Um guia prático de aplicações em BIOINFORMÁTICA

DA BIOLOGIA À BIOTECNOLOGIA

Organizadores:
Pedro Lenz Casa e Scheila de Avila e Silva

EDUCS PESQUISA

DA BIOLOGIA À BIOTECNOLOGIA

DA BIOLOGIA À BIOTECNOLOGIA

Organizadores:
Pedro Lenz Casa e Scheila de Avila e Silva

EDUCS
PESQUISA

1ª edição: 2023
Preparação de texto: Giovana Letícia Reolon
Editoração: Ana Carolina Marques Ramos
Capa: Sabrina Danielli Dani

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

D111 Da biologia à biotecnologia [recurso eletrônico] : um guia prático de aplicações de bioinformática / organizadores Pedro Lenz Casa, Scheila de Avila e Silva. – Caxias do Sul : Educs, 2023.
Dados eletrônicos (1 arquivo)

Apresenta bibliografia.
Vários autores.
Modo de acesso: World Wide Web.
DOI 10.18226/9786558072959
ISBN 978-65-5807-295-9

1. Bioinformática. 2. Biologia. 3. Biotecnologia. I. Casa, Pedro Lenz. II. Silva, Scheila de Avila.

CDU 2. ed.: 57:004

Índice para o catálogo sistemático

1. Bioinformática	57:004
2. Biologia	573
3. Biotecnologia	60

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

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

Sumário

Introdução

Interface entre biologia, bioinformática e biotecnologia

Pedro Lenz Casa[1]

A biotecnologia pode ser sintetizada como o emprego de organismos, células ou moléculas derivadas de seres vivos para o desenvolvimento de novas tecnologias e produtos. Essa área do conhecimento já se encontra difundida em diversos ramos das Ciências da Vida, possuindo aplicações principalmente nas áreas da medicina, da agricultura, da biologia marinha e das ciências ambientais, bem como em inúmeros setores industriais (Gupta *et al*., 2017). Nesse contexto, fica aparente que os genes e os mecanismos de regulação presentes nos organismos de interesse, conhecimentos provenientes da biologia molecular, constituem uma base primordial para a pesquisa em biotecnologia.

O progresso do conhecimento científico acerca da biologia molecular aliado a novas tecnologias para sequenciamento de DNA e engenharia genética têm permitido inúmeras descobertas e conquistas no âmbito científico. Dentre elas, podemos mencionar a técnica de DNA recombinante, que permite a união de sequências de DNA de diferentes organismos. Essa metodologia é amplamente utilizada para a produção comercial de insulina a partir da adição do gene humano produtor de insulina ao plasmídeo da bactéria *Escherichia coli*.

Concomitante aos avanços da biologia molecular nas últimas décadas, uma variedade de tecnologias computacionais e métodos de análise foi desenvolvida em ritmo acelerado, o que possibilitou a interpretação da grande quantidade de dados brutos gerados pelas Ciências da Vida. De fato, o grande

[1] Laboratório de Bioinformática e Biologia Computacional, Instituto de Biotecnologia, Universidade de Caxias do Sul.

volume de dados biológicos provenientes de sequenciamento foi acompanhado de consecutivos avanços computacionais. Essas revoluções tecnológicas culminaram no crescimento de dois grandes campos de estudo, a genômica e a bioinformática (Diniz; Canduri, 2017; Gauthier *et al*., 2019).

O primeiro desses compreende um campo de pesquisa que busca a caracterização e a exploração da estrutura, da função e da evolução de genomas. Para esse objetivo, a genômica se apoia na utilização de ferramentas provenientes da bioinformática, tanto para a montagem e anotação[2] quanto para a investigação e comparação de genomas (De Carvalho *et al*., 2019). Além disso, uma explosão de diversas outras áreas surgem a partir da genômica e seu objeto de estudo.

O conjunto formado por essas vertentes é conhecido como "ômicas" e inclui a transcriptômica, a proteômica, a metabolômica, a filogenômica, entre outras. Embora cada uma dessas possua um foco de estudo biomolecular distinto, os processos biológicos em questão geralmente envolvem interações entre os diferentes níveis ômicos. No momento em que cada ômica é inserida no contexto do dogma central da biologia molecular, no qual cada uma representa um passo diferente e ao mesmo tempo complementar no fluxo de informação genética, a importância de análises integradas pode ser diretamente percebida. Embora esse seja um problema complexo com diversos desafios a serem superados (*e.g.* heterogeneidade dos bancos de dados biológicos e padronização de experimentos e procedimentos de validação de modelos), o planejamento de análises ômicas integradas apresenta grande relevância biológica (Haas *et al*., 2017; Gross; Macleod, 2017).

Já o segundo campo mencionado, a bioinformática, compreende o uso de meios computacionais para o estudo de dados de origem biológica, além de ser considerado um avanço relativamente recente para a pesquisa na área das Ciências da

[2] Montagem e anotação genômica compreendem procedimentos fundamentais que visam, respectivamente, à determinação da sequência de DNA de um organismo e á busca por elementos funcionais presentes nesse genoma, como genes.

Vida. Embora seja frequentemente entendido pelo casamento entre as disciplinas de biologia e informática, um contexto mais abrangente se refere à combinação de conceitos derivados de áreas como a ciência da computação, a matemática, a biologia molecular, a genética, a estatística, bem como outros domínios de conhecimento (Shaik et al., 2019; Sofi; Shafi; Masoodi, 2021). Três principais componentes permeiam a rotina da bioinformática para a resolução de questões biológicas complexas envolvendo grandes volumes de dados, sendo estes: (*i*) infraestrutura computacional para o armazenamento e a segurança dos dados; (*ii*) bancos de dados, que formam repositórios para o armazenamento, a organização e o acesso a dados; e (*iii*) ferramentas e técnicas computacionais para análise, modelamento, visualização, interpretação e comparação de dados (Swiss Institute of Bioinformatics, 2021).

Apesar de a bioinformática possuir uma base bem-estabelecida, não existe um consenso claro quanto ao delineamento de quem é considerado um bioinformata. Enquanto algumas vertentes reservam essa posição apenas para especialistas em todas as facetas da área, outras sugerem a inclusão de qualquer usuário de ferramentas computacionais para fins de pesquisa (Gauthier *et al*., 2019). Independentemente desse debate, o desenvolvimento de habilidades e conhecimentos em bioinformática se tornou essencial para o biólogo e os demais profissionais das Ciências da Vida. Sua importância é tão central para a pesquisa e o mercado de trabalho a ponto de diversas instituições e programas de ensino procurarem a atualização de seus currículos com competências básicas da bioinformática (WELCH *et al*., 2014; WILSON SAYRES *et al*., 2018; SHAIK *et al*., 2019).

Nessa conjuntura, fica evidente uma progressiva aproximação entre a bioinformática e as ciências biológicas, que no futuro poderá culminar na convergência das duas disciplinas (Gauthier *et al*., 2019). De fato, alguns estudos apontam que a biologia passa por um período de transição, caminhando para a direção de se tornar uma ciência da informação, sendo que o

uso de computadores para o estudo da biologia se tornou possível somente após o desenvolvimento e o aprimoramento dos conhecimentos em biologia molecular. Nesse ponto de vista, toda biologia é computacional (Markowetz, 2017; Gauthier *et al*., 2019), o que inclui a biotecnologia. Nessa linha de pensamento, o progresso da biotecnologia somente se torna possível quando apoiado na aplicação dos resultados obtidos pelas pesquisas de caráter bioinformata.

Conforme previamente discutido, a biotecnologia utiliza, em conjunto com abordagens tradicionais, metodologias *in silico* em estudos genômicos, proteômicos, transcriptômicos, de descoberta e desenvolvimento de novos medicamentos, de melhoramento genético, entre outros (Kumar; Chordia, 2017). Novas estratégias e tecnologias surgem em um ritmo regular no mundo da pesquisa, o que cria barreiras para seu entendimento e uso. Assim, serão abordadas neste e-book ferramentas e técnicas desenvolvidas pela bioinformática, compondo um guia prático de utilização. Além disso, ao longo deste e-book será possível perceber a proximidade entre os campos da informática e Ciências da Vida bem como a sua forte relação com a biotecnologia.

Referências

DE CARVALHO, L. M.; BORELLI, G.; CAMARGO, A.P.; DE ASSIS, M.A.; DE FERRAZ, S.M.F.; FIAMENGHI, M.B.; JOSÉ, J.; MOFATTO, L.S.; NAGAMATSU, S.T.; PERSINOTI, G.F.; SILVA, N.V. Bioinformatics applied to biotechnology: A review towards bioenergy research. **Biomass and Bioenergy**, v. 123, p. 195-224, 2019.

DINIZ, W. J. S.; CANDURI, F. Bioinformatics: an overview and its applications. **Genetics and molecular research**, v. 16, n. 1, 2017.

GAUTHIER, J.; VINCENT, A.T.; CHARETTE, S.J.; DEROME, N. A brief history of bioinformatics. **Briefings in Bioinformatics**, v. 20, n. 6, p. 1981-1996, 2019.

GROSS, F.; MACLEOD, M. Prospects and problems for standardizing model validation in systems biology. **Progress in biophysics and molecular biology**, v. 129, p. 3-12, 2017.

GUPTA, V.; SENGUPTA, M.; PRAKASH, J.; TRIPATHY, B.C. An Introduction to Biotechnology. *In*: GUPTA, V. et al. (Eds.). **Basic and Applied Aspects of Biotechnology**. Singapore: Springer, 2017. p. 1-21.

HAAS, R.; ZELEZNIAK, A.; IACOVACCI, J.; KAMRAD, S.; TOWNSEND, S.; RALSER, M. Designing and interpreting 'multi-omic' experiments that may change our understanding of biology. **Current Opinion in Systems Biology**, v. 6, p. 37-45, 2017.

KUMAR, A.; CHORDIA, N. Role of bioinformatics in biotechnology. **Research & Reviews in BioSciences**, v. 12, n. 1, p. 116, 2017.

MARKOWETZ, F. All biology is computational biology. **PLOS Biology**, v. 15, n. 3, p. e2002050, 2017.

SHAIK, N. A.; HAKEEM, K.R.; BANAGANAPALLI, B.; ELANGO, R. **Essentials of Bioinformatics, Volume I:** Understanding Bioinformatics: Genes to Proteins. Cham: Springer International Publishing, 2019.

SOFI, M. Y.; SHAFI, A.; MASOODI, K. Z. **Bioinformatics for Everyone**. London: Academic Press, 2021.

SWISS INSTITUTE OF BIOINFORMATICS. About bioinformatics. **Swiss Institute Of Bioinformatics**, 2021. Disponível em: https://www.sib.swiss/about-sib/what-we-do. Acesso em: 14 out. 2021.

WELCH, L.; LEWITTER, F.; SCHWARTZ, R.; BROOKSBANK, C.; RADIVOJAC, P.; GAETA, B.; SCHNEIDER, M.V. Bioinformatics curriculum guidelines: toward a definition of core competencies. **PLoS computational biology**, v. 10, n. 3, p. e1003496, 2014.

WILSON SAYRES, M. A.; HAUSER, C.; SIERK, M.; ROBIC, S.; ROSENWALD, A.G.; SMITH, T.M.; TRIPLETT, E.W.; WILLIAMS, J.J.; DINSDALE, E.; MORGAN, W.R.; BURNETTE III, J.M. Bioinformatics core competencies for undergraduate life sciences education. **PloS One**, v. 13, n. 6, p. e0196878, 2018.

Capítulo 1

Image J: uma ferramenta para a análise de imagens biológicas

Carine Pedrotti[3]
Clarissa Franzoi[4]
Joséli Schwambach[5]

1. Introdução

Um dos campos em que a computação científica fez incursões particulares foi a área de imagens biológicas, afinal muitos dados biológicos são adquiridos como imagens. Nos últimos anos, o volume e a complexidade de dados nas imagens aumentaram ao ponto de que não é mais viável extrair informações sem o uso de ferramentas computacionais. Em sua forma mais simples, a análise computadorizada de imagens supera as limitações e o viés de um observador humano. Assim, como alternativa para minimizar erros e diminuir o tempo de análise, o uso de programas computacionais de análise de imagem tem sido cada vez mais comum. Dentre os diversos programas disponíveis no mercado, o ImageJ tem se destacado devido à sua facilidade de uso. O interesse em suas aplicações vem principalmente de duas áreas, melhoria na informação da imagem para interpretação humana e processamento de imagens em computador para análise quantitativa e qualitativa, representando um recurso indispensável para a análise de imagens biológicas. O presente capítulo tem como objetivo apresentar informações referentes ao *software* bem como demonstrar o uso básico do ImageJ e suas aplicações

[3] Laboratório de Controle de Doenças de Plantas e Laboratório de Biotecnologia Vegetal, Instituto de Biotecnologia, Universidade de Caxias do Sul.
[4] Idem.
[5] Idem.

para a análise de imagens biológicas, explicado por meio de exemplos concretos.

2. Image J

2.1 O *software*

ImageJ é uma poderosa plataforma frequentemente referenciada para processamento de imagens, desenvolvida por Wayne Rasband no National Institutes of Health. Desde o seu lançamento inicial em 1997, o ImageJ provou-se primordial em muitos projetos científicos, particularmente naqueles das Ciências da Vida (Schneider; Rasband; Eliceiri, 2012; Arena *et al.*, 2017).

O ImageJ é baixado gratuitamente, e, além da sua gratuidade, outra característica fundamental é que o seu código-fonte é de domínio público, ou seja, não está sujeito a direitos autorais, promovendo sua distribuição e modificação apenas citando a fonte (Ferreira; Rasband, 2012). Ele é executado como um miniaplicativo on-line ou como um aplicativo para *download* em qualquer computador com uma máquina virtual Java 1.5 ou posterior. Distribuições para *download* estão disponíveis para Windows, Mac OS X e Linux (https://imagej.nih.gov/ij/download.html). O ImageJ pode exibir, editar, analisar, processar, salvar e imprimir imagens em escala de cinza de 8, 16 e 32 bits e imagens coloridas de 8 e 24 bits. Ele suporta muitos formatos de imagem, incluindo TIFF, GIF, JPEG, BMP, DICOM, FITS e *raw*. As imagens podem ser importadas e lidas como imagens únicas ou como uma série de imagens que compartilham uma única janela. Além disso, é *multithread*, portanto operações demoradas como a leitura de arquivos de imagem podem ser executadas em paralelo com outras operações (Ferreira; Rasband, 2012; Hartig, 2013).

O ImageJ incorpora várias ferramentas úteis para o processamento de imagens. Por exemplo, o programa pode calcular estatísticas de área e valor de *pixel* de seleções definidas pelo usuário, medir distâncias e ângulos, criar

histogramas de densidade e gerar gráficos de perfil de linha. Ele suporta funções padrão de processamento de imagem, como manipulação de contraste, nitidez, suavização, detecção de borda e filtragem mediana, e faz transformações geométricas como escala, rotação e inversão. A imagem pode ser ampliada até 32:1 e 1:32. Todas as funções de análise e processamento estão disponíveis em qualquer fator de ampliação. O programa suporta qualquer número de janelas (imagens) simultaneamente, limitadas apenas pela memória disponível. A calibração espacial está disponível para fornecer medições dimensionais do mundo real em unidades como milímetros, centímetros, etc. (Ferreira; Rasband, 2012).

O ImageJ foi projetado com uma arquitetura aberta que fornece extensibilidade via *plugins*, os quais podem ser escritos pelos usuários e possibilitam a resolução de praticamente qualquer problema de processamento ou análise de imagem, estendendo a funcionalidade do ImageJ para além do seu núcleo básico. As muitas centenas, provavelmente milhares, de *plugins* disponíveis gratuitamente de colaboradores de todo o mundo desempenham um papel fundamental no sucesso do ImageJ. Para que os usuários do programa possam fazer atualizações manuais dos complementos do ImageJ (*plugins*), é conveniente o uso de *ImageJ Distributions*, disponível a partir de várias fontes, que é fornecido com uma coleção pré-organizada de complementos. Também é possível utilizar vários complementos simultaneamente, permitindo ao usuário montar um pacote próprio do ImageJ reunindo os *plugins* que melhor atendam as necessidades do estudo (Ferreira; Rasband, 2012).

2.2 Análise de imagens

Para realizar a análise de uma imagem, é preciso ter um sistema de classificação para obterem-se informações suficientes para diferenciar regiões de interesse e/ou definir um conjunto de características capazes de descrever de maneira efetiva cada região contida em uma imagem. Para isso, faz-se

necessário tratar a imagem evidenciando as regiões de interesse de maneira que o programa possa realizar a análise.

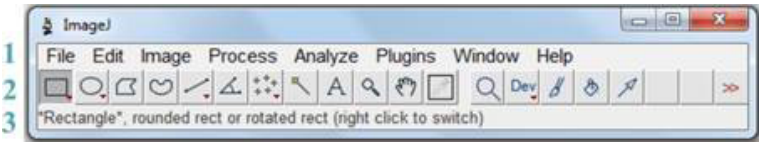
O tratamento de imagem se dá por operações de matrizes que alteram o valor dos seus *pixels*, elementos da imagem que possuem posição e valor de brilho definidos, formando uma matriz correspondente à imagem. Existem diversas operações já descritas na literatura, e a aplicação consecutiva de diferentes operações, ou filtros, pode levar ao mesmo resultado. Dessa maneira, a experiência do operador e o conhecimento básico dos filtros são importantes para se desenvolver um algoritmo com o objetivo de evidenciar as regiões de interesse da imagem (Marcomini; De Souza, 2011). O resultado do tratamento deve ser uma imagem contendo as características que se deseja analisar. Assim, a partir da imagem tratada e após a calibração do *software*, pode-se realizar as medidas.

No entanto, durante o processo de aquisição das imagens a serem analisadas, é importante manter um protocolo uniforme. O foco, a sensibilidade, o tempo de exposição e quaisquer outras condições experimentais devem ser mantidos o mais constantes possível. A alteração de qualquer um desses parâmetros durante a aquisição pode resultar em deturpação de dados. Alterar a sensibilidade da imagem, por exemplo, poderia comprometer seriamente as comparações quantitativas de diferentes imagens (Hartig, 2013).

2.3 Trabalhando com o ImageJ

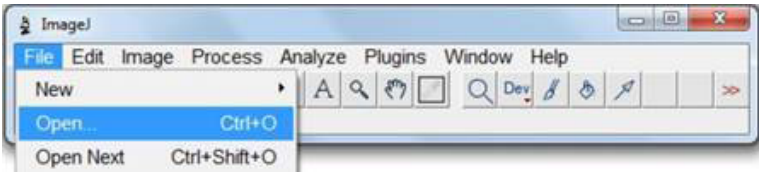
O ImageJ possui uma barra de trabalho com três seções: (*i*) Menu, que exibe mais acessos, como Arquivo, Edição, Imagem, Janela, etc.; (*ii*) Botões de ferramenta, que permite escolher a ferramenta de trabalho, como seleção de área retangular, oval, pontos, *zoom*, etc.; e (*iii*) Barra de informações, que aparece em forma de texto na parte inferior e muda de acordo com a ferramenta selecionada (Figura 1). Para iniciar o trabalho, no menu *File* (Arquivo) > *Open* (Abrir), pode-se abrir uma imagem (Figura 2).

Figura 1 – Interface do programa ImageJ.



Fonte: Elaboração dos autores (2023).

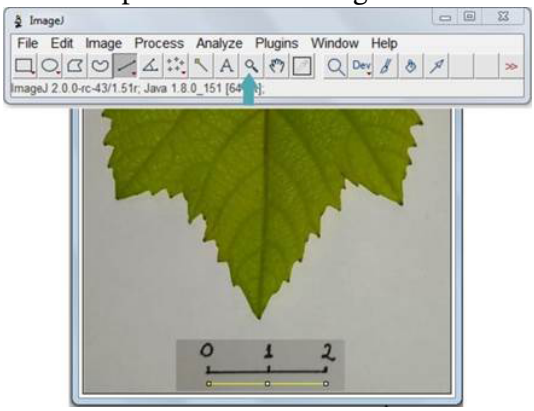
Figura 2 – Comando para abrir um arquivo.



Fonte: Elaboração dos autores (2023).

Independentemente do formato da imagem, as informações são em *pixels*. A primeira coisa a se fazer é calibrar a imagem. O processo de calibração espacial envolve a alteração dos *pixels* de uma imagem para um valor conhecido em mm, cm, mícrons, etc. O usuário precisará de uma referência, como uma régua colocada na imagem ou um objeto de tamanho conhecido na foto (Figura 3).

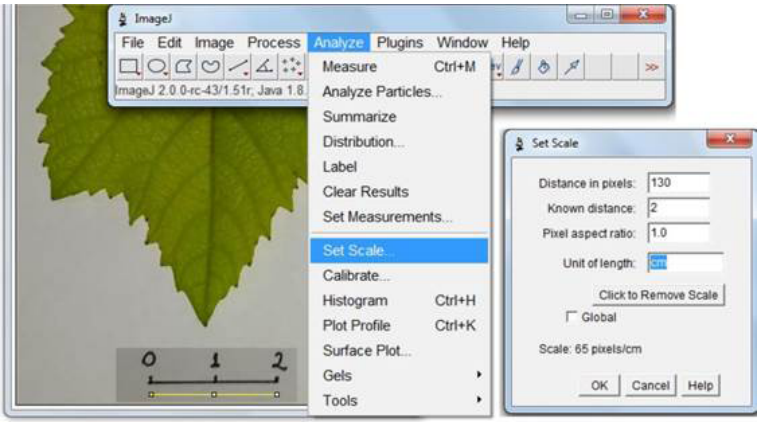
Figura 3 – Desenho de uma linha no objeto de referência para calibrar a imagem.



Fonte: Elaboração dos autores (2023).

Com a ferramenta de seleção de linha, desenha-se uma linha de referência para a escala. Pode-se usar o *zoom* (seta azul na Figura 3) para obter-se a maior precisão possível. Na barra de Menu, nas opções *Analyze* > *Set scale*, digita-se a dimensão da linha desenhada na caixa *Known distance* e define-se a unidade na caixa *Unit of length*. No exemplo, a linha foi desenhada na medida de 2 cm (Figura 4).

Figura 4 – Estabelecendo a calibração e os parâmetros da escala.



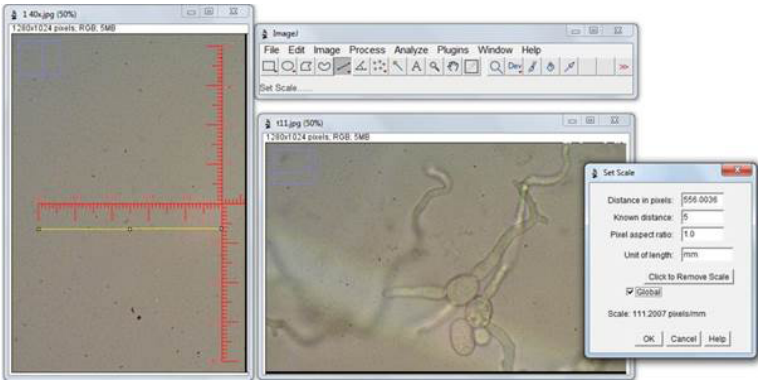
Fonte: Elaboração dos autores (2023).

O ideal é sempre tirar a foto com alguma escala. Se, no ato e tirá-la, a foto tiver um objeto de dimensões conhecidas como moeda, caneta ou até dedos, eles poderão ser usados para calibrar a imagem. Outro caso é quando se trabalha com placas de Petri, as quais possuem dimensões conhecidas que podem ser usadas para a calibração espacial. Se a foto foi tirada com um microscópio óptico e a câmera não tem a função de colocar uma escala, uma opção é tirar uma segunda foto de uma régua (tirada com a mesma ampliação).

Então, se a foto não tiver escala, pode-se abrir as duas imagens no ImageJ, calibrar a imagem com a escala e marcar *Global*. Dessa forma, a calibração será aplicada às imagens a seguir, permitindo a medição, e não será alterada até que

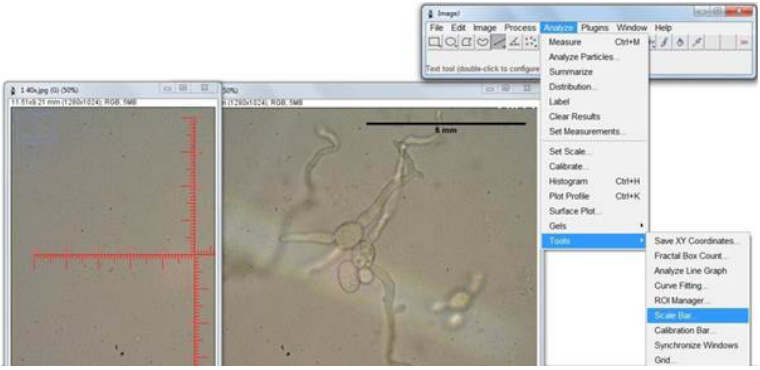
seja recalibrada uma nova imagem ou fechado o programa. Na Figura 5, por exemplo, a primeira foto foi tirada em uma régua graduada com uma objetiva de 40x, e a segunda foto é um conídio germinado tirado com a mesma ampliação. Para adicionar a barra de escala na foto, é preciso acessar o menu *Analyze* > *Tools* > *Scale Bar*. O ImageJ permite escolher o tamanho da barra, a sua posição, a cor e o tipo de letra, entre outros (Figura 6).

Figura 5 – Calibrar imagem sem escala usando duas imagens.



Fonte: Elaboração dos autores (2023).

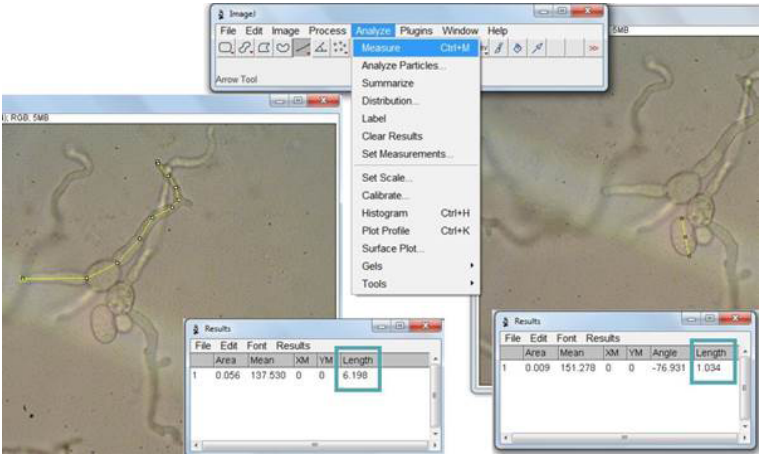
Figura 6 – Colocar a barra de escala na imagem.



Fonte: Elaboração dos autores (2023).

Depois que a imagem estiver calibrada, pode-se fazer qualquer medição, sejam linhas ou áreas. Primeiro, deve-se selecionar o que se deseja medir desenhando uma linha ou uma área no menu *Analyze* > *Measure*. Além de medir linhas retas, também pode-se usar linhas à mão livre ou linhas segmentadas. Uma nova janela *Results* mostrará o valor ponderado em uma tabela, a qual pode ser salva para trabalhos futuros usando *File* > *Save* (Figura 7).

Figura 7 – Medindo o conídio e o tubo germinativo de *Botrytis cinerea*.

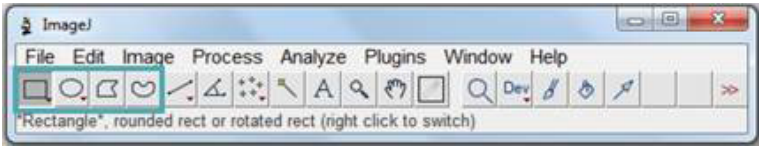


Fonte: Elaboração dos autores (2023).

A maioria dos comandos no ImageJ funcionará em uma região da imagem que precisa ser selecionada ou separada de alguma forma, chamada de seleção de uma área de interesse. Pode-se definir uma área de interesse específica na imagem, usando qualquer uma das ferramentas de seleção na barra de trabalho: retangular, oval, poligonal e à mão livre (Figura 8). Muitas ferramentas têm duas ou mais opções. Clicando com o botão do direito sobre a seta vermelha, a ferramenta exibe um menu de contexto que permite a definição de parâmetros da ferramenta. Por exemplo, na ferramenta *Oval* pode-se alternar entre *Oval selections* (seleção oval), *Elliptical selections*

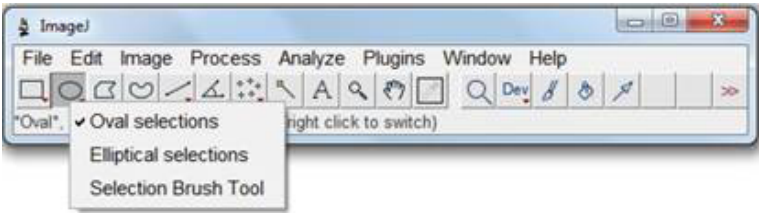
(seleção elíptica) e *Selection Brush Tool* (ferramenta de pincel) (Figura 9).

Figura 8 – Seleção das áreas de interesse.



Fonte: Elaboração dos autores (2023).

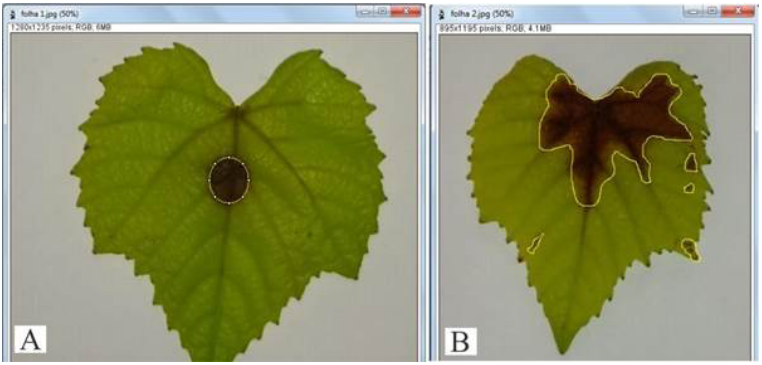
Figura 9 – Selecionando as opções da ferramenta.



