrínseca se concentra apenas na informação que pode ser extraída da própria sequência genômica, como a probabilidade da mesma ser uma sequência codificadora de proteína e o reconhecimento de sítios de *splicing*, a abordagem extrínseca usa a similaridade a outros tipos de sequência como informação (*e.g.*, transcritos e/ou proteínas). Existem vantagens e desvantagens inerentes a cada

uma das abordagens; assim, as duas abordagens podem ser combinadas computacionalmente para melhorar a acurácia das predições, dando origem à abordagem combinada (DOMINGUEZ DEL ANGEL *et al.*, 2018).

Figura 1 – Ilustração das abordagens para predição de genes



Fonte: Adaptada de Dominguez del Angel *et al.* (2018).

A predição intrínseca de genes tenta identificar genes usando modelos estatísticos, os quais necessitam de treinamento e otimização. Esta categoria de algoritmos realiza a busca sistemática na sequência de DNA por certos sinais indicadores de genes codificadores de proteínas. Os modelos estatísticos dependem de características genômicas específicas do organismo, como frequências de códons e distribuições de comprimentos de íntron-éxon, para distinguir genes de regiões intergênicas e para determinar estruturas íntron-éxon. Dessa forma, o objetivo dos preditores *ab initio* é a busca por fases de leitura aberta (ORF, de *open reading frame* em inglês, trechos de sequência sem códons de parada e potencialmente codificando uma proteína).

Um bom conjunto de treinamento é primordial para esta abordagem, ou seja, um conjunto de genes estruturalmente bem anotados, usados para construir modelos e para treinar as ferramentas de predição de genes. Como cada genoma é diferente, esses

modelos e parâmetros devem ser específicos para cada genoma e, portanto, precisam ser reconstruídos e retreinados para cada nova espécie. Esta é, no entanto, também a grande vantagem desta abordagem, já que é capaz de realizar a predição de genes em rápida evolução e genes específicos de uma espécie (DOMINGUEZ DEL ANGEL *et al.*, 2018).

A maioria dos preditores de genes *ab-initio* vem com arquivos de parâmetros pré-calculados para alguns genomas clássicos. No entanto, a menos que o genoma a ser anotado esteja intimamente relacionado a um organismo-modelo, para o qual os arquivos de parâmetros pré-compilados estejam disponíveis, o preditor de genes precisa ser treinado para realizar a predição no genoma que está sendo estudado, pois, mesmo organismos intimamente relacionados podem diferir em relação ao comprimento de íntrons e conteúdo GC (YANDELL; ENCE, 2012).

Outra vantagem dos preditores *ab-initio* é que, em princípio, eles não precisam de evidências externas para identificar um gene ou para determinar sua estrutura íntron-éxon. No entanto, esta categoria de algoritmos geralmente limita-se a encontrar CDSs e não contemplam regiões não traduzidas (UTRs) ou transcritos de *splicing* alternativo. Essa abordagem também tende a falhar na precisão, embora a maioria das ferramentas computacionais possam ser treinadas para ajustar seus parâmetros de predição às características específicas do organismo, melhorando assim a precisão (YANDELL; ENCE, 2012).

A predição extrínseca, por outro lado, baseia-se em evidências externas que podem ser alinhadas sobre o genoma de interesse. Uma das abordagens compreende a predição de genes baseada em homologia de proteínas, sendo realizada a partir do mapeamento de proteínas de outros organismos sobre o genoma de interesse. Esta abordagem de predição de genes permite explorar um vasto número de sequências proteicas revisadas, que encontram-se disponíveis em bancos de dados públicos (*e.g.*, NCBI/RefSeq, UniProtKB/Swiss-Prot) (DOMINGUEZ DEL ANGEL *et al.*, 2018).

Conforme Gabaldón e Koonin (2013), ortólogos são tipicamente os genes mais similares nas respectivas espécies em termos de sequência, estrutura, arquitetura de domínio e função. Assim, sequências de proteínas revisadas de outras espécies fornecem uma boa indicação sobre a presença e localização de genes. Como as sequências polipeptídicas geralmente são mais conservadas do que as sequências de nucleotídeos, elas ainda podem ser alinhadas, mesmo a partir de espécies relacionadas mais distantes.

Embora seja muito útil para determinar a presença de genes, esta abordagem nem sempre fornece informações precisas sobre a estrutura exata de um gene. Portanto, sugere-se a utilização de proteínas bem-anotadas e revisadas de espécies estreitamente

relacionadas para elevar a acurácia dos modelos de genes preditos por essa abordagem (HARIDAS; SALAMOV; GRIGORIEV, 2018; DOMINGUEZ DEL ANGEL *et al.*, 2018).

Outra abordagem extrínseca compreende a predição de genes baseada em homologia de transcriptoma. As informações de transcriptoma, sejam etiquetas de sequências expressas (ESTs, de *expressed sequence tags* em inglês), DNA complementar (cDNA), RNA-Seq ou outros tipos de transcritos disponíveis, desempenham papel ainda mais importante na predição extrínseca, fornecendo informações muito precisas para a predição correta da estrutura dos genes (DOMINGUEZ DEL ANGEL *et al.*, 2018).

O preditor de genes baseado em homologia de transcriptoma utiliza o conjunto completo de transcritos disponíveis para construir modelos de genes, a partir dos transcritos alinhados. Os dados de RNA-Seq, de preferência fita-específica, podem ser usados de duas maneiras (HAAS *et al.*, 2011; CONESA *et al.*, 2016; ZHAO *et al.*, 2011): (i) as sequências geradas pelo RNA-Seq são mapeadas diretamente sobre o genoma de interesse, para realizar a identificação de transcritos; (ii) a montagem do transcriptoma ocorre sem genoma de referência, gerando o conjunto completo de transcritos a partir do RNA-Seq.

A predição extrínseca utiliza diferentes fontes de evidências como ESTs, cDNA e proteínas de espécies intimamente relacionadas. Estas evidências de espécies próximas podem ser obtidas em bancos de dados específicos (*e.g.*, para fungos: MycoCosm). Também é comum o uso de bibliotecas curadas de proteínas (*e.g.*, UniProtKB/Swiss-Prot e NCBI/RefSeq). Devem ser evitadas as proteínas preditas que não tenham sido revisadas e curadas, isto porque modelos não validados podem piorar a precisão dos preditores. Todas as evidências selecionadas devem ser alinhadas ao genoma e, em seguida, os dados de RNA-Seq (transcritos) também devem ser alinhados, quando disponíveis. Os sítios de *splicing* devem então ser identificados, e as evidências devem ser pós-processadas e agrupadas, antes de serem enviadas para a ferramenta computacional inferir o conjunto final de modelos de genes.

De acordo com Yandell e Ence (2012), a predição de genes orientada por evidências tem um grande potencial para melhorar a qualidade da predição de genes em genomas recém-sequenciados, mas na prática pode ser difícil de usar. Esse processo é oneroso e exige o conhecimento aprofundado de diversas ferramentas computacionais especializadas, sendo um dos principais obstáculos que os pipelines de anotação tentam superar.

Devido à estrutura íntron-éxon dos genes de organismos eucariotos, a predição de genes é uma das partes mais desafiadoras da anotação genômica, tendo em vista que

uma das principais dificuldades na anotação genômica é a distinção entre genes codificadores de proteínas, transposons e pseudogenes. Enquanto os preditores *ab-initio* são precisamente caracterizados como preditores de novos genes, os preditores baseados em evidências extrínsecas são geralmente necessários para estabelecer conclusivamente que um gene predito é funcional (EKBLÖM; WOLF, 2014; KEILWAGEN *et al.*, 2018).

Geralmente, os autores recomendam o uso de várias abordagens de predição de genes para combinar diferentes tipos de evidências para a anotação: *ab-initio*, baseada em homologia de proteínas e baseada em homologia de transcriptoma. Nos últimos anos, várias abordagens computacionais foram desenvolvidas, com o intuito de combinar múltiplas fontes, permitindo um incremento significativo na acurácia da predição de genes codificadores de proteínas (HAAS *et al.*, 2011; EKBLÖM; WOLF, 2014; DOMINGUEZ DEL ANGEL *et al.*, 2018; KEILWAGEN *et al.*, 2018).

Ferramentas computacionais que implementam a abordagem combinada são chamadas de Combinadores; elas tomam como entrada uma sequência genômica e implementam um método computacional para construir modelos de genes, a partir de evidências geradas de um conjunto diversificado de fontes (ALLEN; PERTEA; SALZBERG, 2004).

Os Combinadores são provavelmente as ferramentas de predição de genes mais populares e amplamente utilizadas. No entanto, nem todos esses Combinadores são iguais. Enquanto alguns simplesmente escolhem o modelo mais apropriado ou constroem um consenso, a partir das evidências de entrada fornecidas para um determinado *locus*, outros têm uma abordagem mais integrada, na qual a predição intrínseca pode ser modificada pelos dados extrínsecos, resultando em uma predição mais consistente (DOMINGUEZ DEL ANGEL *et al.*, 2018).

Ao executar a anotação genômica, é preciso fazer escolhas, não apenas em relação às ferramentas que serão utilizadas, mas também em relação às fontes e aos tipos de evidências que serão utilizadas em cada etapa. Obviamente, a escolha deve ir em direção às fontes de dados mais confiáveis, implicando muitas vezes evidências menos abrangentes. Por outro lado, o uso de informações de qualidade inferior levará inevitavelmente a um resultado de predição de genes inferior (DOMINGUEZ DEL ANGEL *et al.*, 2018).

Embora as ferramentas de predição geralmente forneçam bons resultados, elas continuam sendo propensas a erros, a validação qualitativa é importante (*e.g.*, avaliando o comprimento das ORFs). A inspeção visual da anotação é outro componente vital para detectar problemas sistemáticos como genes faltantes, predições falsas, vazamento de íntron (íntrons sendo anotados como éxons devido à presença de pré-mRNA), e modelos de genes desmembrados ou agrupados, os quais levam erros ao conjunto final

de genes. Embora a revisão manual dos modelos de genes despense muito tempo, é uma etapa extremamente necessária para gerar um conjunto de genes preciso e confiável. Algumas ferramentas são particularmente úteis, pois permitem que o usuário edite a estrutura do gene predito diretamente por meio de uma interface visual (HAAS *et al.*, 2011; EKBLUM; WOLF, 2014; MCDONNELL; STRASSER; TSANG, 2018).

2.2 Anotação funcional

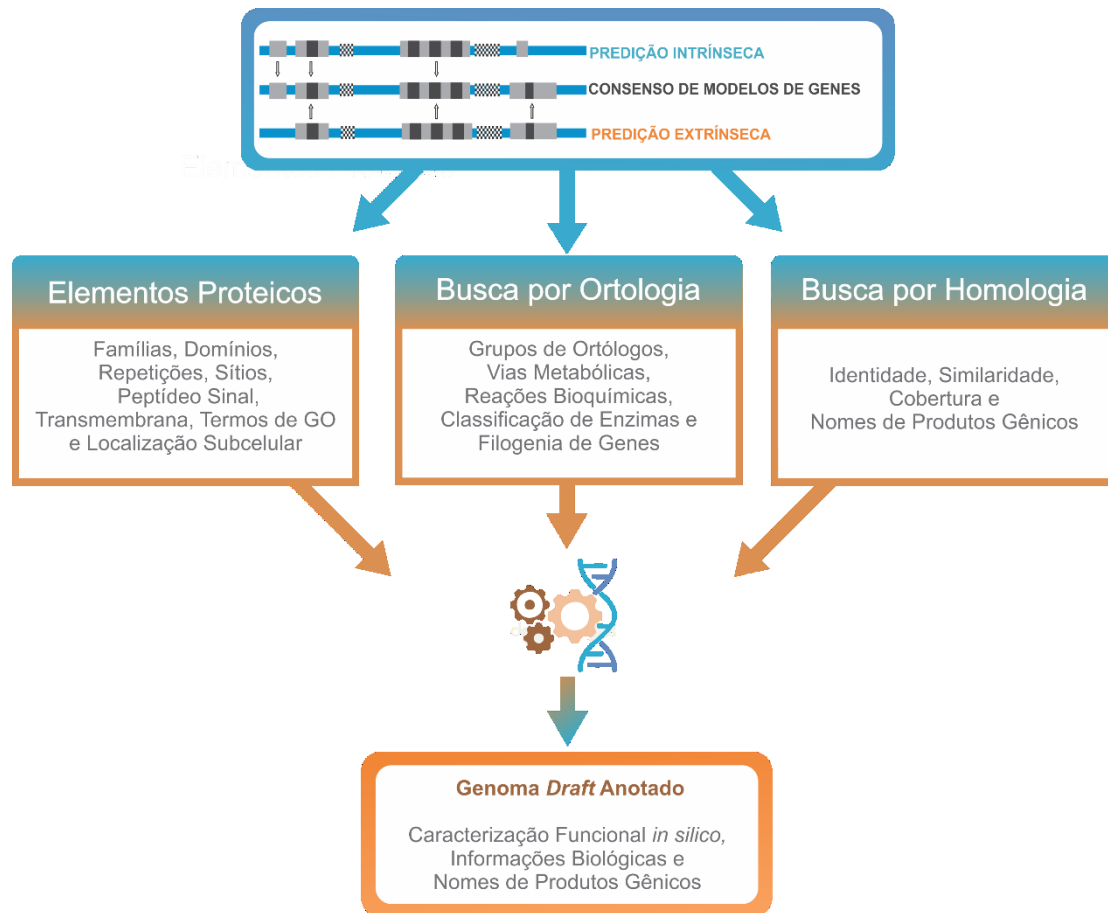
A anotação funcional consiste em atribuir informações biológicas significativas às sequências codificadoras de proteínas e aos seus elementos derivados (*e.g.*, gene, mRNA), analisando a estrutura e a composição da sequência, bem como considerando o que se sabe à respeito de espécies intimamente relacionadas, que podem ser usadas como referência (DOMINGUEZ DEL ANGEL *et al.*, 2018).

O objetivo principal desta fase é a atribuição de nomes de produtos gênicos, geralmente baseados na caracterização funcional *in silico* dos genes preditos. A caracterização funcional dos elementos genômicos compreende diversas informações biológicas como função bioquímica, função biológica, interações proteicas e mecanismos de regulação e expressão. A caracterização destes elementos permite melhor compreensão das propriedades gênicas específicas, como as vias metabólicas e as semelhanças em comparação com espécies intimamente relacionadas (HAAS *et al.*, 2011; DOMINGUEZ DEL ANGEL *et al.*, 2018).

