Universidade de São Paulo
Instituto de Matemática e Estatística
Bacharelado em Ciência da Computação

Um método baseado em multirresolução para análise filoproteômica de venenos de serpentes

Gustavo Mendes Maciel

Monografia Final

mac 0499 — Trabalho de Formatura Supervisionado

Supervisor: Dr. Marcelo da Silva Reis

São Paulo
Dezembro de 2019

Agradecimentos

Faço um agradecimento à pesquisadora Débora Andrade-Silva, do Instituto Butantan, que nos ajudou com a interpretação e a associação entre os parâmetros de configuração de *software* e equipamentos envolvidos neste trabalho.

Agradeço também ao administrador do Centro de Bioinformática do Instituto Butantan, David da Silva Pires, que me proporcionou o acesso à servidora Vital do Laboratório de Ciclo Celular, e configurou o ambiente de trabalho necessário para as últimas etapas deste projeto.

Resumo

Gustavo Mendes Maciel. **Um método baseado em multirresolução para análise filoproteômica de venenos de serpentes**. Monografia (Bacharelado). Instituto de Matemática e Estatística, Universidade de São Paulo, São Paulo, 2019.

Venenos de serpente são misturas complexas de proteínas e peptídeos. Uma estratégia analítica muito utilizada para estudá-los é a proteômica baseada em espectrometria de massas (MS). Análises recentes utilizando dados proteômicos produzidos por MS mostraram que há uma correlação entre a aglomeração hierárquica dos proteomas dos venenos de diferentes espécies do gênero *Bothrops* e a classificação filogenética de tais serpentes. Todavia, a superrepresentação de algumas espécies (e.g., *B. jararaca*) nos bancos de dados comumente utilizados para identificação computacional de peptídeos potencialmente acarreta em vieses nessas análises. Para mitigar este problema, recentemente foram geradas árvores filoproteômicas a partir de sequenciamento *de novo*, isto é, sem o uso de banco de dados; porém, essa abordagem gera muitos peptídeos falsos positivos, o que também introduz ruído nas análises. Nesse trabalho foi desenvolvida uma metodologia com o objetivo de contornar esses problemas, utilizando diretamente os dados brutos provenientes de experimentos de MS para estimar as árvores filoproteômicas. Para isso, foi utilizada uma estratégia de particionamento dos dados proteômicos, que foram representados por matrizes. Tais matrizes foram utilizadas para gerar árvores filoproteômicas por meio de uma abordagem de inferência Bayesiana, empregando métodos de Monte Carlo com cadeias de Markov. Por fim, o teste estatístico CADM foi aplicado para comparar as árvores evolutivas obtidas pelo método desenvolvido nesse trabalho e as obtidas pelo uso de dados de DNA mitocondrial. Os resultados mostraram que esse método conseguiu reafirmar a relação entre o perfil proteômico dos venenos e a filogenia de diferentes espécies de serpentes do gênero *Bothrops*.

Palavras-chave: Espectrometria de massas, Venenos de serpentes, Filoproteômica, Filogenia, Inferência Bayesiana, Multirresolução.

Abstract

Gustavo Mendes Maciel. **A multiresolution based method for phyloproteomic analysis of snake venoms**. Undergraduate Thesis (Bachelor). Institute of Mathematics and Statistics, University of São Paulo, São Paulo, 2019.

Snake venoms are highly complex mixtures of proteins and peptides. An analytical strategy often used to study them is the mass spectrometry-based proteomics (MS). Recent analysis using proteomic data produced through MS showed that there is a correlation between hierarchical clustering of venom proteomes of different snake species from the *Bothrops* genus and its phylogenetic classification. However, the overrepresentation of some species (e.g., *B. jararaca*) in commonly used protein databases likely results in a biased analysis. To mitigate this problem, a recent work generated phyloproteomic trees from *de novo* sequencing, that is, without usage of database; nonetheless, this approach generates a lot of false positive peptide candidates, which also introduces noise in the analysis. In this work, we proposed a novel methodology to tackle these problems, using raw data from MS experiments to infer those phyloproteomic trees. To accomplish this, a partitioning strategy was applied onto proteomic data, that were represented by matrices. Such matrices were utilized to generate phyloproteomic trees from a Bayesian inference approach, applying Markov Chain Monte Carlo methods. Lastly, the CADM statistical test was applied to compare the evolutionary trees obtained from the method developed in this work, with phylogenetic trees obtained from mitochondrial DNA data. The results showed that this method reaffirmed the relationship between the venom proteomic profiles and the phylogeny of different species of *Bothrops* snakes.