Fonte: Elaboração dos autores (2023).

O diâmetro da ferramenta de pincel pode ser ajustado clicando duas vezes no ícone da ferramenta e começando a "pintar" o que se deseja medir. Para ajustar a seleção, deve-se clicar dentro da área de seleção e arrastar ao longo do seu limite, expandindo o limite para o exterior. E, clicando fora da área de seleção e arrastando, será reduzido o tamanho da área de seleção. Alguns exemplos do uso dessa ferramenta (observe a linha que delimita seu perímetro) estão apresentados na Figura 10.

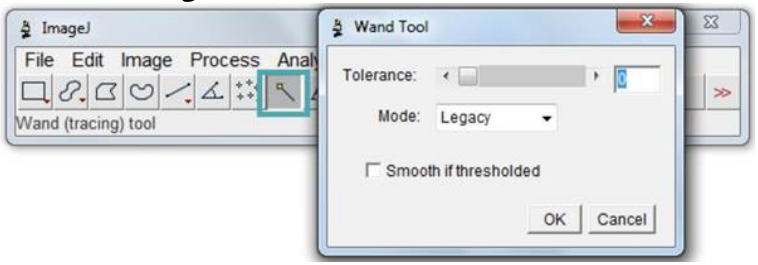
Figura 10 – Zonas determinadas de uma amostra. A: Ferramenta *Oval selections*. B: Ferramenta *Selection Brush Tool*.



Fonte: Elaboração dos autores (2023).

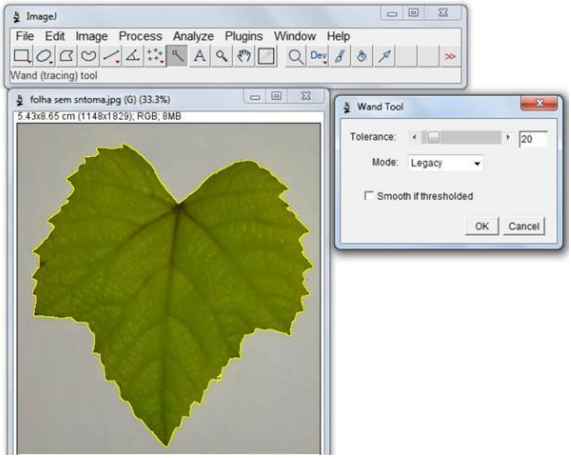
A *Wand Tool* (ferramenta de varinha) é outra maneira de selecionar zonas pela cor, sendo possível criar uma seleção plotando objetos de cores uniformes. Para desenhar um objeto, áreas de cores semelhantes serão selecionadas ao se clicar dentro dele e mover o cursor. Ao clicar duas vezes no ícone da *Wand Tool*, abre-se a caixa de diálogo de configuração, na qual será possível aumentar o valor da tolerância (*Tolerance*). Quanto maior a tolerância, maior a área selecionada (Figura 11). Por exemplo, para calcular a área foliar completa, pode-se aumentar a tolerância, o que permitirá a seleção de toda a área foliar (Figura 12).

Figura 11 – Ferramenta de varinha.



Fonte: Elaboração dos autores (2023).

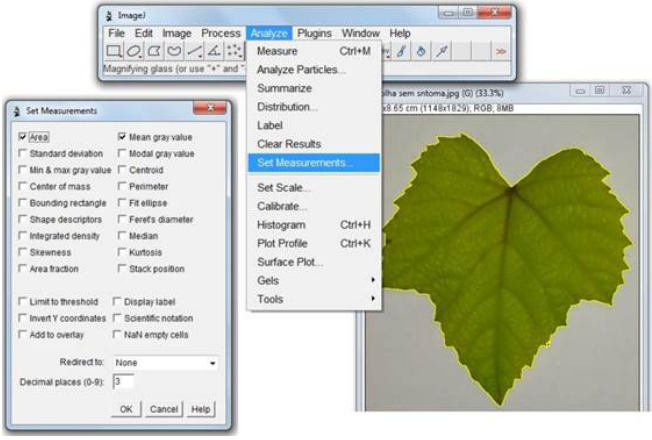
Figura 12 – Ferramenta de varinha para selecionar toda a área foliar.



Fonte: Elaboração dos autores (2023).

Se a foto foi calibrada inicialmente, a área poderá ser medida no menu *Analyze* > *Set Measurements*, selecionando *Area*. Ao realizar-se a medição, poder-se-á obter a área foliar total (Figura 13).

Figura 13 – Estabelecendo a medição da área foliar.

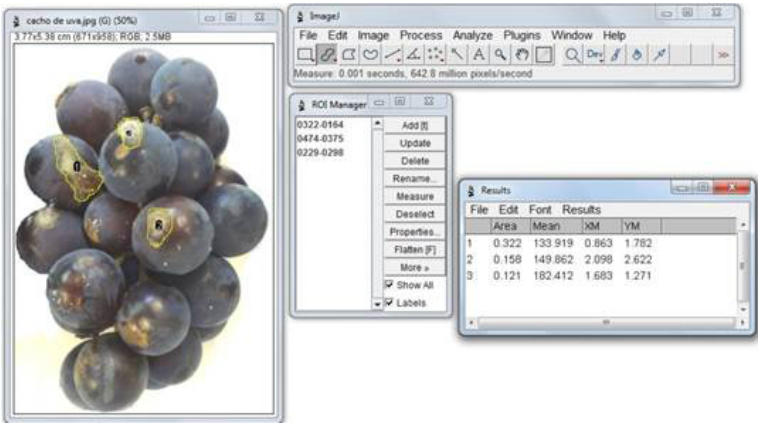


Fonte: Elaboração dos autores (2023).

Também é possível realizar múltiplas medições. Com essa ferramenta, pode-se trabalhar com várias medições, em diferentes locais em uma imagem, de diversos segmentos ou imagens. Todos os tipos de seleção, incluindo pontos, linhas e texto, são compatíveis. É acessado de várias maneiras: Menu *Edit* > *Selection* > *Add to Manager*; Menu *Analyze* > *Tools* > *ROI Manager*; e pressionando a letra "T" no teclado.

A mensuração de áreas de lesão em bagas de uva a partir de uma foto pode, por exemplo, ser realizada com esse recurso. Inicialmente calibra-se a imagem e abre-se o *ROI Manager*, depois faz-se a seleção na estrutura que se deseja medir e clica-se em adicionar (*Add*) para incluir a seleção atual à lista, ou pressiona-se "T". São adicionadas as medidas desejadas e verificado se *Show all* e *Labels* estão marcadas, para que seja possível ver as linhas ao desenhá-las. Ao concluir todas as seleções, pode-se medir todas elas com o comando *Measure*, obtendo-se uma tabela *Results* (Figura 14).

Figura 14 – *ROI Manager* para mediar a área de lesão em bagas de uva.

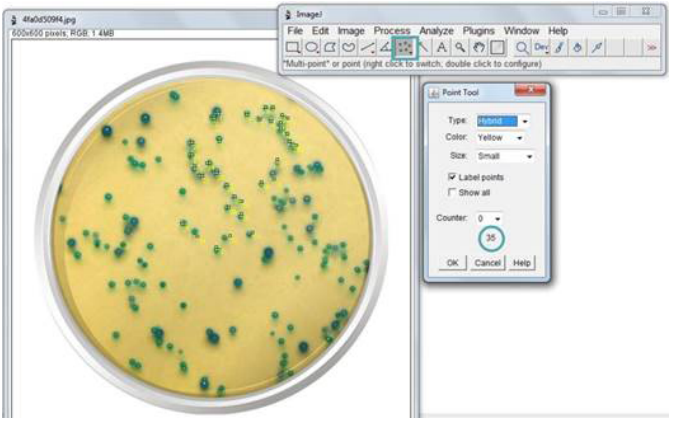


Fonte: Elaboração dos autores (2023).

Outra aplicação do Image J consiste em realizar contagens a partir de uma imagem. Para isso, basta que o usuário abra a imagem, selecione a ferramenta *Point Tool* e clique nas

estruturas que deseja contar. Também é possível escolher o tipo de marca (ponto, círculo, cruz, etc.), a cor e o tamanho. Os pontos contados serão registrados na mesma janela (indicada por um círculo na Figura 15). Se desejar excluir um ponto, pode pressionar ALT e clicar nele. Também é possível movê-los (Figura 15).

Figura 15 – Ferramenta *Point Tool* para contagem de colônias de leveduras em placa de Petri.



Fonte: Elaboração dos autores (2023).

Essas são apenas algumas das ferramentas de uso básico disponíveis no ImageJ. Muitas outras estão disponíveis e podem ser utilizadas para avaliar uma infinidade de imagens biológicas.

Referências

ARENA, E. T.; RUEDEN, C.T.; HINER, M.C.; WANG, S.; YUAN, M.; ELICEIRI, K.W. Quantitating the cell: turning images into numbers with ImageJ. **Wiley Interdisciplinary Reviews Developmental Biology**, v. 6, n. 2, p. e260, 2017.

FERREIRA, T.; RASBAND, W. **The ImageJ User Guide – IJ 1.46r**. 2012. Disponível em: https://imagej.nih.gov/ij/docs/guide/. Acesso em: 5 mar. 2020.

HARTIG, S. M. Basic Image Analysis and Manipulation in ImageJ. **Current Protocols in Molecular Biology**, v. 102, n. 1, p. 14.15.1, 2013.

MARCOMINI, R. F.; DE SOUZA, D. M. P. F. Caracterização microestrutural de materiais cerâmicos utilizando o programa de processamento digital de imagens Image J. **Cerâmica**, v. 57, p. 100-105, 2011.

SCHNEIDER, C. A.; RASBAND, W. S.; ELICEIRI, K. W. NIH Image to ImageJ: 25 years of image analysis. **Nature Methods**, v. 9, n. 7, p. 671-675, 2012.

Capítulo 2

Orange e árvores de decisão: construindo modelos de classificação para dados biológicos de forma intuitiva

Fernanda Pessi de Abreu[6]
Nikael Souza de Oliveira[7]
Pedro Lenz Casa[8]
Scheila de Avila e Silva[9]

1. Introdução

Os recentes avanços das tecnologias computacionais proporcionaram uma revolução em termos de análise de dados, especialmente na área das Ciências da Vida. De fato, o grande volume de dados gerados atualmente impossibilita a realização de análises complexas sem o auxílio de ferramentas computacionais. Ao mesmo tempo, algoritmos de Inteligência Artificial vêm sendo desenvolvidos de modo a auxiliar no trabalho do analista (D'Argenio, 2018; Dall'Alba *et al.*, 2022).

Existem diferentes metodologias de Inteligência Artificial que podem auxiliar na análise de dados, como clusterização, redes neurais e árvores de decisão. Essa última técnica de aprendizado de máquina é uma forma simples e eficiente para classificar os dados, visto que muitos problemas científicos possuem como pressuposto categorizar os casos de acordo com conjuntos finitos de variáveis (Kingsford; Salzberg, 2008; Li; Xu; Deng, 2019). Os taxonomistas, por exemplo, podem

[6] Laboratório de Bioinformática e Biologia Computacional, Instituto de Biotecnologia, Universidade de Caxias do Sul.
[7] Idem.
[8] Idem.
[9] Idem.

classificar as espécies utilizando variáveis químicas, filogenéticas, anatômicas, fisiológicas ou até mesmo morfométricas.

Nesse sentido, Lindbladh *et al.* (2002) utilizaram o algoritmo CART de árvore de decisão para classificar grãos de pólen fósseis de três espécies de Picea. Para a categorização dos dados foram utilizadas variáveis morfométricas. Além disso, como requisito primordial, foi selecionada uma grande quantidade de pólen. Os resultados encontrados no estudo apontaram que a análise por árvore de decisão fornece uma classificação robusta, ao mesmo tempo que mantém a organização conceitual das chaves taxonômicas tradicionais (Lindbladh; O'Connor; Jacobson, 2002).

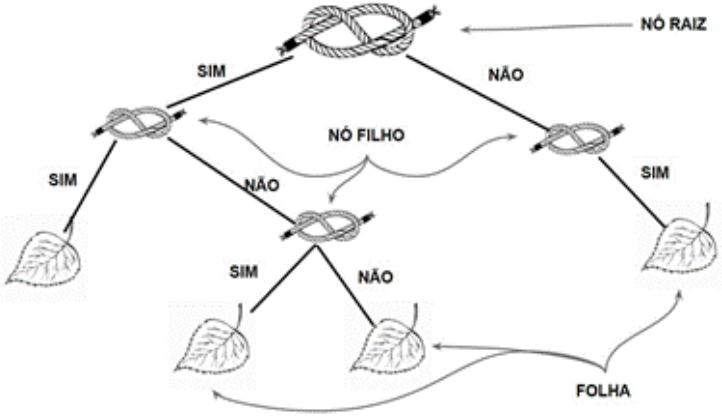
Outro estudo, conduzido por Steinauer e Nickol (2015), utilizou árvores de decisão na revisão taxonômica do complexo de espécies de *Leptorhynchoides thecatus*. Nesse estudo, foram designadas novas espécies para o gênero, evidenciando que o uso de árvore de decisão em conjunto com outras abordagens metodológicas pode possibilitar o reconhecimento e a caracterização dos táxons. Dessa forma, este capítulo busca exemplificar o uso da técnica de mineração de dados conhecida como árvore de decisão aplicada na classificação de espécies utilizando variáveis morfométricas com a ferramenta Orange.

2. Algoritmos de árvore de decisão

Uma árvore de decisão é constituída por um nó-raiz, nós-filhos, ramos e folhas (Figura 1). Os nós são a região em que se realizam os testes e os cálculos para a divisão em novos nós ou folhas. O nó-raiz é o início da árvore e possui todo o conjunto de dados em seu interior. Os nós-filhos são originados do nó-raiz ou de outros nós-filhos. Quanto mais distante do nó raiz, menor é o conjunto de dados dentro de um nó. As folhas são a base da árvore de decisão e representam uma classificação final dos dados. Em outras palavras, elas possuem a decisão em si, um valor ou um rótulo. O conjunto de dados dentro das folhas possui concordância de resultado.

Nós e folhas são conectados pelos ramos, que indicam o valor atribuído à separação do nó ascendente (Sato *et al*., 2013).

Figura 1 – Exemplo de árvore de decisão e sua estrutura contendo nó-raiz, nós-filhos e folhas.



Fonte: Adaptado pelos autores de Sato *et al.* (2013).

Árvores de decisão são comumente empregadas para tarefas de classificação. Nesse sentido, a técnica busca predizer o valor de uma variável "categórica alvo" (resultado final apresentado nas folhas) baseada nas demais variáveis do conjunto de dados. Essa variável-alvo pode ser, por exemplo, um desfecho clínico, um grupo taxonômico, a qualidade da água, entre outros. Vale notar que quando o alvo é uma variável quantitativa contínua a árvore será utilizada para uma tarefa de regressão ao invés de classificação.

Os algoritmos para essa técnica trabalham na forma de "dividir para conquistar", ou seja, um problema complexo é dividido em problemas mais simples, para os quais a mesma estratégia é aplicada até chegar-se em uma solução final (Faceli *et al*., 2011). A decisão de dividir um nó em dois ou mais pode seguir diferentes cálculos e métricas para chegar na melhor opção possível. De forma geral, essas metodologias consideram a pureza dos nós em relação à variável-alvo. Em outras palavras, várias ramificações são testadas considerando-se as

variáveis disponíveis. Ao final, são selecionadas apenas as divisões que resultam em nós-filhos mais homogêneos, os quais possuem uma maior quantidade de casos pertencentes a uma categoria da variável-alvo (Russel; Norvig, 2010).

A fim de exemplificar o processo de construção e funcionamento de uma árvore de decisão, foi adotado um conjunto de dados hipotético, demonstrado na Tabela 1. Cada linha da tabela é uma situação diferente, que possui combinações das variáveis no cabeçalho. O desfecho da situação, indicado pela variável "Reclamação" em cinza, é tomado como variável-alvo ou *target*, sendo esta considerada a base da árvore (folhas). Organizando o conjunto em formato de função, o eixo X representaria cada situação e o Y a variável-alvo. No caso apresentado, a combinação das variáveis (idade, motivo da consulta, tempo de espera, portador de deficiência e vínculo com a unidade) para cada caso (Xn) resulta no desfecho de Reclamação (Y).

Tabela 1 – Conjunto de dados fictícios de situações em uma Unidade Básica de Atendimento, demonstrando exemplos em que houveram ou não reclamações por parte dos pacientes.

Caso	Idade	Motivo da consulta	Tempo de espera (minutos)	Portador de deficiência	Vínculo com a unidade	Reclamação
X_1	60>	Receita	30	Não	Sim	Sim
X_2	50-59	Receita	40	Não	Sim	Sim
X_3	20-29	Urgência	15	Não	Sim	Não
X_4	60>	Rotina	40	Não	Sim	Sim
X_5	60>	Urgência	15	Não	Não	Não
X_6	50-59	Urgência	15	Não	Não	Sim
X_7	20-29	Receita	45	Sim	Sim	Não
X_8	20-29	Rotina	50	Sim	Não	Sim
X_9	50-59	Rotina	50	Não	Sim	Não
X_{10}	50-59	Receita	50	Sim	Sim	Não

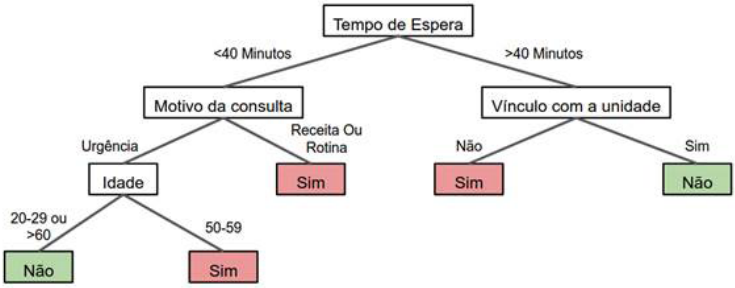
Fonte: Elaboração dos autores (2023).

A partir do conjunto de dados da Tabela 1, originou-se a árvore de decisão demonstrada na Figura 2. Pode-se perceber

que o algoritmo ignorou a variável "portador de deficiência", já que esta não apresenta relevância na tomada de decisão. A leitura da árvore segue uma lógica de condição-resultado (se nó-raiz e nó-filho... então folha) e pode ser feita da seguinte maneira:

- Se "tempo de espera" for menor que 40 minutos, "motivo da consulta" urgência, e "idade" entre 20-29 anos ou maior que 60 anos, o paciente não irá reclamar;
- Se "tempo de espera" for menor que 40 minutos, "motivo da consulta" urgência e "idade" entre 50-59 anos, o paciente irá reclamar;
- Se "tempo de espera" for menor que 40 minutos e "motivo da consulta" receita ou rotina, o paciente irá reclamar;
- Se "tempo de espera" for maior que 40 minutos e o paciente não possui "vínculo com a unidade", o paciente irá reclamar;
- Se "tempo de espera" for maior que 40 minutos e o paciente possui "vínculo com a unidade", o paciente não irá reclamar.

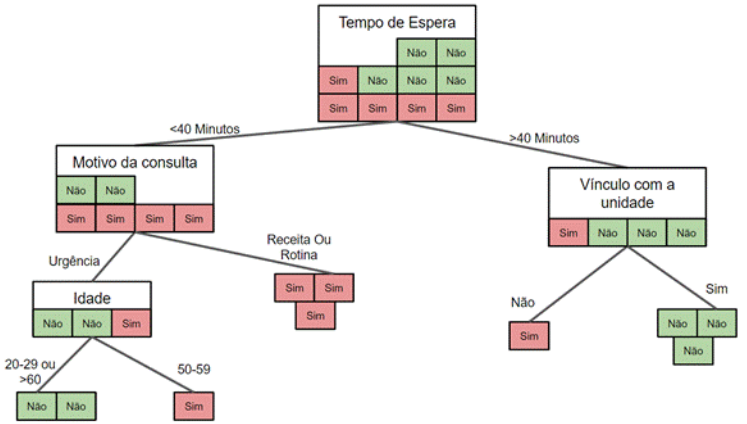
Figura 2 – Árvore de decisão originada a partir dos dados hipotéticos para situações em uma Unidade Básica de Atendimento utilizando a variável "reclamação" como alvo.



Fonte: Elaboração dos autores (2023).

De modo a facilitar a compreensão do modo como ocorre a estratégia de "dividir para conquistar", comentada no início desta seção, elaborou-se a Figura 3, com a qual é possível perceber o conjunto de dados sendo dividido à medida que os nós-filhos são estabelecidos. Ao mesmo tempo, pode-se perceber que todas as situações presentes em cada folha estão em concordância, ou seja, apresentam o mesmo valor da variável-alvo.

Figura 3 – Demonstração de como um conjunto de dados se divide por meio da técnica de árvore de decisão.



Fonte: Elaboração dos autores (2023).

3. Ferramenta Orange

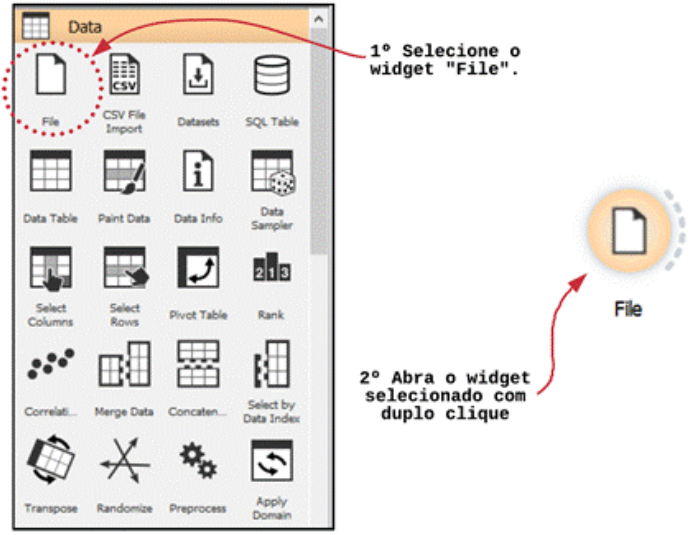
O Orange é um recurso de mineração de dados baseado em componentes (*widgets*) desenvolvido pelos pesquisadores do Laboratório de Bioinformática da Universidade de Ljubljana na Eslovênia. Esse *software* livre inclui diferentes abordagens de visualização, análise, pré-processamento e modelagem dos dados, podendo ser utilizado por meio de uma interface ou um pacote para a linguagem de programação Python. A interface gráfica permite a criação de *workflows* de maneira intuitiva, sem que o usuário possua conhecimentos prévios de programação. A funcionalidade de árvore de decisão implementada

no Orange utiliza as métricas de ganho de informação para classificação e erro quadrático médio para regressão (Demšar *et al.*, 2013). O *download* da ferramenta pode ser realizado pelo link: https://orange.biolab.si/download/#windows. Nesta seção será demonstrado um passo a passo da utilização de árvore de decisão no *software* Orange.

3.1 Começando um novo projeto no Orange

Para iniciar um novo projeto é necessário importar um conjunto de dados. Visto isso, primeiramente selecione o *widget* denominado *File* na aba *Data*. Em seguida, mediante um duplo clique no *widget* adicionado ao *workflow*, será possível selecionar o arquivo desejado (Figura 4). O Orange aceita arquivos de planilha eletrônica nos formatos .xlsx, .csv e .tab, sendo esses últimos delimitados por vírgulas ou tabulações. De modo complementar, o *software* também recebe dados on-line, por exemplo, planilhas Google.

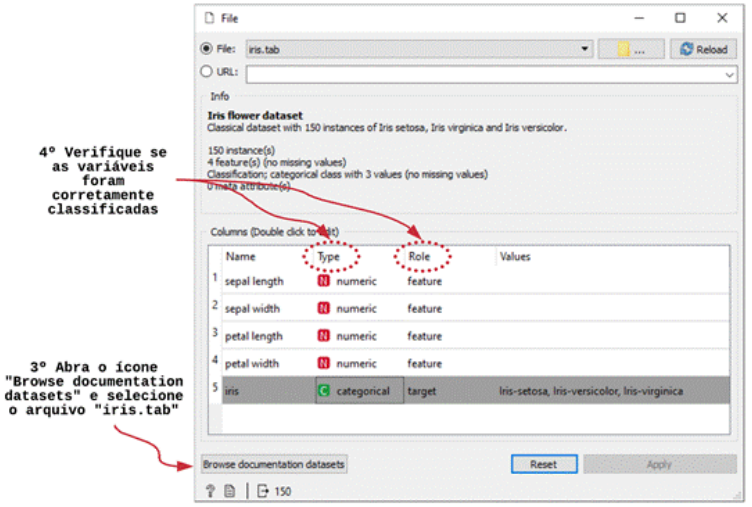
Figura 4 – Seleção do *widget* "*File*" para importação de um arquivo.



Fonte: Elaboração dos autores (2023).

Para este exemplo será utilizado o conjunto de dados multivariado “iris”, o qual representa um dos conjuntos de exemplo disponibilizados pelo Orange. Este contém informações morfométricas de 150 amostras de 3 espécies do gênero *Iris*. Além disso, é composto por quatro variáveis numéricas (comprimento da sépala, largura da sépala, comprimento da pétala e largura da pétala) e um atributo categórico (espécie). Na interface de carregamento do conjunto de dados é necessário abrir a opção *Browse documentation datasets* e selecionar o arquivo “iris.tab”. Com o arquivo já selecionado, é importante verificar na coluna *Type* se as variáveis foram corretamente categorizadas. De modo complementar, a variável de nome “iris” deve estar classificada como *target* na coluna *Role*, ou seja, o alvo das análises; e as demais devem estar selecionadas como *feature* (Figura 5).

Figura 5 – Interface de carregamento dos arquivos.

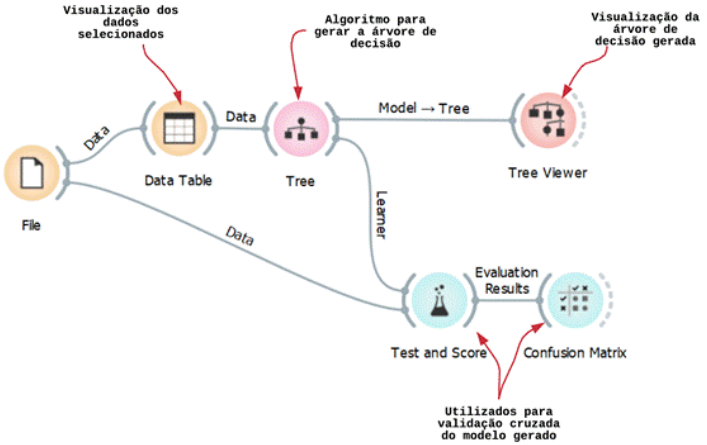


Fonte: Elaboração dos autores (2023).

3.2 Construindo o *workflow* para a árvore de decisão

O *workflow* completo que será utilizado para analisar os dados "iris" está exemplificado na Figura 6. Os *widgets* de *Data Table*, *Tree* e *Tree Viewer* estão localizados, respectivamente, nas abas *Data*, *Model* e *Visualize*. Por fim, *Test and Score* e *Confusion Matrix* estão situados na aba *Evaluate*. Para a adição dos *widgets*, deve-se realizar um duplo clique em cima do ícone correspondente. A organização da interface de trabalho pode ser realizada arrastando os *widgets*. Além disso, é necessário unir os componentes; para isso, selecione a região pontilhada e conecte com o próximo *widget*. A conexão formada deve ser "*Data – Data*", verificada por uma linha contínua. Para isso, deve-se realizar um duplo clique na linha, que abrirá o menu da conexão. Caso não esteja de acordo, o usuário pode selecionar *Clear all* e reconectar as duas colunas. No entanto, caso o objetivo seja analisar apenas parte dos dados, a conexão formada deve ser "*Selected Data – Data*", verificada por uma linha pontilhada.

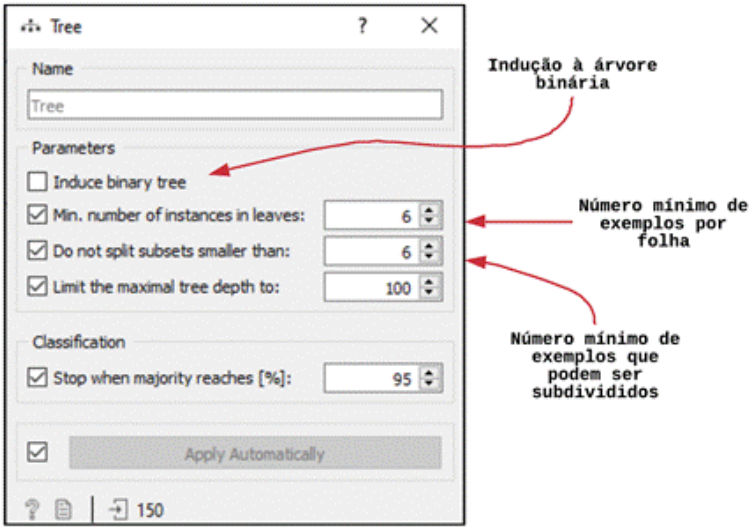
Figura 6 – *Workflow* completo para análise dos dados por meio de árvore de decisão. As descrições localizadas acima das linhas, que estão ligando os componentes, demonstram a relação entre os dados de entrada e saída de cada conexão.



Fonte: Elaboração dos autores (2023).

Existem parâmetros que devem ser ajustados manualmente pelo pesquisador; para isso, é preciso selecionar o *widget Tree*. O parâmetro de indução da árvore binária divide cada um dos nós em apenas dois nós-filhos. Recomenda-se que esse *widget* não seja selecionado para análise de dados biológicos, visto que normalmente são de caráter multivariado. Outros parâmetros, como a delimitação do número mínimo de exemplos por folha e a quantidade mínima de casos para formação de subconjuntos (nós-filhos), devem ser avaliados e testados para cada conjunto de dados (Figura 7). Adicionalmente, pode-se fazer uso dos componentes de validação cruzada para verificar a assertividade do modelo de acordo com os parâmetros estabelecidos.