Existem várias ferramentas disponíveis para anotação funcional que permitem aos usuários obterem anotações para seu conjunto de genes de interesse, por meio de bancos de dados públicos. As ferramentas podem ser executadas individualmente e, em seguida, os resultados são combinados. No entanto, existem fluxos de trabalho disponíveis que fornecem todo o processo de anotação funcional de forma automatizada. Esses pipelines podem incluir a instalação das ferramentas necessárias e os bancos de dados correspondentes, ou os usuários podem fazer essa instalação por conta própria e o pipeline apenas fornece um fluxo estruturado para a análise (DOMINGUEZ DEL ANGEL *et al.*, 2018).

Em um pipeline típico de anotação funcional, conforme ilustrado na Figura 2, informações funcionais são atribuídas às proteínas preditas. O processo implementa três rotas paralelas para a caracterização funcional. A primeira refere-se aos domínios, motivos e às famílias de proteínas, a segunda corresponde à busca de ortólogos e, por fim, a terceira compreende à busca por homologia. Em síntese, as saídas das três fontes de informações são agrupadas para elevar o nível de acurácia da caracterização funcional e da atribuição de nomes de produtos gênicos (DOMINGUEZ DEL ANGEL *et al.*, 2018).

Figura 2 – Ilustração de um pipeline para anotação funcional



Fonte: Adaptada de Dominguez del Angel *et al.* (2018).

Antes da atribuição de nomes de produtos gênicos, é importante caracterizar elementos funcionais adicionais que incluem domínios, motivos, famílias das proteínas, vias metabólicas, localização subcelular da proteína, entre outros. A anotação funcional ainda pode agregar informações específicas que são relevantes para um determinado reino ou filo. Tomados em conjunto, os perfis de função gerais e especializados fornecem uma visão abrangente das características bioquímicas de um genoma, que podem ser correlacionadas com os fenótipos biológicos de uma espécie (HAAS *et al.*, 2011).

A função das proteínas preditas pode ser computacionalmente inferida com base na similaridade entre a sequência de interesse e outras sequências de diferentes repositórios públicos (*e.g.*, BLASTP sobre UniProtKB/Swiss-Prot). Os protocolos de anotação costumam adotar critérios rigorosos para atribuição de nomes de produtos gênicos baseados em similaridade de sequências.

Por exemplo, para McDonnell, Strasser e Tsang (2018), se o modelo do gene e o seu BLASTP corresponderem a uma identidade $\geq 98\%$ ao longo de todo o seu comprimento, então as duas proteínas são consideradas funcionalmente equivalentes, e se a proteína revisada for caracterizada experimentalmente, então pode ser atribuído o mesmo nome de gene e o mesmo nome do produto gênico à proteína que está sendo anotada. Quando a identidade entre as sequências proteicas for $\geq 70\%$ e a cobertura da consulta $\geq 70\%$, é atribuído somente o nome do produto gênico da proteína revisada à proteína que está sendo anotada. Haas *et al.* (2011) ainda adicionam outro critério complementar, indicando que, além da identidade e da cobertura $\geq 70\%$, a diferença de comprimento entre a proteína revisada e a proteína que está sendo anotada deve ser $\leq 10\%$.

Identidade e/ou cobertura inferiores aos citados são abordados de diferentes formas nos protocolos de anotação funcional. Para os genes restantes com nomes de produtos não atribuídos, é comum delegar uma função mais geral com base em seu(s) domínio(s) conservados obtidos a partir da caracterização dos elementos funcionais, principalmente obtidos a partir dos bancos de dados InterPro e/ou Pfam.

Deve-se tomar cuidado ao atribuir resultados baseados em similaridade de sequência, tendo em vista que duas sequências cuja evolução foi independente poderiam ser consideradas homólogas, por compartilharem alguns domínios comuns. Assim, sempre que possível, é aconselhável o uso de sequências ortólogas para fins de anotação, em vez de simplesmente sequências homólogas (DOMINGUEZ DEL ANGEL *et al.*, 2018).

A transferência de anotação funcional baseia-se na “conjectura da função ortogonal”: ortólogos realizam funções idênticas, ou mais precisamente biologicamente equivalentes, em diferentes organismos (GABALDÓN; KOONIN, 2013). Dessa forma, a utilização de proteínas ortólogas, bem anotadas e revisadas de espécies estreitamente relacionadas, podem servir de embasamento para anotação funcional de um genoma. Essa abordagem de anotação funcional, baseada em ortologia, é apoiada por diversos bancos de dados de grupos de ortólogos e ferramentas que dão suporte à identificação de ortólogos.

Por fim, para a nomenclatura de produtos gênicos, comumente são utilizadas as diretrizes internacionais de nomenclatura adotadas tanto pelo GenBank quanto pelo UniProtKB/Swiss-Prot (EMBL-EBI *et al.*, 2018).

Com o crescente número de sequências em repositórios públicos, é possível realizar várias pesquisas e combinar os resultados obtidos para gerar uma anotação consensual. A caracterização funcional dos elementos genômicos é um processo complexo e propenso a erros, os pipelines de anotação funcional automatizada

acumulam e propagam os erros em bases de dados públicas. Portanto, uma curadoria manual é muitas vezes necessária para avaliar vários tipos de evidências e elevar o grau de confiabilidade da anotação funcional (DOMINGUEZ DEL ANGEL *et al.*, 2018).

No entanto, a análise aprofundada de todo genoma exige esforço e tempo, evidenciando a importância da adoção de critérios rigorosos para atribuição de nomes de produtos gênicos. Assim, em muitos casos, a curadoria pode limitar-se a uma porção do genoma que seja de interesse. Em última análise, a verificação experimental é a única maneira de ter certeza de que a caracterização funcional dos produtos gênicos está correta.

Além da curadoria manual, outro fator crucial para elevar o grau de confiabilidade da anotação compreende a escolha de ferramentas computacionais e de fontes de dados confiáveis e revisadas.

3 Ferramentas computacionais e bancos de dados

A escolha das ferramentas computacionais a serem utilizadas em um projeto de anotação genômica é muito importante. As escolhas influenciam diretamente no esforço e no tempo gasto para essa atividade, sendo um dos fatores determinantes para o sucesso do processo de anotação. Vale ressaltar que as ferramentas computacionais geralmente são muito específicas para determinados tipos de dados de entrada e podem não ser capazes de analisar outros formatos, necessitando conversões ou até inviabilizando seu uso.

Nesta seção, são discutidas as principais ferramentas disponíveis e as mais usadas, além de algumas tecnologias de suporte para cada uma das etapas da anotação genômica, descritas na seção anterior. Diversos *softwares* são necessários para a anotação, e as respectivas instalações devem ocorrer em um sistema operacional baseado em Unix, usando a documentação incluída em cada um deles. O uso de pipelines pode reduzir o grau de dificuldade dessa tarefa e garantir a repetitividade e reprodutibilidade do processo de anotação.

3.1 Predição de elementos não codificantes

Antes da predição de genes deve ocorrer a identificação de sequências repetitivas, as quais são muitas vezes malconservadas entre as espécies. O mascaramento das repetições pode ser realizado por um dos *softwares* mais conhecidos para este fim, o RepeatMasker (SMIT; HUBLEY; GREEN, 2013-2015). É aconselhável a criação de uma biblioteca de repetições específica para a espécie, usando ferramentas como RepeatModeler (SMIT; HUBLEY, 2008-2015) ou RepeatExplorer2 (NOVÁK; NEUMANN; MACAS, 2010). Além de bibliotecas curadas de repetições, como Dfam

(biblioteca de elementos transponíveis de DNA) (HUBLEY *et al.*, 2016) e Repbase (biblioteca de sequências consenso originais ou reconstruídas de elementos repetitivos) (BAO; KOJIMA; KOHANY, 2015).

Após o mascaramento, deve ocorrer a predição de outros elementos não codificantes, como a predição de tRNAs usando tRNAscan-SE (LOWE; CHAN, 2016). Para a identificação de snoRNAs pode ser usado o snoSeeker (YANG *et al.*, 2006). Caso tenha sido realizado sequenciamento especializado para outros tipos de ncRNA (como miRNA), as *reads* podem ser alinhadas ao genoma com sRNAtoolbox (RUEDA *et al.*, 2015) ou miRDeep* (AN *et al.*, 2013). Também pode ser utilizada a ferramenta genérica para predição de RNAs não codificantes: Infernal (NAWROCKI; EDDY, 2013), suas predições ocorrem com base no banco de dados RFAM (KALVARI *et al.*, 2017).

Outra etapa importante é a detecção e anotação dos elementos transponíveis. O pacote REPET (FLUTRE *et al.*, 2011) é uma das ferramentas mais utilizadas para grandes genomas eucarióticos, com mais de 50 genomas analisados. PiRATE (BERTHELIER *et al.*, 2018) também pode ser utilizada com a mesma finalidade.

3.2 Predição de genes

Após a predição dos elementos não codificantes relevantes, ocorre a predição de elementos codificantes. Para a etapa de predição intrínseca de genes (ou *ab-initio*), existe uma infinidade de ferramentas, as mais utilizadas e referenciadas são: AUGUSTUS (STANKE; MORGENSTERN, 2005) um dos *softwares* mais utilizados para predição de genes em sequências genômicas eucarióticas, fornece arquivos de treinamento para várias espécies e permite o treinamento para novas espécies; GeneMark-ES (TER-HOVHANNISYAN *et al.*, 2008) identifica genes codificadores de proteínas em genomas de organismos eucariotos, destacando-se por que realiza a predição sem conjuntos de treinamento; FGENESH (SOLOVYEV *et al.*, 2006) é um preditor *ab-initio* bastante preciso e rápido; no entanto, os programas acessíveis no *site* da Softberry são publicados principalmente para uso em volumes leves (inferior a 15 execuções por dia por domínio acadêmico); GlimmerHMM (MAJOROS; PERTEA; SALZBERG, 2004), SNAP (KORF, 2004), GeneId (BLANCO; PARRA; GUIGÓ, 2007); GENSCAN e GenomeScan (YEH; LIM; BURGE, 2001), Twinscan/N-SCAN (GROSS; BRENT, 2006) e mGene (SCHWEIKERT *et al.*, 2009).

O processo de predição extrínseca exige o conhecimento aprofundado de diversas ferramentas computacionais especializadas, tornando-o oneroso e complexo. As *reads* curtas geradas pelo RNA-Seq precisam ser montadas para gerar o conjunto de transcritos. A montagem de RNA-Seq com genoma de referência pode ser realizada

com as ferramentas: minimap2 (LI, 2018), hisat2 (KIM; LANGMEAD; SALZBERG, 2015), Cufflinks (TRAPNELL *et al.*, 2012), Genome-guided Trinity (GRABHERR *et al.*, 2011), etc. Os transcritos também podem ser montados sem genoma de referência (montagem *de novo* de RNA-Seq) com os *softwares*: Trinity (GRABHERR *et al.*, 2011), MEGAHIT (LI *et al.*, 2015), StringTie (PERTEA *et al.*, 2015), Trans-ABYSS (ROBERTSON *et al.*, 2010), Velvet/Oases (SCHULZ *et al.*, 2012) e SOAPdenovo-Trans (XIE *et al.*, 2014).

Em seguida, as evidências de transcritos resultantes do RNA-Seq, bem como as ESTs e cDNA devem ser alinhadas ao genoma. Atualmente, existem mais de 90 alinhadores que podem ser utilizados para esta função, alguns dos principais são listados a seguir: BLAT (KENT, 2002), GMAP-GSNAP (WU; NACU, 2010), BWA (LI; DURBIN, 2009), Bowtie2 (LANGMEAD; SALZBERG, 2012), STAR (DOBIN *et al.*, 2012) e TopHat (TRAPNELL *et al.*, 2012). O pipeline PASA (HAAS *et al.*, 2003) também pode ser útil nessa fase, é uma ferramenta que explora alinhamentos de transcritos expressos para modelar estruturas de genes. Além do alinhamento de transcritos, na predição extrínseca ocorre o alinhamento de evidências de proteínas revisadas ao genoma. As proteínas também podem ser alinhadas a partir de diversas ferramentas: BLASTN (CAMACHO *et al.*, 2009), GeneWise (BIRNEY; CLAMP; DURBIN, 2004), Exonerate (SLATER; BIRNEY, 2005), Diamond (BUCHFINK; XIE; HUSON, 2015) e FGenesh+ (SOLOVYEV, 2004).

Os Combinadores, por sua vez, produzem um consenso para cada modelo gênico predito, integrando múltiplas fontes de evidências geradas pelas ferramentas de predição intrínsecas e extrínsecas. Um dos Combinadores mais utilizados é o EvidenceModeler (EVM) (HAAS *et al.*, 2008); destacam-se também: Evigan (LIU *et al.*, 2008) e JIGSAW (ALLEN; SALZBERG, 2005).

Nota-se que existe uma infinidade de ferramentas relacionadas ao processo de anotação genômica. Escolher o conjunto de ferramentas mais adequado para os dados disponíveis nem sempre é uma tarefa fácil. Nesse sentido, surgem os pipelines de anotação que sugerem fluxos de trabalho e ferramentas para cada etapa do processo. Os pipelines podem ser genéricos (*e.g.* para eucariotos em geral), ou podem ser mais especializados (*e.g.* para um determinado reino ou filo). Pipelines mais genéricos geralmente são menos automatizados, muitas vezes sugerindo apenas os passos do fluxo de trabalho e algumas ferramentas, como no caso do NCBI (2018). Os pipelines procuram facilitar o processo de predição de genes, entre os pipelines para eucariotos destacam-se: Eugène (SCHIEX; MOISAN; ROUZÉ, 2001), BRAKER1 (HOFF *et al.*, 2016), MAKER2 (HOLT; YANDELL, 2011), GeMoMa (KEILWAGEN *et al.*, 2018) e ENSEMBL (AKEN *et al.*, 2016).

Os pipelines específicos podem englobar características genômicas de um reino ou filo, elevando a acurácia do processo. Para fungos, por exemplo, destacam-se: Broad Institute (HAAS *et al.*, 2011), Joint Genome Institute (JGI) (NORDBERG *et al.*, 2013) e SnowyOwl (REID *et al.*, 2014); Já para plantas, destacam-se: Joint Genome Institute (JGI) (NORDBERG *et al.*, 2013), TriAnnot (LEROY *et al.*, 2012) e MAKER-P (CAMPBELL *et al.*, 2014).