Keywords: Mass spectrometry, Snake venom, Phyloproteomics, Phylogenetics, Bayesian Inference, Multiresolution.

Sumário

Apêndices

Capítulo 1

Introdução

Venenos de serpentes são misturas complexas de proteínas e peptídeos, fundamentais para a sobrevivência das espécies venenosas, podendo ser utilizados como defesa ou para imobilizar e matar presas. Proteomas dos venenos (i.e., conjunto de todas essas proteínas) são estudados com o objetivo de identificar as composições dos venenos e assim entender a relação entre seus componentes e os efeitos que os mesmos acarretam em um outro organismo. A espectrometria de massas, uma técnica analítica que mede a razão massa/carga de moléculas ionizadas, vem sendo amplamente utilizada na identificação das proteínas presentes nos venenos de serpentes. Trabalhos recentes utilizando essa técnica (Andrade-Silva, Zelanis *et al.*, 2016; Andrade-Silva, Ashline *et al.*, 2018; Raposo, 2018) mostraram que há uma relação muito forte entre o proteoma dos venenos das diferentes espécies do gênero *Bothrops* e a classificação filogenética de tais serpentes.

As proteínas que compõem os venenos podem sofrer glicosilações, modificações pós-traducionais nas quais um glicano (i.e., um polissacarídeo) é ligado a essa proteína, formando uma glicoproteína. Os trabalhos de Andrade-Silva, Zelanis *et al.* (2016) e Andrade-Silva, Ashline *et al.* (2018) mostraram que as variações de glicanos e glicoproteínas nos venenos de diferentes espécies contribuem para a caracterização dos fenótipos desses venenos. Cladogramas obtidos por meio de aglomerações hierárquicas sobre os glicoproteomas se mostraram semelhantes aos obtidos com uso de DNA mitocondrial. Porém, não foram utilizadas informações de peptídeos na construção dos cladogramas e não houve nenhuma quantificação na comparação deles.

Com o objetivo de quantificar e melhorar esses resultados, o trabalho de Raposo (2018) fez o uso de métodos de inferência Bayesiana para gerar os cladogramas a partir de informações trazidas pelos peptídeos identificados na espectrometria de massas. Para identificar os peptídeos, os dados provenientes da espectrometria de massas foram comparados aos dados

teóricos de um banco de dados. Todavia, embora esse método tenha apresentado progressos, ele ainda possui alguns problemas. Para mitigar a superrepresentação de algumas espécies no banco de dados, foi utilizada uma estratégia de sequenciamento *de novo*, que dispensa o uso do banco. Porém, tal estratégia apresenta limitações, principalmente porque são gerados muitos peptídeos candidatos que são falsos positivos, o que impacta se não na topologia das árvores obtidas, no tamanho dos ramos dessas árvores.

Dessa forma, uma possibilidade de contornar os problemas apresentados acima seria fazer uso das informações adquiridas da espectrometria de massas sem a identificação de peptídeos. Uma maneira seria utilizar matrizes que são obtidas quando consideram-se as intensidades de detecção dos íons como função dos valores de massa/carga e do tempo de eluição (i.e., do tempo do experimento) obtidos pela espectrometria. Exemplos dessas matrizes são apresentados na Figura 1.1. Nesses exemplos, observamos que esse tipo de matriz é esparsa e com uma distribuição não-homogênea das informações. Para casos como esse, uma abordagem potencialmente adequada envolve o uso de classificadores multirresolução. Essa categoria de classificadores já se mostrou adequada para esse tipo de dado de treinamento no contexto de processamento de imagens, como visto em Vaquero *et al.* (2005). Todavia, tal técnica nunca foi explorada em análises filoproteômicas.