Figura 7 – Parâmetros importantes para a geração da árvore de decisão.



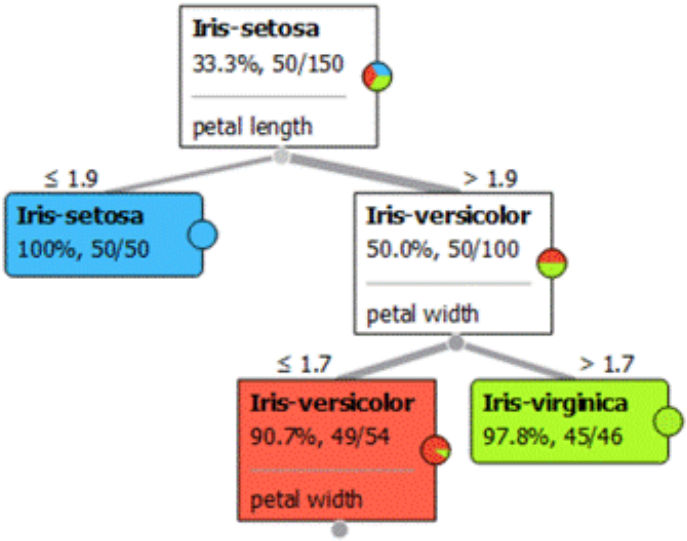
Fonte: Elaboração dos autores (2023).

3.3 Visualizando a árvore de decisão gerada

A árvore gerada pelo modelo pode ser visualizada com o componente *Tree Viewer* (Figura 8). Ainda mais, alguns pa-

râmetros podem ser ajustados para melhorar a visualização e o entendimento, como largura, quantidade de níveis, largura da borda e classe-alvo. Para salvar a árvore de decisão existem duas formas: a primeira é selecionando o ícone de salvar presente no canto inferior esquerdo do menu *Tree Viewer*; e a segunda é adicionando o componente *Save Data* ao *workflow*, presente na aba *Data*, sendo necessário conectar uma linha entre *Tree Viewer* e o novo *widget*.

Figura 8 – Visualização da árvore de decisão gerada.



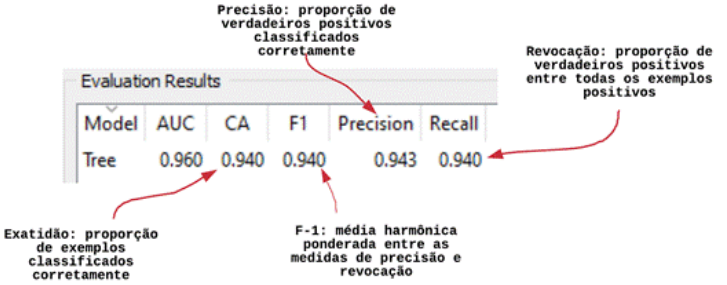
Fonte: Elaboração dos autores (2023).

3.4 Analisando o modelo da árvore de decisão

A avaliação do modelo predito pode ocorrer por meio da validação cruzada. Nesse método, são formados subconjuntos aleatórios a partir dos dados, chamados de teste e treino. Para que cada caso faça parte de ambos os subconjuntos em ocasiões diferentes, esse processo de divisão ocorrerá algumas vezes. As principais métricas de avaliação utilizadas podem ser geradas pelo *widget Test and Score* (Figura 9). Em complemento, a verificação da assertividade do modelo pode ocorrer

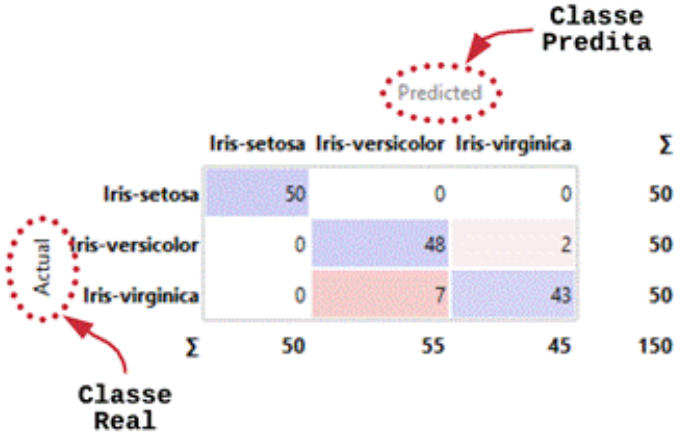
por meio da matriz de confusão. Na matriz de confusão é apresentada a frequência da classificação para cada classe do modelo; na diagonal principal estão representados os verdadeiros positivos (Figura 10).

Figura 9 – Exemplo das medidas de avaliação geradas pelo *widget Test and Score*.



Fonte: Elaboração dos autores (2023).

Figura 10 – Exemplo de matriz de confusão gerada pelo *widget Confusion Matrix*.



Fonte: Elaboração dos autores (2023).

4. Considerações finais

Visto o grande volume de dados gerados em conjunto com múltiplas variáveis, a utilização de ferramentas que auxiliem na mineração de dados torna-se fundamental. Nesse sentido, o uso de *softwares* como o Orange proporciona ao pesquisador uma maneira intuitiva de explorar seus dados. Além disso, algoritmos de árvore de decisão para análise de dados morfométricos das espécies biológicas podem possibilitar novos *insights* sobre a sua classificação e categorização. De modo complementar, a metodologia apresentada também pode ser aplicada para a construção de chaves dicotômicas com características contínuas ou discretas.

Referências

DALL'ALBA, G.; CASA, P.L.; ABREU, F.P.; NOTARI, D.L.; DE AVILA E SILVA, S. A Survey of Biological Data in a Big Data Perspective. **Big Data**, v. 10, n. 4, p. 279-297, 2022.

D'ARGENIO, V. The High-Throughput Analyses Era: Are We Ready for the Data Struggle? **High-Throughput**, v. 7, n. 1, p. 8, 2018.

DEMŠAR, J.; CURK, T.; ERJAVEC, A.; GORUP, Č.; HOČEVAR, T.; MILUTINOVIČ, M.; MOŽINA, M.; POLAJNAR, M.; TOPLAK, M.; STARIČ, A.; ŠTAJDOHAR, M. Orange: data mining toolbox in Python. **Journal of machine Learning research**, v. 14, n. 1, p. 2.349-2.353, 2013.

FACELI, K.; LORENA, A.C.; GAMA, J.; DE ALMEIDA, T.A.; DE CARVALHO, A.C.P.L.F. **Inteligência artificial:** uma abordagem de aprendizado de máquina. Rio de Janeiro: LTC, 2011.

KINGSFORD, C.; SALZBERG, S. L. What are decision trees? **Nature Biotechnology**, v. 26, n. 9, p. 1.011-1.013, 2008.

LI, M.; XU, H.; DENG, Y. Evidential Decision Tree Based on Belief Entropy. **Entropy**, v. 21, n. 9, p. 897, 2019.

LINDBLADH, M.; O'CONNOR, R.; JACOBSON, G. L. Morphometric analysis of pollen grains for paleoecological studies: classification of Picea from eastern North America. **American Journal of Botany**, v. 89, n. 9, p. 1.459-1.467, 2002.

RUSSELL, S.; NORVIG, P. **Artificial Intelligence: A Modern Approach**. 3. ed. Upper Saddle River: Prentice Hall, 2010.

SATO, L. Y.; SHIMABUKURO, Y.E.; KUPLICH, T.M.; GOMES, V.C.F. 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** [...]. São José dos Campos: Instituto Nacional de Pesquisas Espaciais (INPE), 2013. p. 2.353-2.360.

STEINAUER, M. L.; NICKOL, B. B. Revision of Leptorhynchoides thecatus (Acanthocephala: Illiosentidae), with morphometric analysis and description of six new species. **The Journal of Parasitology**, v. 101, n. 2, p. 193-211, 2015.

Capítulo 3

Triagem virtual de pequenas moléculas

Gustavo Machado das Neves[10]
Luciano Porto Kagami[11]
Luis Fernando Saraiva Macedo Timmers[12]
Rafael Andrade Caceres[13]

1. Introdução

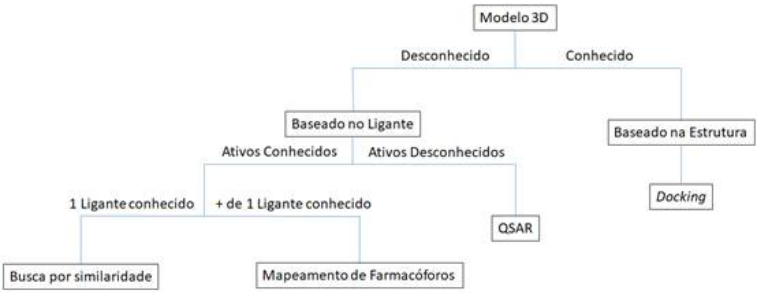
Durante o século XX e o início do século XXI as inovações tecnológicas nos diversos campos da ciência, a interdisciplinaridade e o intercâmbio de ideias levaram a um aumento exponencial na disponibilidade de informações acerca de novos alvos moleculares e novas moléculas, algumas com possível fim terapêutico. Apesar dos avanços tecnológicos das macromoléculas (proteínas) na terapia gênica e na obtenção de anticorpos monoclonais, é pouco provável que as pequenas moléculas percam seu espaço de mercado. As macromoléculas são degradadas se administradas por via oral, na maioria das vezes nem conseguem atravessar as membranas celulares e possuem um alto custo de produção (Robert *et al.*, 2019). O equilíbrio de mercado entre as pequenas moléculas e as macromoléculas depende da necessidade de alta eficiência para o tratamento de doenças a um custo mínimo. Por esse motivo, vários métodos foram desenvolvidos com a finalidade de encontrar atividade farmacológica para essas pequenas moléculas, ou seja, compostos com massa molecular abaixo de 900 daltons (Macielag, 2012).

A triagem de pequenas moléculas em larga escala teve início nos anos 90, com a introdução da triagem de alto ren-

[10] Universidade Federal do Rio Grande do Sul.
[11] Idem.
[12] Universidade do Vale do Taquari.
[13] Universidade Federal de Ciências da Saúde de Porto Alegre.

dimento ou *High Throughput Screening* (HTS). Esse método automatizado é capaz de avaliar a atividade de milhares de compostos em centenas de proteínas-alvo ou diferentes linhagens celulares. Entendem-se como proteínas-alvo as proteínas previamente validadas experimentalmente cuja modulação ocasiona um efeito biológico esperado. Os ensaios de Triagem Virtual, ou *Virtual Screening* (VS), utilizam métodos computacionais para analisar bancos de dados com milhares de compostos. A base do VS difere do HTS em seus objetivos, pois o método computacional pretende identificar um pequeno número de compostos com uma probabilidade acima da média de ser ativo para testes biológicos (Stahura; Bajorath, 2004). Assim, o VS é uma opção quando os ensaios são complexos ou difíceis de serem adaptados ao HTS. Tanto o método HTS quanto o VS possuem limitações. A principal limitação do HTS é o grande número de falsos positivos, enquanto os métodos computacionais possuem limitações inerentes a imperfeições de algoritmos e dependência dos dados experimentais (Pagadala; Syed; Tuszynski, 2017; Yang *et al*., 2020). No entanto, esses dois métodos podem ser utilizados de forma complementar, se aplicados em conjunto de maneira significativa (Stahura; Bajorath, 2004). Basicamente, a rotina de VS é dividida em duas abordagens: uma baseada na estrutura do alvo (*Structure Based Virtual Screening*) e outra baseada na semelhança de ligantes (*Ligand Based Virtual Screening*). A escolha entre a abordagem baseada no ligante ou na estrutura é avaliada de acordo com os dados conhecidos do alvo a ser triado. A Figura 1 mostra o esquema para a determinação do método de VS a ser utilizado.

Figura 1 – Esquema para decisão de escolha do melhor método para a rotina de triagem virtual.



Fonte: Elaboração dos autores (2023).

Este capítulo abordará conceitos, rotinas e cuidados para uma efetiva utilização das ferramentas *in silico* na triagem de pequenas moléculas.

2. Triagem virtual baseada na estrutura

A triagem virtual baseada na estrutura é um método que utiliza um modelo tridimensional de um alvo farmacológico (*e.g.* proteínas, ácidos nucléicos), comumente chamado de receptor, para prever a sua interação com diferentes ligantes, por meio de programas de atracamento molecular (*docking*). Os graus de liberdade (GL) envolvidos no processo variam conforme o ensaio de *docking* for configurado. Entende-se como flexibilidade a capacidade de rotação e estiramento das ligações entre os átomos que constituem uma molécula. A configuração que exige menos esforço computacional mantém o ligante e o receptor rígidos. Já uma configuração mais próxima da realidade biológica utiliza o ligante e o receptor flexíveis. No entanto, a velocidade de processamento é diminuída e o tempo para obterem-se os resultados é aumentado. A configuração com uma exigência computacional mediana mantém o ligante flexível e o receptor rígido (Rosenfeld; Vajda; Delisi, 1995).

Os programas de *docking* fazem uso de dois algoritmos para modelar a interação entre o ligante e o alvo: o algoritmo de

amostragem e o algoritmo de pontuação (Quadro 1) (Guedes; De Magalhães; Dardenne, 2014). O algoritmo de amostragem é responsável por buscar um conjunto de poses do ligante dentro de um determinado sítio (ortostérico ou alostérico), enquanto o algoritmo de pontuação avalia numericamente as interações com o receptor. A sinergia entre os algoritmos de amostragem e de pontuação faz possível a previsão tanto da pose do ligante quanto da sua posição no ranking entre os compostos candidatos. No entanto, o ensaio de *docking* por si não é suficiente para um resultado satisfatório e confiável. É preciso fazer um planejamento prévio, com escolhas certas para um bom desempenho da modelagem.

Quadro 1 – Principais algoritmos de amostragem e de pontuação utilizados no *docking*.

Algoritmos de amostragem			**Algoritmos de pontuação**		
Classificação	**Características**	**Exemplos**	**Classificação**	**Características**	**Exemplos**
Sistemáticos	Exploração de todos os GL durante a busca.	Busca exaustiva (Glide) Construção incremental (FlexX) Conjunto de conformações (Dock 6.0)	Baseado em campos de força	Forças de van der Waals, interações eletrostáticas (ligante-receptor)	GoldScore
Estocásticos	Modificações randômicas avaliadas por critérios probabilísticos	Monte Carlo (Glide) Algoritmos evolucionários (GOLD, Autodock)	Baseado em dados empíricos	Diversas interações ponderadas a partir de dados empíricos	ChemScore; GlideScore
Determinísticos	Atual estado do sistema determina as alterações a serem feitas	Dinâmica Molecular (CDOCKER)	Baseado em conhecimento prévio	Utilizam análises estatísticas (inversa de Boltzmann)	DrugScore; PMF

Fonte: Elaboração dos autores (2023).

Como os compostos são avaliados quanto à interação com um receptor, a escolha de modelos de proteínas viáveis para o ensaio de *docking* é essencial para o sucesso da predição. É possível obter modelos tridimensionais oriundos da cristalo-

grafia e da difração de raios x, Ressonância Magnética Nuclear (RMN), microscopia eletrônica e modelagem comparativa. Devem-se priorizar os modelos com resolução inferior a 2,0 ångströms (Å), pois resoluções acima desse valor começam a perder informações precisas das posições atômicas, principalmente nas porções mais flexíveis das proteínas, chamadas de alças ou *loops* (Tovchigrechko; Walls; Vakser, 2002). Processos prévios de preparo do ligante e do receptor são necessários para uma boa prática na rotina de *docking*. Basicamente, o preparo do receptor se resume à verificação dos estados de protonação dos resíduos de aminoácidos (Sastry *et al*., 2013) e à minimização de energia do receptor (Clark; Webster-Clark, 2012). A adequação dos estados de protonação proporciona uma interpretação dos algoritmos de pontuação mais correta devidos às interações eletrostáticas. A minimização de energia faz uso de campos de força para reposicionar os átomos do receptor, evitando o impedimento estérico e adequando a distância entre os seus átomos.

Dois métodos de avaliação da acurácia dos programas de *docking* são amplamente utilizados: o *re-docking* e o *cross-docking*. Entende-se como *re-docking* (Figura 2A) o processo de retirar o ligante do complexo e, após, utilizar o programa de *docking* para inseri-lo novamente (Dallakyan; Olson, 2015). Então, o desvio da raiz quadrada média (do inglês *Root-Mean-Square Deviation* – RMSD) entre as posições atômicas do ligante do modelo experimental e do ligante inserido pelo *docking* é avaliado. Abaixo, a equação utilizada para o cálculo do RMSD:

Equação 1 – Root-Mean-Square Deviation – RMSD

$$RMSD(v,w) = \sqrt{\frac{1}{n}\sum_{i=1}^{n}\left|v_i - w_i\right|^2}$$
$$= \sqrt{\frac{1}{n}\sum_{i=1}^{n}\left(\left(v_{ix}-w_{ix}\right)^2+\left(v_{iy}-w_{iy}\right)^2+\left(v_{iz}-w_{iz}\right)^2\right)}$$

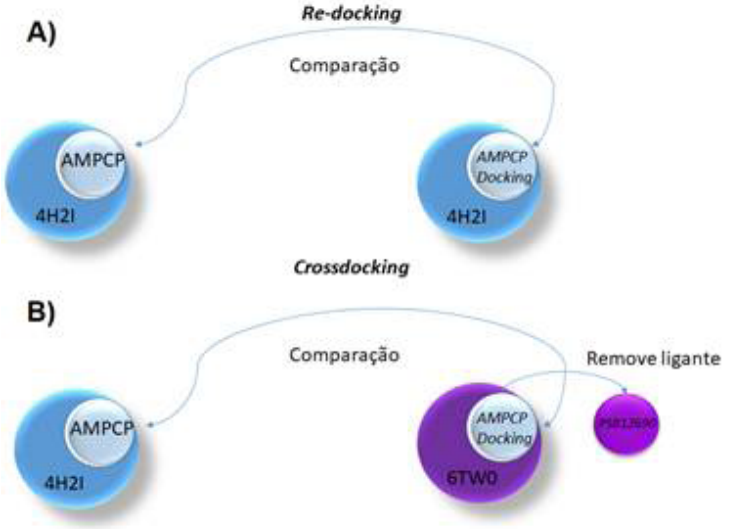
Em que:

v é a posição atômica do ligante do modelo experimental;

w a posição atômica do ligante reposicionado pelo programa de *docking*.

O processo de *cross-docking* (Figura 2B) consiste em realizar o atracamento molecular em outros receptores não nativos que sejam iguais ou similares e avaliar o RMSD. Esse método é utilizado para tentar identificar o deslocamento dos resíduos de aminoácidos pela influência do ligante (efeito induzido). Os resultados obtidos tanto para o *re-docking* quanto para o *cross-docking* podem ser avaliados individualmente (valores de RMSD < 2,0 Å) (Caroli *et al*., 2014) ou frente a outros programas de *docking*, para determinar qual possui maior poder preditivo (Daeyaert *et al*., 2004).

Figura 2 – Esquema dos processos de *re-docking* (A) e *crossdocking* (B). Os códigos de quatro algarismos alfanuméricos são códigos de identificação das estruturas no banco de dados *Protein Data Bank* (PDB).



Fonte: Elaboração dos autores (2023).

A triagem virtual baseada na estrutura do receptor ainda possui muitas limitações. A previsão do efeito de solvatação, a flexibilidade dos resíduos de aminoácidos e as funções de pontuação ainda estão limitadas às capacidades computacionais, por isso a precisão é afetada pela simplificação dos cálculos (Huang; Zou, 2010). No futuro, com o avanço tecnológico, será possível desenvolver métodos de modelagem mais precisos, que incorporem dados bioquímicos e biofísicos e consigam lidar com uma gama maior de graus de liberdade.

3. Triagem virtual baseada no ligante

Um dos maiores desafios relacionados ao desenvolvimento de fármacos deriva da necessidade de inovação, tanto do ponto de vista farmacológico (mecanismo de ação) quanto do ponto de vista sintético (estrutura química). Sendo assim, pesquisadores das mais diversas áreas comparam moléculas a fim de analisar semelhanças e eventuais diferenças, de modo a projetar novos e melhores compostos. As técnicas de triagem virtual baseadas no ligante fundamentam-se na informação química contida em cada molécula/fármaco, uma vez que se desconhece o alvo molecular. Uma das principais vantagens desses métodos reside na velocidade de processamento de informações, já que não dependem dos graus de liberdade associados aos métodos baseados na estrutura (Gimeno *et al*., 2019). Por outro lado, tais métodos podem ficar restritos ao espaço químico das estruturas comparadas (referência e bancos de dados) (Rodrigues *et al*., 2012).

Os métodos baseados em ligante levam em consideração uma premissa fundamental: moléculas semelhantes, em geral, apresentam propriedades (*e.g.* atividades biológicas e parâmetros físico-químicos) semelhantes (Maggiora *et al*., 2014; Gimeno *et al*., 2019). Dentre os principais métodos utilizados para triagem virtual podemos destacar: (a) Pesquisa por Similaridade; (b) Triagem Virtual por Farmacóforo; (c) Relação Estrutura-Atividade Quantitativa (do inglês *Quantitative Structure-Activity Relationship* – QSAR).

3.1 Pesquisa por similaridade

Considerada como o berço das técnicas baseadas em ligantes, a pesquisa por similaridade respalda-se na comparação de um composto com atividade biológica conhecida (referência) e um ou mais compostos de interesse, os quais podem ser encontrados em base de dados (*e.g.* ZINC e ChEMBL) ou em quimiotecas construídas pelo usuário. O conceito-chave é a similaridade; no entanto, tal termo pode ser de difícil interpretação. Estabelecer similaridade e dissimilaridade entre compostos muitas vezes é uma tarefa árdua, em virtude dos diferentes tipos de similaridade que podem ser avaliadas: similaridade química, similaridade molecular, similaridade biológica, entre outras (Maggiora *et al*., 2014; Cereto-Massagué *et al*., 2015; Da Silva Rocha *et al*., 2019). Abordaremos aqui os conceitos relacionados à similaridade molecular, devido à sua relação com o desenvolvimento de fármacos.

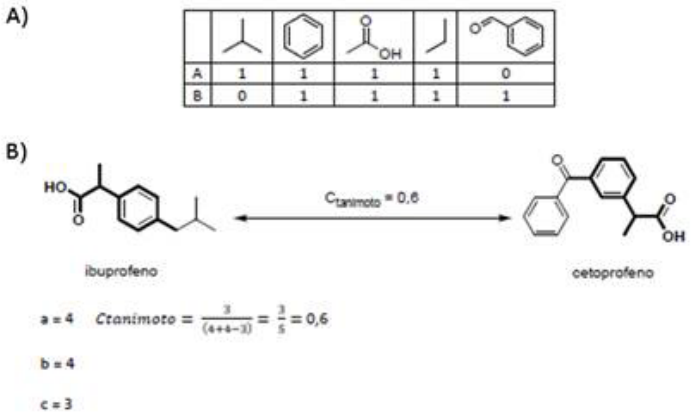
A similaridade molecular leva em consideração dois principais aspectos: (a) a representação molecular e (b) os coeficientes de similaridade (Willett, 2003; Maggiora *et al*., 2014). Os compostos analisados precisam ser representados de uma forma prática, de modo que os programas possam identificar a estrutura química e compará-la adequadamente.

Os métodos de representação molecular levam em consideração os dados obtidos a partir dos grafos moleculares. Tais métodos dividem-se, ainda, conforme os níveis de informação relacionados: bidimensionais (2D), ou seja, dados topológicos, subestruturais e fragmentos moleculares; e tridimensionais (3D), definidos por pontos farmacofóricos, conformações e formas/volume (Liu; Jiang; Li, 2011; Maggiora *et al*., 2014; Cereto-Massagué *et al*., 2015; Shin *et al*., 2015; Gimeno *et al*., 2019). Uma das formas mais utilizadas para representação envolve a criação de impressões digitais moleculares (*molecular fingerprints*) (Figura 3A), as quais se caracterizam por transformar a estrutura molecular em uma sequência de dígitos binários (bits), conforme a presença ou a ausência de um determinado grupo químico (Rodrigues *et al*., 2012; Da Silva

Rocha *et al*., 2019). Os *fingerprints* podem ser classificados, de acordo com a forma de obtenção dos bits, em: (a) *fingerprints* baseados na codificação por subestrutura; (b) *fingerprints* topológicos ou baseados em caminhos; e (c) *fingerprints* circulares (Cereto-Massagué *et al*., 2015; Gimeno *et al*., 2019).

A partir da representação molecular, as funções de similaridade permitem a comparação dos compostos e traduzem as informações em valores numéricos, geralmente de 0 a 1, os quais relacionam o nível de similaridade entre os compostos (Rodrigues *et al*., 2012; Da Silva Rocha *et al*., 2019) (Figura 3B). Existe uma série de coeficientes matemáticos que podem ser utilizados para quantificar a similaridade estrutural dos compostos (Quadro 2).

Figura 3 – (A) Representação de um *fingerprint* hipotético dos compostos ibuprofeno e cetoprofeno. (B) Cálculo do respectivo coeficiente de Tanimoto, estabelecendo, a partir da fórmula, que a molécula de ibuprofeno possui 60% de similaridade estrutural se comparada a moléculas de cetoprofeno.



Fonte: Elaboração dos autores (2023).

Quadro 2 – Principais coeficientes matemáticos usados para relações de similaridade

Coeficientes	Expressões matemáticas	Coeficientes	Expressões matemáticas
Coeficiente de combinação simples	$\frac{(c+d)}{(a+b+c+d)}$	Rogers/Tanimoto	$\frac{(c+d)}{(2a+2b+c+d)}$
Tanimoto (Jaccard)	$\frac{c}{(a+b-c)}$	Baroni-Urbani/Buser	$\frac{\left(\sqrt{cd}+c\right)}{\sqrt{cd}+a+b+c}$
Cosine, Ochiai, Carbo	$\frac{c}{\sqrt{ab}}$	Kulczynski-2	$\frac{1}{2}\left(\frac{c}{a+c}+\frac{c}{b+c}\right)$
Dice (Sørensen)	$\frac{2c}{(a+b)}$	Distância de Soergel	$\frac{(a+b-2c)}{(a+b-c)}$
Forbes	$\frac{cm}{ab}$	Distância de Hamming (Distancia de Manhattan)	$a+b-2c$
Russell/Rao	$\frac{c}{(a+b+c+d)}$	Distância Euclidiana	$\sqrt{(a+b-2c)}$
Legenda			
a = nº características presentes no composto A e ausentes no composto B b = nº características presentes no composto B e ausentes no composto A c = nº características comuns aos dois compostos A e B d = nº características ausentes em ambos os compostos m = nº total de *bits* presentes nos *fingerprints*			

Fonte: Elaboração dos autores (2023).

3.2 Triagem virtual por farmacóforo

O conceito de farmacóforo, elaborado pela União Internacional de Química Pura e Aplicada (IUPAC), pode ser compreendido como "o conjunto de características eletrônicas e estéricas necessárias para garantir interações supramoleculares ótimas com um alvo biológico específico e para ativar (ou bloquear) sua resposta biológica" (Wermuth *et al*., 1998). Esse conceito encontra-se relacionado a múltiplas técnicas de quimioinformática, como: triagem virtual por farmacóforo baseado em ligantes; triagem virtual utilizando farmacóforos obtidos a partir da estrutura 3D do alvo farmacológico (técni-

ca conhecida também como farmacóforo reverso); emprego do farmacóforo para obter *fingerprints*; e utilização de pontos farmacofóricos para auxiliar o *docking* (Sliwoski *et al*., 2014; Schaller *et al*., 2020). Daremos uma atenção mais especial às técnicas relacionadas ao farmacóforo clássico obtido por meio da comparação de três ou mais ligantes.

O processo de obtenção do farmacóforo inicia-se pela geração dos confôrmeros compostos ativos. Muitas vezes a conformação 3D de um composto pode ser obtida a partir de um complexo cristalográfico da molécula com seu alvo (conformação bioativa). Em outros casos, existe a necessidade de explorar o espaço conformacional do composto, de modo a se obter um maior número de conformações para comparação (Sliwoski *et al*., 2014). Esse processo apresenta uma das maiores limitações para geração do modelo farmacofórico, uma vez que a conformação bioativa, nesse caso, é desconhecida (Langer; Wolber, 2004). Em seguida, os confôrmeros são alinhados de acordo com algoritmos que consideram diferentes graus de flexibilidade: alinhamento rígido, semiflexível e flexível. Por fim, o programa extrai as características comuns (grupos hidrofóbicos, doadores/aceptores de ligação de hidrogênio, grupos aromáticos, grupos carregados positivamente/negativamente) e avalia a pontuação conforme critérios que podem envolver: número de características comuns, sobreposição de volume, energia, dentre outros (Leach *et al*., 2010; Sliwoski *et al*., 2014).

3.3 Relação estrutura-atividade quantitativa (QSAR)

Antes de introduzirmos o conceito de QSAR, precisaremos definir a ideia basal de relação estrutura-atividade (REA), a qual sustenta e norteia o campo da química medicinal/farmacêutica. Segundo a IUPAC, a REA pode ser entendida como "a associação entre os aspectos específicos da estrutura molecular e uma ação biológica definida" (Nordberg; Duffus; Templeton, 2004).

Uma das primeiras observações referentes a esse tema remonta à pesquisa de Crum-Brown e Fraser, publicada em 1869, sobre a mudança nas atividades farmacológicas de certos compostos após alterações químicas (Zavod; Knittel, 2012). Tais descobertas foram importantes para demonstrar que a estrutura molecular influencia na atividade biológica desempenhada por um composto e que uma mudança na estrutura molecular pode levar a uma mudança na atividade. Sendo assim, estudos de REA são elaborados para verificar quais grupos químicos são importantes para a atividade e quais podem ser modificados, de modo a melhorar a eficácia e a segurança dos compostos. Segundo a IUPAC, QSAR pode ser definido como: "relações matemáticas que ligam a estrutura química e a atividade farmacológica numa forma quantitativa para uma série de compostos" (Van De Waterbeemd *et al.*, 1997).