Uma dificuldade persistente na anotação do genoma é distinguir genes codificadores de proteínas funcionais dos pseudogenes. Pseudogenes são elementos genéticos hereditários formalmente definidos por duas propriedades: sua semelhança com genes funcionais e sua presumida falta de atividade. No entanto, sua caracterização precisa, particularmente, no que diz respeito à última qualidade, é muito imprecisa. PseudoPipe é um pipeline computacional baseado em homologia para identificação de pseudogenes (ZHANG *et al.*, 2006). Outra ferramenta útil para busca de pseudogenes é PPFINDER (VAN BAREN; BRENT, 2006).

A inspeção visual de genes preditos é uma tarefa importante para avaliar vários tipos de evidências e elevar o grau de confiabilidade das predições. Essa tarefa é facilitada por ferramentas web colaborativas para edição de anotação genômica como JBROWSE (BUELS *et al.*, 2016) e Apollo (DUNN *et al.*, 2019) ou por versões *desktop* como Artemis (RUTHERFORD *et al.*, 2000).

3.3 Anotação funcional

A primeira etapa da anotação funcional identifica domínios, sítios funcionais e famílias de proteínas, geralmente obtidos a partir de grandes bases de dados de famílias de proteínas e suas respectivas anotações (InterPro, Pfam e GO). A anotação pode ocorrer a partir de consultas via servidor web ou a partir do *download* e instalação dos bancos de dados e de ferramentas específicas como InterProScan (JONES *et al.*, 2014), HMMER (FINN; CLEMENTS; EDDY, 2011) e CD-Search (MARCHLER-BAUER, A.; BRYANT, S. H., 2004).

Outros domínios e sítios específicos podem ser identificados a partir de ferramentas especializadas. Os peptídeos sinais de secreção, sequências localizadas na porção N-terminal da proteína, determinam a secreção da mesma e podem ser preditos pela ferramenta SignalP (PETERSEN *et al.*, 2011). As proteínas transmembrana são proteínas firmemente aderidas aos lipídios da membrana e formam canais de transporte de substâncias, seus domínios e topologia podem ser preditos por ferramentas como TMHMM (KROGH *et al.*, 2001) ou Phobius (KÄLL; KROGH; SONNHAMMER, 2007). TargetP (EMANUELSSON *et al.*, 2007) ou WoLF PSORT (HORTON *et al.*, 2007) podem ser utilizadas para prever a localização subcelular.

Os números EC de *Enzyme Commission Numbers* em inglês, compreendem um esquema de classificação numérica para enzimas, baseado nas reações químicas que elas catalisam. A anotação de enzimas pode ser obtida juntamente com a identificação de vias metabólicas, reações bioquímicas, grupos de ortólogos, etc. Utilizam-se as ferramentas KEGG Mapper (KANEHISA *et al.*, 2016) ou KAAS (MORIYA *et al.*, 2007) para mapear os identificadores KEGG relacionados a cada proteína.

A segunda etapa da anotação funcional realiza buscas por ortologia em bancos de dados especializados (OrthoDB, EggNOG e PhylomeDB) a partir de consulta web, ou a partir da instalação dos *softwares* BUSCO (WATERHOUSE *et al.*, 2017) e EggNOG-mapper (HUERTA-CEPAS *et al.*, 2017). Outra forma de mapeamento de ortologia consiste em utilizar *softwares* especializados (NICHIO; MARCHAUKOSKI; RAITTZ, 2017). Nesse caso, a ortologia é identificada, a partir de bancos de dados específicos como Subconjuntos Taxonômicos ou Proteomas de espécies intimamente relacionadas, ambas disponíveis no UniProtKB. Por fim, a busca por homologia, terceira etapa da anotação funcional, geralmente ocorre sobre bases de dados de proteínas revisadas, utilizando a ferramenta BLASTP (CAMACHO *et al.*, 2009).

Os pipelines de anotação funcional pretendem agrupar informações de diversas ferramentas citadas para facilitar a anotação das proteínas preditas; destacam-se a ferramenta comercial Blast2GO (CONESA *et al.*, 2005) e a ferramenta para anotação de transcriptoma Trinotate (BRYANT *et al.*, 2017). Nota-se uma carência de pipelines de código-aberto para anotação funcional de proteínas preditas.

Quando reinos ou filos específicos estão sob análise, categorias de funções especializadas são de particular interesse. Para fungos, por exemplo, pode-se utilizar diversas ferramentas de predição: proteínas quinases utilizando Kinannotate (GOLDBERG *et al.*, 2013), enzimas ativas em carboidratos (CAZy) utilizando dbCAN2 (ZHANG *et al.*, 2018), proteínas ancoradas por GPI utilizando PredGPI (PIERLEONI; MARTELLI; CASADIO, 2008), transportadores utilizando BioV Suite (REDDY; SAIER, 2012), *clusters* de metabólitos secundários (PKS, NRPS, etc.) utilizando as ferramentas SMURF (KHALDI *et al.*, 2010) ou antiSMASH (BLIN *et al.*, 2017), peptidases (MEROPS) utilizando BLASTP (CAMACHO *et al.*, 2009) e fatores de transcrição via SUPERFAMILY (PANDURANGAN *et al.*, 2018).

3.4 Bancos de dados

A quantidade de bancos de dados genômicos é abundante e as informações contidas são variadas. Muitas vezes esses dados não são precisos e contêm erros. Assim, recomenda-se, sempre que possível, o uso de fontes confiáveis e revisadas. Alguns dos principais bancos de dados necessários para anotação genômica são listados a seguir:

O Consórcio InterPro (MITCHELL *et al.*, 2018) é uma plataforma com acesso a 14 bancos de dados: CATH-Gene3D, HAMAP, PANTHER, Pfam, PIRSF, PRINTS, ProDom, PROSITE Patterns, PROSITE Profiles, SMART, SUPERFAMILY, TIGRFAMs, CDD e SFLD. Os elementos do InterPro são classificados em famílias, domínios, repetições ou sítios, dependendo da entidade biológica que representam. Cada elemento também está associado aos termos do Gene Ontology (GO), que fornecem um vocabulário controlado para descrever a função da proteína, localização celular e envolvimento em vias e processos biológicos mais amplos.

O UniProt Knowledgebase (UniProtKB) (UNIPROT CONSORTIUM, 2018) é o componente principal do consórcio Universal Protein Resource (UniProt). UniProtKB é uma base de conhecimento de sequências de proteínas abrangente e anotada, que consiste em duas seções: UniProtKB/Swiss-Prot, contendo entradas manualmente anotadas por especialistas, e UniProtKB/TrEMBL, contendo entradas anotadas automaticamente extraídas do *European Molecular Biology Laboratory* (EMBL), bem como sequências e anotações importadas do ENSEMBL, EnsemblGenomes, EnsemblPlants e a partir do NCBI (RefSeq). Além das duas seções citadas, é possível escolher subconjuntos taxonômicos ou proteomas de espécies específicas no UniProtKB, conforme necessidade para o processo de anotação.

O *National Center for Biotechnology Information* (NCBI) fornece um grande conjunto de recursos *online* para informações e dados biológicos, incluindo o GenBank: o principal banco de dados de sequências de ácidos nucleicos (SAYERS *et al.*, 2018). Destacam-se também:

- NCBI proteínas de referência (RefSeq);
- NCBI sequências de proteínas não redundantes (nr);
- NCBI banco de dados de domínios conservados (CDD);
- NCBI genomas de referência (RefSeq);
- NCBI coleção de nucleotídeos (nr/nt).

O *European Bioinformatics Institute* (EMBL-EBI) hospeda dados de experimentos de ciências da vida e fornece acesso livre e irrestrito aos dados em todas as principais áreas da biologia e da biomedicina (SQUIZZATO *et al.*, 2015). Destacam-se os seguintes bancos de dados:

- ENSEMBL: é uma base de dados que gerencia recursos de anotação de genomas de referência e provê genômica comparativa (CUNNINGHAM *et al.*, 2018);
- Enzyme Portal: é uma base de dados para enzimas e proteínas com atividade enzimática (ALCANTARA *et al.*, 2012);
- MEROPS: é uma base de dados para peptidases e inibidores (RAWLINGS *et al.*, 2017);

– ChEMBL: é um banco de dados de moléculas bioativas curadas manualmente com propriedades semelhantes a drogas (MENDEZ *et al.*, 2018).

O Enciclopédia de Genes e Genoma de Kioto (KEGG) é uma coleção de bancos de dados que abrange funções de alto nível e utilidades de sistemas biológicos, como células, organismos e ecossistemas. As proteínas são associadas a vias metabólicas, reações bioquímicas, sua relação com doenças e com a área médica, nomenclatura de enzimas, grupos de ortólogos, módulos, hierarquias, entre outros (KANEHISA *et al.*, 2016).

Muitos grupos desenvolveram métodos e bancos de dados para mapear ortólogos, que podem ser muito úteis para transferência de anotação funcional, baseada na conjectura da função ortogonal. Destacam-se os seguintes bancos de dados e classificações:

– OrthoDB: é um catálogo abrangente de anotações evolutivas e funcionais de ortólogos (KRIVENTSEVA *et al.*, 2018);

– EggNOG: é um banco de dados de grupos ortólogos e anotação funcional (HUERTA-CEPAS *et al.*, 2018), inclui também a classificação KOG (de EuKaryotic Orthologous Groups em inglês), uma versão eucarioto-específica para identificar ortólogos e parálogos (TATUSOV *et al.*, 2003);

– PhylomeDB: é um banco de dados público para catálogos completos de filogenias de genes (HUERTA-CEPAS *et al.*, 2013).

Além dos bancos de dados citados, existe uma infinidade de bancos de dados específicos para as mais variadas classificações de sequências de nucleotídeos e proteínas, bem como para reinos, filos, classes, ordens, famílias e gêneros. A confiabilidade das fontes de dados é fundamental para anotação, bancos de dados específicos e curados podem elevar a confiabilidade das anotações. Seguem alguns exemplos de bancos de dados importantes: projeto genomas de vertebrados (VGP) (KOEPLI *et al.*, 2015), banco de dados de genomas de *Aspergillus* (AspGD) (CERQUEIRA *et al.*, 2013), genomas de fungos em MycoCosm e genomas de plantas verdes em Phytozome (NORDBERG *et al.*, 2013) e banco de dados de enzimas ativas em carboidratos (CAZy) (LOMBARD *et al.*, 2013).

Os resultados dos projetos de anotação genômica podem ser submetidos a bancos de dados públicos, como GenBank (SAYERS *et al.*, 2018), ENA (HARRISON *et al.*, 2018) e ENSEMBL (CUNNINGHAM *et al.*, 2018). A submissão do genoma e anotação garante a disponibilidade e consulta dos resultados a partir da divulgação para a comunidade científica. Essas publicações utilizam-se de ferramentas auxiliares como *Genome Annotation Generator* (GAG), que realiza a leitura de um genoma e sua

anotação, convertendo-o para o formato .tbl do NCBI, e Tbl2asn que automatiza a criação de registros de sequência para envio ao GenBank (GEIB *et al.*, 2018).

4 Perspectivas, limitações e desafios

A profundidade de anotação associada aos dados da sequência do genoma pode ser muito rica, especialmente para organismos-modelo, nos quais muitos genes foram bem caracterizados. Para obter informações adicionais sobre a função do gene, as anotações podem ser ligadas aos dados de expressão gênica, às principais vias nos mapas metabólicos e à literatura mais recente que elucida ainda mais o conhecimento sobre uma determinada função gênica. Manter esses dados especializados vai além da missão dos arquivos de sequência e, assim, os bancos de dados especializados surgiram ao longo dos anos para atender melhor à comunidade científica.

A anotação automatizada não é um problema resolvido, e as ferramentas de predição são suscetíveis a erros. Os custos de sequenciamento rapidamente decrescentes estão produzindo níveis de dados sem precedentes, gerando erros que podem facilmente aumentar de tamanho e se propagar ao longo de muitos conjuntos de dados. Torna-se essencial a tomada de medidas para resolver esses problemas.

A curadoria manual é frequentemente usada para avaliar vários tipos de evidências e melhorar as predições automatizadas. Em última análise, a verificação experimental é a única maneira de garantir que uma estrutura gênica está correta. A anotação é desafiadora, altamente subestimada em dificuldade e altamente subvalorizada, até que uma comunidade de impacto utilize as sequências genômicas anotadas. A anotação pode ser feita com alta precisão em um único nível de genes, por investigadores com experiência em famílias de genes. O desafio é como extrapolar isso para todo o genoma.

Uma combinação de anotações automatizadas, semiautomatizadas e manuais talvez seja a melhor maneira de abordar genomas nos quais não há grandes comunidades envolvidas. A anotação genômica é iterativa, nunca perfeita, sempre pode ser melhorada com novas evidências e melhores algoritmos.

O crescente escopo da anotação genômica apresenta os maiores desafios. Atualmente, a anotação genômica vai além da mera identificação de genes codificadores de proteínas, enfatiza-se cada vez mais a anotação de transposons, regiões reguladoras, pseudogenes e genes de ncRNA.