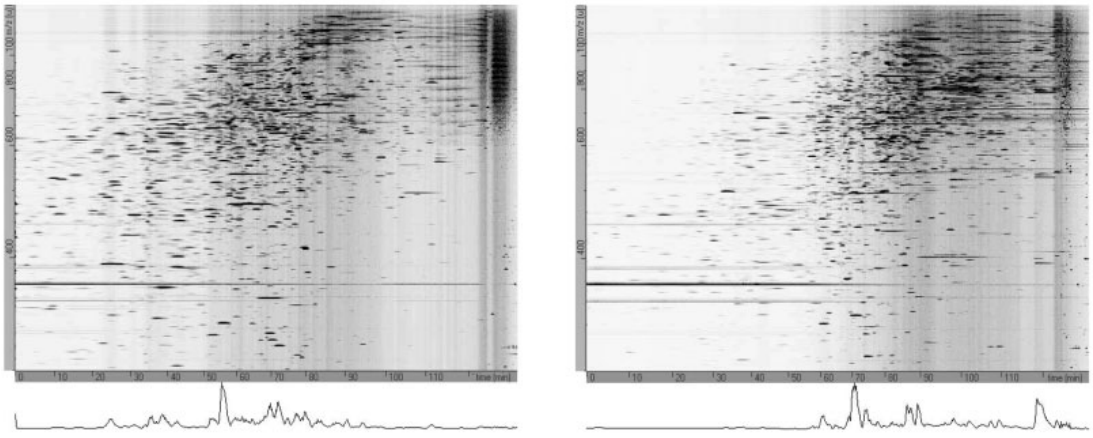


Figura 1.1: ***Exemplo de visualização de um experimento de espectrometria de massas.*** *Neste gráfico são exibidas as intensidades das detecções de íons como função dos valores de massa/carga (eixo y) e do tempo de eluição (eixo x). As áreas mais escuras do gráfico representam uma maior abundância de íons. Figura extraída de Fox e Serrano (2008).*

1.1 Objetivos

Este trabalho teve como objetivo geral desenvolver e implementar uma metodologia para construir árvores filoproteômicas a partir de dados proteômicos obtidos pela espectrometria de massas, sem a identificação de peptídeos, aplicando também a estratégia

de classificadores multirresolução, mitigando assim os problemas presentes nas técnicas mencionadas na seção anterior.

Como objetivo específico, aplicamos essa metodologia na construção de árvores evolutivas, a partir de informações proteômicas adquiridas dos venenos de sete serpentes do gênero *Bothrops* por meio da espectrometria de massas, e comparamo-as com as árvores filogenéticas obtidas com o uso de dados genômicos.

1.2 Organização da monografia

O restante dessa monografia está organizada da seguinte maneira:

No Capítulo 2 (Conceitos fundamentais) são apresentados alguns conceitos centrais de biologia e da proteômica baseada em espectrometria de massas, necessários para um bom entendimento desse trabalho. Além disso, são introduzidas algumas formalidades envolvendo operações sobre matrizes.

No Capítulo 3 (Metodologia) é feita uma descrição completa do *pipeline* desenvolvido, detalhando cada etapa, e também incluindo os programas escritos e as ferramentas utilizadas.

O Capítulo 4 (Resultados) expõe e discute os resultados obtidos pela aplicação da metodologia desenvolvida sobre os dados proteômicos de sete espécies de serpentes do gênero *Bothrops*.

No Capítulo 5 (Conclusão), é feita uma recapitulação do que foi realizado neste trabalho, discutindo os resultados obtidos e propondo futuras continuações nessa linha de pesquisa.

Por fim, temos os Apêndices A e B. O primeiro expõe as configurações tanto do *software* utilizado para processar os dados dos arquivos gerados nos experimentos de espectrometria de massas, como do espectrômetro de massa que gerou tais arquivos. O segundo, por sua vez, descreve modificações feitas no *software* mencionado, necessárias para o desenvolvimento deste trabalho.