Os estudos de QSAR, portanto, são modelos matemáticos desenvolvidos de modo a estabelecer uma relação entre os dados químicos dos compostos e determinados desfechos biológicos (atividade). Tais modelos são construídos por meio da utilização de duas classes de métodos: (a) supervisionados e (b) não supervisionados. Tais métodos podem ser oriundos da estatística ou provenientes de aprendizado de máquina (Quadro 3). Os métodos supervisionados utilizam os valores da variável dependente (atividade) para a criação do modelo, enquanto para os métodos não supervisionados essa informação não é necessária. Em virtude da disponibilidade das variáveis de desfecho, os métodos supervisionados são mais utilizados em QSAR.

Quadro 3 – Principais métodos utilizados para a elaboração de estudo de QSAR.

Tipo de variável	**Método supervisionado**	**Método não supervisionado**
Variável dependente contínua	Regressão: – Linear (MLR) – Polinomial – Minimos quadrados parciais Árvores de decisão (DT) Florestas randômicas (RF)	Agrupamento e redução de dimensionalidade: – Análise de componentes principais (PCA) – Análise de clusters hierárquico (HCA) – *k-means* – Mapeamento não linear (NLM) – Mapeamento de Kohonen (KM)
Variável dependente discreta/ categórica	Classificação – Análise discriminante linear (LDA); – Késimo vizinho mais próximo (kNN); – Máquina de vetores de suporte (SVM) – Árvores de decisão (DT) – Redes Neurais (NN)	Análises de associação – *A priori* – Padrão de associação frequente (FP-Growth) Modelo oculto de Markov

Fonte: Elaboração dos autores (2023).

O estudo de QSAR inicia-se pela transformação dos dados químicos das moléculas em variáveis numéricas (descritores moleculares ou descritores químicos) que expressam um conjunto de propriedades físico-químicas apresentadas pelo composto. Os descritores podem ser classificados conforme os níveis de representação molecular (dimensionalidade) utilizados para obtê-los (Quadro 4) (Rodrigues *et al*., 2012; Sliwoski *et al*., 2014; Danishuddin; Khan, 2016). No entanto, é importante frisar que nem todos os descritores calculados serão utilizados para a obtenção do modelo de QSAR, sendo muitas vezes necessária a utilização de um método de redução de variáveis para aperfeiçoá-lo (Shahlaei, 2013), assim como as variáveis de desfecho podem precisar de tratamento prévio

(normalização e transformações logarítmicas). Em seguida, utilizam-se programas para produzir e testar os modelos conforme os diferentes métodos apresentados anteriormente.

Quadro 4 – Principais classes de descritores utilizados no QSAR.

Representação molecular (dimensionalidade)	**Características**	**Principais descritores**
Constitucionais (1D)	Obtidos simplesmente pela informação química da molécula	Massa molecular (MW), fórmula molecular, número de átomos/ grupos funcionais
Topológicos (2D)	Obtidos por meio de grafos moleculares; informam a conectividade interatômica	Índice de Zagreb, índice de Wiener, vetores de autocorrelação 2D, descritores BCUT
Geométricos (3D)	Obtidos a partir das coordenadas 3D dos átomos	WHIM, MoRSE, GETAWAY, vetores de autocorrelação 3D
Eletrônicos (3D)	Descrevem aspectos eletrônicos da molécula	Energia dos orbitais HOMO-LUMO, potencial eletrostático
Campos de interação molecular (3D)	Mapeamento das possíveis interações favoráveis e desfavoráveis à atividade	CoMFA, CoMSIA, GRID

Fonte: Elaboração dos autores (2023).

Para a realização dos estudos de QSAR, existe a necessidade de os compostos químicos apresentarem elevado grau de pureza (idealmente compostos enantiopuros), bem como de os dados biológicos serem obtidos preferencialmente por um mesmo estudo ou mensurados pelo mesmo método. Além disso, estabelece-se que o número de moléculas estudadas seja o maior possível (divididas entre grupo treinamento e teste/ validação), e que elas apresentem adequadas representatividade biológica (três ordens de magnitude: moléculas inativas, fracamente ativas, moderadamente ativas e fortemente ativas) e diversidade estrutural. Um estudo interessante mostra 21

causas mais comuns de erros em QSAR (Dearden; Cronin; Kaiser, 2009).

Em 2004, a Organização para Cooperação e Desenvolvimento Econômico (OECD) reuniu especialistas para criar orientações para a validação de modelos de QSAR para propósitos regulatórios (OECD, 2004). Foram definidos cinco princípios que norteiam os estudos de QSAR: (1) desfecho definido; (2) algoritmo claro e não ambíguo; (3) domínio de aplicabilidade definido; (4) métricas apropriadas; e (5) interpretação mecanística (quando possível).

4. Métodos de avaliação do desempenho da triagem virtual

O maior problema da utilização da triagem virtual é a escolha de um limite de seleção de um conjunto de compostos ideais para testes experimentais. Normalmente, essa seleção é realizada por meio da análise de resultados da triagem virtual retrospectiva para um conjunto de compostos ativos conhecidos e compostos inativos putativos (*decoys*), ou compostos conhecidamente inativos. Os *decoys* são físico-quimicamente semelhantes, no entanto, topologicamente diferentes dos compostos ativos (Wallach; Lilien, 2011) e podem ser obtidos em servidores como DUD-E (Mysinger *et al*., 2012) e programas como Decoyfinder (Cereto-Massagué *et al*., 2012).

As ferramentas comumente utilizadas para poder preditivo do método utilizado na triagem virtual são o Fator de Enriquecimento (*Enrichment Factor* – EF) e a Área sob a Curva ROC (*Area Under Curve ROC* – ROC-AUC). O EF mede quantos compostos mais ativos são encontrados dentro de uma fração de uma lista classificada, em comparação a uma seleção aleatória, pode ser é definido como (Truchon; Bayly, 2007):

Equação 2 – Cálculo do fator de enriquecimento.

$$EF = \frac{Hits_a}{N_a} / \frac{Hits_t}{N_t}$$

Em que:

$Hits_a$ é o número de compostos ativos contidos na amostra da base de dados;

$Hits_t$ é o número total de compostos ativos da base de dados;

N_a é o número de compostos da mostra e N_t é o número total de compostos da base de dados.

Apesar da sua facilidade de cálculo, o EF possui alguns problemas devido à dependência do número de verdadeiros positivos e verdadeiros negativos, tal como o valor de corte, fazendo dessa ferramenta mais uma medida de desempenho experimental do que método (Truchon; Bayly, 2007). Ao contrário, a ROC-AUC não depende do número de ativos e inativos, ou da razão entre eles (Truchon; Bayly, 2007). A curva ROC é um gráfico construído pela sensibilidade (*Se*) (Equação 2) ou pela especificidade (*Sp*) (Equação 3). ROC-AUC, assim como o EF, representa a probabilidade de um composto ativo escolhido aleatoriamente estar melhor colocado na lista ordenada do que um composto inativo também escolhido aleatoriamente (Clark; Webster-Clark, 2008).

Equações 3 e 4 – Cálculos para obtenção da sensibilidade (*Se*) e da especificidade (*Sp*).

$$Se = \frac{TP}{TP+FN}$$

$$Sp = \frac{TN}{TN+FP}$$

Em que:

TP é o número de verdadeiros positivos;

FN falsos negativos;

FP falsos positivos;

TN verdadeiros negativos.

A área sob a curva, que em muitos casos varia entre 0,5 (classificação aleatória) e 1,0 (classificação perfeita), é uma medida objetiva do desempenho geral do método, que independe do limite selecionado.

A ROC-AUC também possui um viés denominado "reconhecimento precoce" (*early recognition*), que é evidenciado especialmente quando as bibliotecas de compostos são grandes e apenas uma pequena fração dos compostos pode ser testada (Truchon; Bayly, 2007). Devido a esse viés, outras ferramentas foram elaboradas para a avaliação do poder preditivo do método utilizado na triagem virtual. Somente duas dessas ferramentas serão descritas: o aprimoramento inicial robusto (*Robust Initial Enhancement* – RIE) e a discriminação aprimorada por Boltzmann do ROC (*Boltzmann-enhanced discrimination of ROC* – BEDROC).

O RIE considera a classificação completa da lista de compostos ativos e inativos bem como utiliza uma ponderação das posições decrescente, dependendo do parâmetro α. A Equação 5 mostra como o RIE é calculado (ZHAO *et al.*, 2009).

Equação 5 – Cálculo do aprimoramento inicial robusto

Em que:

$$RIE = \frac{\sum_{i=1}^{n} e^{-\alpha r_i}}{\left\langle \sum_{i=1}^{n} e^{-\alpha r_i} \right\rangle_r}$$

r_i é a classificação relativa (ou seja, a classificação dividida pelo tamanho da lista de classificação);

$1/\alpha$ é a fração da lista mais importante para a pontuação final, semelhante ao ponto de corte.

O denominador é o valor médio quando os ativos são distribuídos aleatoriamente na lista de classificação.

Assim como RIE, a BEDROC é uma ferramenta em que os compostos ativos são ponderados dependendo da sua posição no *ranking* usando um parâmetro α (distribuição de Boltzmann), variando de 1 para o composto com melhor classificação para próximo a 0 (Zhao *et al.*, 2009). Além disso, há uma correlação direta entre RIE e BEDROC, como pode ser visto na Equação 6 (Truchon; Bayly, 2007).

Equação 6 – Relação de RIE com BEDROC.

$$BEDROC = \frac{RIE - RIE_{min}}{RIE_{max} - RIE_{min}}$$

Para BEDROC, o parâmetro α é determinante do estreitamento da faixa na lista de classificação dos compostos. Para valores altos de α, os compostos ativos devem ser encontrados mais no topo da lista, sendo o valor de BEDROC próximo de 1 (com um valor de $\alpha = 20$, por exemplo, os compostos ativos devem ser encontrados nos primeiros 8% da lista de classificação, enquanto com um valor de $\alpha = 50$ devem ser encontrados nos primeiros 3%). Essas métricas são adequadas quando a razão entre o número de compostos ativos e o total de compostos lista multiplicados pelo parâmetro α é muito menor que 1 (Truchon; Bayly, 2007). Apesar de a RIE e a BEDROC conseguirem lidar melhor com o reconhecimento precoce, a presença do parâmetro α nessas métricas é um fator de complicação a mais na análise de desempenho, uma vez que comparar valores de RIE e BEDROC para parâmetros α diferentes é errado. Além disso, conjuntos com diferentes proporções entre compostos ativos e inativos não podem ser

comparados diretamente usando essas métricas (Truchon; Bayly, 2007).

Além dessas, existem outras métricas para a avaliação da triagem por QSAR, como *Bootstrapping*, *leave-one-out* (LOO), *leave-N-out* (LNO) e Q2, que servem, basicamente, para verificar a qualidade da regressão linear (Wold; Eriksson; Clementi, 1995; Van De Waterbeemd; Rose, 2008). Além disso, as métricas de índice κ, sensibilidade, especificidade, precisão e acurácia são utilizadas para avaliar modelos de classificação (Van De Waterbeemd; Rose, 2008).

Não há métrica adequada para todos os ensaios de triagem virtual. Cada caso deve ser analisado antes da utilização de qualquer uma das métricas citadas. Além disso, nenhum método automatizado substitui a avaliação visual dos resultados. Dessa forma, um olhar treinado é sem dúvidas o melhor método.

5. Banco de pequenas moléculas

Apesar de ser possível planejar um novo composto e testá-lo computacionalmente, essa prática requer conhecimentos prévios de síntese orgânica para avaliar a sua viabilidade sintética. Dessa maneira, a maioria dos estudos de triagem virtual utiliza bancos de compostos previamente sintetizados. Existem muitos desses bancos (públicos e privados) que contém entre dezenas de milhares e milhões de entradas de compostos químicos (Southan; Várkonyi; Muresan, 2009). Entre os bancos de dados públicos mais utilizados, estão: PubChem (Wang *et al*., 2009), ZINC (Irwin *et al*., 2012), BindingDB (Liu; Jiang; Li, 2007) e ChEMBL (Gaulton *et al*., 2012). O ZINC possui estruturas de compostos previamente virtualizadas e disponíveis comercialmente. Já o PubChem, o BindingDB e o ChEMBL possuem informações de atividade biológica. Nas bases de dados BindingDB e ChEMBL, as atividades dos compostos podem ser classificadas, o que é muito útil para o *benchmarking* de novas metodologias computacionais (Banegas-Luna; Cerón-Carrasco; Pérez-Sánchez, 2018).

6. Conclusão

Quando utilizada de forma adequada, a triagem virtual de pequenas moléculas pode ser uma ferramenta muito útil, trazendo economia financeira e otimização do tempo. O progresso computacional e dos algoritmos de processamento de dados ainda não substituem o *know-how* do modelista no sucesso da triagem virtual, sendo essa uma metodologia amplamente utilizada por grupos de pesquisa e grandes laboratórios farmacêuticos que buscam prospectar e desenvolver moléculas com potencial terapêutico. Atualmente, o conceito de desenvolvimento racional de fármacos incorpora a utilização de recursos computacionais para a criação e o aprimoramento de compostos, sendo inúmeros os exemplos de fármacos desenvolvidos (*e.g.* Imatinib, Tamiflu, etc.). No entanto, é fundamental o contínuo entendimento de quais são as forças que guiam a interação de uma pequena molécula com o seu alvo molecular e de algoritmos e funções mais eficientes na simulação dessa interação.

Referências

BANEGAS-LUNA, A.-J.; CERÓN-CARRASCO, J. P.; PÉREZ-SÁNCHEZ, H. A review of ligand-based virtual screening web tools and screening algorithms in large molecular databases in the age of big data. **Future Medicinal Chemistry**, v. 10, n. 22, p. 2.641-2.658, 2018.

CAROLI, A.; BALLANTE, F.; WICKERSHAM III, R.B.; CORELLI, F.; RAGNO, R. Hsp90 inhibitors, part 2: combining ligand-based and structure-based approaches for virtual screening application. **Journal of Chemical Information and Modeling**, v. 54, n. 3, p. 970-977, 2014.

CERETO-MASSAGUÉ, A.; GUASCH, L.; VALLS, C.; MULERO, M.; PUJADAS, G.; GARCIA-VALLVÉ, S. DecoyFinder: an easy-to-use python GUI application for building target-specific decoy sets. **Bioinformatics**, v. 28, n. 12, p. 1.661-1.662, 2012.

CERETO-MASSAGUÉ, A.; OJEDA, M.J.; VALLS, C.; MULERO, M.; GARCIA-VALLVÉ, S.; PUJADAS, G. Molecular fingerprint similarity search in virtual screening. **Methods**, v. 71, p. 58-63, 2015.

CLARK, P.; GRANT, J.; MONASTRA, S.; JAGODZINSKI, F.; STREINU, I. Periodic rigidity of protein crystal structures. *In*: IEEE INTERNATIONAL CONFERENCE ON COMPUTATIONAL ADVANCES IN BIO AND MEDICAL SCIENCES (ICCABS), 2., 2012, Las Vegas. **Proceedings of the 2012 IEEE 2nd International Conference on Computational Advances in Bio and medical Sciences (ICCABS)**. Las Vegas: Institute of Electrical and Electronics Engineers (IEEE), 2012. p. 1-6.

CLARK, R. D.; WEBSTER-CLARK, D. J. Managing bias in ROC curves. **Journal of Computer-Aided Molecular Design**, v. 22, n. 3-4, p. 141-146, 2008.

DA SILVA ROCHA, S. F. L.; OLANDA, C.G.; FOKOUE, H.H.; SANT'ANNA, C.M. Virtual Screening Techniques in Drug Discovery: Review and Recent Applications. **Current Topics in Medicinal Chemistry**, v. 19, n. 19, p. 1.751-1.767, 2019.

DAEYAERT, F.; DE JONGE, M.; HEERES, J.; KOYMANS, L.; LEWI, P.; VINKERS, M.H.; JANSSEN, P.A. A pharmacophore docking algorithm and its application to the cross-docking of 18 HIV-NNRTI's in their binding pockets. **Proteins:** Structure, Function, and Bioinformatics, v. 54, n. 3, p. 526-533, 2004.

DALLAKYAN, S.; OLSON, A. J. Small-Molecule Library Screening by Docking with PyRx. *In*: HEMPEL, J. E.; WILLIAMS, C. H.; HONG, C. C. (Eds.). **Chemical Biology:** Methods and Protocols. New York: Humana Press, 2015. p. 243-250.

DANISHUDDIN, NULL; KHAN, A. U. Descriptors and their selection methods in QSAR analysis: paradigm for drug design. **Drug Discovery Today**, v. 21, n. 8, p. 1291-1302, 2016.

DEARDEN, J. C.; CRONIN, M. T. D.; KAISER, K. L. E. How not to develop a quantitative structure-activity or structure-property relationship (QSAR/QSPR). **SAR and QSAR in environmental research**, v. 20, n. 3-4, p. 241-266, 2009.

GAULTON, A.; BELLIS, L.J.; BENTO, A.P.; CHAMBERS, J.; DAVIES, M.; HERSEY, A.; LIGHT, Y.; MCGLINCHEY, S.; MICHALOVICH, D.; AL-LAZIKANI, B.; OVERINGTON, J.P. ChEMBL: a large-scale bioactivity database for drug discovery. **Nucleic Acids Research**, v. 40, n. D1, p. D1100-D1107, 2012.

GIMENO, A.; OJEDA-MONTES, M.J.; TOMÁS-HERNÁNDEZ, S.; CERETO-MASSAGUÉ, A.; BELTRÁN-DEBÓN, R.; MULERO, M.; PUJADAS, G.; GARCIA-VALLVÉ, S. The Light and Dark Sides of Virtual Screening: What Is There to Know? **International Journal of Molecular Sciences**, v. 20, n. 6, p. 1.375, 2019.

GUEDES, I. A.; DE MAGALHÃES, C. S.; DARDENNE, L. E. Receptor-ligand molecular docking. **Biophysical Reviews**, v. 6, n. 1, p. 75-87, 2014.

HUANG, S.-Y.; ZOU, X. Advances and challenges in protein-ligand docking. **International Journal of Molecular Sciences**, v. 11, n. 8, p. 3.016-3.034, 2010.

IRWIN, J. J.; STERLING, T.; MYSINGER, M.M.; BOLSTAD, E.S.; COLEMAN, R.G. ZINC: a free tool to discover chemistry for biology. **Journal of Chemical Information and Modeling**, v. 52, n. 7, p. 1.757-1.768, 2012.

LANGER, T.; WOLBER, G. Pharmacophore definition and 3D searches. **Drug Discovery Today:** Technologies, v. 1, n. 3, p. 203-207, 2004.

LEACH, A. R.; GILLET, V.J.; LEWIS, R.A.; TAYLOR, R. Three-dimensional pharmacophore methods in drug discovery. **Journal of Medicinal Chemistry**, v. 53, n. 2, p. 539-558, 2010.

LIU, T.; LIN, Y.; WEN, X.; JORISSEN, R.N.; GILSON, M.K. BindingDB: a web-accessible database of experimentally determined protein-ligand binding affinities. **Nucleic Acids Research**, v. 35, n. suppl_1, p. D198-D201, 2007.

LIU, X.; JIANG, H.; LI, H. SHAFTS: a hybrid approach for 3D molecular similarity calculation. 1. Method and assessment of virtual screening. **Journal of Chemical Information and Modeling**, v. 51, n. 9, p. 2.372-2.385, 2011.

MACIELAG, M. J. Chemical Properties of Antimicrobials and Their Uniqueness. *In*: DOUGHERTY, T. J.; PUCCI, M. J. (Eds.). **Antibiotic Discovery and Development**. Boston: Springer, 2012. p. 793-820.

MAGGIORA, G.; VOGT, M.; STUMPFE, D.; BAJORATH, J. Molecular similarity in medicinal chemistry. **Journal of Medicinal Chemistry**, v. 57, n. 8, p. 3.186-3.204, 2014.

MYSINGER, M. M.; CARCHIA, M.; IRWIN, J.J.; SHOICHET, B.K. Directory of useful decoys, enhanced (DUD-E): better ligands and decoys for better benchmarking. **Journal of Medicinal Chemistry**, v. 55, n. 14, p. 6582-6594, 2012.

ORGANISATION FOR ECONOMIC CO-OPERATION AND DEVELOPMENT (OECD). OECD principles for the validation, for regulatory purposes, of (quantitative) structure-activity relationship models. *In:* MEETING OF THE CHEMICALS COMMITTEE AND WORKING PARTY ON CHEMICALS,

PESTICIDES AND BIOTECHNOLOGY, 37., 2004, Paris. **Anais** [...]. Paris: Organisation for Economic Co-operation and Development (OECD), 2004.

NORDBERG, M.; DUFFUS, J.; TEMPLETON, D. M. Glossary of terms used in toxicokinetics (IUPAC Recommendations 2003). **Pure and Applied Chemistry**, v. 76, n. 5, p. 1.033-1.082, 2004.

PAGADALA, N. S.; SYED, K.; TUSZYNSKI, J. Software for molecular docking: a review. **Biophysical Reviews**, v. 9, n. 2, p. 91-102, 2017.

ROBERT, A.; BENOIT-VICAL, F.; LIU, Y.; MEUNIER, B. Small Molecules: The Past or the Future in Drug Innovation? *In*: CARVER, P.L. (Ed.). **Essential Metals in Medicine:** Therapeutic Use and Toxicity of Metal Ions in the Clinic. Berlin: De Gruyter, 2019. p. 17-48.

RODRIGUES, R. P.; MANTOANI, S.P.; DE ALMEIDA, J.R.; PINSETTA, F.R.; SEMIGHINI, E.P.; DA SILVA, V.B.; DA SILVA, C.H.T. Estratégias de Triagem Virtual no Planejamento de Fármacos. **Revista Virtual de Química**, v. 4, n. 6, p. 739-776, 2012.

ROSENFELD, R.; VAJDA, S.; DELISI, C. Flexible docking and design. **Annual Review of Biophysics and Biomolecular Structure**, v. 24, p. 677-700, 1995.

SASTRY, G. M.; ADZHIGIREY, M.; DAY, T.; ANNABHIMOJU, R.; SHERMAN, W. Protein and ligand preparation: parameters, protocols, and influence on virtual screening enrichments. **Journal of Computer-Aided Molecular Design**, v. 27, n. 3, p. 221-234, 2013.

SCHALLER, D.; ŠRIBAR, D.; NOONAN, T.; DENG, L.; NGUYEN, T.N.; PACH, S.; MACHALZ, D.; BERMUDEZ, M.; WOLBER, G. Next generation 3D pharmacophore modeling. **WIREs Computational Molecular Science**, v. 10, n. 4, p. e1468, 2020.

SHAHLAEI, M. Descriptor Selection Methods in Quantitative Structure-Activity Relationship Studies: A Review Study. **Chemical Reviews**, v. 113, n. 10, p. 8.093-8.103, 2013.

SHIN, W.-H.; ZHU, X.; BURES, M.G.; KIHARA, D. Three-dimensional compound comparison methods and their application in drug discovery. **Molecules**, v. 20, n. 7, p. 12.841-12.862, 2015.

SLIWOSKI, G.; KOTHIWALE, S.; MEILER, J.; LOWE, E.W. Computational methods in drug discovery. **Pharmacological Reviews**, v. 66, n. 1, p. 334-395, 2014.

SOUTHAN, C.; VÁRKONYI, P.; MURESAN, S. Quantitative assessment of the expanding complementarity between public and commercial databases of bioactive compounds. **Journal of Cheminformatics**, v. 1, n. 1, p. 10, 2009.

STAHURA, F. L.; BAJORATH, J. Virtual screening methods that complement HTS. **Combinatorial Chemistry & High Throughput Screening**, v. 7, n. 4, p. 259-269, 2004.

TOVCHIGRECHKO, A.; WELLS, C. A.; VAKSER, I. A. Docking of protein models. **Protein Science:** A Publication of the Protein Society, v. 11, n. 8, p. 1.888-1.896, 2002.

TRUCHON, J.-F.; BAYLY, C. I. Evaluating virtual screening methods: good and bad metrics for the "early recognition" problem. **Journal of Chemical Information and Modeling**, v. 47, n. 2, p. 488-508, 2007.

VAN DE WATERBEEMD, H.; CARTER, R.E.; GRASSY, G.; KUBINYI, H.; MARTIN, Y.C.; TUTE, M.S.; WILLETT, P. Glossary of terms used in computational drug design (IUPAC Recommendations 1997). **Pure and Applied Chemistry**, v. 69, n. 5, p. 1.137-1.152, 1997.

VAN DE WATERBEEMD, H.; ROSE, S. Quantitative Approaches to Structure-Activity Relationships. *In*: WERMUTH, C. G. (Ed.). **The Practice of Medicinal Chemistry**. 3. ed. New York: Academic Press, 2008. p. 491-513.

WALLACH, I.; LILIEN, R. Virtual Decoy Sets for Molecular Docking Benchmarks. **Journal of Chemical Information and Modeling**, v. 51, n. 2, p. 196-202, 2011.

WANG, Y.; XIAO, J.; SUZEK, T.O.; ZHANG, J.; WANG, J.; BRYANT, S.H. PubChem: a public information system for analyzing bioactivities of small molecules. **Nucleic Acids Research**, v. 37, p. W623-W633, 2009.

WERMUTH, C. G.; GANELLIN, C.R.; LINDBERG, P.; MITSCHER, L.A. Glossary of terms used in medicinal chemistry (IUPAC Recommendations 1998). **Pure and Applied Chemistry**, v. 70, n. 5, p. 1.129-1.143, 1998.

WILLETT, P. Similarity-based approaches to virtual screening. **Biochemical Society Transactions**, v. 31, n. 3, p. 603-606, 2003.

WOLD, S.; ERIKSSON, L.; CLEMENTI, S. Statistical Validation of QSAR Results. *In*: VAN DE WATERBEEMD, H. (Ed.). **Chemometric Methods in Molecular Design**. Weinheim: VCH, 1995. p. 309-338.

YANG, Z.-Y.; HE, J.H.; LU, A.P.; HOU, T.J.; CAO, D.S. Frequent hitters: nuisance artifacts in high-throughput screening. **Drug Discovery Today**, v. 25, n. 4, p. 657-667, 2020.

ZAVOD, R. M.; KNITTEL, J. J. Drug design and relationship of functional groups to pharmacologic activity. *In*: LEMKE, T. L.; WILLIAMS, D. A.; ROCHE, V. F.; ZITO, S. W. (Eds.). **Foye's Principles of Medicinal Chemistry**. 7. ed. Philadelphia: Wolters Kluwer Health/Lippincott Williams & Wilkins, 2012. p. 29-60.

ZHAO, W.; HEVENER, K.E.; WHITE, S.W.; LEE, R.E.; BOYETT, J.M. A statistical framework to evaluate virtual screening. **BMC bioinformatics**, v. 10, p. 225, 2009.

Capítulo 4

Métodos de detecção de seleção natural com dados genômicos

Henrique Vieira Figueiró[14]

1. Introdução

Seleção natural é a força evolutiva mais conhecida dentro da biologia. Sua importância foi primeiramente reconhecida por Charles Darwin e Alfred Wallace de forma independente, mas desde então esse conceito foi sendo desenvolvido e aprimorado com a evolução do campo (Darwin, 1859). Novas áreas como a genética, a biotecnologia e, mais recentemente, a genômica têm utilizado análises de avaliação de seleção natural de forma cada vez mais elaborada. Os resultados observados são fundamentais em estudos de resistência a antibióticos, adaptação humana e tomada de decisão para conservação de espécies, apenas para citar alguns exemplos (Allendorf; Hohenlohe; Luikart, 2010; Bouzat, 2010; Park *et al.*, 2012).

Apesar de muito difundido, o conceito de seleção natural ainda é alvo de muito debate, dentro e fora da academia. Seleção natural é o fenômeno de indivíduos que apresentam alguma vantagem adaptativa no ambiente que habitam e, consequentemente, deixam mais descendentes, fazendo com que tal característica, ao longo de gerações, acabe se fixando na população. Com o passar do tempo, ela foi se tornando mais elaborada, e os mecanismos fisiológicos, ecológicos e genéticos foram sendo esclarecidos. A primeira distinção necessária é que quando falamos em seleção natural geralmente estamos associando-a ao termo "seleção positiva". Com o avanço da

[14] Laboratório de Genômica e Biologia Molecular, Escola de Ciências da Saúde e da Vida, Pontifícia Universidade Católica do Rio Grande do Sul.

genética foi possível perceber três tipos distintos de seleção: (*i*) positiva, (*ii*) negativa e (*iii*) neutra ou balanceadora.

Seleção positiva, também conhecida como seleção darwiniana, é quando uma mutação altera um fenótipo que traz vantagens reprodutivas a um indivíduo, provocando o aumento de frequência dessa mutação na população ao longo de gerações. Modos de detecção de seleção positiva se baseiam no fato de que grande parte do genoma é neutro, ou seja, sem efeito de seleção. A partir disso, fica estabelecido um modelo neutro para comparação com possíveis regiões que se encontram sob seleção positiva. Tanto métodos populacionais – por exemplo, aqueles baseados em frequência alélica – quanto métodos filogenéticos – baseados nas alterações de códon – utilizam o modelo neutro como base de comparação.