Referências

- AKEN, B. L. *et al.* The ensembl gene annotation system. **Database: the journal of biological databases and curation**, Oxford, 2016, baw093. doi:10.1093/database/baw093 Disponível em: <http://www.ensembl.org>. Acesso em: 3 abr. 2019.
- ALCANTARA, R. *et al.* The EBI enzyme portal. **Nucleic acids research**, v. 41, n. D1, p. D773-D780, 2012. doi:10.1093/nar/gks1112 Disponível em: <https://www.ebi.ac.uk/enzymeportal>. Acesso em: 4 abr. 2019.
- ALLEN, J. E.; PERTEA, M.; SALZBERG, S. L. Computational gene prediction using multiple sources of evidence. **Genome research**, v. 14, n. 1, p. 142-148, 2004. doi:10.1101/gr.1562804.
- ALLEN, J. E.; SALZBERG, S. L. JIGSAW: integration of multiple sources of evidence for gene prediction. **Bioinformatics**, v. 21, n. 18, p. 3596-3603, 2005. doi:10.1093/bioinformatics/bti609 Disponível em: <http://www.cbcb.umd.edu/software/jigsaw>. Acesso em: 3 abr. 2019.
- AN, J. *et al.* miRDeep*: an integrated application tool for miRNA identification from RNA sequencing data. **Nucleic acids research**, v. 41, n. 2, p. 727-737, 2013. doi:10.1093/nar/gks1187 Disponível em: <https://sourceforge.net/projects/mirdeepstar>. Acesso em: 2 abr. 2019.
- ARENAS, M. *et al.* Forensic genetics and genomics: much more than just a human affair. **PLoS genetics**, v. 13, n. 9, p. e1006960, 2017. doi:10.1371/journal.pgen.1006960.
- BAO, W.; KOJIMA, K. K.; KOHANY, O. Repbase Update, a database of repetitive elements in eukaryotic genomes. **Mobile DNA**, v. 6, n. 11, 2015. doi:10.1186/s13100-015-0041-9 Disponível em: <https://www.girinst.org/repbase>. Acesso em: 2 abr. 2019.
- BERG, J. S. *et al.* Newborn sequencing in genomic medicine and public health. **Pediatrics**, v. 139, n. 2, e20162252, 2017. doi:10.1542/peds.2016-2252.
- BERTHELIER, J. *et al.* PiRATE: a Pipeline to Retrieve and Annotate Transposable Elements. **SEANOE**, 2018. doi: 10.17882/51795 Disponível em: <https://www.seanoe.org/data/00406/51795/data/53095.ova>. Acesso em: 2 abr. 2019.
- BIRNEY, E.; CLAMP, M.; DURBIN, R. GeneWise and Genomewise. **Genome research**, v. 14, n. 5, 988-995, 2004. doi:10.1101/gr.1865504 Disponível em: <https://www.ebi.ac.uk/Tools/psa/genewise>. Acesso em: 3 abr. 2019.
- BLANCO, E.; PARRA, G.; GUIGÓ, R. Using geneid to Identify Genes. **Current Protocols in Bioinformatics**, v. 18, n. 1, p. 4.3.1-4.3.28, 2007. doi:10.1002/0471250953.bi0403s18. Disponível em: <http://genome.crg.es/software/geneid>. Acesso em: 3 abr. 2019.
- BLIN, K. *et al.* antiSMASH 4.0—improvements in chemistry prediction and gene cluster boundary identification. **Nucleic acids research**, v. 45, n. W1, p. W36-W41, 2017. Disponível em: <https://fungismash.secondarymetabolites.org>. Acesso em: 4 abr. 2019.
- BRYANT, D. M. *et al.* A tissue-mapped axolotl de novo transcriptome enables identification of limb regeneration factors. **Cell reports**, v. 18, n. 3, p. 762-776, 2017. doi:10.1016/j.celrep.2016.12.063 Disponível em: <https://trinode.github.io>. Acesso em: 4 abr. 2019.
- BUCHFINK, B.; XIE, C.; HUSON, D. H. Fast and sensitive protein alignment using DIAMOND. **Nature Methods**, v. 12 n. 1, p. 59-60, 2015. doi: 10.1038/nmeth.3176 Disponível em: <https://ab.inf.uni-tuebingen.de/software/diamond>. Acesso em: 3 abr. 2019.
- BUELS, R. *et al.* JBrowse: a dynamic web platform for genome visualization and analysis. **Genome biology**, v. 17, p. 66, 2016. doi:10.1186/s13059-016-0924-1 Disponível em: <https://jbrowse.org>. Acesso em: 4 abr. 2019.
- CAMACHO, C. *et al.* BLAST+: architecture and applications. **BMC bioinformatics**, v. 10, p. 421, 2009. doi:10.1186/1471-2105-10-421 Disponível em: <https://blast.ncbi.nlm.nih.gov/Blast.cgi>. Acesso em: 3 abr. 2019.
- CAMPBELL, M. S. *et al.* MAKER-P: a tool kit for the rapid creation, management, and quality control of plant genome annotations. **Plant physiology**, v. 164, n. 2, p. 513-524, 2013. doi:10.1104/pp.113.230144 Disponível em: <http://www.yandell-lab.org/software/maker-p.html>. Acesso em: 3 abr. 2019.

- CERQUEIRA, G. C. *et al.* The Aspergillus Genome Database: multispecies curation and incorporation of RNA-Seq data to improve structural gene annotations. **Nucleic acids research**, v. 42, n. D1, p. D705-D710, 2013. doi:10.1093/nar/gkt1029 Disponível em: <http://www.aspergillusgenome.org>. Acesso em: 4 abr. 2019.
- CONESA, A. *et al.* A survey of best practices for RNA-seq data analysis. **Genome biology**, v. 17, n. 1, p. 1-19, 2016. doi:10.1186/s13059-016-0881-8.
- CONESA, A. *et al.* Blast2GO: a universal tool for annotation, visualization and analysis in functional genomics research. **Bioinformatics**, v. 21, n. 18, p. 3674-3676, 2005. doi:10.1093/bioinformatics/bti610 Disponível em: <https://www.blast2go.com>. Acesso em: 4 abr. 2019.
- CUNNINGHAM, F. *et al.* Ensembl 2019. **Nucleic acids research**, v. 47, n. D1, p. D745-D751, 2018. doi:10.1093/nar/gky1113 Disponível em: <https://www.ensembl.org>. Acesso em: 4 abr. 2019.
- DOBIN, A. *et al.* STAR: ultrafast universal RNA-seq aligner. **Bioinformatics**, Oxford, England, v. 29, n. 1, p. 15-21, 2012. doi:10.1093/bioinformatics/bts635 Disponível em: <https://github.com/alexdobin/STAR>. Acesso em: 3 abr. 2019.
- DOMINGUEZ DEL ANGEL, V. *et al.* Ten steps to get started in Genome Assembly and Annotation. **F1000Research**, v. 7, ELIXIR, p. 148, 2018. doi:10.12688/f1000research.13598.1
- DUNN, N. A. *et al.* Apollo: Democratizing genome annotation. **PLoS computational biology**, v. 15, n. 2, e1006790, 2019. doi:10.1371/journal.pcbi.1006790 Disponível em: <http://genomearchitect.org>. Acesso em: 4 abr. 2019.
- EKBLOM, R.; WOLF, J. B. A field guide to whole-genome sequencing, assembly and annotation. **Evolutionary applications**, v. 7, n. 9, p. 1026-1042, 2014. doi:10.1111/eva.12178.
- EMANUELSSON, O. *et al.* Locating proteins in the cell using TargetP, SignalP and related tools. **Nature protocols**, v. 2, n. 4, p. 953, 2007. doi:10.1038/nprot.2007.131 Disponível em: <http://www.cbs.dtu.dk/services/TargetP>. Acesso em: 4 abr. 2019.
- EMBL-EBI *et al.* **International Protein Nomenclature Guidelines**. 2018. Disponível em: https://www.uniprot.org/docs/International_Protein_Nomenclature_Guidelines.pdf e https://www.ncbi.nlm.nih.gov/genome/doc/internatprot_nomenguide. Acesso em: 2 abr. 2019.
- FIGUEIRÓ, H. V. *et al.* Genome-wide signatures of complex introgression and adaptive evolution in the big cats. **Science Advances**, v. 3, n. 7, p. e1700299, 2017. doi:10.1126/sciadv.1700299.
- FINN, R. D.; CLEMENTS, J.; EDDY, S. R. HMMER web server: interactive sequence similarity searching. **Nucleic acids research**, v. 39, n. suppl_2, p. W29-W37, 2011. doi:10.1093/nar/gkr367 Disponível em: <http://hmmer.org>. Acesso em: 4 abr. 2019.
- FLUTRE, T. *et al.* Considering transposable element diversification in de novo annotation approaches. **PLoS one**, v. 6, n. 1, e16526, 2011. doi:10.1371/journal.pone.0016526 Disponível em: <https://urgi.versailles.inra.fr/index.php/repet>. Acesso em: 2 abr. 2019.
- GABALDÓN, T.; KOONIN, E. V. Functional and evolutionary implications of gene orthology. **Nature Reviews Genetics**, v. 14, n. 5, p. 360-366, 2013. doi:10.1038/nrg3456.
- GEIB, S. M. *et al.* Genome Annotation Generator: a simple tool for generating and correcting WGS annotation tables for NCBI submission. **GigaScience**, v. 7, n. 4, p. giy018, 2018. doi:10.1093/gigascience/giy018 Disponível em: <https://github.com/genomeannotation/GAG>. Acesso em: 4 abr. 2019.
- GOLDBERG, J. M. *et al.* Kinannotate, a computer program to identify and classify members of the eukaryotic protein kinase superfamily. **Bioinformatics**, v. 29, n. 19, p. 2387-2394, 2013. doi:10.1093/bioinformatics/btt419 Disponível em: <https://sourceforge.net/projects/kinannotate>. Acesso em: 4 abr. 2019.
- GRABHERR, M. G. *et al.* Full-length transcriptome assembly from RNA-Seq data without a reference genome. **Nature biotechnology**, v. 29, n. 7, p. 644-652, 2011. doi:10.1038/nbt.1883 Disponível em: <https://github.com/trinityrnaseq>. Acesso em: 3 abr. 2019.

- GROSS, S. S.; BRENT, M. R. Using multiple alignments to improve gene prediction. **Journal of Computational Biology**, v. 13, n. 2, 2006 doi: 10.1089/cmb.2006.13.379 Disponível em: <http://mblab.wustl.edu/software.html>. Acesso em: 3 abr. 2019.
- GUSELLA J. F. *et al.* A polymorphic DNA marker genetically linked to Huntington's disease. **Nature**, v. 306, n. 5940, p. 234, 1983. doi:10.1038/306234a0.
- HAAS, B. J. *et al.* Approaches to Fungal Genome Annotation. **Mycology**, v. 2, n. 3, p. 118-141, 2011. doi:10.1080/21501203.2011.606851.
- HAAS, B. J. *et al.* Automated eukaryotic gene structure annotation using EVIDENCEModeler and the Program to Assemble Spliced Alignments. **Genome biology**, v. 9, n. 1, p. R7, 2008. doi:10.1186/gb-2008-9-1-r7 Disponível em: <https://evidencemodeler.github.io>. Acesso em: 2 abr. 2019.
- HAAS, B. J. *et al.* Improving the Arabidopsis genome annotation using maximal transcript alignment assemblies. **Nucleic acids research**, v. 31, n. 19, 5654-5666, 2003. doi:10.1093/nar/gkg770 Disponível em: <https://github.com/PASApipeline>. Acesso em: 3 abr. 2019.
- HARIDAS, S.; SALAMOV, A.; GRIGORIEV, I.V. **Fungal Genome Annotation. Fungal Genomics: Methods in Molecular Biology** 1775, c. 15, p. 171-184, 2018. doi:10.1007/978-1-4939-7804-5_15.
- HARRISON, P. W. *et al.* The European Nucleotide Archive in 2018. **Nucleic acids research**, v. 47, n. D1, p. D84-D88, 2018. doi:10.1093/nar/gky1078 Disponível em: <https://www.ebi.ac.uk/ena>. Acesso em: 4 abr. 2019.
- HOFF, K. J. *et al.* BRAKER1: Unsupervised RNA-Seq-Based Genome Annotation with GeneMark-ET and AUGUSTUS. **Bioinformatics**, Oxford, England, v. 32, n. 5, p. 767-769, 2016. doi:10.1093/bioinformatics/btv661 Disponível em: <https://github.com/Gaius-Augustus/BRAKER>. Acesso em: 3 abr. 2019.
- HOLT, C.; YANDELL, M. MAKER2: an annotation pipeline and genome-database management tool for second-generation genome projects. **BMC bioinformatics**, v. 12, p. 491, 2011. doi:10.1186/1471-2105-12-491. Disponível em: <http://www.yandell-lab.org/software/maker.html>. Acesso em: 3 abr. 2019.
- HOOD, L.; ROWEN, L. The human genome project: big science transforms biology and medicine. **Genome medicine**, v. 5, n. 9, p. 79, 2013. doi:10.1186/gm483.
- HORTON, P. *et al.* WoLF PSORT: protein localization predictor. **Nucleic acids research**, v. 35, n. suppl_2, p. W585-W587, 2007. doi:10.1093/nar/gkm259 Disponível em: <https://wolfpsort.hgc.jp>. Acesso em: 4 abr. 2019.
- HUBLEY, R. *et al.* The Dfam database of repetitive DNA families, **Nucleic Acids Research**, v. 44, n. D1, p. D81-D89, 2016. doi:10.1093/nar/gkv1272. Disponível em: <https://dfam.org>. Acesso em: 2 abr. 2019.
- HUERTA-CEPAS, J. *et al.* eggNOG 5.0: a hierarchical, functionally and phylogenetically annotated orthology resource based on 5090 organisms and 2502 viruses. **Nucleic acids research**, v. 47, n. D1, p. D309-D314, 2018. doi:10.1093/nar/gky1085. Disponível em: <http://eggnoget.embl.de>. Acesso em: 4 abr. 2019.
- HUERTA-CEPAS, J. *et al.* Fast genome-wide functional annotation through orthology assignment by eggNOG-mapper. **Molecular biology and evolution**, v. 34, n. 8, p. 2115-2122, 2017. doi:10.1093/molbev/msx148. Disponível em: <http://eggnogetdb.embl.de/#/app/emapper>. Acesso em: 4 abr. 2019.
- HUERTA-CEPAS, J. *et al.* PhylomeDB v4: zooming into the plurality of evolutionary histories of a genome. **Nucleic acids research**, v. 42, n. D1, p. D897-D902, 2013. doi:10.1093/nar/gkt1177. Disponível em: <http://phylomedb.org>. Acesso em: 4 abr. 2019.
- JONES, P. *et al.* InterProScan 5: genome-scale protein function classification. **Bioinformatics**, v. 30, n. 9, p. 1236-1240, 2014. doi:10.1093/bioinformatics/btu031. Disponível em: <https://www.ebi.ac.uk/interpro>. Acesso em: 4 abr. 2019.
- KÄLL, L.; KROGH, A.; SONNHAMMER, E. L. L. Advantages of combined transmembrane topology and signal peptide prediction – the Phobius web server. **Nucleic acids research**, v. 35, n. suppl_2, p.

W429-W432, 2007. doi:10.1093/nar/gkm256. Disponível em: <http://phobius.sbc.su.se>. Acesso em: 4 abr. 2019.

KALVARI, I. *et al.* Rfam 13.0: shifting to a genome-centric resource for non-coding RNA families. **Nucleic acids research**, v. 46, n. D1, p. D335-D342, 2017. doi:10.1093/nar/gkx1038. Disponível em: <http://rfam.org>. Acesso em: 2 abr. 2019.

KANEHISA, M. *et al.* KEGG: new perspectives on genomes, pathways, diseases and drugs. **Nucleic acids research**, v. 45, n. D1, p. D353-D361, 2016. doi:10.1093/nar/gkw1092. Disponível em: <https://www.genome.jp/kegg/mapper.html>. Acesso em: 4 abr. 2019.

KEILWAGEN, J. *et al.* Combining RNA-seq data and homology-based gene prediction for plants, animals and fungi. **BMC bioinformatics**, v. 19, n. 1, p. 189, 2018. doi:10.1186/s12859-018-2203-5.