Capítulo 2

Conceitos fundamentais

Neste capítulo serão introduzidos alguns conceitos fundamentais para um bom entendimento deste trabalho. Inicialmente serão apresentados os conceitos de biologia e espectrometria de massas que serão usados extensivamente ao longo deste texto. Na sequência, serão descritos os métodos estatísticos empregados na construção e comparação de árvores filoproteômicas e filogenéticas. Por último, serão formalizadas algumas ideias envolvendo matrizes que foram utilizadas no desenvolvimento deste projeto.

2.1 Conceitos biológicos

O material genético de um ser vivo, que é propagado de forma hereditária, é codificado por uma molécula chamada ácido desoxirribonucleico (DNA, do inglês *deoxyribonucleic acid*). O DNA é composto por duas fitas, que se dispõem em um formato conhecido como dupla hélice, constituindo uma longa cadeia de nucleotídeos. Essa cadeia é dividida em genes, unidades funcionais que fornecem instruções para a síntese de proteínas ou de moléculas de ácido ribonucleico (RNA, do inglês *ribonucleic acid*). A grande maioria da informação genética de um organismo eucarioto se encontra no núcleo das células, mas parte dela também está presente nas mitocôndrias.

Proteínas são formadas por pelo menos um polipeptídeo, que é uma longa cadeia de aminoácidos. O processo de síntese das proteínas através das informações contidas em um gene consiste, simplificadamente, em dois passos:

1. Transcrição: a partir da sequência de DNA de um gene é produzida uma molécula de RNA mensageiro (mRNA), que leva a informação genética até os ribossomos, onde são produzidas as proteínas.

2. Tradução: a sequência de mRNA é decodificada, de modo que cada tripla de nucleotídeos, chamada de códon, é associada ou à produção de um aminoácido, ou a um sinal de início ou parada. Esses sinais determinam o início e o fim da sequência de códons utilizada para sintetizar um polipeptídeo.

O processo acima descrito é conhecido como o caso geral do dogma central da biologia molecular.

2.1.1 DNA mitocondrial

A mitocôndria é uma organela exclusiva de organismos eucariotos, presente em praticamente todas as células da grande maioria desses organismos, e que se trata da "usina de força" celular. Sua principal função é a produção de adenosina trifosfato (ATP, do inglês *adenosine triphosphate*), molécula que fornece a energia necessária para o funcionamento das células, pelo processo conhecido como respiração celular.

Essa organela possui seu próprio genoma, que consiste no chamado DNA mitocondrial (mtDNA), muito utilizado no estudo das relações filogenéticas entre espécies próximas. O uso extensivo do mtDNA se deve ao fato dele evoluir mais rapidamente que os marcadores genéticos do núcleo das células, em animais, sendo possível obter mais variabilidade entre sequências de organismos próximos. Além disso, é uma molécula simples, abundante (estando presente em grande quantidade em diversos tipos de células) e bem conservada, pois, por ser herdada exclusivamente da fêmea, não sofre recombinação no processo de herança (Hassanin *et al.*, 2013).

Assim, sequências de genes mitocondriais, como por exemplo do citocromo b (Cytb) e da NADH dehidrogenase, subunidade 4 (ND4), são o "*gold standard*" para análises filogenômicas, sendo amplamente utilizadas como marcadores genéticos; por exemplo, tais sequências foram utilizadas para a construção de árvores filogenéticas de serpentes do gênero *Bothrops* (Fenwick *et al.*, 2009).

2.1.2 Venenos de serpentes

Venenos de serpentes são misturas altamente complexas de biomoléculas, compostas principalmente por proteínas e peptídeos. Dentre elas, estão presentes uma variedade de toxinas glicosiladas, como metaloproteinases, serino proteinases e fosfolipases (e.g., PLA), que essencialmente são enzimas proteolíticas e lipolíticas (Andrade-Silva, Zelanis *et al.*, 2016). A sinergia entre esses diferentes compostos tem uma importância fundamental na sobrevivência das serpentes peçonhentas, resultando em venenos que podem ser utilizados tanto de forma defensiva como para imobilizar e matar presas.