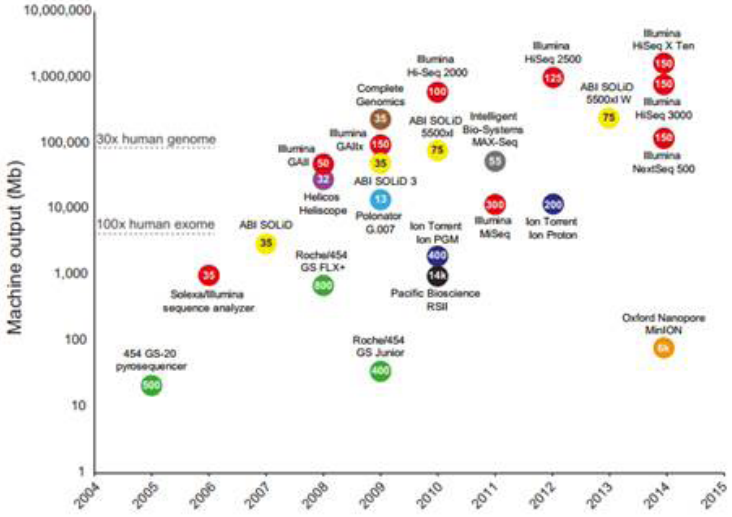
Seleção negativa ou purificadora é quando uma mutação desfavorável é mantida em baixa frequência pela pressão do ambiente. A remoção de alelos deletérios é um fenômeno importante para a sobrevivência de populações e pode ter consequências importantes na variabilidade das espécies. Um efeito da seleção negativa é a chamada seleção de fundo, ou seja, a remoção da variabilidade de regiões ligadas fisicamente a mutações deletérias que são expurgadas da população. Espécies com alta taxa de mutação e baixa taxa de recombinação são mais propensas a sofrer a perda de variabilidade causada por seleção de fundo.

Por fim, seleção balanceadora é a manutenção de mutações favoráveis na população, geralmente dando preferência à manutenção de sítios heterozigotos. Esse é um dos modos de seleção de mais difícil detecção. Na ausência de seleção, dizemos que o sítio é neutro, ou seja, nenhum alelo tem preferência, e as alterações em sua frequência são causadas por outras forças evolutivas.

Outro ponto deste capítulo diz respeito ao uso de dados genômicos. Até o início dos anos 2000, o acesso a genomas completos era algo completamente impensável. Essa limitação fazia com que o número de regiões genômicas analisadas

fossem muito limitadas. Com exceção de organismos procariotos, a maioria dos trabalhos de seleção natural em organismos com genomas maiores acabava se limitando a poucos genes candidatos. Esse cenário foi alterado com o surgimento de sequenciadores de alto desempenho, capazes de sequenciar múltiplos genomas completos em uma questão de dias (Figura 1).

Figura 1 – Evolução da capacidade de sequenciamento. O eixo X indica o ano de lançamento do sequenciador, enquanto o eixo Y indica a capacidade de geração de dados em megabases.



Fonte: Reuter, Spacek e Snyder (2015).

De forma resumida, esses aparelhos têm como base a fragmentação do genoma (de forma mecânica ou enzimática), a ligação de adaptadores, a amplificação e a leitura em sistema de imagens de alta definição que registram a leitura das bases do DNA ao longo de inúmeros ciclos (Meyer; Kircher, 2010; Van Dijk; Jaszczyszyn; Thermes, 2014). Após o sequenciamento, sequências curtas de DNA, em torno de

100 pares de base (pb), são mapeadas e alinhadas para formação do genoma do alvo. Uma limitação é a taxa de erro ainda considerada alta para sequências repetitivas, pois, por serem sequências curtas, torna-se difícil estimar o começo e o fim de tais regiões. Métodos mais recentes com capacidade de sequenciamento de regiões longas do genoma (milhares de bases) aprimoraram muito os resultados obtidos (Reuter; Spacek; Snyder, 2015).

Considerando isso, escolher regiões candidatas não é mais necessário, e podemos analisar o padrão de seleção ao longo do genoma completo. Para isso, existem diversos métodos propostos. A sua escolha vai depender da pergunta a ser respondida e do tipo de dado analisado. Análises de seleção positiva podem ser feitas em nível de indivíduo, população ou espécie (Nielsen, 2005). Isso significa que podem englobar diferentes intervalos de tempo, desde uma escala curta, como no caso de vírus, até milhões de anos em análises comparativas entre famílias ou ordens distintas. Os próximos parágrafos irão descrever alguns desses métodos e explicar para quais tipos de dados são recomendados.

Análises com foco em populações têm como principal objetivo a identificação de padrões de adaptação local. Esse tipo de análise avalia alterações na frequência alélica de uma população em relação a outras. Para isso é feita a polarização dos dados, em que se identifica quais são os alelos ancestrais e derivados. Quando uma população apresenta uma frequência alélica diferente das demais, é possível que o fenótipo expressado por aquela região gênica esteja sob seleção. É importante destacar que alterações na frequência alélica podem ser causadas por outras forças evolutivas, como a deriva genética. Então, também é importante que outras análises sejam feitas para descartar esse tipo de evento.

Existem algumas métricas que avaliam alterações de frequência, como diferenciação populacional (F_{ST}) ou diversidade genética (D de Tajima e diversidade nucleotídica) (Tajima, 1989; Akey *et al.*, 2002). A diferenciação populacional

é uma medida de fluxo gênico entre populações: quanto maior o valor médio, maior o isolamento. No caso de trabalhos que quantificam seleção natural, é verificada a diferenciação ao longo do genoma, sendo que valores altos muito superiores à média indicam que o *locus* em questão se encontra sob seleção.

Outra forma de identificar sinais de seleção é por meio da quantificação de diversidade genética de uma determinada região. O genoma possui um valor médio de diversidade, e desvios para baixo são um dos indicativos de seleção. Quando uma mutação benéfica ocorre, ela acaba fazendo com que todo o bloco de recombinação de que ela faz parte fique fixado. Ao longo do tempo, esse bloco acaba se fixando na população como um todo, causando a perda de diversidade genética para aquele bloco. Esse fenômeno é conhecido como varredura seletiva. Programas como Plink, VCFtools e ANGSD são alguns dos mais usados para análises desse tipo (Purcell *et al*., 2007; Danecek *et al*., 2011; Korneliussen; Albrechtsen; Nielsen, 2014).

Um exemplo bastante conhecido é de um gene envolvido no controle da hipóxia em humanos, o qual apresentou sinais de seleção positiva em populações no Tibete e, mais recentemente, nos Andes. Os pesquisadores analisaram alterações no espectro de frequência utilizando dados de exoma, que é o conjunto de éxons (região expressa do gene). Eles compararam populações de regiões montanhosas contra populações de outras regiões e observaram alterações na frequência alélica do gene EPAS1 (Yi *et al.*, 2010). Além disso, esse foi um dos primeiros trabalhos a estudar seleção natural incluindo DNA antigo.

Outro método amplamente utilizado que foca na região codificante do genoma mensura a razão entre mutações sinônimas (dS) e não sinônimas (dN) ($\omega = dN/dS$). Aqui vale um lembrete de biologia molecular básica. Mutações sinônimas são aquelas que não alteram a função da proteína, enquanto mutações não sinônimas são aquelas em que uma alteração no DNA vai causar alteração do aminoácido traduzido a

partir do códon. Como mutações desse tipo tem uma baixa frequência, esse método é destinado para mutações já fixadas em uma espécie. Por esse motivo, o método é utilizado em genômica comparativa, quando diversas espécies são analisadas ao mesmo tempo, geralmente do ponto de vista filogenético. A interpretação dos valores obtidos pode ser vista na Tabela 1.

Tabela 1 – Valores do teste de seleção e sua interpretação.

Valor de ω	Interpretação
$\omega > 1$	Seleção positiva
$\omega = 1$	Evolução neutra
$\omega < 1$	Seleção negativa

Fonte: Elaboração do autor (2023).

Esse método vem sendo utilizado desde o início do sequenciamento tradicional, com genes candidatos. Além disso, esse teste também era usado na seleção de genes neutros para uso em inferência filogenética. O programa mais utilizado para essa análise é o PAML (Yang, 2007). Vale citar também o programa HYPHY, que utiliza os mesmos princípios, mas conta com mais opções de modelos evolutivos para teste (Kosakovsky Pond *et al*., 2020).

Apesar de parecer simples, esse método emprega princípios estatísticos robustos para validação dos resultados. A utilização de modelos evolutivos que comparam eventos neutros com eventos seletivos talvez seja o principal deles. O princípio desse método é que dN/dS segue uma distribuição estatística entre os sítios amostrados. Quando a distribuição que permite $dN/dS>1$, por exemplo, se encaixa significativamente melhor aos dados do que aquela que não permite, isso quer dizer que os dados apresentam seleção positiva (Nielsen, 2005; Yang, 2007). Para testar a significância, é feito um teste de *likelihood ratio*, em que se compara um modelo nulo, sem seleção, com o observado, com seleção. Baseado em uma tabela de qui-quadrado com um determinado grau de liberdade, o valor de significância é estimado.

2. Teste de seleção para genômica comparativa

O guia a seguir descreve uma rotina de análises para detecção de seleção em nível de espécie dentro de uma filogenia. As ferramentas computacionais necessárias para os procedimentos se encontram no Quadro 1. Uma versão dessa rotina de análise foi utilizada no artigo "Genome-wide signatures of complex introgression and adaptive evolution in the big cats" (Figueiró *et al.*, 2017).

Quadro 1 – Programas necessários para execução da rotina de análise.

Programa*	**Endereço eletrônico**	**Referência**
BWA	bio-bwa.sourceforge.net/	(Li; Durbin, 2009)
ANGSD	www.popgen.dk/angsd/	(Korneliussen; Albrechtsen; Nielsen, 2014)
gffread	github.com/gpertea/gffread	
ete3	etetoolkit.org	(Huerta-Cepas; Dopazo; Gabaldón, 2010)
PAML	abacus.gene.ucl.ac.uk/software/paml.html	(Yang, 2007)
Fasttree	www.microbesonline.org/fasttree/	
padjust	Pacote do R	
*Todos os programas rodam em sistema Linux.		

Fonte: Elaboração do autor (2023).

As análises podem ser desempenhadas com base nos passos descritos no Quadro 2 e utilizando os seguintes arquivos necessários: genoma referência no formato fasta; sequências curtas (*reads*) dos organismos de interesse no formato fastq; arquivo de anotação com as regiões de interesse no formato gff3; e filogenia do grupo-alvo no formato *newick* (notação

parentética). Além disso, pode ser feita a escolha do modelo evolutivo para o teste de seleção (Figura 2).

Quadro 2 – Passo a passo para execução do *workflow* de análise de seleção

1. Mapear as *reads* das espécies de interesse utilizando um genoma completo com boa notação das regiões codificantes como referência.

2. Gerar o consenso no formato fasta de cada espécie no programa ANGSD, com o comando "*-dofasta*". Utilizar filtros de cobertura conforme as características do dado analisado.

3. Com o programa gffread e a anotação do genoma referência, extrair a região codificante (CDS) mais longa de cada gene para cada uma das espécies analisadas.

4. Caso não exista uma filogenia estimada na literatura para o grupo de estudo, é necessário fazê-lo. Programas como fastrree e RAxML são os mais indicados. Importante ressaltar que não é necessário enraizar a árvore, e todos as espécies analisadas devem estar presentes no alinhamento com o mesmo nome.

5. O teste de seleção é feito no programa ete3, que roda o programa PAML por meio de uma interface gráfica.

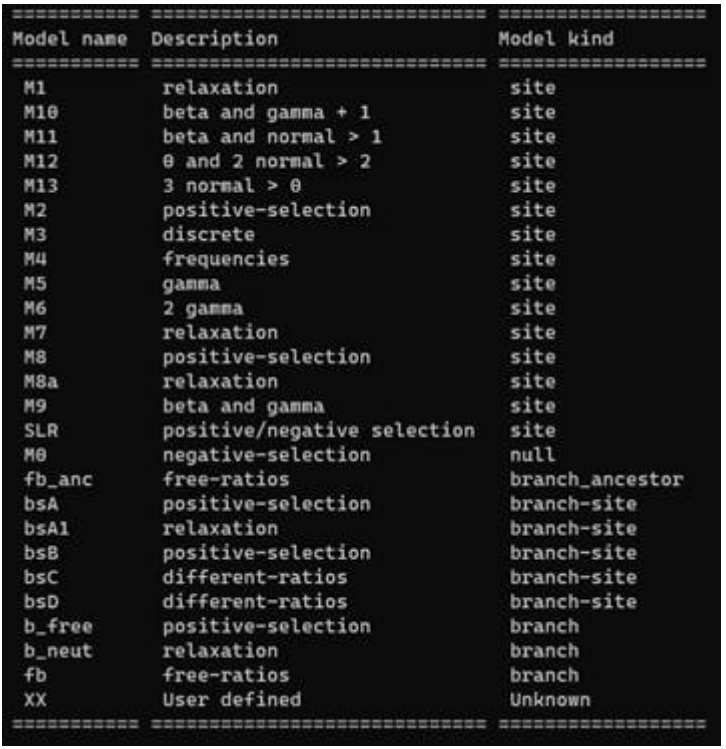
   a) Os arquivos de entrada são a árvore filogenética e o alinhamento do gene. Além disso, deve ser escolhido um modelo evolutivo e um organismo alvo caso seja o objetivo.

   b) É importante destacar que devem sempre ser escolhidos dois modelos, um neutro e outro que assume seleção, ambos do mesmo tipo de teste (M7 vs. M8 ou bsA vs. bsA1, por exemplo).

   c) Quando múltiplos genes são analisados ao mesmo tempo, é possível fazer um *script* na linguagem *bash* para rodar um por um.

Fonte: Elaboração do autor (2023).

Figura 2 – Opções de modelo evolutivo para o teste de seleção. *Site* indica testes que analisam apenas os sítios do alinhamento, sem levar em conta a filogenia. *Branch-site* considera tanto o alinhamento quanto a filogenia. *Branch* dá um peso maior para as relações filogenéticas. *Branch ancestor* leva em consideração o ancestral nas comparações.



Model name	Description	Model kind
M1	relaxation	site
M10	beta and gamma + 1	site
M11	beta and normal > 1	site
M12	0 and 2 normal > 2	site
M13	3 normal > 0	site
M2	positive-selection	site
M3	discrete	site
M4	frequencies	site
M5	gamma	site
M6	2 gamma	site
M7	relaxation	site
M8	positive-selection	site
M8a	relaxation	site
M9	beta and gamma	site
SLR	positive/negative selection	site
M0	negative-selection	null
fb_anc	free-ratios	branch_ancestor
bsA	positive-selection	branch-site
bsA1	relaxation	branch-site
bsB	positive-selection	branch-site
bsC	different-ratios	branch-site
bsD	different-ratios	branch-site
b_free	positive-selection	branch
b_neut	relaxation	branch
fb	free-ratios	branch
XX	User defined	Unknown

Fonte: Elaboração do autor (2023).

O resultado é apresentado na tela ou pode ser enviado para um arquivo. Deve-se extrair os valores de significância e realizar uma correção, visto que foram realizados múltiplos testes. O pacote p.adjust no R pode ser utilizado, basta utilizar como entrada a lista de valores. O teste de correção a ser utilizado é o *false discovery rate* (FDR).

Referências

AKEY, J. M.; ZHANG, G.; ZHANG, K.; JIN, L.; SHRIVER, M.D. Interrogating a high-density SNP map for signatures of natural selection. **Genome Research**, v. 12, n. 12, p. 1.805-1.814, 2002.

ALLENDORF, F. W.; HOHENLOHE, P. A.; LUIKART, G. Genomics and the future of conservation genetics. **Nature Reviews Genetics**, v. 11, n. 10, p. 697-709, 2010.

BOUZAT, J. L. Conservation genetics of population bottlenecks: the role of chance, selection, and history. **Conservation Genetics**, v. 11, n. 2, p. 463-478, 2010.

DANECEK, P.; AUTON, A.; ABECASIS, G.. ALBERS, C.A.; BANKS, E.; DEPRISTO, M.A.; HANDSAKER, R.E.; LUNTER, G.; MARTH, G.T.; SHERRY, S.T.; MCVEAN, G. The variant call format and VCFtools. **Bioinformatics**, v. 27, n. 15, p. 2.156-2.158, 2011.

DARWIN, C. **On the Origin of Species**. London: John Murray, 1859.

FIGUEIRÓ, H. V.; LI, G.; TRINDADE, F.J.; ASSIS, J.; PAIS, F.; FERNANDES, G.; SANTOS, S.H.; HUGHES, G.M.; KOMISSAROV, A.; ANTUNES, A.; TRINCA, C.S. Genome-wide signatures of complex introgression and adaptive evolution in the big cats. **Science Advances**, v. 3, n. 7, p. e1700299, 2017.

HUERTA-CEPAS, J.; DOPAZO, J.; GABALDÓN, T. ETE: a python Environment for Tree Exploration. **BMC bioinformatics**, v. 11, p. 24, 2010.

KORNELIUSSEN, T. S.; ALBRECHTSEN, A.; NIELSEN, R. ANGSD: Analysis of Next Generation Sequencing Data. **BMC bioinformatics**, v. 15, p. 356, 2014.

KOSAKOVSKY POND, S. L.; POON, A.F.; VELAZQUEZ, R.; WEAVER, S.; HEPLER, N.L.; MURRELL, B.; SHANK, S.D.; MAGALIS, B.R.; BOUVIER, D.; NEKRUTENKO, A.; WISOTSKY, S. HyPhy 2.5-A Customizable Platform for Evolutionary Hypothesis Testing Using Phylogenies. **Molecular Biology and Evolution**, v. 37, n. 1, p. 295-299, 2020.

LI, H.; DURBIN, R. Fast and accurate short read alignment with Burrows-Wheeler transform. **Bioinformatics**, v. 25, n. 14, p. 1754-1760, 2009.

MEYER, M.; KIRCHER, M. Illumina sequencing library preparation for highly multiplexed target capture and sequencing.

Cold Spring Harbor Protocols, v. 2010, n. 6, p. pdb.prot5448, 2010.

NIELSEN, R. Molecular signatures of natural selection. **Annual Review of Genetics**, v. 39, p. 197-218, 2005.

PARK, D. J.; LUKENS, A.K.; NEAFSEY, D.E.; SCHAFFNER, S.F.; CHANG, H.H.; VALIM, C.; RIBACKE, U.; VAN TYNE, D.; GALINSKY, K.; GALLIGAN, M.; BECKER, J.S. Sequence-based association and selection scans identify drug resistance loci in the Plasmodium falciparum malaria parasite. **Proceedings of the National Academy of Sciences**, v. 109, n. 32, p. 13.052-13.057, 2012.

PURCELL, S.; NEALE, B.; TODD-BROWN, K.; THOMAS, L.; FERREIRA, M.A.; BENDER, D.; MALLER, J.; SKLAR, P.; DE BAKKER, P.I.; DALY, M.J.; SHAM, P.C. PLINK: a tool set for whole-genome association and population-based linkage analyses. **American Journal of Human Genetics**, v. 81, n. 3, p. 559-575, 2007.

REUTER, J. A.; SPACEK, D. V.; SNYDER, M. P. High-throughput sequencing technologies. **Molecular Cell**, v. 58, n. 4, p. 586-597, 2015.

TAJIMA, F. Statistical method for testing the neutral mutation hypothesis by DNA polymorphism. **Genetics**, v. 123, n. 3, p. 585-595, 1989.

VAN DIJK, E. L.; JASZCZYSZYN, Y.; THERMES, C. Library preparation methods for next-generation sequencing: Tone down the bias. **Experimental Cell Research**, v. 322, n. 1, p. 12-20, 2014.

YANG, Z. PAML 4: phylogenetic analysis by maximum likelihood. **Molecular Biology and Evolution**, v. 24, n. 8, p. 1.586-1.591, 2007.

YI, X.; LIANG, Y.; HUERTA-SANCHEZ, E.; JIN, X.; CUO, Z.X.P.; POOL, J.E.; XU, X.; JIANG, H.; VINCKENBOSCH, N.; KORNELIUSSEN, T.S.; ZHENG, H. Sequencing of 50 human exomes reveals adaptation to high altitude. **Science**, v. 329, n. 5987, p. 75-78, 2010.

Capítulo 5

Clusterização para promotores bacterianos: ferramenta DNA Sequences Clusterizer

Gabriel Dall'Alba[15]

1. Introdução

A pesquisa científica em áreas como a biologia e a medicina encontra-se em uma era chamada de pós-genômica (Marx, 2013), compreendida por meio dos avanços na genômica desde a conclusão do Projeto Genoma Humano em 2003 (Hanson, 2020). De alguns anos para cá, tem-se atrelado mais a noção de pós-genômica a avanços em plataformas de sequenciamento de alto rendimento (*high-throughput sequencing*, como SORT-Seq, MFA-Seq, RNA-Seq, etc.), desenvolvimento de bancos de dados e biobancos mais sofisticados e avanços na tecnologia de microarranjos de DNA (Batrakou *et al*., 2020; Hanson, 2020).

Tais avanços têm contribuído expressivamente para a pesquisa, permitindo o sequenciamento de genomas completos – de bactérias, vírus, plantas e animais – com maior eficiência e menor custo. Contudo, a estrada à frente da genômica ainda está repleta de desafios a serem batidos: a anotação genômica frente a esse cenário ainda é um processo de elevado custo monetário e temporal, bem como complexo na sua essência (Coelho *et al*., 2020), e, sendo feita com o mínimo apoio da bioinformática, pode provar-se um desafio maior do que o imaginado. Na assistência ao combate desses desafios, a bioinformática traz o aporte computacional – com uma diversa gama de técnicas e métodos –, a fim de reduzir os

[15] University of British Columbia.

elevados custos bem como auxiliar na redução de eventuais erros que as novas técnicas ainda trazem consigo.

Um campo bastante explorado no contexto genômico são os esforços computacionais para predizer/identificar elementos gênicos. Nisso estão incluídas a predição de genes no processo de anotação de um genoma e a predição de elementos não codificantes (como sequências promotoras e sítios de ligação de fatores de transcrição) (Dominguez Del Angel *et al*., 2018). Além disso, a expressão gênica e a sua regulação são importantes objetos de estudo para potenciais descobertas em áreas como a medicina, a biotecnologia e a biologia molecular (Coelho *et al*., 2020; Dall'alba *et al*., 2019).

Não existe uma única fórmula para enfrentamento do desafio de mapear os elementos gênicos de um genoma recém-sequenciado ou investigar padrões e modelos dentro de um conjunto de dados biológicos. Abordagens baseadas em redes neurais artificiais, s*upport vector machines*, clusterização, árvores de decisão, entre outras, são bem-vindas e encontradas em larga escala na literatura – para revisões no assunto, ver Singh, Kaur e Goel (2015), Chen e Zhang, (2014) e Goés *et al*. (2014).

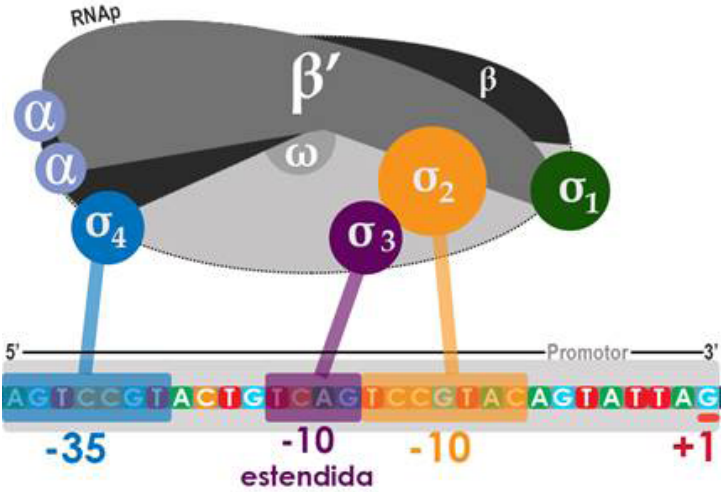
Neste capítulo, o foco será destinado à clusterização não hierárquica para promotores bacterianos, utilizando o algoritmo *k-means* na elucidação de possíveis padrões entre as sequências promotoras. Nas próximas seções, a técnica de clusterização e os promotores bacterianos serão descritos, seguidos da ferramenta que será abordada como exemplo, com sua execução e apontamentos finais.

2. Clusterização para promotores bacterianos

Antes de entendermos os conceitos de clusterização, é importante compreender o dado biológico que será trabalhado aqui. Promotores bacterianos – que podem também ser chamados de sequências promotoras ou simplesmente promotores – são pequenas sequências de DNA localizadas a algumas dezenas de pares de base anteriores (*upstream*) ao

sítio de início de transcrição (TSS) (Figura 1). Nessas pequenas sequências, ocorre uma etapa fundamental da transcrição gênica: a interação entre uma enzima denominada RNA Polimerase (RNAP) e a sequência promotora. O promotor é um dos elementos gênicos responsáveis por permitir que a RNAP – responsável pela transcrição – consiga atracar na fita de DNA. Em bactérias, é classicamente caracterizado por duas regiões consensuais de nucleotídeos conservados entre diferentes promotores, localizadas a 10 e 35 pares de base distantes do TSS – sendo, assim, denominadas -10 e -35 (Krebs; Goldstein; Kilpatrick, 2017).

Figura 1 – Representação esquemática e simplificada de um promotor bacteriano e uma holoenzima RNA Polimerase com as suas cinco subunidades principais ($\alpha_2\beta\beta'\omega$) e os diferentes sítios de ligação da subunidade σ. Cada sítio de ligação faz referência a um sítio de interação entre a holoenzima e a sequência promotora. Além das ligações que aparecem na imagem, as subunidades α interagem com elementos UP e σ_1 com sequências discriminadoras localizadas *downstream* da região -10. À direita (*downstream*) do nucleotídeo "+1" está a região codificante do gene.



Fonte: Adaptado de Ruff, Record e Artsimovitch (2015).

Essa interação entre as duas partes só é possível por intermédio de uma pequena subunidade enzimática denominada sigma (σ). A RNAP recruta uma subunidade disponível no meio intracelular e, por ação desta, é guiada até o promotor específico. Na bactéria Gram-negativa modelo, *Escherichia coli*, sete distintos fatores σ são conhecidos. Amplamente investigado, o σ^{70} (também conhecido como *housekeeping factor* ou sigma vegetativo/constitutivo) está relacionado a genes essenciais de manutenção da célula. Além desse, seis fatores chamados de alternativos compõem a gama de subunidades encontradas na bactéria: σ^{24}, σ^{28}, σ^{32}, σ^{38}, σ^{54} e σ^{19}. Enquanto os genes relacionados ao sigma vegetativo estão ligados à manutenção da célula, os genes relacionados aos sigmas alternativos são muitas vezes responsáveis por respostas específicas da bactéria ao meio onde ela está, incluindo funções de assimilação de nitrogênio (σ^{54}), transporte de ferro (σ^{19}), motilidade e patogenicidade (σ^{28}) e resposta a estresses por choque térmico (σ^{24} e σ^{32}) (Dall'alba *et al.*, 2019; Davis *et al.*, 2017).

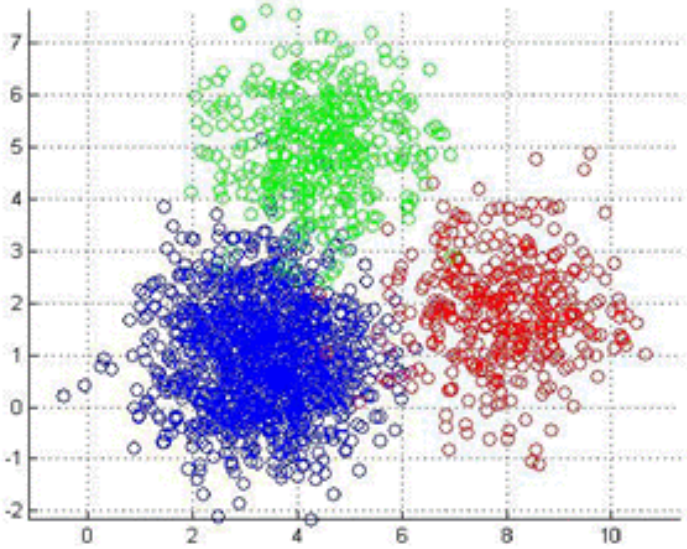
2.1 A técnica de clusterização

Clusterização (Figura 2) é uma técnica de classificação não supervisionada, recomendada quando não há classes em um conjunto de dados a serem preditas, mas é possível dividir suas instâncias em grupos naturais, ou clusters (Witten; Frank; Hall, 2011). Em outras palavras, a clusterização é interessante quando buscamos tendências ou padrões dentro de um conjunto de dados – com o devido cuidado de não confundir a busca por padrões com o reconhecimento de padrões (Basso, 2015). Para Fontana (2013, p. 19), a clusterização pode ser explicada como:

> A análise de cluster, conhecida também como clusterização (clustering), é uma tarefa de mineração de dados que consiste em um processo de analisar dados, através de seus atributos, objetivando a identificação de padrões para a organização de conjuntos físicos ou abstratos de objetos similares (também denominados

elementos ou componentes). Através dela, é possível criar agrupamentos que identificam uma classe de objetos similares entre si e dissimilares entre as classes. Estas classes de objetos, ou grupos de objetos similares, são denominadas cluster. A clusterização é uma tarefa não supervisionada, ou seja, não necessita de um aprendizado prévio para a identificação de padrões, diferentemente da classificação convencional utilizada em redes neurais (inteligência artificial), na qual é necessário treinar a rede com um escopo resumido de dados para que a esta possa identificar padrões em dados em escopos diferentes. Esta forma, auto-organizável de selecionar os dados, determina uma aprendizagem por observação, na qual o algoritmo aprende observando um único escopo dados, ao mesmo tempo em que o classifica. Em consequência disso, ao iniciar o processo de análise, não é necessário determinar etiquetas (classes) previamente conhecidas.

Figura 2 – Exemplificação de três clusters posicionados em um plano.



Fonte: Fontana (2013, p. 31).

2.2 O algoritmo k-means

O algoritmo *k-means* é não hierárquico e requer um valor de entrada denominado *k*, juntamente com o conjunto de dados a ser processado, sendo *k* o número de agrupamentos (ou clusters) gerados. Recebendo essa informação, o algoritmo atribui a *k* objetos a função de centróides iniciais. Essa atribuição pode ser heurística, arbitrária (e dada pelo usuário) ou feita respeitando alguma técnica específica (Fontana, 2013). Em seguida, cada entrada do conjunto de dados tem sua distância dos centróides calculadas e é alocada a um cluster de acordo com a menor distância encontrada. Após esse procedimento, cada cluster tem seu centróide recalculado a partir da posição média de suas entradas e a operação anterior de cálculo das distâncias é repetida com os novos centróides. O processo segue várias iterações, verificando se houve a alocação de um ou mais objetos para algum cluster diferente. Quando não houver mais alterações de clusters (ou atingir-se um valor *T* de iterações, caso este tenha sido informado pelo usuário) finaliza-se a execução do algoritmo.