KENT, W. J. BLAT--the BLAST-like alignment tool. **Genome research**, v. 12, n. 4, p. 656-664, 2002. doi:10.1101/gr.229202.

KHALDI, N. *et al.* SMURF: genomic mapping of fungal secondary metabolite clusters. **Fungal Genetics and Biology**, v. 47, n. 9, p. 736-741, 2010. doi:10.1016/j.fgb.2010.06.003. Disponível em: <https://www.jcvi.org/smurf>. Acesso em: 4 abr. 2019.

KIM, D.; LANGMEAD, B.; SALZBERG, S. L. HISAT: a fast spliced aligner with low memory requirements. **Nature Methods**, v. 12, p. 357-360, 2015. doi: 10.1038/nmeth.3317. Disponível em: <https://ccb.jhu.edu/software/hisat2>. Acesso em: 3 abr. 2019.

KOEPFLI, K. P. *et al.* The Genome 10K Project: a way forward. **Annual Review of Animal Biosciences**, v. 3, n. 1, p. 57-111, 2015. doi:10.1146/annurev-animal-090414-014900. Disponível em: <https://vertebrategenomesproject.org>. Acesso em: 4 abr. 2019.

KORF, I. Gene finding in novel genomes. **BMC bioinformatics**, v. 5, n. 59, 2004. doi:10.1186/1471-2105-5-59. Disponível em: <https://github.com/KorfLab/SNAP>. Acesso em: 3 abr. 2019.

KRIVENTSEVA, E. V. *et al.* OrthoDB v10: sampling the diversity of animal, plant, fungal, protist, bacterial and viral genomes for evolutionary and functional annotations of orthologs. **Nucleic acids research**, v. 47, n. D1, p. D807-D811, 2018. doi:10.1093/nar/gky1053. Disponível em: <https://www.orthodb.org>. Acesso em: 4 abr. 2019.

KROGH, A. *et al.* Predicting transmembrane protein topology with a hidden Markov model: application to complete genomes. **Journal of molecular biology**, v. 305, n. 3, p. 567-580, 2001. doi:10.1006/jmbi.2000.4315. Disponível em: <http://www.cbs.dtu.dk/services/TMHMM>. Acesso em: 4 abr. 2019.

LAMPA, S. *et al.* Lessons learned from implementing a national infrastructure in Sweden for storage and analysis of next-generation sequencing data. **GigaScience**, v. 2, n. 1, p. 9, 2013. doi:10.1186/2047-217X-2-9.

LANGMEAD, B.; SALZBERG, S. L. Fast gapped-read alignment with Bowtie 2. **Nature methods**, v. 9, n. 4, p. 357-359, 2012. doi:10.1038/nmeth.1923. Disponível em: <http://bowtie-bio.sourceforge.net/bowtie2>. Acesso em: 3 abr. 2019.

LEROY, P. *et al.* TriAnnot: A Versatile and High Performance Pipeline for the Automated Annotation of Plant Genomes. **Frontiers in plant science**, v. 3, n. 5, 2012. doi:10.3389/fpls.2012.00005.

LI, D. *et al.* MEGAHIT: an ultra-fast single-node solution for large and complex metagenomics assembly via succinct de Bruijn graph. **Bioinformatics**, v. 31, n. 10, p. 1674-1676, 2015. doi: 10.1093/bioinformatics/btv033. Disponível em: <https://github.com/voutcn/megahit>. Acesso em: 3 abr. 2019.

LI, H. Minimap2: pairwise alignment for nucleotide sequences, **Bioinformatics**, v. 34, n. 18, p. 3094-3100, 2018. doi: 10.1093/bioinformatics/bty191. Disponível em: <https://github.com/lh3/minimap2>. Acesso em: 3 abr. 2019.

LI, H.; DURBIN, R. Fast and accurate short read alignment with Burrows-Wheeler transform. **Bioinformatics**, Oxford, England, v. 25, n. 14, p. 1754-1760, 2009. doi:10.1093/bioinformatics/btp324. Disponível em: <http://bio-bwa.sourceforge.net>. Acesso em: 3 abr. 2019.

LIU, Q. *et al.* Evigan: a hidden variable model for integrating gene evidence for eukaryotic gene prediction. **Bioinformatics**, v. 24, n. 5, p. 597-605, 2008. doi: 10.1093/bioinformatics/btn004. Disponível em: <https://www.seas.upenn.edu/~strctlrn/evigan/evigan>. Acesso em: 2 abr. 2019.

LOMBARD, V. *et al.* The carbohydrate-active enzymes database (CAZy) in 2013. **Nucleic acids research**, v. 42, n. D1, p. D490-D495, 2013. doi:10.1093/nar/gkt1178 Disponível em: <http://www.cazy.org>. Acesso em: 4 abr. 2019.

LOWE, T. M.; CHAN, P. P. tRNAscan-SE On-line: integrating search and context for analysis of transfer RNA genes. **Nucleic acids research**, v. 44, n. W1, p. W54-W57, 2016. doi:10.1093/nar/gkw413. Disponível em: <http://lowelab.ucsc.edu/tRNAscan-SE>. Acesso em: 2 abr. 2019.

MAJOROS, W. H.; PERTEA, M.; SALZBERG, S. L. TigrScan and GlimmerHMM: two open source ab initio eukaryotic gene-finders, **Bioinformatics**, v. 20, n. 16, p. 2878-2879, 2004. doi: 10.1093/bioinformatics/bth315. Disponível em: <https://ccb.jhu.edu/software/glimmerhmm>. Acesso em: 3 abr. 2019.

MARCHLER-BAUER, A.; BRYANT, S. H. CD-Search: protein domain annotations on the fly. **Nucleic acids research**, v. 32, n. suppl_2, p. W327-W331, 2004. Disponível em: <https://www.ncbi.nlm.nih.gov/Structure/cdd/wrpsb.cgi>. Acesso em: 4 abr. 2019.

MCDONNELL, E.; STRASSER, K.; TSANG, A. Manual Gene Curation and Functional Annotation. **Fungal Genomics: Methods in Molecular Biology** 1775, c. 16, p. 185-208, 2018. doi: 10.1007/978-1-4939-7804-5_16.

MENDEZ, D. *et al.* ChEMBL: towards direct deposition of bioassay data. **Nucleic acids research**, v. 47, n. D1, p. D930-D940, 2018. doi:10.1093/nar/gky1075. Disponível em: <https://www.ebi.ac.uk/chembl>. Acesso em: 4 abr. 2019.

MITCHELL, A. L. *et al.* InterPro in 2019: improving coverage, classification and access to protein sequence annotations. **Nucleic acids research**, v. 47, n. D1, p. D351-D360, 2018. doi:10.1093/nar/gky1100. Disponível em: <https://www.ebi.ac.uk/interpro>. Acesso em: 4 abr. 2019.

MORIYA, Y. *et al.* KAAS: an automatic genome annotation and pathway reconstruction server. **Nucleic acids research**, v. 35, n. suppl_2, p. W182-W185, 2007. doi:10.1093/nar/gkm321. Disponível em: <https://www.genome.jp/kegg/kaas>. Acesso em: 4 abr. 2019.

NAWROCKI, E. P.; EDDY, S. R. Infernal 1.1: 100-fold faster RNA homology searches. **Bioinformatics**, Oxford, England, v. 29, n. 22, p. 2933-2935, 2013. doi:10.1093/bioinformatics/btt509. Disponível em: <http://eddylab.org/infernal>. Acesso em: 2 abr. 2019.

NCBI. **Eukaryotic Annotated Genome Submission Guide**. 2018. Disponível em: https://www.ncbi.nlm.nih.gov/genbank/eukaryotic_genome_submission. Acesso em: 3 abr. 2019.

NICHIO, B. T. L.; MARCCHIAUKOSKI, J. N.; RAITTZ, R. T. New tools in orthology analysis: A brief review of promising perspectives. **Frontiers in genetics**, v. 8, p. 165, 2017. doi: 10.3389/fgene.2017.00165.

NIELSEN, R. *et al.* Tracing the peopling of the world through genomics. **Nature**, v. 541, n. 7637, p. 302, 2017. doi:10.1038/nature21347.

NORDBERG, H. *et al.* The genome portal of the Department of Energy Joint Genome Institute: 2014 updates. **Nucleic acids research**, v. 42, n. D1, p. D26-D31, 2013. doi:10.1093/nar/gkt1069. Disponível em: <https://genome.jgi.doe.gov/mycocosm/home> e <http://phytozome.jgi.doe.gov>. Acesso em: 4 abr. 2019.

NOVÁK, P.; NEUMANN, P.; MACAS, J. Graph-based clustering and characterization of repetitive sequences in next-generation sequencing data. **BMC bioinformatics**, v. 11, n. 1, p. 378, 2010. doi:10.1186/1471-2105-11-378. Disponível em: <http://repeatexplorer.org>. Acesso em: 2 abr. 2019.

PANDURANGAN, A. P. *et al.* The SUPERFAMILY 2.0 database: a significant proteome update and a new webserver. **Nucleic acids research**, v. 47, n. D1, p. D490-D494, 2018. doi:10.1093/nar/gky1130. Disponível em: <http://supfam.org>. Acesso em: 4 abr. 2019.

PERTEA, M. *et al.* StringTie enables improved reconstruction of a transcriptome from RNA-seq reads. **Nature biotechnology**, 33(3), 290-295, 2015. doi:10.1038/nbt.3122. Disponível em: <https://ccb.jhu.edu/software/stringtie>. Acesso em: 3 abr. 2019.

- PETERSEN, T. N. *et al.* SignalP 4.0: discriminating signal peptides from transmembrane regions. **Nature methods**, v. 8, n. 10, p. 785, 2011. doi:10.1038/nmeth.1701. Disponível em: <http://www.cbs.dtu.dk/services/SignalP>. Acesso em: 4 abr. 2019.
- PIERLEONI, A.; MARTELLI, P. L.; CASADIO, R. PredGPI: a GPI-anchor predictor. **BMC bioinformatics**, v. 9, n. 1, p. 392, 2008.
- RAWLINGS, N. D. *et al.* The MEROPS database of proteolytic enzymes, their substrates and inhibitors in 2017 and a comparison with peptidases in the PANTHER database. **Nucleic acids research**, v. 46, n. D1, p. D624-D632, 2017. doi:10.1093/nar/gkx1134. Disponível em: <https://www.ebi.ac.uk/merops>. Acesso em: 4 abr. 2019.
- REDDY, V. S.; SAIER, M. H. BioV Suite—a collection of programs for the study of transport protein evolution. **The FEBS journal**, v. 279, n. 11, p. 2036-2046, 2012. doi:10.1111/j.1742-4658.2012.08590.x. Disponível em: <http://biov.tcdb.org>. Acesso em: 4 abr. 2019.
- REID, I. *et al.* SnowyOwl: accurate prediction of fungal genes by using RNA-Seq and homology information to select among ab initio models. **BMC bioinformatics**, v. 15, p. 229, 2014. doi:10.1186/1471-2105-15-229 Disponível em: <http://sourceforge.net/projects/snowyowl>. Acesso em: 3 abr. 2019.
- ROBERTSON, G. *et al.* De novo assembly and analysis of RNA-seq data. **Nature Methods**, v. 7, p. 909-912, 2010. doi: 10.1038/nmeth.1517. Disponível em: <https://github.com/begsc/transabyss>. Acesso em: 3 abr. 2019.
- RUEDA, A. *et al.* sRNAtoolbox: an integrated collection of small RNA research tools. **Nucleic acids research**, v. 43, n. W1, p. W467-W473, 2015. doi:10.1093/nar/gkv555. Disponível em: <https://bioinfo5.ugr.es/srnatoolbox>. Acesso em: 2 abr. 2019.
- RUTHERFORD, K. *et al.* Artemis: sequence visualization and annotation. **Bioinformatics**, v. 16, n. 10, p. 944-945, 2000. doi:10.1093/bioinformatics/16.10.944. Disponível em: <https://www.sanger.ac.uk/science/tools/artemis>. Acesso em: 4 abr. 2019.
- SAYERS, E. W. *et al.* Database resources of the national center for biotechnology information. **Nucleic acids research**, v. 47, n. D1, p. D23-D28, 2018. doi:10.1093/nar/gky1069. Disponível em: <https://www.ncbi.nlm.nih.gov>. Acesso em: 4 abr. 2019.
- SCHIEX, T.; MOISAN, A.; ROUZÉ, P. Eugène: An Eukaryotic Gene Finder That Combines Several Sources of Evidence. *In: COMPUTATIONAL BIOLOGY, JOBIM 2000. Lecture Notes in Computer Science*, vol. 2066. Springer, Berlin, Heidelberg, 2001. doi: 10.1007/3-540-45727-5_10. Disponível em: <http://eugene.toulouse.inra.fr>. Acesso em: 3 abr. 2019.
- SCHULZ, M. H. *et al.* Oases: robust de novo RNA-seq assembly across the dynamic range of expression levels. **Bioinformatics**, Oxford, England, v. 28, n. 8, p. 1086-1092, 2012. doi:10.1093/bioinformatics/bts094. Disponível em: <https://www.ebi.ac.uk/~zerbino/oases>. Acesso em: 3 abr. 2019.
- SCHWEIKERT, G. *et al.* mGene.web: a web service for accurate computational gene finding. **Nucleic acids research**, v. 37, Web Server issue, p. W312-W316, 2009. doi:10.1093/nar/gkp479. Disponível em: <http://www.mgene.org>. Acesso em: 3 abr. 2019.
- SLATER, G. S.; BIRNEY, E. Automated generation of heuristics for biological sequence comparison. **BMC bioinformatics**, v. 6, p. 31, 2005. doi:10.1186/1471-2105-6-31. Disponível em: <https://www.ebi.ac.uk/about/vertebrate-genomics/software/exonerate>. Acesso em: 3 abr. 2019.
- SMIT, A.F.A.; HUBLEY, R. **RepeatModeler Open-1.0**. 2008-2015. Disponível em: <http://www.repeatmasker.org>. Acesso em: 2 abr. 2019.
- SMIT, A.F.A.; HUBLEY, R.; GREEN, P. **RepeatMasker Open-4.0**. 2013-2015. Disponível em: <http://www.repeatmasker.org>. Acesso em: Acesso em: 2 abr. 2019.