Nesse trabalho, a análise será focada em algumas espécies do gênero *Bothrops*. Esse gênero apresenta uma enorme diversidade de espécies, englobando a maioria das serpentes venenosas que estão distribuídas pelo território da América do Sul e América Central. Os trabalhos de Andrade-Silva, Zelanis *et al.* (2016), Andrade-Silva, Ashline *et al.* (2018) e Raposo (2018) apresentam evidências de que os perfis proteômicos e glicoproteômicos dos venenos de serpentes desse gênero correlacionam com a filogenia delas.

2.1.3 Árvores evolutivas

Árvores evolutivas são diagramas que representam as relações evolutivas entre diversas entidades de uma unidade taxonômica, como por exemplo as espécies de serpentes do gênero *Bothrops*. Também podem ser chamadas de árvores filogenéticas, no caso em que são levadas em conta as similaridades e diferenças genéticas entre as entidades, ou de árvores filoproteômicas, quando os proteomas são utilizados como referência para construí-las. A Figura 2.1 apresenta um exemplo de árvore evolutiva.

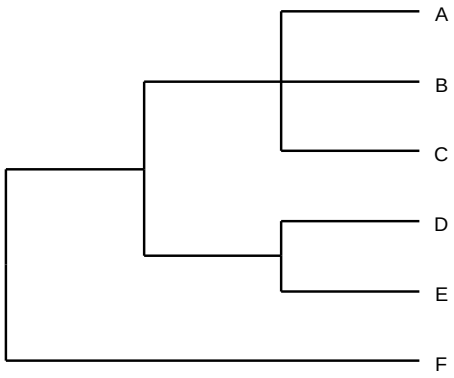


Figura 2.1: ***Exemplo de árvore evolutiva com seis entidades, rotuladas de A a F.***

O padrão de ramificação das árvores reflete uma hipótese acerca da evolução dos organismos a partir de ancestrais comuns. As folhas (que na Figura 2.1 estão rotuladas de A a F) representam os organismos que estão sendo estudados, e os pontos de ramificação (ou nós internos), indicam o ancestral comum, que não é conhecido, mais recente dos organismos presentes na ramificação em questão. Dois organismos são mais relacionados quando têm um ancestral comum recente, e menos relacionadas caso contrário. Por exemplo, D e E estão mais relacionadas do que E e F.

As ramificações denotam possíveis divergências a partir de sua raiz, que geraram descendentes diferentes. Os comprimentos de suas arestas geralmente são interpretados como o número de substituições por campo, que denotam a quantidade de substituições que ocorreram em uma sequência de aminoácidos, por exemplo. As arestas e nós internos de uma árvore também podem ser rotulados com informações pertinentes.

Podem também ocorrer multifurcações (ou politomias), como no caso das entidades A, B e C no exemplo mostrado. Em geral, isso significa que não há informação suficiente para determinar a ordem das ramificações.

2.2 Proteômica baseada em espectrometria de massas

A proteômica consiste na análise em larga escala do proteoma de um organismo ou sistema, e é considerada complementar à genômica, a nível de proteína. As técnicas e métodos utilizados permitem a identificação e quantificação de proteínas e peptídeos em uma amostra, bem como o estudo da expressão proteica e de sua influência na atividade dos genes e das células.

Uma técnica muito utilizada na proteômica é a espectrometria de massas (MS, do inglês *mass spectrometry*), que mede a razão massa/carga (m/z) das moléculas ionizadas de uma amostra. Tipicamente, um espectrômetro de massas consiste de três partes: uma fonte de ionização, um analisador de massa e um detector.