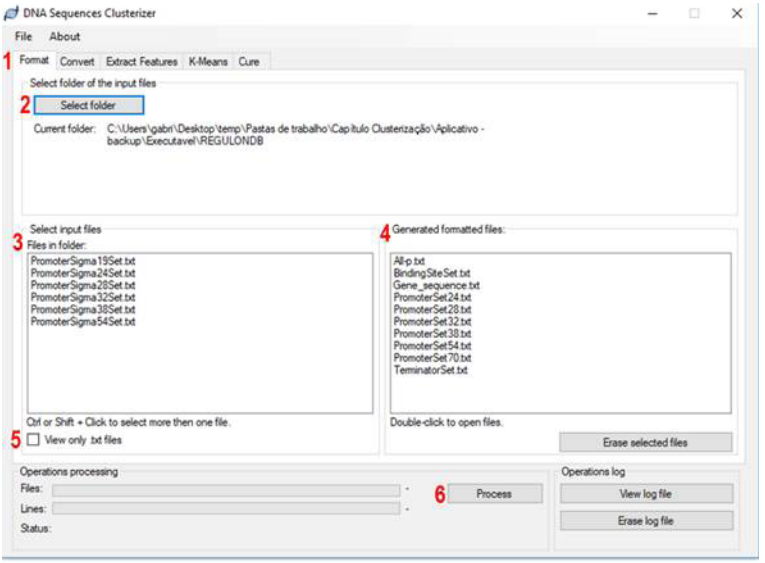
A atribuição do valor *k*, tanto na pesquisa em promotores bacterianos quanto em outras aplicações, se dá por conhecimento prévio do objeto de estudo. Uma metodologia comumente empregada para formalizar a determinação do *k* leva o nome de *Elbow*: um método heurístico em que a variância explicada em função do número de clusters é plotada em um gráfico de eixo X = número de clusters e Y = percentual de variância explicada (Kodinariya; Makwana, 2013). Determina-se o *k* ótimo no ponto em que há a estabilização da curva, de maneira similar se compararmos com a curva do coletor na prática de biologia de campo.

3. Exemplo de uso

Para exemplificar a clusterização aplicada a promotores bacterianos, a ferramenta DNA Sequences Clusterizer (Figura 3) (Fontana, 2013; Basso, 2015) será utilizada. Será feito o passo a passo de uma análise utilizando dados rela-

cionados a promotores da bactéria Gram-negativa *E. coli* extraídos do banco de dados RegulonDB.

Figura 3 – Interface da ferramenta DNA Sequences Clusterizer. (1) Abas de execução da clusterização por meio da ferramenta. (2) Seleção de pasta com dados a serem trabalhados (recomendado fazer uma pasta em diretório próximo ao da ferramenta em si). (3) Exibição dos arquivos de *input* (localizados na pasta indicada pelo item 2). (4) Visualização da pasta de arquivos formatados. (5) Opção de exibir apenas arquivos .txt (de nosso interesse). (6) Botão para efetuar o processamento dos arquivos de *input*. Os arquivos processados aparecem na aba do item 4 – arquivos formatados.



Fonte: Elaboração do autor (2023).

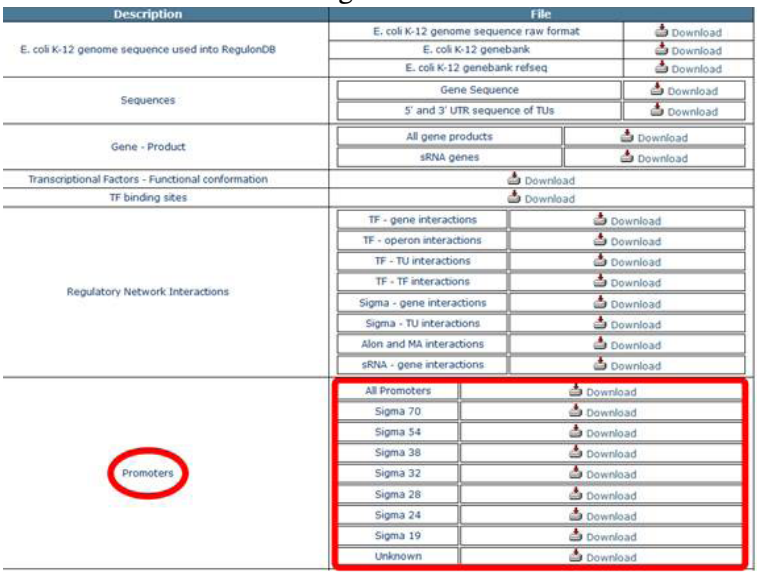
3.1 A hipótese a ser testada

Queremos testar a hipótese de que as características naturais entre sequências promotoras relacionadas a diferentes fatores σ são suficientes para separá-las em clusters distintos: um cluster para cada fator σ.

3.2 Extração de dados do RegulonDB

Acessando o RegulonDB (Gama-Castro *et al.*, 2016) pelo link http://regulondb.ccg.unam.mx/, é preciso navegar até os arquivos de texto dos promotores de interesse localizados no menu *Experimental Datasets*, dentro da aba de *Downloads* (Figura 4).

Figura 4 – Conjuntos de dados para promotores de *E. coli* disponíveis em *Experimental Datasets* no banco de dados RegulonDB.



Fonte: Elaboração do autor (2023).

Figura 5 – Exemplo de arquivo com um conjunto de promotores baixado do RegulonDB.

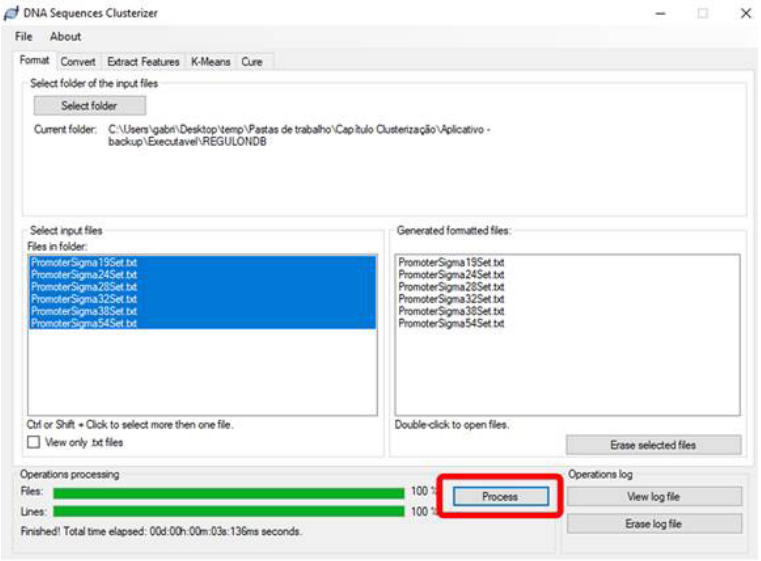
Fonte: Elaboração do autor (2023).

O arquivo baixado do banco de dados (no nosso exemplo, o "PromoterSigma54Set.txt") deve ter a mesma aparência que na imagem acima (Figura 5). Para o exemplo, serão utilizados os arquivos dos σ^{19}, σ^{24}, σ^{28}, σ^{32}, σ^{38} e σ^{54}, totalizando 1.306 sequências promotoras. No RegulonDB todas as sequências possuem um tamanho padrão de 81 nucleotídeos.

3.3 Formatação dos arquivos

Com a ferramenta aberta, deve-se ter certeza de que a pasta de arquivos a serem trabalhados (ver Figura 3, item 2) contém os arquivos.txt extraídos do RegulonDB mencionados ao fim da seção anterior. Selecionados todos os arquivos da aba da esquerda (Shift + Clique), deve-se clicar em *Process* (Figura 6). Nessa etapa ocorre a formatação dos dados, transformando os nucleotídeos "A", "T", "G" e "C" em respectivos valores numéricos de acordo com a metodologia descrita na literatura de referência da ferramenta.

Figura 6 – Formatação de arquivos na ferramenta DNA Sequences Clusterizer.



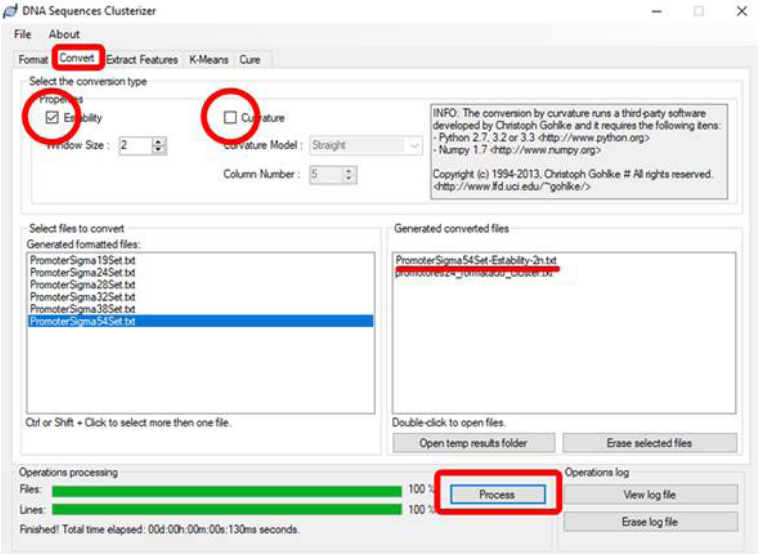
Fonte: Elaboração do autor (2023).

3.4 Conversão para estabilidade ou curvatura

Esse passo é opcional, recomendado caso o usuário opte por fazer a clusterização de promotores bacterianos utilizando codificações de dados em valores de estabilidade ou curvatura, não simplesmente em nucleotídeos. Essencialmente, as sequências promotoras são transformadas em valores numéricos calculados com base nas propriedades físicas do DNA. O princípio de investigação permanece o mesmo: encontrar padrões e possíveis inferências de relevância para o entendimento dessas sequências nos mecanismos biológicos em que elas participam. Para realizar essa operação, é necessário avançar para a aba *Convert*, selecionar a caixa que indica curvatura ou estabilidade e determinar os demais parâmetros de interesse (*e.g.* a janela de nucleotídeos a ser calculada e o modelo de curvatura a ser usado) (Figura 7). Um conhecimento prévio sobre as metodologias de estabilidade e curvatura é importan-

te para a melhor compreensão da execução desse passo (para mais informações dentro dessa temática, conferir o capítulo 7 do presente e-book).

Figura 7 – Passo opcional de conversão das sequências promotoras em valores de estabilidade ou curvatura.

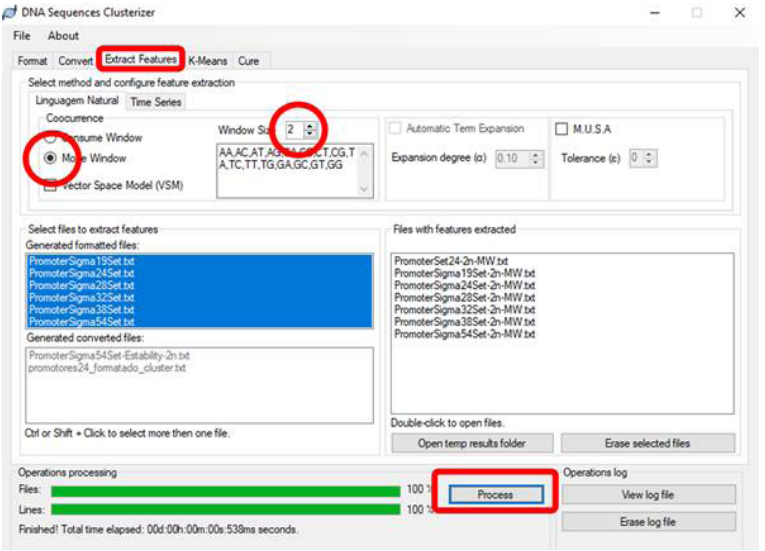


Fonte: Elaboração do autor (2023).

3.5 Extração de atributos

Uma série de diferentes parâmetros pode ser escolhida na aba *Extract Features*. Para nosso exemplo, ficaremos com as seguintes opções: *Coocurrence*, *Move Window* e um *Window Size* = 2 para todos os arquivos (Figura 8). Se optar por usar a opção *Time Series*, o usuário perceberá que a segunda janela de arquivos está liberada, não a primeira. Uma série temporal só pode ser utilizada se a conversão para valores de curvatura, estabilidade ou outro parâmetro similar for utilizado.

Figura 8 – Parâmetros para extração de atributos.

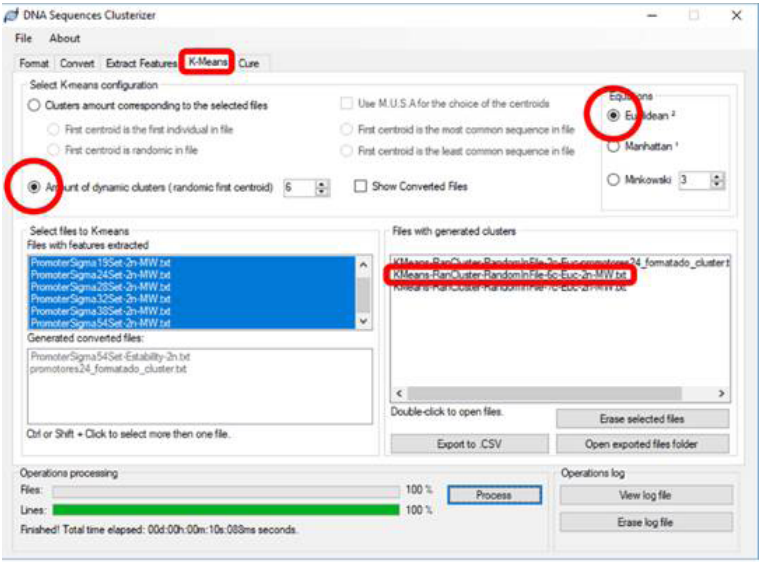


Fonte: Elaboração do autor (2023).

3.6 Clusterização com k-means

Por fim, faz-se a seleção da aba com o algoritmo de clusterização: *k-means* ou CURE. Aqui, utilizaremos o *k-means* e os seguintes parâmetros: *Amount of dynamic clusters* = 6 e o cálculo da distância será euclidiana (Figura 9). O parâmetro *Amount of dynamic clusters* é o nosso k, sendo atribuído o valor escolhido pelo pesquisador. Aqui, o valor de k utilizado corresponde ao número de fatores sigma incluídos no exemplo. Isso significa que estaremos gerando seis clusters, cada um com seu centróide inicial aleatório. Devem ser selecionados todos os arquivos desejados na análise (no nosso caso, todos que vêm sido trabalhados) para em seguida iniciar-se o processamento dos dados.

Figura 9 – Algoritmo *k-means.*



Fonte: Elaboração do autor (2023).

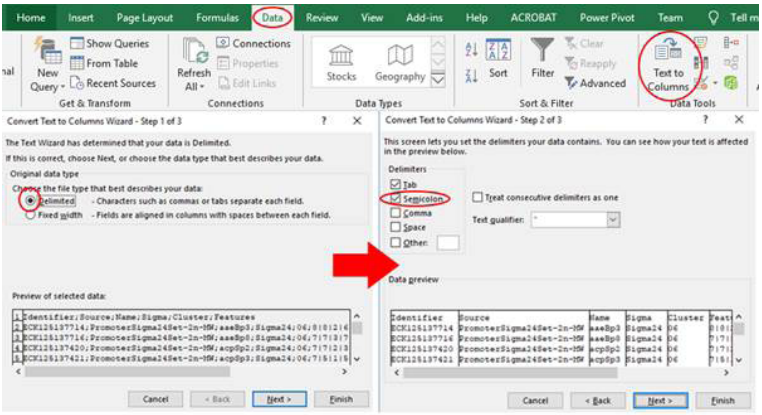
O arquivo de saída leva o nome dos parâmetros que foram utilizados no modelo, seguindo a notação: "Algoritmo utilizado – Parâmetro de quantidade de clusters utilizado – Tipo de determinação de centróide – Número de clusters – Cálculo de distância utilizado – Janela de nucleotídeos na extração de atributos – Tipo de extração de atributos – Extensão do arquivo". Dessa forma, o nosso arquivo final leva o nome de "Kmeans-RanCluster-RandomInFile-6c-Euc-2n-MW.txt". Selecionamos o arquivo de saída na janela da direita e clicamos em "*Export to.CSV*". Encontramos o arquivo na pasta *ExportedFiles*, que é gerada na mesma pasta em que está a ferramenta. O arquivo .csv pode ser aberto com qualquer editor de planilhas eletrônicas.

3.7 Visualização e análise dos resultados

Para facilitar a visualização dos resultados, é importante delimitar as colunas por ponto e vírgula. Utilizando o Microsoft Excel, é possível fazer isso pelo menu *Data* em *Text*

to Columns (texto para colunas), conforme demonstrado na Figura 10. É importante garantir que, antes de fazer isso, toda a coluna A esteja selecionada.

Figura 10. Opções para organização da planilha



Fonte: Elaboração do autor (2023).

A partir disso, é possível filtrar os resultados por cluster ou por fator σ. Devido à sensibilidade do cluster ao centróide e ao fato de termos empregado o *k-means* com centróides aleatórios, é extremamente difícil que os resultados de uma execução sejam iguais aos de outra.

Algumas ferramentas úteis para a visualização dos resultados utilizando promotores bacterianos são a WebLogo (Crooks *et al*., 2004) e a StringDB (Franceschini *et al*., 2012) (Figura 11). Na primeira, pode-se visualizar os motivos consensuais -10, -35, presença de -10 estendido, discriminadores e TSS (+1), testando a hipótese de o critério para a geração de um cluster ser a similaridade entre motivos consensuais de sequências promotoras distintas. Na segunda, pode-se buscar explicações para um dado cluster por intermédio das proteínas dos genes que são regulados pelos promotores encontrados no cluster. Um exemplo de pergunta nessa linha investigativa poderia ser, por exemplo: são todos genes dos promotores de um cluster ligados ao sistema de secreção tipo

III? Ou quimiotaxia? O interessante é que, a partir dos resultados da clusterização, o pesquisador deve inferir por meio da observação e da comparação com a literatura, buscando dar forma e contexto para os padrões que vão sendo identificados.

Figura 11 – Exemplos de visualização de resultado utilizando as ferramentas Weblogo (à esquerda) e StringDB (à direita).

Fonte: Dall'Alba *et al.* (2016).

Na Tabela 1 estão os resultados obtidos seguindo o passo a passo aqui descrito. Nota-se que todos os clusters apresentam sequências relacionadas ao fator σ^{24} em maior quantidade. Ao observarmos atentamente os conjuntos de dados utilizados para esse exemplo, percebemos que o σ^{24} possui mais sequências que os demais fatores σ, e utilizamos apenas seis clusters. Logo, podemos perceber que nossa hipótese não se sustenta, uma vez que um único fator σ diluiu-se para todos os clusters (cenário que se repete para todos os fatores). Isso nos levaria a repensar o problema: esgotadas as explicações de cunho biológico para os resultados apresentados, seis é um número adequado de clusters para esse tipo de investigação? Logo, é hora de voltar e repetir a execução, alterando o valor de *k*.

Tabela 1 – Resultados obtidos no exemplo de clusterização.

Cluster	Quantidade de sequências	Fator σ predominante
1	187	σ^{24} (100/187)
2	142	σ^{24} (50/142)
3	184	σ^{24} (65/184)
4	191	σ^{24} (105/191)
5	233	σ^{24} (99/233)
6	239	σ^{24} (94/239)

Fonte: Elaboração do autor (2023).

4. Conclusão

A clusterização é uma técnica interessante quando se busca um "norte" para o objeto de estudo. Aqui, exploramos a execução do ponto de vista do usuário, usando um dos diversos algoritmos disponíveis. No caso de promotores bacterianos, busca-se compreender as suas particularidades a nível estrutural – particularidades que são toleradas na regulação gênica *in vivo*, mas que geram os mais diversos desafios na hora da pesquisa *in silico*. A heterogeneidade encontrada em promotores bacterianos impede que as abordagens de predição ou anotação genômica voltada para esses elementos sejam de execução trivial, por isso encontra-se mérito na pesquisa de caráter investigativo como a permitida pela clusterização.

Referências

BASSO, T. A. **Clusterização aplicada na análise genômica**. 2015. Trabalho de Conclusão de Curso (Bacharelado em Ciência da Computação) – Universidade de Caxias do Sul, Caxias do Sul, 2015.

BATRAKOU, D.G.; MÜLLER, C.A.; WILSON, R.H.; NIEDUSZYNSKI, C.A. DNA copy-number measurement of genome replication dynamics by high-throughput sequencing: the sort-seq, sync-seq and MFA-seq family. **Nature Protocols**, v. 15, n. 3, p. 1255-1284, 2020.

CHEN, C. L. P.; ZHANG, C.-Y. Data-intensive applications, challenges, techniques and technologies: A survey on Big Data. **Information Sciences**, v. 275, p. 314-347, 2014.

COELHO, R. V.; DALL'ALBA, G.; DE AVILA E SILVA, S.; ECHEVERRIGARAY, S.; DELAMARE, A.P.L. Toward Algorithms for Automation of Postgenomic Data Analyses: Bacillus subtilis Promoter Prediction with Artificial Neural Network. **Omics:** A Journal of Integrative Biology, v. 24, n. 5, p. 300-309, 2020.

CROOKS, G. E.; HON, G.; CHANDONIA, J.M.; BRENNER, S.E. WebLogo: a sequence logo generator. **Genome Research**, v. 14, n. 6, p. 1188-1190, 2004.

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.

DALL'ALBA, G.; DE AVILA E SILVA, S.; ADAMI, A.G.; ECHEVERRIGARAY, S. Análise in silico de promotores de Escherichia coli reconhecidos pelo fator σ28. **SaBios-Revista de Saúde e Biologia**, v. 11, n. 1, p. 31-40, 2016.

DAVIS, M. C.; KESTHELY, C.A.; FRANKLIN, E.A.; MACLELLAN, S.R. The essential activities of the bacterial sigma factor. **Canadian Journal of Microbiology**, v. 63, n. 2, p. 89-99, 2017.

DOMINGUEZ DEL ANGEL, V.; HJERDE, E.; STERCK, L.; CAPELLA-GUTIERREZ, S.; NOTREDAME, C.; PETTERSSON, O.V.; AMSELEM, J.; BOURI, L.; BOCS, S.; KLOPP, C.; GIBRAT, J.F. Ten steps to get started in Genome Assembly and Annotation. **F1000Research**, v. 7, p. 148, 2018.

FONTANA, E. A. **Algoritmos de clusterização aplicados na análise genômica da bactéria Escherichia coli**. 2013. Trabalho de Conclusão de Curso (Bacharelado em Ciência da Computação) – Universidade de Caxias do Sul, Caxias do Sul, 2013.

FRANCESCHINI, A.; SZKLARCZYK, D.; FRANKILD, S.; KUHN, M.; SIMONOVIC, M.; ROTH, A.; LIN, J.; MINGUEZ, P.; BORK, P.; VON MERING, C.; JENSEN, L.J. STRING v9.1: protein-protein interaction networks, with increased coverage and integration. **Nucleic Acids Research**, v. 41, n. D1, p. D808-D815, 2013.

GAMA-CASTRO, S.; SALGADO, H.; SANTOS-ZAVALETA, A.; LEDEZMA-TEJEIDA, D.; MUÑIZ-RASCADO, L.; GARCÍA-SOTELO, J.S.; ALQUICIRA-HERNÁNDEZ, K.; MARTÍNEZ-FLORES, I.; PANNIER, L.; CASTRO-MONDRAGÓN, J.A.; MEDINA-RIVERA, A. RegulonDB version 9.0: high-level integration of gene regulation, coexpression, motif clustering and

beyond. **Nucleic Acids Research**, v. 44, n. D1, p. D133-D143, 2016.

GÓES, F.; ALVES, R.; CORRÊA, L.; CHAPARRO, C.; THOM, L. A Comparison of Classification Methods for Gene Prediction in Metagenomics. *In*: THE INTERNATIONAL WORKSHOP ON NEW FRONTIERS IN MINING COMPLEX PATTERNS (NFMCP), 3., 2014, Nancy, France. **Proceedings of the 3rd Workshop on New Frontiers in Mining Complex Patterns**. Nancy, France: Springer, 2015. p. 136-147.

HANSON, C. **Genetics and the Literary Imagination**. New York: Oxford University Press, 2020.

KODINARIYA, T.; MAKWANA, P. Review on Determining of Cluster in k-means Clustering. **International Journal of Advance Research in Computer Science and Management Studies**, v. 1, n. 6, p. 90-95, 2013.

KREBS, J. E.; GOLDSTEIN, E. S.; KILPATRICK, S. T. **Lewin's genes XII**. Burlington: Jones & Bartlett Learning, 2017.

MARX, V. The big challenges of big data. **Nature**, v. 498, n. 7453, p. 255-260, 2013.

RUFF, E. F.; RECORD, M. T.; ARTSIMOVITCH, I. Initial events in bacterial transcription initiation. **Biomolecules**, v. 5, n. 2, p. 1035-1062, 2015.

SINGH, S.; KAUR, S.; GOEL, N. A Review of Computational Intelligence Methods for Eukaryotic Promoter Prediction. **Nucleosides, Nucleotides & Nucleic Acids**, v. 34, n. 7, p. 449-462, 2015.

WITTEN, I. H.; FRANK, E.; HALL, M. A. **Data mining:** practical machine learning tools and techniques. 3. ed. Burlington: Morgan Kaufmann, 2011.

Capítulo 6

Ferramenta BacPP: predizendo promotores de bactérias Gram-negativas e seus fatores sigma associados

Gustavo Sganzerla Martinez[16]

1. Inteligência Artificial e aprendizado de máquina

A informática é uma ciência relativamente nova, tendo seu início não muito mais do que meio século atrás. Desde o seu início, essa ciência buscava automatizar cálculos que demorariam um grande tempo quando realizados por humanos (Rojas, 1997). Sendo assim, a evolução da computação teve diversos estágios, um deles sendo a introdução do conceito de Inteligência Artificial (IA).

Desde a sua concepção, a IA se pautava em simular processos humanos para a solução de problemas (Russel; Norvig, 2010). Nesse ponto, é necessário separar duas vertentes da ciência da computação que podem ser um tanto confusas: a IA, que busca criar máquinas "humanas", e o aprendizado de máquina, que busca criar máquinas que sejam capazes de aprender. Como exemplo da primeira instância, há sistemas que buscam simular a maneira como um humano conversa por meio de mensagens digitadas, em que os desenvolvedores tentam atribuir um comportamento humano à sua máquina, como emoções e imperfeições. Em segundo lugar, pode ser mencionado o BacPP – alvo deste capítulo –, o qual compreende um sistema que analisa a complexa relação entre componentes do DNA de bactérias para classificá-los; isso apenas se torna possível com a alta escalabilidade que as máquinas empregam, garantindo a elas a capacidade de aprender.

[16] Laboratório de Bioinformática e Biologia Computacional, Instituto de Biotecnologia, Universidade de Caxias do Sul.

2. Uma tarefa simples na natureza e complicada na computação

Regiões promotoras do DNA são o alvo de estudo da ferramenta a ser explorada. Essas regiões são áreas do DNA que antecedem a região codificante – cuja transcrição resultará em uma proteína funcional, garantindo à célula que suas necessidades básicas sejam supridas. Esse tipo de sequência reguladora é fundamental nos três domínios de vida devido à sua função no processo de transcrição de DNA em RNA. Para que tal processo ocorra, uma enzima denominada RNA polimerase DNA dependente (RNAP) irá ler a informação contida apenas na região codificante do DNA e irá sintetizar uma molécula de RNA (Krebs; Goldstein; Kilpatrick, 2014). É importante ressaltar que a RNAP não irá transcrever em RNA a informação genética contida na região promotora. Para que isso ocorra de maneira apropriada, existem aparatos de transcrição que auxiliam a RNAP a identificar a sequência que deve ser transcrita em formato de RNA e a que não. Esses segmentos não sintetizados são as regiões reguladoras, que atuam como sinalizadores de onde começa a região a ser transcrita (promotor) e onde termina (terminador).

Ao analisarmos seres eucariontes e arqueias, há a presença de um conjunto de nucleotídeos TATA localizado na posição -28 *upstream* (ou 28 nucleotídeos antes do +1 ou *Transcription Start Site* – TSS, ou sítio de início de transcrição); embora nem todos os promotores possuam tal sequência de nucleotídeos, indícios biofísicos apontam que essa região é diferente das demais áreas do DNA (Martinez *et al*., 2021b). Já as bactérias possuem mecanismos de transcrição distintos, em que a RNAP conta com cinco subunidades. Uma dessas subunidades é o fator sigma (σ), responsável pela identificação de regiões promotoras que gerarão RNAs transcritos com diferentes funções. A especificidade garantida pelo σ garante que a RNAP seja direcionada de maneira específica, garantindo à célula que sua necessidade seja atendida. Por exemplo, em *Escherichia coli*, o σ^{70} (o número corresponde ao peso mo-

lecular) entra em cena direcionando a enzima RNAP quando genes fundamentais para o funcionamento das atividades celulares precisam ser expressos. Mais além, em *E. coli*, há o favorecimento da presença de nucleotídeos TATAAT localizados na posição -10 e TTGACA na posição -35, garantindo um aspecto conservado ao promotor bacteriano.

Até então, o processo de identificação de promotores parece ser uma tarefa relativamente simples, podendo ser facilmente realizado por computadores. Afinal, um nível elevado na conservação de sequências em determinados segmentos do DNA é o que garante que um promotor seja um promotor.

Entretanto, não é tão simples assim. Bactérias são organismos que habitam os mais diversos meios. Dessa forma, a evolução causou mudanças na localização exata de seus aparatos de transcrição. Ocorrências como mutações, deleções e inserções de nucleotídeos dificultam o processo de localizar promotores *in silico*, requerendo sistemas muito mais complexos do que aqueles que apenas verificam a presença de TATA, TATAAT ou TTGACA em suas determinadas posições. É a área perfeita para a Inteligência Artificial entrar em cena.