- SOLOVYEV, V. *et al.* Automatic annotation of eukaryotic genes, pseudogenes and promoters. **Genome biology**, v. 7, s. 1, p. S10.1-S10.12, 2006. doi:10.1186/gb-2006-7-s1-s10. Disponível em: <http://www.softberry.com>. Acesso em: 2 abr. 2019.
- SOLOVYEV, V. Statistical Approaches in Eukaryotic Gene Prediction. **Handbook of Statistical Genetics**, 2004. doi: 10.1002/0470022620.bbc06. Disponível em: <http://www.softberry.com>. Acesso em: 2 abr. 2019.
- SQUIZZATO, S. *et al.* The EBI Search engine: providing search and retrieval functionality for biological data from EMBL-EBI. **Nucleic acids research**, v. 43, n. W1, p. W585-W588, 2015. doi:10.1093/nar/gkv316. Disponível em: <https://www.ebi.ac.uk>. Acesso em: 4 abr. 2019.
- STANKE, M.; MORGENSTERN, B. AUGUSTUS: a web server for gene prediction in eukaryotes that allows user-defined constraints. **Nucleic acids research**, v. 33, Web Server issue, s. 2, p. W465-W467, 2005. doi:10.1093/nar/gki458. Disponível em: <http://bioinf.uni-greifswald.de/webaugustus>. Acesso em: 2 abr. 2019.
- TATUSOV, R. L. *et al.* The COG database: an updated version includes eukaryotes. **BMC bioinformatics**, v. 4, n. 1, p. 41, 2003. doi:10.1186/1471-2105-4-41.
- TER-HOVHANNISYAN, V. *et al.* Gene prediction in novel fungal genomes using an ab initio algorithm with unsupervised training. **Genome research**, v. 18, n. 12, p. 1979-1990, 2008. doi:10.1101/gr.081612.108. Disponível em: <http://exon.biology.gatech.edu/GeneMark>. Acesso em: 2 abr. 2019.
- TRAPNELL, C. *et al.* Differential gene and transcript expression analysis of RNA-seq experiments with TopHat and Cufflinks. **Nature protocols**, v. 7, n. 3, p. 562-578, 2012. doi:10.1038/nprot.2012.016. Disponível em: <http://ccb.jhu.edu/software/tophat> e <http://cole-trapnell-lab.github.io/cufflinks>. Acesso em: 3 abr. 2019.
- UNIPROT CONSORTIUM. UniProt: a worldwide hub of protein knowledge. **Nucleic acids research**, v. 47, n. D1, p. D506-D515, 2018. doi:10.1093/nar/gky1049. Disponível em: <http://www.uniprot.org>. Acesso em: 4 abr. 2019.
- VAN BAREN, M. J.; BRENT, M. R. Iterative gene prediction and pseudogene removal improves genome annotation. **Genome research**, v. 16, n. 5, p. 678-685, 2006. doi:10.1101/gr.4766206 Disponível em: <http://mblab.wustl.edu/software.html#ppfinderLink>. Acesso em: 4 abr. 2019.
- WATERHOUSE, R. M. *et al.* BUSCO applications from quality assessments to gene prediction and phylogenomics. **Molecular biology and evolution**, v. 35, n. 3, p. 543-548, 2017. doi:10.1093/molbev/msx319. Disponível em: <https://busco.ezlab.org>. Acesso em: 4 abr. 2019.
- WU, T. D.; NACU, S. Fast and SNP-tolerant detection of complex variants and splicing in short reads. **Bioinformatics**, Oxford, England, v. 26, n. 7, p. 873-881, 2010. doi:10.1093/bioinformatics/btq057. Disponível em: <http://research-pub.gene.com/gmap>. Acesso em: 3 abr. 2019.
- XIE, Y. *et al.* SOAPdenovo-Trans: de novo transcriptome assembly with short RNA-Seq reads. **Bioinformatics**, v. 30, n. 12, p. 1660-1666, 2014. doi:10.1093/bioinformatics/btu077. Disponível em: <https://github.com/aquaskyline/SOAPdenovo-Trans>. Acesso em: 3 abr. 2019.
- YANDELL, M.; ENCE, D. A beginner's guide to eukaryotic genome annotation. **Nature Reviews Genetics**, v. 13, n. 5, p. 329-342, 2012. doi:10.1038/nrg3174.
- YANG, J. H. *et al.* snoSeeker: an advanced computational package for screening of guide and orphan snoRNA genes in the human genome. **Nucleic acids research**, v. 34, n. 18, p. 5112-5123, 2006. doi:10.1093/nar/gkl672.
- YEH, R. F.; LIM, L. P.; BURGE, C. B. Computational inference of homologous gene structures in the human genome. **Genome research**, v. 11, n. 5, p. 803-816, 2001. doi:10.1101/gr.175701. Disponível em: <http://genes.mit.edu/genomescan.html>. Acesso em: 3 abr. 2019.

ZHANG, H. *et al.* dbCAN2: a meta server for automated carbohydrate-active enzyme annotation. **Nucleic acids research**, v. 46, n. W1, p. W95-W101, 2018. doi:10.1093/nar/gky418. Disponível em: <http://bcb.unl.edu/dbCAN2>. Acesso em: 4 abr. 2019.

ZHANG, Z. *et al.* PseudoPipe: an automated pseudogene identification pipeline. **Bioinformatics**, v. 22, n. 12, p. 1437-1439, 2006. doi:10.1093/bioinformatics/btl116. Disponível em: <http://www.pseudogene.org/pseudopipe>. Acesso em: 4 abr. 2019.

ZHAO, Q. Y. *et al.* Optimizing de novo transcriptome assembly from short-read RNA-Seq data: a comparative study. **BMC bioinformatics**, v. 12, Supl. 14, S2, 2011. doi:10.1186/1471-2105-12-S14-S2.

APLICAÇÕES DE COMPUTAÇÃO PARALELA E DISTRIBUÍDA EM BIOINFORMÁTICA

Clodis Boscarioli,¹ Guilherme Galante,² Luiz Antonio Rodrigues³

1 Introdução

A Bioinformática representa uma convergência de várias áreas do conhecimento que envolvem modelagem de fenômenos biológicos, genômica, biotecnologia e tecnologia da informação, análise e interpretação de dados e desenvolvimento de novos algoritmos para análise de conjuntos de dados biológicos, nos quais são empregadas diversas técnicas da Ciência da Computação (DUMANCAS, 2015).

Em geral, a Bioinformática tem três objetivos principais (LUSCOMBE *et al.*, 2001): o primeiro é organizar e armazenar os dados de forma que permita aos pesquisadores o acesso às informações existentes e o envio de novas entradas à medida que são produzidas. Embora o armazenamento de dados seja tarefa essencial, as informações guardadas nesses bancos de dados são essencialmente inúteis até serem analisadas. Nesse sentido, o segundo objetivo é desenvolver ferramentas e recursos que auxiliem na análise desses dados. Por consequência, o terceiro objetivo é, a partir de tais ferramentas, analisar os dados e interpretar os resultados de maneira biologicamente significativa.

As tarefas realizadas por essas ferramentas de Bioinformática são geralmente intensivas em Computação e utilizam grandes quantidades de memória e armazenamento, sendo quase impossível processar todos os dados sequencialmente usando uma única máquina (MERELLI, 2019a). Além disso, conjuntos de dados cada vez maiores estão se tornando comuns, e acredita-se que o tamanho só aumentará no futuro (MIKAILOV *et al.*, 2017). Um exemplo disso pode ser observado nas estatísticas apresentadas pelo GenBank, que mostram que o número de bases de dados nesse repositório praticamente dobra a cada 18 meses, desde o ano de 1982.⁴

Neste cenário, torna-se fundamental o uso de ambientes de computação de alto desempenho (*High Performance Computing*, HPC), juntamente com técnicas de paralelismo para processar todos os dados produzidos em tempo razoável e com a

¹ Universidade Estadual do Oeste do Paraná, Ciência da Computação. *E-mail*: clodis.boscarioli@unioeste.br

² Universidade Estadual do Oeste do Paraná, Ciência da Computação. *E-mail*: guilherme.galante@unioeste.br

³ Universidade Estadual do Oeste do Paraná, Ciência da Computação. *E-mail*: luiz.rodrigues@unioeste.br

⁴ Disponível em: <https://www.ncbi.nlm.nih.gov/genbank/statistics/>. Acesso em: 28 fev. 2020.

qualidade adequada (XU *et al.*, 2007). Vários projetos de Bioinformática de grande escala já se beneficiam de técnicas de paralelismo em infraestruturas de HPC, como *clusters* e grades computacionais, unidades de processamento gráfico (GPU) e, mais recentemente, das nuvens computacionais.

O objetivo deste capítulo é apresentar as principais aplicações de Computação Paralela e Distribuída em Bioinformática. A Seção 2 discute os ambientes paralelos e distribuídos mais utilizados na Bioinformática. A Seção 3 apresenta e compara ferramentas e soluções utilizadas em cada ambiente. As potencialidades e limitações da área são apresentadas na Seção 4.

2 Ambientes paralelos e distribuídos na bioinformática

A computação de alto desempenho (HPC) tem sido utilizada tradicionalmente em diversas áreas, como Física, Matemática e Engenharias, nas quais o poder computacional intensivo é necessário. Mais recentemente, a HPC passa a ser utilizada na Bioinformática, no intuito de processar as quantidades cada vez maiores de dados gerados (ALMEIDA *et al.*, 2003). Ensaios envolvendo tecnologias de sequenciamento de próxima geração, estudos de associação genômica ampla e espectrometria de massa são exemplos de aplicações que geram grandes quantidades de dados em um único experimento.

O armazenamento e a análise destes dados estão se tornando um gargalo para o seu uso efetivo, e para possibilitar novos avanços nas descobertas da área. O sequenciamento de DNA, RNA, o epigenoma, o metaboloma e o proteoma de numerosas células em milhões de indivíduos, e sequenciamento de amostras coletadas ambientalmente de milhares de locais por dia, são exemplos de aplicações que nos levarão para a escala dos *exabytes* de dados nos próximos 5-10 anos (SCHADT *et al.*, 2010). Dessa forma, faz-se indispensável o uso de arquiteturas de alto desempenho para que a massa de dados possa ser processada e os resultados possam originar em inovações nas áreas clínicas e farmacêuticas. Um bom exemplo é a descoberta de novos genes associados a diferentes doenças. Com a descoberta desses genes, os cientistas podem chegar a um entendimento mais profundo da etiologia de várias doenças de causas genéticas. Consequentemente, várias drogas e tratamentos podem ser desenvolvidos para neutralizar tais doenças (DUMANCAS, 2015).

Vários projetos de Bioinformática de grande escala já se beneficiam de técnicas de paralelismo em infraestruturas de HPC, em ambientes como *clusters*, grades, unidades de processamento gráfico (GPU) e nuvens, que têm algo em comum: o código serial não será executado mais rapidamente nesses computadores do que em qualquer

outro computador normal. O poder dos ambientes de HPC é, de fato, obtido pela possibilidade de computação paralela ou distribuída.

Nesse cenário, a Bioinformática oferece oportunidades interessantes para pesquisa em aplicações de HPC para os próximos anos. Algumas das grandes áreas da Bioinformática, ricas e complexas, relacionadas à genômica também podem se beneficiar de infraestruturas de HPC e técnicas de paralelismo, como NGS, proteômica, transcriptômica, metagenômica e bioinformática estrutural (OCAÑA; OLIVEIRA, 2015).

2.1 Dos multicores para as nuvens

Uma das formas mais básicas do uso do paralelismo é o emprego de processadores com múltiplos núcleos (multicores), nos quais as computações são distribuídas entre os vários núcleos da CPU, que compartilham um mesmo endereçamento de memória. Bibliotecas de *software*, como *Pthreads*, *OpenMP*, e linguagens de programação, como Java, oferecem suporte para a programação neste tipo de arquitetura.

Considerando a natureza paralela das pesquisas de banco de dados de sequência genética, cada cálculo de pontuação de pares é independente de todos os outros cálculos e, portanto, pode ser executado por um único núcleo de processamento. Cada parte do processamento pode ser tratado separadamente e, no final, as pontuações devem ser combinadas.

Embora a exploração do paralelismo em máquinas de memória compartilhada seja simples, a capacidade de processamento está limitada a um número pequeno de núcleos que pode não ser suficiente para as grandes demandas das aplicações. Neste sentido, pode-se interligar por meio de uma rede rápida um conjunto de computadores individuais (chamados de nós) e criar *clusters*. Teoricamente, com essa abordagem é possível obter um sistema com capacidade arbitrária, de acordo com o número de nós disponível. O paralelismo nesse tipo de arquitetura pode ser explorado pelo uso de bibliotecas e *frameworks* como MPI, *MapReduce* e *Spark* (ALNASIR; SHANARAN, 2018). Nos *clusters*, o hardware é usado de forma semelhante às máquinas *multicore*, dividindo e distribuindo o problema entre os nós e coletando os resultados (LIGHTBODY *et al.*, 2016).

Analisando o *ranking* de novembro de 2018 do TOP500,⁵ que lista as 500 máquinas com maior capacidade de processamento do mundo, 88,4% das máquinas são *clusters*. Dois *clusters* classificados entre os 10 nessa lista, Summit e Titan (1º e 9º,

⁵Disponível em: <https://www.top500.org/>. Acesso em: 28 fev. 2020.

respectivamente) pertencentes ao *Oak Ridge National Laboratory*, nos Estados Unidos, são utilizados para pesquisas biológicas, que reúnem em seu programa de biologia computacional, membros da Divisão de Biociências, Divisão de Ciências da Computação e Matemática e Divisão de Ciências Ambientais, para conduzirem pesquisas colaborativas focadas em vários aspectos da análise de genoma e proteoma e biologia de sistemas moleculares.⁶

Utilizando princípios similares aos dos *clusters*, as grades computacionais (ou *computational grids*) são uma combinação de computadores em rede fracamente acoplados (com alta latência de rede) que administrados de forma independente trabalham juntos em tarefas computacionais comuns com baixo ou nenhum custo, pois proprietários de computadores individuais oferecem seus sistemas para tais esforços (MERELLI, 2019b). A computação em grade, como uma estrutura de computação distribuída, oferece um poderoso ambiente de computação de alto desempenho, particularmente para aplicações paralelas de dados de granularidade grossa. A área da Bioinformática é uma que pode ser facilmente adequada aos ambientes de grade, uma vez que as aplicações são geralmente paralelas e toleram as grandes latências (KRISHNAN, 2005). Grades como EGI,⁷ Opensciencegrid⁸ e CSGRID⁹ são exemplos de infraestruturas utilizadas para a realização de experimentos biológicos.