3. Um guia prático para predizer promotores

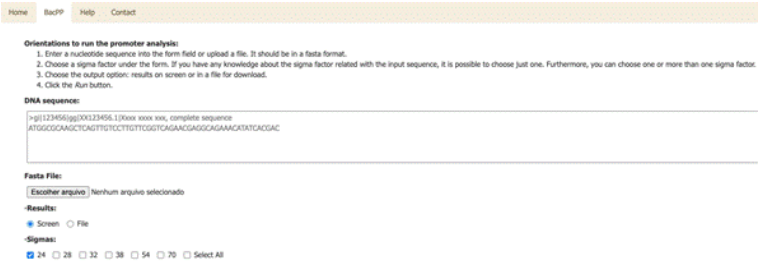
A ferramenta computacional *Bacterial Promoter Prediction* (BacPP) teve início em 2011, e a sua evolução acontece até hoje. O invento surgiu da necessidade de expansão na predição de promotores bacterianos. Na época, apenas a predição de regiões promotoras de *E. coli* associadas à subunidade de RNAP σ^{70} era difundida. Porém, existiam também outras subunidades σ que necessitavam de atenção e poderiam, de fato, aumentar a precisão ao classificar promotores de *E. coli*, um organismo modelo em genética bacteriana. Com essa premissa em mente, o BacPP foi desenvolvido. Em seus primórdios (considerando que o artigo inicial da aplicação foi publicado em 2011), a ferramenta contava apenas com uma versão desenvolvida na linguagem de programação Python, de maneira local. Posteriormente, em 2016, ganhou sua

versão *web* (De Avila e Silva *et al*., 2016), tendo em mente que grande parte do acesso à internet atualmente vem de dispositivos móveis. Nesse tipo de situação, os aplicativos com versão *web* costumam a se sobrepor devido ao seu desprendimento de plataformas específicas, requerendo somente conexão à internet para rodar (ENGE, 2019).

Ao entrar na página do aplicativo (http://www.bacpp.bioinfoucs.com/home), o usuário depara-se com os objetivos da ferramenta e uma segunda aba contendo um FAQ (*Frequently Asked Questions*, questões mais comuns), fornecendo ajuda sobre o funcionamento do aplicativo. A última aba da página inicial contém informação de contato com a equipe de cientistas responsáveis pelo BacPP.

A ferramenta somente pode ser acessada por um cadastro que fornece ao usuário um *login* e oportuniza a criação de uma senha. Essa decisão de desenvolvimento foi tomada visando à possibilidade de fazer um levantamento estatístico dos usuários – outros fatores como país de origem, instituição de afiliação também são considerados. Após o acesso, o usuário ganha uma nova aba, denominada BacPP, na qual é possível iniciar a predição de promotores. A Figura 1 apresenta a tela de execução do BacPP. Primeiramente, o usuário informa a sequência de nucleotídeos que deseja saber se corresponde a um promotor e a qual fator σ. O usuário, além de fazer a entrada manual, pode fazer o *upload* de algum arquivo .FASTA no aplicativo, contendo *n* sequências para serem analisadas. Um ponto importante é a necessidade de trabalhar-se com o formato .FASTA no BacPP – em sequências inseridas manualmente ou com o *upload* de um arquivo contendo as sequências (pela opção "Escolher arquivo"). Tal formato é o padrão em bioinformática para sequências genômicas e peptídicas.

Figura 1 – Tela inicial de execução do BacPP, na qual o usuário cadastrado faz a inserção (manual ou por *upload*) de sua(s) sequência(s) de interesse.



Fonte: Elaboração do autor (2023).

O usuário tem a opção de escolher o formato de saída de sua análise, que pode ser na própria aplicação *web* ou em um arquivo .txt. O usuário também pode informar à aplicação o fator σ da sequência inserida – o que refina mais ainda a busca. Por fim, pelo botão "*Run*", a análise da(s) sequência(s) é feita. É possível filtrar os resultados de acordo com a precisão desejada. Demonstramos na Figura 2A a inserção manual de sequências no BacPP. As sequências de teste correspondem a três promotores de *E. coli* (σ^{54}) obtidos do banco de dados RegulonDB (Santos-Zavaleta *et al.*, 2016). As sequências .FASTA empregadas no teste aqui demonstrado estão disponíveis no Quadro 1. Selecionamos a opção de deixar os resultados demonstrados na tela. Ainda há a opção de ter os resultados carregados num arquivo para *download*. Por fim, todos os fatores σ foram selecionados para que a probabilidade de a sequência pertencer a cada um deles seja avaliada.

Pelo botão "*Run*", a execução é feita e os resultados são mostrados na tela (Figura 2B). A janela de leitura da ferramenta compreende 80 nucleotídeos, logo, como demonstrado na Figura 2B, são feitas duas leituras (nucleotídeo 1 ao 80 e nucleotídeo 2 ao 81; já que cada uma das sequências de entrada contém 81 nucleotídeos). Por fim, destacado em vermelho, cada um dos σs sobre qual a sequência de entrada foi filtrada é apresentado. Isolamos apenas o σ^{54} como alvo (retângulo

verde) e analisamos a probabilidade retornada pelo BacPP de uma sequência pertencer ao fator σ alvo. Pode-se observar que a sequência prpBp foi predita com 100% de chance de pertencer ao σ70, sinalizando uma classificação errônea da ferramenta. As demais sequências (ffsp5 e glnAp2) foram preditas com 99% de probabilidade de pertencer ao fator σ que elas de fato pertencem.

Figura 2 – Execução do BacPP. (A) Processo de inserção manual de três sequências promotoras reguladas pelo σ^{54} na bactéria *E. coli*. (B) Resultado da execução do BacPP, em que a seta azul mostra o identificador extraído da entrada FASTA de cada sequência. Destacadas em retângulos azuis estão as sequências originais em janelas de leitura de 80 nucleotídeos. As linhas vermelhas correspondem aos fatores σ selecionados como alvo de busca. Os retângulos verdes correspondem à probabilidade de a sequência pertencer ao fator σ alvo.

Fonte: Elaboração do autor (2023).

Quadro 1 – Sequências .FASTA utilizadas no exemplo do presente capítulo.

```
> ffsp5
cgagtgaagtcgcattgcgcaagaaaccagcatctggcacgcgatgggttgcaattagccGgggcagcagtgataatgcgc
> glnAp2
gcatgataacgccttttaggggcaatttaaaagttggcacagatttcgctttatcttttTacggcgacacggccaaaata
> prpBp
tgaaattaaacatttaattttattaaggcaattgtggcacacccctttgctttgtctttatCaacgcaaataacaagttgat
```

Fonte: Elaboração do autor (2023).

4. BacPP e redes neurais artificiais

O método de predição do BacPP é baseado em conhecimento extraído do treinamento de redes neurais artificiais (RNs), um tipo de técnica que já foi aplicado com sucesso em outras aplicações de bioinformática (Rani; Bhavani; Bapi, 2007; Janga; Collado-Vides, 2007). O funcionamento de ferramentas dessa natureza é baseado no comportamento do cérebro humano. RNs possuem neurônios artificiais, sendo que estes possuem canais de entrada e saída. Os valores recebidos na entrada são ponderados pelos pesos e pelas funções matemáticas. O resultado é dado como valor de saída do neurônio. Dessa forma, uma RN pode aprender a separar classes de dados; no caso do BacPP, promotores e não promotores. A extração de regras de uma RN permite que padrões biológicos sejam derivados dos dados de entrada (Faceli *et al.*, 2011).

A extração de regras das RNs treinadas para a implementação do BacPP mostrou-se bastante efetiva ao classificar promotores. Mostramos na Tabela 1 os valores de acurácia, especificidade e sensibilidade, separado por cada fator σ.

Tabela 1 – Resultados de aprendizado do BacPP[17].

Fator Sigma	Acurácia (%)	Sensibilidade (%)	Especificidade (%)
σ^{24}	86,9	95,6	78,2
σ^{28}	92,8	90,4	95,2
σ^{32}	91,5	92,9	90,1
σ^{38}	89,3	83	93,9
σ^{54}	97	100	94,11
σ^{70}	83,6	85,4	81,8

Fonte: De Avila e Silva *et al.* (2011).

A capacidade preditiva do BacPP é equiparável a ferramentas similares. Klauck *et al.* (2020) realizaram um comparativo com a capacidade preditiva de diversas ferramentas utilizando como entrada promotores de *E. coli* do σ^{70}. Dentre seus resultados, os autores concluíram que o BacPP tem sua performance similar aos demais classificadores. Além disso, o fato de o BacPP empregar outros sigmas além do σ^{70} configura um diferencial à ferramenta.

5. Trabalhos futuros envolvendo o BacPP

O BacPP vem sendo incrementado constantemente. O núcleo de pesquisas do BacPP, o Laboratório de Bioinformática da Universidade de Caxias do Sul, conta com alunos de Iniciação Científica, Mestrado e Doutorado.

A grande frente de pesquisa relacionada ao BacPP diz respeito ao uso de características físicas da fita de DNA, as quais são dependentes da sequência de nucleotídeos e podem aumentar ainda mais a taxa de acerto na classificação da ferramenta (Ryasik *et al.*, 2018). Alguns dos parâmetros existentes utilizados para a codificação da informação genética são ental-

[17] Acurácia refere-se aos promotores corretamente identificados como promotores em todo o conjunto de dados (promotores e não promotores). Sensibilidade é a capacidade de identificar como promotor todos os promotores, excluindo os não promotores da equação. Especificidade mede a capacidade de classificar como não promotor os dados de controle da ferramenta.

pia, entropia, estabilidade, empilhamento e curvatura, sendo que as quatro primeiras já foram empregadas ao incrementar a capacidade de classificação da ferramenta (Martinez *et al.*, 2021).

Além de melhorar a classificação com bactérias Gram-negativas, os integrantes do projeto também trabalham para a inclusão de outras bactérias – incluindo as Gram-positivas e as arqueias.

Referências

BOYER, C. B.; MERZBACH, U. C. **História da matemática**. 3. ed. São Paulo: Editora Blucher, 2019.

DE AVILA E SILVA, S.; NOTARI, D.L.; NEIS, F.A.; RIBEIRO, H.G.; ECHEVERRIGARAY, S. BacPP: a web-based tool for Gram-negative bacterial promoter prediction. **Genetics and molecular research**, v. 15, n. 2, 2016.

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, p. 92-99, 2011.

ENGE, E. Mobile vs. Desktop Usage in 2020. **Perficient**, 2019. Disponível em: https://www.perficient.com/insights/research-hub/mobile-vs-desktop-usage. Acesso em: 28 jun. 2020.

FACELI, K.; LORENA, A. C.; GAMA, J.; DE CARVALHO, A. C. P. L. F. **Inteligência artificial:** uma abordagem de aprendizado de máquina. Rio de Janeiro: LTC, 2011.

JANGA, S. C.; COLLADO-VIDES, J. Structure and evolution of gene regulatory networks in microbial genomes. **Research in Microbiology**, v. 158, n. 10, p. 787-794, 2007.

KLAUCK, H. A.; DALL'ALBA, G.; DE AVILA E SILVA, S.; DELAMARE, A.P.L. Prediction and Recognition of Gram-Negative Bacterial Promoter Sequences: An Analysis of Available Web Tools. **Journal of Biotechnology Research**, v. 6, n. 7, p. 90-97, 2020.

KREBS, J. E.; GOLDSTEIN, E. S.; KILPATRICK, S. T. **Lewin's genes XI. Burlington:** Jones & Bartlett Learning, 2014.

MARTINEZ, G. S.; DE AVILA E SILVA, S.; KUMAR, A.; PÉREZ-RUEDA, E. DNA structural and physical properties reveal

peculiarities in promoter sequences of the bacterium Escherichia coli K-12. **SN Applied Sciences**, v. 3, n. 8, p. 740, 2021a.

MARTINEZ, G. S.; SARKAR, S.; KUMAR, A.; PÉREZ-RUEDA, E.; DE AVILA E SILVA, S. Characterization of promoters in archaeal genomes based on DNA structural parameters. **MicrobiologyOpen**, v. 10, n. 5, p. e1230, 2021b.

MCDERMOTT, J. R1: A rule-based configurer of computer systems. **Artificial Intelligence**, v. 19, n. 1, p. 39-88, 1982.

RANI, T. S.; BHAVANI, S. D.; BAPI, R. S. Analysis of E. coli promoter recognition problem in dinucleotide feature space. **Bioinformatics**, v. 23, n. 5, p. 582-588, 2007.

ROJAS, R. Konrad Zuse's legacy: the architecture of the Z1 and Z3. **IEEE Annals of the History of Computing**, v. 19, n. 2, p. 5-16, 1997.

RUSSELL, S. J.; NORVIG, P. **Artificial intelligence: a modern approach**. 3. ed. Upper Saddle River: Prentice Hall, 2010.

RYASIK, A.; ORLOV, M.; ZYKOVA, E.; ERMAK, T.; SOROKIN, A. Bacterial promoter prediction: Selection of dynamic and static physical properties of DNA for reliable sequence classification. **Journal of Bioinformatics and Computational Biology**, v. 16, n. 1, p. 1840003, 2018.

SANTOS-ZAVALETA, A.; SALGADO, H.; GAMA-CASTRO, S.; SÁNCHEZ-PÉREZ, M.; GÓMEZ-ROMERO, L.; LEDEZMA-TEJEIDA, D.; GARCÍA-SOTELO, J.S.; ALQUICIRA-HERNÁNDEZ, K.; MUÑIZ-RASCADO, L.J.; PEÑA-LOREDO, P.; ISHIDA-GUTIÉRREZ, C. RegulonDB v 10.5: tackling challenges to unify classic and high throughput knowledge of gene regulation in E. coli K-12. **Nucleic Acids Research**, v. 47, n. D1, p. D212-D220, 2019.

SMOLENSKY, P. On the Proper Treatment of Connectionism. *In*: COLE, D. J.; FETZER, J. H.; RANKIN, T. L. (Eds.). **Philosophy, Mind, and Cognitive Inquiry:** Resources for Understanding Mental Processes. Dordrecht: Springer, 1990. p. 145-206.

Capítulo 7

A curvatura intrínseca do código genético

Pedro Lenz Casa[18]
Fernanda Pessi de Abreu[19]
Nikael Souza de Oliveira[20]
Scheila de Avila e Silva[21]

1. Propriedades físicas e estruturais do DNA

A área da genômica procura explorar a estrutura, a função e a evolução do material genético dos seres vivos. Com o aprimoramento de técnicas de sequenciamento, a genômica produziu uma imensa quantidade de dados, sendo que o número de genomas completos publicados no banco de dados NCBI ultrapassa 264 mil considerando apenas organismos procariotos. A informação armazenada proveniente da genômica pode ser subdividida em duas classes: genômica funcional, que se preocupa com a função de determinadas regiões do DNA, e genômica estrutural, que mantém foco na caracterização da organização e da estrutura do DNA (De Carvalho *et al*., 2019; NCBI, 2014).

Esse segundo nível de informação é geralmente retratado em termos da própria sequência nucleotídica. No entanto, o material genético celular *in vivo* não demonstra um aspecto uniforme e linear. De fato, podem existir variações locais na sua estrutura, as quais podem ser dependentes da composição da sequência (intrínsecas) ou até decorrentes da interação com proteínas (extrínsecas). Essas variações compõem uma série de propriedades estruturais da molécula de DNA, para

[18] Laboratório de Bioinformática e Biologia Computacional, Instituto de Biotecnologia, Universidade de Caxias do Sul.
[19] Idem.
[20] Idem.
[21] Idem.

as quais podem ser atribuídos valores numéricos a partir de uma base teórica e/ou experimental (Florquin *et al*., 2005).

Ainda mais, Meysman, Marchal e Engelen (2012) propuseram uma classificação das propriedades estruturais, dividindo estas em conformacionais e físico-químicas (Quadro 1). O autor considera como características conformacionais aquelas que denotam detalhes da estrutura estática do DNA e a influência das interações entre os pares de bases (pb) sobre esta. De um nível organizacional mais restrito para um mais abrangente, são contemplados os desvios entre nucleotídeos, como *tilt*, *roll* e *twist*, e em seguida propriedades como a curvatura e *A/Z-philicity*. Em contraste, as características físico-químicas aludem ao potencial dinâmico da estrutura do DNA em termos de energia livre armazenada e geralmente retratam a extensão da resistência demonstrada pela molécula diante de diversas variáveis (Meysman; Marchal; Engelen, 2012). No decorrer deste capítulo será abordada a propriedade estrutural de curvatura do DNA. Nesse sentido, será composto um guia para a utilização da ferramenta DNA Curvature Analysis desenvolvida pelo pesquisador Christoph Gohlke, que se encontra disponível on-line pelos links: https://www.lfd.uci.edu/~gohlke/dnacurve/, interface *web*, e https://pypi.org/project/dnacurve/, *script* na linguagem Python (Gohlke, 2020).

Quadro 1 – Exemplos de propriedades estruturais do DNA. Foram mantidos os nomes originais na língua inglesa para evitar equívocos de tradução.

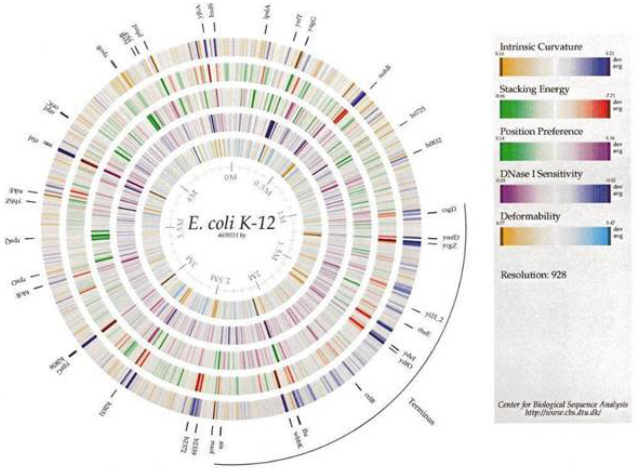
Propriedade estrutural	**Descrição**	**Categoria**
Slide-rise-tilt-roll-twist-shift	Desvios translacionais e rotacionais entre dinucleotídeos.	Conformacional
Curvature	Curvas feitas pela molécula de DNA que são geralmente derivadas dos desvios entre nucleotídeos adjacentes.	Conformacional

Minor/major groove depth and width	Profundidade e largura dos ângulos torcionais, sendo que espaços maiores podem permitir um acesso facilitado às bases na dupla hélice.	Conformacional
A/Z-philicity	Propensão de a molécula adotar conformações de A-DNA e Z-DNA.	Conformacional
Propeller twist	Rotação entre as bases de um nucleotídeo.	Conformacional
Persistence length	Comprimento no qual a molécula de DNA mantém sua direcionalidade. Em outras palavras, trata-se do tamanho esperado para um fragmento resistir a deformações.	Físico-Química
DNA stability	Enumeração da energia teoricamente necessária para "disromper" ou formar a dupla hélice de DNA. Existem diversas escalas para a medida de estabilidade, sendo algumas apresentadas abaixo.	Físico-Química
Stress-induced duplex stability	Estresse torcional resultante da compactação do DNA. Essa propriedade pode ser considerada uma medida de estabilidade da dupla hélice.	Físico-Química
Base stacking energy	Empilhamento da energia das bases em sequência, que contribui para a estabilidade geral da dupla hélice de DNA.	Físico-Química
Deformability	Desvios adotados pelo DNA em resposta à ligação com proteínas. O inverso das escalas de deformabilidade (*deformability*) são escalas de rigidez (*rigidity*), que mensuram a resistência a esses desvios.	Físico-Química
Bendability	Propensão de a molécula de DNA dobrar ou ser dobrada em uma direção específica. A escala de Brunker, por exemplo, enumera a dobrabilidade em direção ao *major groove*.	Físico-Química

Fonte: Adaptado de Meysman, Marchal e Engelen (2012).

Para fins deste capítulo, serão utilizadas sequências promotoras da bactéria Gram-negativa *Escherichia coli* como entrada para a ferramenta de análise de curvatura. Assim, é inconcebível deixar de fora da discussão um dos trabalhos mais abrangentes envolvendo a caracterização estrutural do genoma desse organismo. Na construção do denominado "Atlas Estrutural para *E. coli*", Pedersen *et al.* (2000) descreveram as propriedades de curvatura, flexibilidade e estabilidade do DNA da bactéria, sendo que no cálculo da curvatura intrínseca foi utilizado o modelo de *Bolshoi & Trifonov*, na avaliação da flexibilidade foram utilizadas as escalas de *DNaseI sensitivity*, *position preference* e *deformability* e para a descrição da estabilidade foi utilizado um modelo de *base stacking energy* (Figura 1). Por meio dessa metodologia, os autores identificaram um padrão para o DNA de sequências promotoras que aparenta possuir relevância biológica. De maneira geral, promotores apresentaram elevada curvatura, baixa flexibilidade e menor estabilidade quando comparados ao DNA codificante e regiões intergênicas sem promotores (Pedersen *et al*., 2000).

Figura 1 – Atlas estrutural do genoma de *E. coli*, sendo cada circunferência a representação do genoma completo em termos de propriedades estruturais do DNA. Cada escala representada foi organizada de forma que a média no genoma admite ausência de coloração (porção central das escalas na direita da imagem) e os desvios adotam presença de coloração (extrema direita e esquerda das escalas).



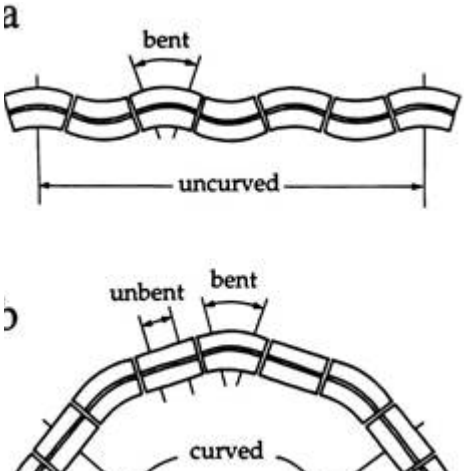
Fonte: Pedersen *et al.* (2000).

2. Curvatura do DNA

A característica estrutural de curvatura do DNA foi primeiramente observada em estudos do comportamento do material genético em gel eletroforético. Nesse sentido, fragmentos de comprimento idêntico, os quais se espera que migrem de maneira aproximadamente semelhante na matriz do gel, ocasionalmente demonstravam discrepâncias significativas no seu deslocamento. Investigando esse curioso fenômeno, foi determinado que curvas na molécula de DNA poderiam causar desvios nos resultados experimentais de eletroforese (Sinden, 2012). Analisando extensamente as moléculas com comportamento fora do esperado, verificou-se a emergência de curvas na dupla hélice diante da presença de séries repetitivas de adenina (*A-tracts*). Além disso, esse efeito

de curva em B-DNA necessita da periodicidade dos *A-tracts* a cada repetição helicoidal, ou seja, que essas séries de adenina estejam espaçadas em torno de 10-11 pb (Figura 2). Dessa forma, cada pequena contribuição para a curvatura é conferida com a mesma direcionalidade (Koo; Wu; Crothers, 1986; Haran; Mohanty, 2009). A partir dessa premissa, diversos modelos para predição da curvatura foram desenvolvidos levando em consideração os desvios de *tilt*, *roll* e *twist*. Esses três parâmetros denotam, respectivamente, rotações ao redor dos eixos *x*, *y* e *z* entre dinucleotídeos (Dickerson, 1989; Goodsell; Dickerson, 1994).

Figura 2. Representação da propriedade de curvatura do DNA. (A) Uma molécula com desvios locais, porém sem curvatura. (B) Uma molécula com desvios locais espaçados por regiões sem curvas locais que produzem um efeito de curvatura macroscópica.



Fonte: Goodsell e Dickerson (1994).

Os recursos disponíveis até então prediziam a conformação correta para a maioria dos fragmentos de DNA, contudo havia exceções. Considerando essa perspectiva, algumas sequências específicas contendo *A-tracts* não demonstravam aspecto curvado. Além disso, eventuais moléculas com conte-

údo exclusivo de G/C (guanina e citosina) podiam adotar uma leve curvatura. Em consequência, começou a ser levada em consideração a influência dos nucleotídeos que flanqueiam os *A-tracts*, não se limitando apenas a A/T (adenina e timina). Nesse sentido, pesquisas subsequentes marcaram mudanças significativas na base dos modelos de cálculo de curvatura do DNA (Kanhere; Bansal, 2004). Um dos trabalhos mais relevantes da área foi de Bolshoy *et al.* (1991), o qual avaliou a contribuição de 16 dinucleotídeos para a curvatura do DNA. Para isso, o autor comparou dados experimentais com predições computacionais de 54 fragmentos sintéticos de DNA, evidenciando a contribuição de dinucleotídeos diferentes de AA na deflexão do eixo da molécula (Bolshoy *et al.*, 1991).

3. Ferramenta DNA Curvature Analysis

A DNA Curvature Analysis é um recurso computacional desenvolvido na linguagem de programação Python pelo pesquisador Christoph Gohlke que permite ao usuário realizar diversas operações a partir de uma sequência de DNA, incluindo cálculos de curvatura. Para isso, é considerado o inverso do raio de um círculo passando pelas coordenadas da dupla hélice de DNA em uma janela de 10 nucleotídeos *upstream* e *downstream* do ponto analisado. Além disso, também são analisados o ângulo da curvatura entre os vetores normais dos nucleotídeos a 15 pb *upstream* e *downstream* do ponto analisado bem como o ângulo local da curva entre os vetores normais dos nucleotídeos a 2 pb *upstream* e *downstream* do ponto analisado. Todos os valores descritos são calculados a partir da premissa do modelo teórico *dinucleotide wedge* e normalizados relativo à curvatura de um nucleossomo (curvatura de 0,0234 e ângulo de 2,3728), a qual compõe 1 unidade de curvatura.

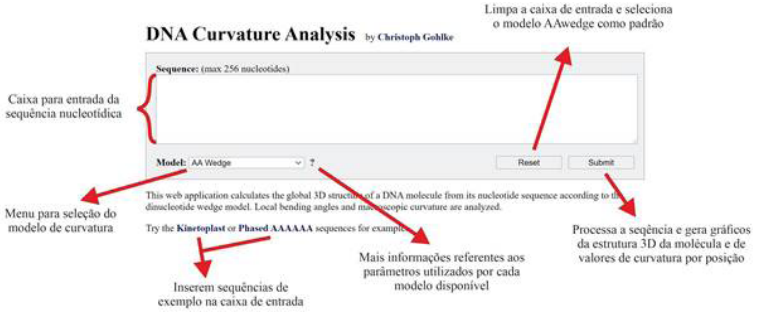
Além disso, a ferramenta dispõe de sete modelos preditivos, sendo eles: *Nucleosome Positioning* (Satchwell; Drew; Travers, 1986), *AA Wedge* (Jernigan *et al.*, 1986), *Bolshoi & Trifonov* (Bolshoy *et al.*, 1991), *Calladine & Drew* (Calladine;

Drew; Mccall, 1988), *Reversed Calladine & Drew* (Goodsell; Dickerson, 1994), *Cacchione & De Santis* (Cacchione *et al.*, 1989; De Santis *et al.*, 1990) e o *DNAse I Consensus* (Gabrielian; Pongor, 1996). Cada um desses apresenta uma tabela de valores obtidos em análises experimentais para as variáveis *tilt*, *roll* e *twist* de cada dinucleotídeo (ou trinucleotídeo). A partir desses valores, o algoritmo reproduz a trajetória da molécula em um espaço tridimensional, com a qual os valores de curvatura são obtidos em sequência.

3.1 Uso da interface on-line

A ferramenta dispõe de uma interface on-line disponível no endereço https://www.lfd.uci.edu/~gohlke/dnacurve/. Embora a usabilidade seja simplificada e convidativa, essa versão do programa contém algumas limitações. Nesse sentido, o tamanho máximo da sequência a ser analisada é restringido a 256 nucleotídeos. Além disso, não é incluída a opção de *upload* de arquivo para análise e a inserção de múltiplas sequências não é suportada. Assim, essa versão tem um papel demonstrativo, sendo que ilustra a funcionalidade e os resultados da ferramenta. No emprego desse instrumento computacional, o pesquisador conta com uma caixa para a inserção da sua sequência de DNA e um menu para a seleção do modelo de curvatura (Figura 3).

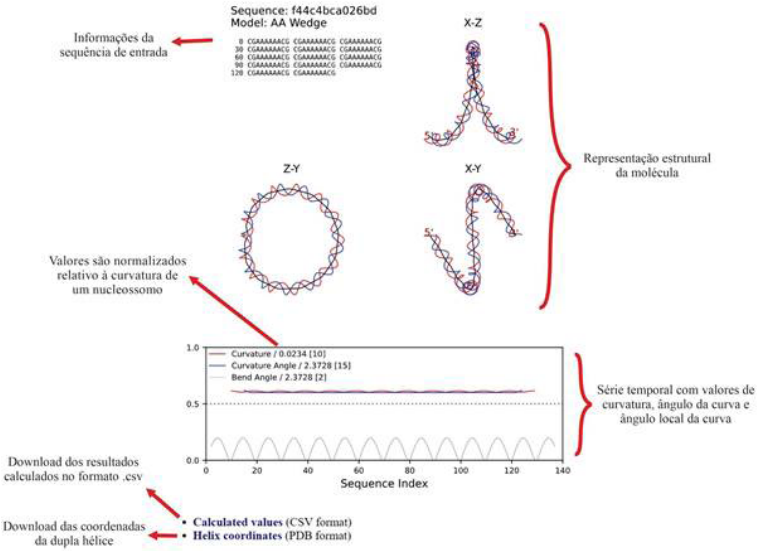
Figura 3 – Tela inicial da interface *web*.



Fonte: Elaboração dos autores (2023).

Após o processamento de uma sequência de DNA, o pesquisador é direcionado para uma tela de resultados que contém a representação da trajetória da molécula e um gráfico com os valores de curvatura, ângulo de curvatura e ângulo local da curva para cada nucleotídeo. Os valores estão em unidades de curvatura. Além disso, existe a opção de *download* dos resultados em formato .csv e das coordenadas da dupla hélice no formato .pdb (Figura 4).

Figura 4 – Tela de resultados da interface *web*.



Fonte: Elaboração dos autores (2023).

3.2 Uso como script

A ferramenta também possui uma versão no formato de biblioteca em Python, disponível em https://pypi.org/project/dnacurve/. Em contraste com a interface on-line, a biblioteca "dnacurve" permite uma customização rebuscada da operação. Para o *download* e o uso desta, é necessário, como pré-requisito, efetuar a instalação da linguagem de programação Python (https://www.python.org/downloads/) e os

pacotes NumPy (https://numpy.org/) e Matplotlib (https://matplotlib.org/). Após a instalação da Python, a obtenção dos módulos adicionais pode ser facilitada utilizando o recurso "pip" no *prompt* de comando, de acordo com os passos apresentados no Quadro 2.

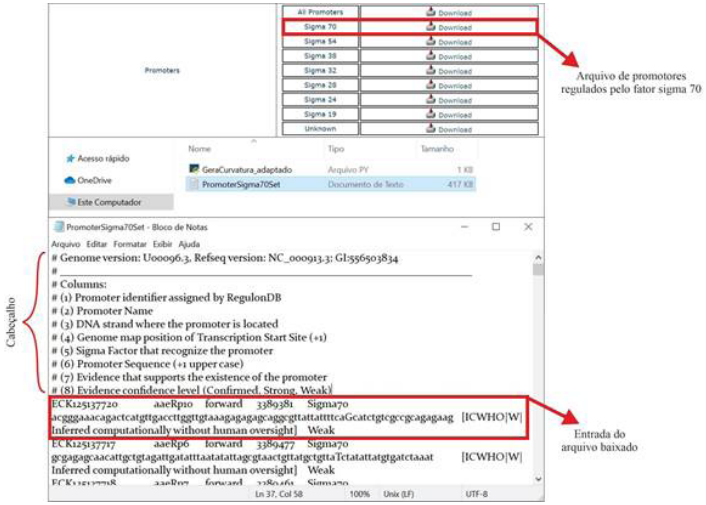
Quadro 2 – Passos para a obtenção das bibliotecas NumPy e Matplotlib utilizando o recurso pip.

1. Abrir um terminal, o qual pode ser encontrado digitando "*prompt* de comando" na barra de pesquisa do sistema operacional.
2. Executar a linha "python -m ensurepip --default-pip", a fim de verificar se o módulo pip está funcionando adequadamente.
3. Caso o pip não tenha sido encontrado, executar "python get-pip.py" para a sua obtenção e retornar ao passo de verificação.
4. Executar em sequência os comandos "pip install NumPy", "pip install Matplotlib" e "pip install dnacurve".

Fonte: Elaboração dos autores (2023).

Para o nosso exemplo, vamos utilizar sequências promotoras de *E. coli K-12* reguladas pelo fator sigma 70, disponíveis no banco de dados RegulonDB pelo endereço http://regulondb.ccg.unam.mx/menu/download/datasets/ (Santos-Zavaleta *et al*., 2019). Ao realizar o *download* do arquivo, este deve ser aberto para a remoção manual do cabeçalho (Figura 5).

Figura 5 – Arquivo para *download* no RegulonDB (parte superior) e arquivo .csv aberto com o bloco de notas (parte inferior).



Fonte: Elaboração dos autores (2023).

Para o processamento de um arquivo com múltiplas sequências utilizando o módulo de análise de curvatura, foi elaborado um algoritmo na linguagem de programação Python (Quadro 3). A continuidade do algoritmo foi organizada com comentários (#) a cada bloco de comandos, que delineiam a funcionalidade do conjunto de linhas abaixo dele. Além disso, cada comentário foi numerado para facilitar seu referenciamento no capítulo. O uso efetivo do *script* (extensão .py) requer que o arquivo com sequências (extensão .txt) esteja incluído na mesma pasta de trabalho. Embora tenha sido desenvolvida para o exemplo do capítulo, a sequência do algoritmo pode ser facilmente adaptada de acordo com o objetivo da pesquisa.

Quadro 3 – *Script* para cálculo da curvatura.

```
import os
from dnacurve import CurvedDNA
#-------------------------------------------------------------------------
#1 Nome do arquivo com promotores
arq_promotores = "PromoterSigma70Set.txt"
#2 recupera o local do script
raiz = os.getcwd()
#3 abre arquivo para guardar apenas a curvatura de todas sequencias
final = open(raiz + "/arq_curvatura.csv",'w')
#4 cria pasta de saída e salva o caminho
saida = os.mkdir(raiz + "/" + "dados_gerados")
caminho_saida = raiz + "/" + "dados_gerados"
#5 abre arquivo com promotores e faz a leitura de cada linha
ent = open(raiz + "/" + arq_promotores,'r')
linhas = ent.readlines()
#6 Laço de repetição para cada promotor no arquivo, separado em colunas
for linha in linhas:
colunas = linha.split()
#7 condição para filtrar linhas sem sequência de DNA (considerando tamanho de 81 pb do
RegulonDB)
if len(colunas[5]) > 80:
#8 Processa a sequência nucleotídica de cada linha do arquivo e salva valores de curvatura
como .csv
resultado = CurvedDNA(colunas[5], 'TRIFONOV', name = colunas[1])
resultado.save_csv(caminho_saida + "/" + colunas[1] + ".csv")
#9 Transpõe e concatena os resultados apenas da curvatura em um mesmo arquivo
arq_salvo = open(caminho_saida + "/" + colunas[1] + ".csv", 'r')
posicao = arq_salvo.readlines()[1:]
for l in posicao:
col_curv = l.split(",")
final.write(col_curv[2]+";")
final.write("\n")
```

Fonte: Elaboração dos autores (2023).

As linhas abaixo de #8 demonstram a usabilidade do módulo de análise de curvatura, sendo o comando para cálculo expresso pelo seguinte argumento: CurvedDNA ("sequência de DNA", "modelo de curvatura", *name* = "nome da sequência"). O parâmetro "sequência de DNA" deve ser uma *string* contendo a sequência a ser analisada, por exemplo: "AATAGCTGAGATC". O campo de "modelo de curvatura" deve ser preenchido com um dos modelos disponíveis na ferramenta, referidos pelas seguintes *strings*: "AAWEDGE", "CALLADINE", "TRIFONOV", "DESANTIS", "REVERSED" (*Reversed Calladine & Drew*), "NUCLEOSOME" (*Nucleosome Positioning*) e "TRINUCLEOTIDE" (*DNAse I Consensus*). Por fim, deve ser informado um nome para a sequência. De forma complementar, existem parâmetros numéricos opcionais que

podem ser definidos após o "nome da sequência", sendo eles: "maxlen", que determina o comprimento máximo da sequência de DNA aceita pela ferramenta (o valor padrão caso não alterado pelo usuário é igual a 510); e "curvature_window", "curve_window" e "bend_window", que determinam a janela de pb em que são medidos a curvatura, o ângulo da curvatura e o ângulo local da curva, respectivamente (valores padrão são iguais a 10, 15 e 2 pb, respectivamente).

Uma variável contendo a curvatura de uma molécula seguida pela comando .save_pdb resulta nas coordenadas da dupla hélice sendo salvas em um arquivo .pdb, o qual pode ser visualizado utilizando *softwares* de análise de estruturas moleculares, como, por exemplo, o Chimera (Pettersen *et al*., 2004). Como parâmetro, é necessária a especificação do local completo em que o arquivo deve ser salvo, incluindo o nome do arquivo e a extensão .pdb. Já o comando .plot possibilita ao pesquisador salvar gráficos com valores de curvatura juntamente com uma representação estrutural da molécula analisada; assim como a especificação do local completo do arquivo, incluindo o nome do arquivo e a extensão, a qual pode ser .png, .pdf ou .ps. Além disso, deve-se informar a resolução da imagem em pontos por polegada (dpi), sendo recomendado um mínimo de 300 dpi. Os comandos, tanto para salvar a trajetória da dupla hélice quanto para plotar os resultados, estão no Quadro 4.

Quadro 4 – Comandos adicionais de customização.

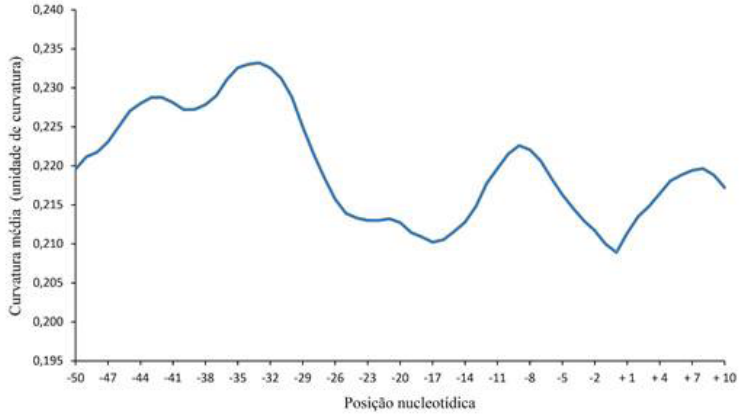
```
#Comando para salvar a trajetória.
variável.save_pdb('caminho do arquivo de saída')
#Comando para plotar a imagem
variável.plot('caminho do arquivo de saída', dpi= resolução da imagem)
```

Fonte: Elaboração dos autores (2023).

Ao executar o algoritmo desenvolvido, os resultados são salvos separadamente em arquivos .csv para cada sequência de DNA (comando ".save_csv('caminho completo')" no bloco

#8). Na eventualidade de todas as sequências de entrada possuírem o mesmo tamanho, que é o caso dos promotores do RegulonDB, as planilhas geradas podem ser transpostas e concatenadas (linhas abaixo de #3 e #9) com o valor de apenas uma das variáveis (no nosso caso, estamos salvando apenas os valores de curvatura). Após esse procedimento, obtemos uma planilha final com os valores de curvatura organizados por posição nucleotídica (arquivo "arq_curvatura.csv" gerado na pasta de trabalho). Abrindo o arquivo com Excel ou LibreOfficeCalc, a média de curvatura por posição pode ser calculada. Por fim, é possível construir uma série temporal com essa informação (Figura 6).

Figura 6 – Exemplificação gráfica dos resultados obtidos por meio de valores de curvatura.



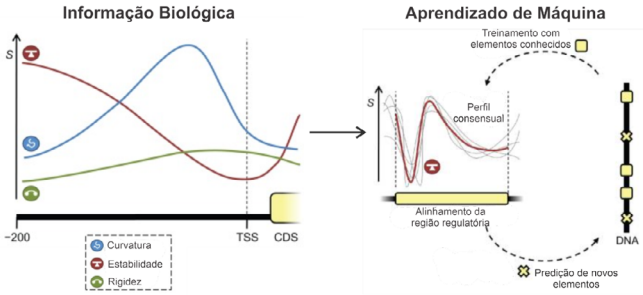
Fonte: Elaboração dos autores (2023).

4. O uso da curvatura do DNA como informação biológica

As propriedades estruturais do DNA contêm informações importantes sobre os processos biomoleculares. Nesse sentido, abordagens múltiplas utilizando tanto os perfis estruturais quanto as próprias sequências nucleotídicas podem fornecer subsídios para o aperfeiçoamento de metodologias.

A predição dos sítios de ligação de proteínas nas regiões regulatórias dos genes é um desafio para a era genômica. Visto isso, as informações biológicas fornecidas a partir da análise de características como a curvatura podem ser empregadas como atributos adicionais em técnicas de aprendizado de máquina. Desse modo, é possível, por meio do treinamento com elementos conhecidos, a reconstrução de perfis consensuais e a predição de novos elementos regulatórios (Figura 7) (Meysman; Marchal; Engelen, 2012).

Figura 7 – Predição de sítios regulatórios a partir de propriedades estruturais.



Fonte: Adaptado de Meysman, Marchal e Engelen (2012).

A disposição espacial da dupla hélice de DNA também pode fornecer informações sobre a interação com outros elementos contidos no meio, além das próprias proteínas. Duan *et al.* (2018) demonstraram que a curvatura intrínseca do DNA é o principal determinante *cis* para a taxa de mutação local em leveduras e humanos. O estudo apontou que menores valores de curvatura estão correlacionados negativamente com eventos de mutação, uma vez que esse desvio pode promover a ligação com proteínas tornando a sequência nucleotídica menos acessível a elementos mutagênicos (Duan *et al.*, 2018). Conforme demonstrado ao longo deste capítulo, a interação física entre pares de bases vizinhos pode prover informações valiosas sobre o DNA, permitindo que pesquisadores obtenham novos *insights* na área genômica.

Referências

BOLSHOY, A.; MCNAMARA, P.; HARRINGTON, R.E.; TRIFONOV, E.N. Curved DNA without A-A: experimental estimation of all 16 DNA wedge angles. **Proceedings of the National Academy of Sciences**, v. 88, n. 6, p. 2.312-2.316, 1991.

CACCHIONE, S.; DE SANTIS, P.; FOTI, D.; PALLESCHI, A.; SAVINO, M. Periodical polydeoxynucleotides and DNA curvature. **Biochemistry**, v. 28, n. 22, p. 8706-8713, 1989.

CALLADINE, C. R.; DREW, H. R.; MCCALL, M. J. The intrinsic curvature of DNA in solution. **Journal of molecular biology**, v. 201, n. 1, p. 127-137, 1988.

DE CARVALHO, L. M.; BORELLI, G.; CAMARGO, A.P.; DE ASSIS, M.A.; DE FERRAZ, S.M.F.; FIAMENGHI, M.B.; JOSÉ, J.; MOFATTO, L.S.; NAGAMATSU, S.T.; PERSINOTI, G.F.; SILVA, N.V. Bioinformatics applied to biotechnology: A review towards bioenergy research. **Biomass and Bioenergy**, v. 123, p. 195-224, 2019.

DE SANTIS, P.; PALLESCHI, A.; SAVINO, M.; SCIPIONI, A. Validity of the nearest-neighbor approximation in the evaluation of the electrophoretic manifestations of DNA curvature. **Biochemistry**, v. 29, n. 39, p. 9269-9273, 1990.

DICKERSON, R. E. Definitions and nomenclature of nucleic acid structure components. **Nucleic Acids Research**, v. 17, n. 5, p. 1.797-1.803, 1989.

DUAN, C.; HUAN, Q.; CHEN, X.; WU, S.; CAREY, L.B.; HE, X.; QIAN, W. Reduced intrinsic DNA curvature leads to increased mutation rate. **Genome Biology**, v. 19, n. 1, p. 132, 14 set. 2018.

FLORQUIN, K.; SAEYS, Y.; DEGROEVE, S.; ROUZE, P.; VAN DE PEER, Y. Large-scale structural analysis of the core promoter in mammalian and plant genomes. **Nucleic Acids Research**, v. 33, n. 13, p. 4.255-4.264, 2005.

GABRIELIAN, A.; PONGOR, S. Correlation of intrinsic DNA curvature with DNA property periodicity. **FEBS letters**, v. 393, n. 1, p. 65-68, 1996.

GOHLKE, C. **DNACurve:** DNA Curvature Analysis. v. 2020.1.1. [s. l.], 2020. Disponível em: https://pypi.org/project/dnacurve/. Acesso em: 25 maio 2020.

GOODSELL, D. S.; DICKERSON, R. E. Bending and curvature calculations in B-DNA. **Nucleic Acids Research**, v. 22, n. 24, p. 5.497-5.503, 1994.

HARAN, T. E.; MOHANTY, U. The unique structure of A-tracts and intrinsic DNA bending. **Quarterly Reviews of Biophysics**, v. 42, n. 1, p. 41-81, 2009.

JERNIGAN, R. L.; SARAI, A.; TING, K.-L.; NUSSINOV, R. Hydrophobic interactions in the major groove can influence DNA local structure. **Journal of Biomolecular Structure and Dynamics**, v. 4, n. 1, p. 41-48, 1986.

KANHERE, A.; BANSAL, M. DNA bending and curvature: a "turning point" in DNA function?. **Proceedings of the Indian National Science Academy**, v. 70, n. 2, p. 239-254, 2004.

KOO, H.-S.; WU, H.-M.; CROTHERS, D. M. DNA bending at adenine thymine tracts. **Nature**, v. 320, n. 6062, p. 501-506, 1986.

MEYSMAN, P.; MARCHAL, K.; ENGELEN, K. DNA structural properties in the classification of genomic transcription regulation elements. **Bioinformatics and Biology Insights**, v. 6, p. 155-168, 2012.

NATIONAL CENTER FOR BIOTECHNOLOGY INFORMATION (NCBI). **Genome**. Disponível em: https://www.ncbi.nlm.nih.gov/genome/. Acesso em: 25 ago. 2020.

PEDERSEN, A. G.; JENSEN, L.J.; BRUNAK, S.; STÆRFELDT, H.H.; USSERY, D.W. A DNA structural atlas for Escherichia coli. **Journal of Molecular Biology**, v. 299, n. 4, p. 907-930, 2000.

PETTERSEN, E. F.; GODDARD, T.D.; HUANG, C.C.; COUCH, G.S.; GREENBLATT, D.M.; MENG, E.C.; FERRIN, T.E. UCSF Chimera – A visualization system for exploratory research and analysis. **Journal of Computational Chemistry**, v. 25, n. 13, p. 1.605-1.612, 2004.

SANTOS-ZAVALETA, A.; SALGADO, H.; GAMA-CASTRO, S.; SÁNCHEZ-PÉREZ, M.; GÓMEZ-ROMERO, L.; LEDEZMA-TEJEIDA, D.; GARCÍA-SOTELO, J. S.; ALQUICIRA-HERNÁNDEZ, K.; MUÑIZ-RASCADO, L. J.; PEÑA-LOREDO, P.; ISHIDA-GUTIÉRREZ, C. RegulonDB v 10.5: tackling challenges to unify classic and high throughput knowledge of gene regulation in E. coli K-12. **Nucleic Acids Research**, v. 47, n. D1, p. D212-D220, 2019.

SATCHWELL, S. C.; DREW, H. R.; TRAVERS, A. A. Sequence periodicities in chicken nucleosome core DNA. **Journal of molecular biology**, v. 191, n. 4, p. 659-675, 1986.

SINDEN, R. R. **DNA Structure and Function**. San Diego: Academic Press, 2012.

Biografia dos revisores

André Luis Martinotto (almartin@ucs.br)

Graduado em Ciência da Computação pela Universidade de Caxias do Sul, mestre em Computação pela Universidade Federal do Rio Grande do Sul e doutor em Ciências dos Materiais pela Universidade Federal do Rio Grande do Sul. Tem experiência na área de Ciência da Computação, atuando principalmente nas áreas de programação paralela e distribuída, matemática computacional e simulação computacional.

Johnatan Vilasboa (johnatan.vilasboa@gmail.com)

Licenciado em Ciências Biológicas pela Universidade de Caxias do Sul e doutor em Ciências: Biologia Celular e Molecular pela Universidade Federal do Rio Grande do Sul. Atualmente é Research Fellow na Universidade de Nottingham, Reino Unido. Possui experiência em Biotecnologia de bioprocessos e Bioquímica e Fisiologia de plantas, com foco na propagação clonal de espécies arbóreas por estaquia.

Marcos Vinicius Rossetto (mvrossetto@ucs.br)

Graduado em sistemas de informação pela Universidade de Caxias do Sul, é doutorando no Programa de Pós-Graduação em Biotecnologia da Universidade de Caxias do Sul e mestre em Biotecnologia pelo mesmo programa. A experiência acadêmica está relacionada a pesquisas de desenvolvimento de *softwares* aplicados à análise de dados biológicos.

Nicole Anne Modena (modena.nicole@gmail.com)

Bacharela e Licenciada em Ciências Biológicas pela Universidade de Caxias do Sul e mestra em Biotecnologia

pela mesma instituição. Tem experiência em Biologia molecular, Bioinformática, Informática Educativa e Docência das disciplinas de Ciências da Natureza, Matemática e Ensino Religioso nos Anos Finais do Ensino Fundamental. Atualmente, atua como servidora pública da Rede Municipal de Ensino de Caxias do Sul.

Ricardo Vargas Dorneles
(rvdornel@ucs.br)

Engenheiro eletricista e tecnólogo em processamento de dados, mestre e doutor em Ciência da Computação pela Universidade Federal do Rio Grande do Sul, na área de Arquiteturas Paralelas e Processamento de Alto Desempenho. Experiência em modelagem e simulação computacional e análise de desempenho.

Biografia dos autores

Carine Pedrotti
(carine_pedrotti@yahoo.com.br)

Graduada em Ciências Biológicas pela Universidade de Caxias do Sul. Possui mestrado e doutorado em Biotecnologia pela Universidade de Caxias do Sul. Possui experiência em controle alternativo de doenças em plantas e pós-colheita, microbiologia agrícola e enológica, vitivinicultura e nanotecnologia.

Clarissa Franzoi
(cfranzoi@ucs.br)

Graduada em Ciências Biológicas pela Universidade de Caxias do Sul, mestranda no Programa de Pós-Graduação em Botânica da Universidade Federal do Rio Grande do Sul. Possui Experiência em controle alternativo de fungos e ervas daninhas utilizando óleos essenciais.

Fernanda Pessi de Abreu
(fpabreu1@ucs.br)

Graduada em Ciências Biológicas pela Universidade de Caxias do Sul. Mestranda no Programa de Pós-Graduação em Genética e Biologia Molecular da Universidade Federal do Rio Grande do Sul. Tem experiência em mineração e aplicação de técnicas de inteligência artificial em dados biológicos, genética de fungos e expressão gênica diferencial em neoplasias. Atualmente, trabalha com citotaxonomia de Hymenophyllaceae.

Gabriel Dall'Alba
(gdalba@phas.ubc.ca)

Graduado em Ciências Biológicas, mestrando no programa Genome Science and Technology pela University of British Columbia. Possui experiência em Bioinformática e Biologia

Evolutiva. Atualmente, investiga as origens da multicelularidade através do genoma do ctenóforo *Mnemiopsis leidyi*.

Gustavo Machado das Neves (gustavo.neves@ufrgs.br)

Graduado em Farmácia pela Universidade Federal do Rio Grande do Sul e em Biomedicina pela UFCSPA. Possui mestrado e doutorado em Ciências Farmacêuticas pelo Programa de Pós-Graduação Ciências Farmacêuticas da Universidade Federal do Rio Grande do Sul. Possui experiência na área de Química Farmacêutica/Medicinal com ênfase em planejamento de fármacos assistido por computador e reposicionamento de fármacos com o uso de técnicas de modelagem molecular e triagem virtual, tais como: ancoramento molecular, derivação de padrões farmacofóricos e QSAR.

Gustavo Sganzerla Martinez (gsmartinez@ucs.br)

Cientista de dados cujo foco é a resolução de problemas biológicos. O trabalho de Gustavo envolve a aplicação de inteligência artificial para extrair padrões de conjuntos de dados provenientes das áreas das ciências da vida. Atualmente, Gustavo realiza pesquisa de nível pós-doutorado na Dalhousie University (Canadá) com foco em doenças infecciosas.

Henrique Vieira Figueiró (henriquevf@gmail.com)

Bacharel e licenciado em Ciências Biológicas pela Pontifícia Universidade Católica do Rio Grande do Sul. Possui mestrado e doutorado em Zoologia pela mesma instituição. Possui experiência na aplicação de ferramentas de bioinformática na análise de dados genômicos de mamíferos. Suas linhas de pesquisa envolvem genômica evolutiva, com especial enfoque em estudos de seleção natural bem como genômica da conservação, com enfoque em espécies Neotropicais.

Joséli Schwambach
(jschwambach@ucs.br)

Graduada em Ciências Biológicas pela Universidade Federal do Rio Grande do Sul. Possui mestrado e doutorado em Biologia Celular e Molecular pela Universidade Federal do Rio Grande do Sul. Possui experiência em propagação vegetal, no controle biológico de doenças de plantas através do uso de *Bacillus* sp. e *Trichoderma* sp. e no controle de doenças de plantas utilizando óleos essenciais de plantas.

Luciano Porto Kagami
(lucianopkagami@hotmail.com)

Graduado em Química Industrial pela Pontifícia Universidade Católica do Rio Grande do Sul, é mestre e doutor no Programa de Pós-Graduação em Ciências Farmacêuticas pela Universidade Federal do Rio Grande do Sul. Atualmente é servidor do Hospital de Clínicas de Porto Alegre, colaborador do Laboratório de Síntese Orgânica Medicinal – LaSOM, da Universidade Livre de Bruxelas – "*Structural Biology Brussels*". Possui experiência em quimioinformática, bioinformática e linguagem de programação.

Luis Fernando Saraiva Macedo Timmers
(luis.timmers@univates.br)

Graduado em Ciências Biológicas pela Pontifícia Universidade Católica do Rio Grande do Sul. Possui mestrado e doutorado em Biologia Celular e Molecular pela Pontifícia Universidade Católica do Rio Grande do Sul. Possui experiência em biologia molecular, biofísica molecular computacional, biologia estrutural, química medicinal e bioinformática, atuando principalmente em temas como: flexibilidade proteica, modelagem molecular, dinâmica molecular, docking e processo de interação proteína-ligante. Atualmente, atua como docente na Universidade do Vale do Taquari e é membro do corpo permanente do Programa de Pós-Graduação em

Biotecnologia e coordenador do Programa de Pós-Graduação em Ciências Médicas.

Nikael Souza de Oliveira
(nsoliveira4@ucs.br)

Graduado em Ciências Biológicas pela Universidade de Caxias do Sul. Técnico em enfermagem pela faculdade Fátima. Mestrando em Biotecnologia pela Universidade de Caxias do Sul. Possui experiência na área clínica de atendimento ao paciente como técnico em enfermagem. Além disso, possui experiência na área de pesquisa com triagem de enzimas e análises de Bioinformática com genômica e transcriptômica.

Pedro Lenz Casa
(plcasa@ucs.br)

Graduado em Ciências Biológicas (Bacharelado e Licenciatura) pela Universidade de Caxias do Sul. Tem experiência na análise de características estruturais do DNA utilizando abordagens computacionais, bem como na predição de sequências regulatórias em bactérias. Além disso, tem conhecimento de técnicas de inteligência artificial aplicadas para a mineração de dados de origem biológica.

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

Graduado em Química pela Universidade Luterana do Brasil. Possui mestrado em Biologia Celular e Molecular e doutorado em Medicina e Ciências da Saúde com ênfase em Farmacologia Molecular Bioquímica, ambos pela Pontifícia Universidade Católica do Rio Grande do Sul. Atua principalmente na área de biofísica molecular computacional (bioinformática estrutural, modelagem, docagem, dinâmica molecular e QSAR). Possui experiência em cristalização de proteínas e difração de raios X. Atualmente é Professor Adjunto da Universidade Federal de Ciências da Saúde de Porto Alegre, membro do corpo permanente do Programa de

Pós-Graduação em Biociências da Universidade Federal de Ciências da Saúde de Porto Alegre.

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

Graduada em Gestão da Tecnologia da Informação pela Universidade do Vale do Rio dos Sinos e em Ciências Biológicas pela Universidade de Caxias do Sul. Possui mestrado em Computação Aplicada pela Universidade do Vale do Rio dos Sinos e doutorado em Biotecnologia pela Universidade de Caxias do Sul. Possui experiência em análise de dados, integração de bases de dados biológicas e aplicação de técnicas de inteligência artificial em dados genômicos.

FUNDAÇÃO UNIVERSIDADE DE CAXIAS DO SUL

UCS UNIVERSIDADE DE CAXIAS DO SUL

EDUCS
PESQUISA
COMUNIDADE
ENSINO
INFANTOJUVENIL
LITERÁRIO
POCKETS
TRADUÇÕES
INTERNACIONAL
TESES & DISSERTAÇÕES

A Universidade de Caxias do Sul é uma Instituição Comunitária de Educação Superior (ICES), com atuação direta na região nordeste do estado do Rio Grande do Sul. Tem como mantenedora a Fundação Universidade de Caxias do Sul, entidade jurídica de Direito Privado. É afiliada ao Consórcio das Universidades Comunitárias Gaúchas - COMUNG; à Associação Brasileira das Universidades Comunitárias - ABRUC; ao Conselho de Reitores das Universidades Brasileiras - CRUB; e ao Fórum das Instituições de Ensino Superior Gaúchas.

Criada em 1967, a UCS é a mais antiga Instituição de Ensino Superior da região e foi construída pelo esforço coletivo da comunidade.

Uma história de tradição

Em meio século de atividades, a UCS marcou a vida de mais de 120 mil pessoas, que contribuem com o seu conhecimento para o progresso da região e do país.

A universidade de hoje

A atuação da Universidade na atualidade também pode ser traduzida em números que ratificam uma trajetória comprometida com o desenvolvimento social.

Localizada na região nordeste do Rio Grande do Sul, a Universidade de Caxias do Sul faz parte da vida de uma região com mais de 1,2 milhão de pessoas.

Com ênfase no ensino de graduação e pós-graduação, a UCS responde pela formação de milhares de profissionais, que têm a possibilidade de aperfeiçoar sua formação nos programas de Pós-Graduação, Especializações, MBAs, Mestrados e Doutorados. Comprometida com excelência acadêmica, a UCS é uma instituição sintonizada com o seu tempo e projetada para além dele.

Como agente de promoção do desenvolvimento a UCS procura fomentar a cultura da inovação científica e tecnológica e do empreendedorismo, articulando as ações entre a academia e a sociedade.

A Editora da Universidade de Caxias do Sul

O papel da EDUCS, por tratar-se de uma editora acadêmica, é o compromisso com a produção e a difusão do conhecimento oriundo da pesquisa, do ensino e da extensão. Nos mais de 1.500 títulos publicados é possível verificar a qualidade do conhecimento produzido e sua relevância para o desenvolvimento regional.

Conheça as possibilidades de formação e aperfeiçoamento vinculadas às áreas de conhecimento desta publicação acessando o QR Code:


EDUCS