Um exemplo de *middleware* que dá suporte às aplicações fracamente acopladas nas plataformas de grade é o *Berkeley Open Infrastructure for Network Computing* (BOINC), que se baseia na abordagem de computação voluntária, no qual cada computador voluntário escolhe uma parte do problema para processar, baixa os dados do servidor, resolve o problema e envia os resultados para o servidor. A computação em grade baseada em BOINC pode fornecer a capacidade computacional necessária a muitas análises de bioinformática, como apresentado no trabalho de Pinthong *et al.* (2016), que implementa a ferramenta de alinhamento de sequências BLAST usando essa abordagem. Aplicações de Bioinformática também foram portadas para a plataforma BOINC no estilo “@home”, como Rosetta@home, para predição de estruturas de proteínas¹⁰ e DOCKING@home,¹¹ para acoplamento simulações de *docking* de proteínas.

Uma outra abordagem de HPC, neste caso não distribuída, tem ganhado atenção em diversas áreas do conhecimento, que é o uso de Unidades de Processamento Gráfico

⁶ Disponível em: <https://www.olcf.ornl.gov/leadership-science/biology/>. Acesso em: 28 fev. 2020.

⁷ Disponível em: <https://www.egi.eu/>. Acesso em: 28 fev. 2020.

⁸ Disponível em: <https://opensciencegrid.org/docs/>. Acesso em: 28 fev. 2020.

⁹ Disponível em: <https://jira.tecgraf.puc-rio.br/confluence/display/CN/CSGrid+Overview>. Acesso em: 28 fev. 2020.

¹⁰ Disponível em: <https://boinc.bakerlab.org/>. Acesso em: 28 fev. 2020.

¹¹ Disponível em: <http://docking.cis.udel.edu/>. Acesso em: 28 fev. 2020.

(GPU). As GPU são coprocessadores paralelos de múltiplos núcleos e extremamente eficientes em termos de energia e de baixo custo para a obtenção de ganhos significativos de desempenho, tanto que grandes *clusters* estão adotando o uso desses dispositivos como forma de acelerar partes das aplicações que estão executando (NOBILE *et al.*, 2016). Desde junho de 2011, as GPU estão presentes nos supercomputadores do TOP 500, sendo que, da edição de novembro de 2018, cerca de 130 máquinas possuem GPU.

A arquitetura *Compute Unified Device Architecture* (CUDA) da Nvidia é o padrão de *facto* mais utilizado para o desenvolvimento de ferramentas baseadas em GPU nas áreas de Bioinformática, Biologia Computacional e Biologia de Sistemas. A CUDA só pode explorar as GPU da Nvidia, mas existem soluções alternativas, como o Microsoft DirectCompute (Microsoft Windows) e a biblioteca independente de plataforma OpenCL.

Nobile *et al.* (2016) é um bom ponto de partida para o estudo do uso de GPU na Bioinformática, uma vez que os autores revisam o estado da arte do uso de GPU na área da Bioinformática, Biologia Computacional e Biologia de Sistemas, e ainda apresentam uma coleção de ferramentas desenvolvidas para realizar análises computacionais em disciplinas de ciências da vida, enfatizando as vantagens e desvantagens no uso dessa arquitetura paralela.

herdando algumas características e funcionalidades das arquiteturas apresentadas, a computação em nuvem surgiu como uma alternativa para resolver problemas de computação científica, com a promessa de provisionar recursos virtualmente infinitos. Esse modelo de computação oferece aos usuários finais uma variedade de recursos, desde o *hardware* até o nível da aplicação, cobrando-os com base em pagamento por uso (*pay-per-use*), permitindo acesso imediato aos recursos necessários e sem a necessidade de aquisição de infraestrutura adicional (GALANTE *et al.*, 2016).

Por meio da computação em nuvem, máquinas com grande poder de processamento podem ser adquiridas em regime de aluguel, dependendo dos requisitos e do uso do usuário. Ao instanciar um conjunto de máquinas virtuais, um *cluster* pode ser implantado sob demanda, no qual abordagens de exploração de paralelismo, tais como a MPI, MapReduce e BOINC, podem ser utilizadas. Também é comum que provedores ofereçam tipos de instâncias virtuais que já agregam GPU em suas configurações. A flexibilidade e o custo/benefício proporcionados pela computação em nuvem são extremamente atraentes à biologia computacional, em particular para laboratórios de biotecnologia de pequeno e médio porte que precisam realizar análises de bioinformática sem abordar todas as questões de ter uma infraestrutura interna de computação (PERÉZ-SÁNCHEZ; CECILIA; MERELLI, 2014).

Os dados e sistemas necessários para realizar a pesquisa também podem ser colocados na nuvem e acessados como um serviço, de acordo com as demandas e necessidades. Soluções de dados como serviço (DaaS) oferecem acesso dinâmico aos dados sob demanda e disponibilizam os dados mais recentes disponíveis. Um exemplo é representado pelo DaaS da Amazon *Web Services* (AWS),¹² que fornece um repositório centralizado de conjunto de dados públicos, como Ensembl, 1000 Data Genome, UniGene e Freebase, no qual os dados podem ser compartilhados e integrados em aplicações desenvolvidas para a própria nuvem da Amazon.

Também já é possível encontrar na literatura vários esforços para desenvolver ferramentas baseadas na nuvem para executar diferentes tarefas de Bioinformática, como, aplicações genômicas, alinhamento de sequências e análise de expressão gênica (SHAKIL; ALAM, 2018). Com isso, todo o processo de instalação de *software* passa a ser desnecessário e o cientista não precisa mais desenvolver conhecimento especializado para instalar sistemas operacionais, bibliotecas, *software*, etc. Pode-se apenas realizar o acesso na respectiva nuvem e usar o *software* exigido por eles, disponibilizado sem nenhuma configuração inicial. Nesse modelo, nenhuma preocupação com a infraestrutura é necessária, pois o provedor da nuvem se responsabiliza por fornecer os recursos para dar suporte às aplicações.

3 Ferramentas e aplicações

A evolução das aplicações paralelas e distribuídas é muito próxima da evolução dos sistemas e do *hardware*, variando desde as primeiras soluções em grades computacionais, que visavam a aproveitar os recursos de processamento ociosos, até a utilização de poderosos supercomputadores, construídos especificamente para a computação de alto desempenho. Sistemas *multicore* e *hardware* específico para processamento paralelo complementam o rol de ferramentas e aplicações. Esta seção apresenta exemplos de soluções nessas diversas categorias.

3.1 Sistemas multicore

O *Parallel Association Rules Extractor from SNPs* (PARES) (AGAPITO; GUZZI; CANNATARO, 2019) permite extrair os fatores responsáveis para o desenvolvimento de doenças multifatoriais por meio de análises em sequências genômicas utilizando o algoritmo *Frequent Pattern Growth* (FP-Growth). A ferramenta, desenvolvida em Java, utiliza um modelo mestre-escravo com múltiplas *threads* para executar tarefas paralelas em sistemas multiprocessados. Dado um computador com n

¹² Disponível em: <https://registry.opendata.aws/>. Acesso em: 28 fev. 2020.

núcleos, o sistema cria uma *thread* de controle, responsável pelo pré-processamento, particionamento e distribuição de carga entre as $n-1$ *threads* de processamento, e coleta dos resultados. Inicialmente, a *thread* de controle transforma o conjunto de dados de entrada em um conjunto de transações, filtrando as estatisticamente insignificantes por meio do Teste de Fisher, um teste de significância estatística na análise de tabelas de contingência, para testar a hipótese de duas variáveis. Após a etapa de filtragem, a tabela resultante é transformada em um banco de dados de transações. O próximo passo é remover as dependências entre transações para criar tarefas totalmente independentes, que serão executadas pelas *threads* de processamento.

Parallelized Split Merge Sampling on Dirichlet Process Mixture Model (ParDPMM) (DUAN; PINTO; XIE, 2019) propõe uma solução paralela para agrupamento de células utilizando OpenMP. Uma estratégia de dividir e agrupar é empregada para melhorar a convergência e a eficácia do resultado.

A disponibilidade da plataforma multiprocessada Xeon Phi a partir de 2012 (baseada na arquitetura Intel *Many Integrated Core*, MIC), é explorada por Liu e Schmidt (2014) para aumentar a eficiência do algoritmo Smith-Waterman. Outras soluções na mesma linha incluem XSW (WANG, 2014), SWAPHI (LIU; SCHMIDT, 2014) e SWIMM (RUCCI *et al.*, 2015) e SWIMM 2.0 (RUCCI *et al.*, 2019). SWAPHILS, XSW e SWAPHI apresentaram desempenho de 30, 62 e 70 GCUPS, respectivamente. SWIMM 2.0, utilizando conjunto de instruções AVX2 (*Advanced Vector Extensions 2*) e capacidades mais amplas de processamento de vetores alcançou 511 GCUPS em um único *Intel's Knights Landing* (KNL).

3.2 Sistemas de grades computacionais

Um dos grandes desafios da análise genética e da biologia molecular é a análise filogenética de famílias de genes individuais para a reconstrução da árvore da vida, isto é, do relacionamento evolutivo entre espécies ou entre indivíduos da mesma espécie. A máxima verossimilhança (*Maximum likelihood*) é um dos métodos mais precisos para este tipo de análise. MultiPhyl (KEANE; NAUGHTON; MCINERNEY, 2007) foi a primeira solução de plataforma de filogenética distribuída apresentada, com dezenas de modelos e métodos estatísticos. A proposta era usar o processamento ocioso de computadores voluntários para realizar tarefas distribuídas em um modelo cliente-servidor. Em um formato de *webserver*, o usuário submetia a tarefa via portal web e recebia o resultado por *e-mail*. Os cálculos eram feitos utilizando a biblioteca *Phylogenetic Analysis Library* (PAL), escrita na linguagem de programação Java.

Distributed phylogeny reconstruction by maximum likelihood (DPRml) (KEANE *et al.*, 2005) utiliza uma solução semelhante para grades computacionais. No primeiro

passo, um processo constrói uma árvore inicial durante um determinado período, sendo 30 minutos o tempo padrão. Em seguida, o conjunto de árvores geradas é distribuído como tarefas entre as demais máquinas do sistema. As tarefas são executadas com baixa prioridade em cada máquina, visando a aproveitar apenas recursos ociosos sem atrapalhar as demais atividades dos usuários.

3.3 Sistemas de clusters de computadores

A busca por sequências em cadeias é uma das tarefas mais triviais em Bioinformática. Dois algoritmos conhecidos neste segmento são Needleman-Wunsch (NEEDLEMAN; WUNSCH, 1970) e Smith-Waterman (SMITH; WATERMAN, 1981). No entanto, estes algoritmos são extremamente custosos para grandes volumes de dados. Assim, soluções paralelas e distribuídas são comumente encontradas. DSEARCH (KEANE; NAUGHTON, 2005), por exemplo, é uma plataforma distribuída, escrita em Java, que utiliza o paradigma mestre-escravo e a biblioteca NeoBio¹³ para realizar os cálculos em blocos de dados distribuídos entre os escravos.

O trabalho de Nowicki, Bzhalava e Bała (2018) utiliza a biblioteca PCJ (*Parallel Computing in Java* (PCJ)),¹⁴ para implementar a divisão de tarefas e execução das buscas para problemas de sequenciamento genético com base no pacote NCBI-BLAST, do *National Center for Biotechnology Information*. De acordo com seus autores, PCJ permite a execução da aplicação em diversas plataformas paralelas com escalabilidade para até 200 mil *cores*.

3.4 Soluções em GPU (*Graphics Processing Unit*)

O trabalho de Liu *et al.* (2007) foi pioneiro em utilizar GPU e a API OpenGL para solucionar o problema de alinhamento de sequências genéticas. Com o surgimento da plataforma CUDA da Nvidia, diversas outras soluções foram propostas, em especial as da série CUDASW++, cuja versão 3.0 (LIU; WIRAWAN; SCHMIDT, 2013) ainda é considerada o estado da arte em implementações para GPU e utiliza técnicas avançadas de alinhamento de dados, juntamente com as instruções *Single instruction multiple data* (SIMD), para obter o máximo de desempenho. Os testes utilizando o banco de dados Swiss-Prot obtiveram um aumento de desempenho de 2,9 e 3,2 sobre a versão 2.0 alcançando 119.0 e 185.6 GCUPS (*Giga Cell Updates Per Second*) nos modelos *single-GPU* GeForce GTX 680 e *dual-GPU* GeForce GTX 690, respectivamente. A solução

¹³ Disponível em: <http://www.neobio.sourceforge.net>. Acesso em: 28 fev. 2020.

¹⁴ Disponível em: <https://github.com/hpdcj/PCJ>. Acesso em: 28 fev. 2020.

também conseguiu bons resultados na comparação com as bibliotecas SWIPE e BLAST+.

3.5 Soluções em nuvens computacionais

O uso das nuvens na execução de aplicações de bioinformática pode variar desde ser feito de diversas formas, desde uma única ferramenta sendo executada em uma máquina virtual, até *workflows* e *pipelines* que se utilizam de *clusters* formados por conjuntos de instâncias na nuvem (NAVALE; BOURNE, 2018).

O *Basic Local Alignment Search Tool* (BLAST) é um exemplo de ferramenta que está disponível na nuvem. Imagens do servidor BLAST estão disponíveis nos repositórios (*marketplaces*) das principais nuvens públicas, como AWS, Azure e GCP, permitindo que os usuários possam instanciar máquinas virtuais com o ambiente já configurado e pronto para uso. Os usuários também podem executar o BLAST diretamente através de uma plataforma disponibilizada pelo *National Center for Biotechnology Information* (NCBI). O servidor NCBI BLAST destina-se principalmente ao uso interativo, com a expectativa de que os usuários executem um número moderado de pesquisas, entre 10 e 20, por dia.¹⁵

Wang *et al.* (2018) desenvolveram o SciApps, uma plataforma baseada na web para *workflows* de bioinformática. A plataforma foi projetada para automatizar a execução de aplicações e suportar a execução de *workflows* em inúmeras máquinas de um *cluster* local ou na nuvem. O *workflow* apresentado como estudo de caso, implementa um pipeline de anotação iterativa de três passos com dois aplicativos, MAKER, um *pipeline* portátil de anotação de genoma com um conjunto integrado de ferramentas de previsão de genes e SNAP, um analisador de ácido nucleico baseado em cadeias ocultas de Markov.

Outro exemplo de plataforma para a execução de *workflows* de bioinformática é apresentado por Novella *et al.* (2019). A plataforma Pachyderm utiliza-se de contêineres Docker e Kubernetes com o intuito de permitir a execução paralela das aplicações, prometendo uma boa escalabilidade, e garantindo interoperabilidade e reprodutibilidade por meio da containerização das ferramentas. O trabalho apresenta estudos de caso na área de metabolômica para a avaliação da plataforma.

Guo *et al.* (2018) apresentam um *survey* sobre o uso de aplicações Spark em ambientes de *cluster* e nuvem para o sequenciamento de próxima geração e outros domínios biológicos, como epigenética, filogenia e descoberta de drogas, fornecendo

¹⁵ Disponível em: <https://ncbi.github.io/blast-cloud/>. Acesso em: 28 fev. 2020.

diretrizes para permitir aos pesquisadores de bioinformática aplicar o Spark em seus próprios campos.

4 Potencialidades e limitações

Em geral, as soluções de bioinformática precisam lidar com grandes quantidades de dados e longos períodos de processamento. Um dos principais desafios é justamente a otimização dos algoritmos para diminuir o tempo de execução dos processos. Por possuírem alta complexidade computacional, técnicas de indexação, estruturas hierárquicas e filtragem são indicadas para a otimização das soluções (YIN *et al.*, 2017).

HPC tem sido a estratégia mais utilizada para diminuir o tempo de obtenção dos resultados. No entanto, as soluções paralelas costumam ser altamente dependentes da arquitetura em que são executadas, especialmente em questões de memória e transferência de dados. Neste sentido, há um vasto campo de pesquisa para a adaptação dos algoritmos de acordo com as especificidades de cada arquitetura, podendo reduzir significativamente o seu tempo de execução, como é o caso das soluções baseadas em GPU e Xeon Phi, e das *Field Programmable Gate Array* (FPGA), que são circuitos integrados programáveis. Liu *et al.* (2014) e Di Tucci *et al.* (2017) são exemplos do uso destas tecnologias em bioinformática. No entanto, estas otimizações geralmente exigem adaptações consideráveis no algoritmo sequencial, especialmente pela natureza das soluções em bioinformática, que lidam com estruturas de dados irregulares, como árvores de sufixo, matrizes esparsas e representação em grafos. Xeon Phi, por exemplo, possui unidades de processamento vetorial de 512-bits, mas capacidade limitada de memória, exigindo que as soluções sejam adaptadas para o melhor uso do *hardware* (RUCCI *et al.*, 2019).

Outro grande desafio para a adaptação das soluções em ambientes paralelos e distribuídos é a solução/diminuição de dependências entre as tarefas, evitando grandes trocas de informações entre os processos e o bloqueio de tarefas mais rápidas em função daquelas mais complexas, que consomem mais tempo de processamento. O alinhamento de sequências, por exemplo, costuma utilizar estruturas de dados com até 2 GB para um único genoma humano. O uso de memória compartilhada (*Pthreads* e *OpenMP*) pode ser adequado para este tipo de problema. Por outro lado, o surgimento de novas arquiteturas *multicore* e linguagens específicas, como Intel Cilk Plus, podem ser mais eficientes por explorarem melhor as especificidades do *hardware* comprometendo, entretanto, a portabilidade.

A proliferação da computação em nuvem tem direcionado as soluções para este tipo de ambiente, especialmente para o uso de MapReduce. No entanto, a representação

de grafos no sequenciamento de genoma, por exemplo, não se adéqua naturalmente ao paralelismo de dados dos modelos. Neste sentido, novos *frameworks*, como Pregel (MALEWICZ *et al.*, 2010) e DiGraph (YU *et al.*, 2019), têm tentado explorar o processamento distribuído de grafos.

Outra abordagem que tem se mostrado promissora na nuvem é o uso de contêineres orquestrados via Kubernetes. Essa abordagem permite que os usuários criem e empacotem *software* em contêineres sem levar em consideração a arquitetura das máquinas subjacentes, facilitando o desenvolvimento e a implementação de aplicações e *workflow* de bioinformática (NOVELLA *et al.*, 2019). Além disso, vários provedores de nuvem dão suporte nativo a essa tecnologia.

Bioinformática é uma área emergente em franca expansão, que carece ainda de muita pesquisa e aplicações na área computacional para seu desenvolvimento. Há um crescimento exponencial na quantidade de dados biológicos produzidos, que por sua vez estão espalhados geograficamente em vários laboratórios e repositórios de dados, e existe um grande potencial para os usuários finais e pesquisadores; e a gestão dessa informação heterogênea produzida em grandes volumes de dados pela área de bioinformática requer o desenvolvimento de ambientes distribuídos para seu processamento, capazes de prover serviços eficientes de comparação de sequências, disponibilizando também ferramentas auxiliares de anotação, para facilitar o trabalho do pesquisador da área.

Referências

AGAPITO, G.; GUZZI, P. H.; CANNATARO, M. Parallel extraction of association rules from genomics data. **Applied Mathematics and Computation**, v. 350, p. 434-446, 2019.

ALMEIDA, N. F.; ALVES, C. E. R.; CÁCERES E. N.; SONG, S. W. Comparison of genomes using high-performance parallel computing. *In: Proceedings of 15th Symposium on Computer Architecture and High Performance Computing*. 2003.

ALNASIR, J. J.; SHANAHAN, H. P. The application of Hadoop in structural bioinformatics. **Briefings in Bioinformatics**. v. 0, bby106, p. 1-10. 2018.

DI TUCCI, L.; O'BRIEN, K.; BLOTT, M; SANTAMBROGIO, M. D. Architectural optimizations for high performance and energy efficient Smith-Waterman implementation on FPGAs using OpenCL, **Proceedings of Design Automation & Test in Europe Conference & Exhibition (DATE)**, p. 716-721, 2017.

DUAN, T.; PINTO, J. P.; XIE, X. Parallel clustering of single cell transcriptomic data with split-merge sampling on Dirichlet process mixtures, **Bioinformatics**, v. 35, Issue 6, p. 953-961. 2019.

DUMANCAS, G. G. Applications of supercomputers in sequence analysis and genome annotation. *In: SEGALL, R. S.; COOK, J. S.; ZHANG Q. (org.) Research and applications in global supercomputing*. Hershey, PA, USA: IGI Global, 2015. p.149-175.

GALANTE, G.; BONA, L. C. E.; MURY, A. R. *et al.* An analysis of public clouds elasticity in the execution of scientific applications: a survey. **Journal of Grid Computing**, v. 14, p. 193-216, 2016.

- GUO, R.; ZHAO, Y.; ZOU, Q. *et al.* Bioinformatics applications on apache spark, **GigaScience**, v. 7, n. 8, p. giy098, 2018.
- KEANE, T. M.; NAUGHTON, T. J.; MCINERNEY, J. O. MultiPhyl: a high-throughput phylogenomics webserver using distributed computing, **Nucleic Acids Research**, v. 35, Issue suppl_2, July, p. W33-W37, 2007.
- KEANE, T. M.; NAUGHTON, T. J. DSEARCH: sensitive database searching using distributed computing, **Bioinformatics**, v. 21, Issue 8, p. 1705-1706, 2005.
- KEANE, T. M.; NAUGHTON, T. J.; TRAVERS, S. A. A.; MCINERNEY, J. O.; MCCORMACK, G. P. DPRml: distributed phylogeny reconstruction by maximum likelihood, **Bioinformatics**, v. 21, Issue 7, p. 969-974, 2005.
- KRISHNAN, A. GridBLAST: a globus-based high-throughput implementation of blast in a grid computing framework. **Concurrency and Computation: Practice and Experience**, v. 17, n. 13, p. 1607-1623, 2005.
- LIGHTBODY, G.; BROWNE, F.; ZHENG, H. HABERLAND, BLAYNEY, V. J. The role of high performance, grid and cloud computing in high-throughput sequencing. *In*: 2016 IEEE INTERNATIONAL CONFERENCE ON BIOINFORMATICS AND BIOMEDICINE (BIBM). **Proceedings of 2016 IEEE International Conference on Bioinformatics and Biomedicine**, 2016.
- LIU, Y.; B. SCHMIDT, B. SWAPHI: Smith-waterman protein database search on Xeon Phi coprocessors. **Proceedings of IEEE 25th International Conference on Application-Specific Systems, Architectures and Processors**, Zurich, p. 184-185, 2014.
- LIU, W.; SCHMIDT, B.; VOSS, G.; MÜLLER-WITTIG, W. Streaming algorithms for biological sequence alignment on GPUs. **IEEE Trans Parallel Distrib Syst**, v. 18, n. 9, p. 1270-1281, 2007.
- LIU, Y.; TRAN, T.; LAUENROTH, F.; SCHMIDT, B. SWAPHI-LS: Smith-Waterman Algorithm on Xeon Phi coprocessors for Long DNA Sequences, **Proceedings of IEEE International Conference on Cluster Computing (CLUSTER)**, p. 257-265, 2014.
- LIU, Y.; WIRAWAN, A.; SCHMIDT, B. CUDASW++ 3.0: accelerating Smith-Waterman protein database search by coupling CPU and GPU SIMD instructions. **BMC Bioinforma**, v.14, n. 117, p. 1-10, 2013.
- LUSCOMBE, N. M.; GREENBAUM, D.; GERSTEIN, M. What is bioinformatics? A proposed definition and overview of the field. **Methods of Information in Medicine**, v. 40, n.4, p. 346-358, 2001.
- MALEWICZ, G.; AUSTERN, M. H.; BIK, A. J.; DEHNERT, J. C.; HORN, I.; LEISER, N.; *et al.* Pregel: a system for large-scale graph processing. **Proceedings of the 2010 ACM SIGMOD International Conference on Management of Data**, p. 135-146, 2010.
- MERELLI, I. Parallel architectures for bioinformatics. *In*: RANGANATHAN, S. *et al.* (org.) **Encyclopedia of Bioinformatics and Computational Biology**. Cambridge, MA, USA: Elsevier, p. 209-214, 2019a.
- MERELLI, I. Infrastructure for high-performance computing: grids and grid computing. *In*: RANGANATHAN, S. *et al.* (org.) **Encyclopedia of Bioinformatics and Computational Biology**. Cambridge, MA, USA: Elsevier, p. 230-235. 2019b.
- MIKAILOV M.; LUO F. J.; BARKLEY S.; *et al.* Scaling bioinformatics applications on HPC. **BMC Bioinformatics**, v. 18, Supplement 14, p. 164-169, 2017.
- NAVALE, V.; BOURNE, P. E. Cloud computing applications for biomedical science: a perspective. **PLoS Computatioanl Biology**, v. 14, n. 6, p. e1006144, 2018.
- NEEDLEMAN, S.; WUNSCH, C. A general method applicable to the search for similarities in the amino acid sequences of two proteins. **Journal of Molecular Biology**, v. 48, p. 443-453, 1970.
- NOBILE, M. S.; CAZZANIGA, P.; TANGHERLONI, A.; BESOZZI, D. Graphics processing units in bioinformatics, computational biology and systems biology. **Briefings in bioinformatics**, v. 18, n.5, p. 870-885, 2016.

- NOVELLA, J. A.; KHOONSARI, P. E.; HERMAN, S. *et al.* Container-based bioinformatics with Pachyderm. **Bioinformatics**, v. 35, n. 5, p. 839-846, 2019.
- NOWICKI, M.; BZHALAVA, D.; BAŁA, P. Massively parallel implementation of sequence alignment with basic local alignment search tool using parallel computing in java library. **Journal of Computational Biology**, v. 25, n. 8, p. 871-881, 2018.
- OCAÑA K.; OLIVEIRA D. Parallel computing in genomic research: advances and applications. **Advances and Applications in Bioinformatics and Chemistry**, v. 8, p. 23-35, 2015.
- PERÉZ-SÁNCHEZ, H.; CECILIA, J. M.; MERELLI, I. The role of high-performance computing in bioinformatics. *In: 2ND INTERNATIONAL WORK-CONFERENCE ON BIOINFORMATICS AND BIOMEDICAL ENGINEERING. Proceedings of 2nd International Work-Conference on Bioinformatics and Biomedical Engineering*, 2014.
- PINTHONG, W.; MUANGRUEN, P.; SURIYAPHOL, P.; MAIRIANG, D. A simple grid implementation with Berkeley Open Infrastructure for Network Computing using BLAST as a model. **PeerJ**, v. 4, p. e2248, 2016.
- RUCCI, E.; GARCÍA, C.; BOTELLA, G.; DE GIUSTI, A.; NAIOUF, M.; PRIETO-MATÍAS, M. **An energy-aware performance analysis of SWIMM: Smith-Waterman implementation on Intel's multicore and manycore architectures. Concurrency Comput Pract Experience**, v. 27, n. 18, p. 5517-5537, 2015.
- RUCCI, E.; SANCHEZ, C. G.; GUILLERMO BOTELLA JUAN, ARMANDO DE GIUSTI, MARCELO NAIOUF, MANUEL PRIETO-MATIAS. SWIMM 2.0: Enhanced Smith-Waterman on Intel's Multicore and Manycore Architectures Based on AVX-512 Vector Extensions. **Int J Parallel Programming**, v. 47, n. 2. p. 47: 296, 2019.
- SCHADT, E. E; LINDERMAN, M. D.; SORENSON, J.; LEE, L.; NOLAN, G. P. Computational solutions to large-scale data management and analysis. **Nature Reviews Genetics**. v. 11, p. 647-657, 2010.
- SHAKIL, K. A.; ALAM, M. Cloud computing in bioinformatics and big data analytics: current status and future research. *In: AGGARWAL V.; BHATNAGAR V.; MISHRA D. (org.). Big Data Analytics. Advances in Intelligent Systems and Computing*. Singapore: Springer, 2018. p. 629-640. v. 654.
- SMITH, T; WATERMAN, M. Identification of common molecular subsequences. **Journal of Molecular Biology**, v. 147, p. 195-197, 1981.
- WANG, L.; CHAN, Y.; DUAN, X.; LAN, H.; MENG, X.; LIU W. XSW: accelerating biological database search on Xeon Phi. **Proceedings of IEEE International Parallel & distributed processing symposium workshops (IPDPSW)**, p. 950-957, 2014.
- WANG, L.; LU, Z.; VAN BUREN, P.; WARE, D. SciApps: a cloud-based platform for reproducible bioinformatics workflows. **Bioinformatics**, v. 34, n. 22, p. 3917-3920, 2018.
- XU, G.; LU, F.; YU, H.; XU, Z. A distributed parallel computing environment for bioinformatics problems. *In: 6TH INTERNATIONAL CONFERENCE ON GRID AND COOPERATIVE COMPUTING. Proceedings of Sixth International Conference on Grid and Cooperative Computing*, 2007.
- YIN, Z.; LAN, H.; TAN, G.; LU, M.; VASILAKOS, A. V.; LIU, W. Computing platforms for big biological data analytics: perspectives and challenges. **Computational and Structural Biotechnology Journal**, v. 15, p. 403-411, 2017.