As partículas da amostra são ionizadas na fonte de ionização, adquirindo assim uma carga, positiva ou negativa. No analisador de massa, esses íons são submetidos a um campo elétrico e/ou magnético, o que permite o cálculo dos seus valores de massa/carga a partir das seguintes leis:

$$\begin{cases} F = Q(E + v \times B) \\ F = ma. \end{cases} \tag{2.1}$$

Aqui, F é a força aplicada ao íon, m e Q são a massa e a carga do íon, respectivamente, a é a aceleração, E é o campo elétrico e $v \times B$ é o produto vetorial entre a velocidade do íon e o campo magnético. Juntando as duas expressões obtemos

$$\frac{m}{Q}a = E + v \times B. \tag{2.2}$$

Com essa equação é possível encontrar a razão massa/carga de um íon. Passando pelo analisador de massa, os íons vão de encontro ao detector, que capta e amplifica os sinais obtidos pela colisão (ou por algum outro evento) entre os íons e o detector. A partir desses sinais é produzido um espectro de massas (ver Subseção 2.2.1).

Existem vários tipos de analisadores de massa, cada um com suas vantagens e desvantagens. Apesar das diferenças entre eles, todos partem da ideia do movimento de um íon em um campo elétrico ou magnético, regido pela Equação 2.2. Um exemplo simples e bem

conhecido de analisador de massa é o TOF (do inglês *time-of-flight*), esquematizado na Figura 2.2. Ele utiliza um campo elétrico para acelerar os íons e, então, calcula a massa/carga com base no tempo que os íons levam pra colidir com o detector.

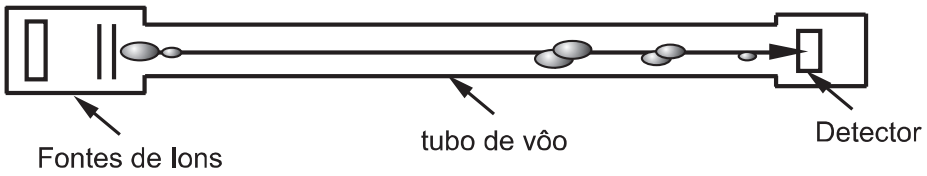


Figura 2.2: ***Esquema de um analisador de massa TOF.*** *Figura extraída de* Chiaradia *et al. (2008).*

Uma técnica de espectrometria de massas que vale mencionar é a *Tandem* (MS/MS ou MS^2). Nesse procedimento, dois ou mais espectrômetros de massas são acoplados com o objetivo de deixar a análise mais refinada. O primeiro (MS1) separa os íons da amostra ionizada pelos valores de m/z obtidos. Seleciona-se então uma razão massa/carga específica, e os íons que possuem esse valor, chamados precursores, são fragmentados e introduzidos no segundo espectrômetro (MS2), que calcula os valores de m/z dos fragmentos. Com isso, é possível fazer uma análise mais precisa e minuciosa da mistura, diferenciando íons com valores muito próximos de massa/carga.

Para uma descrição geral e mais detalhada sobre a proteômica baseada em espectrometria de massas, bem como sobre as diferentes abordagens tecnológicas que geralmente são utilizadas, ver Aebersold e Mann (2003). O artigo de Colinge e Bennett (2007) também aborda esse assunto, focando mais nos aspectos computacionais da proteômica baseada em MS.

2.2.1 Espectro de massas

O resultado de um experimento de espectrometria de massas é geralmente apresentado na forma de um espectro de massas, que é um gráfico no qual as intensidades das detecções (também chamadas de abundância relativa) dos íons são exibidas como função dos valores de massa/carga. Um exemplo de espectro de massa pode ser visto na Figura 2.3.

Cada pico (eixo y) representa a abundância relativa de um íon, que tem sua massa/carga indicada no eixo x. A abundância relativa está relacionada com a frequência de detecção do íon pelo detector, ou seja, quanto maior a altura do pico, maior é a abundância relativa e, consequentemente, a frequência de detecção do íon.

Com esses dados é possível identificar os peptídeos presentes na mistura, utilizando dados teóricos armazenados em bancos de dados ou usando uma estratégia de sequenciamento *de novo*. Porém, essas duas abordagens possuem alguns problemas. Na primeira,

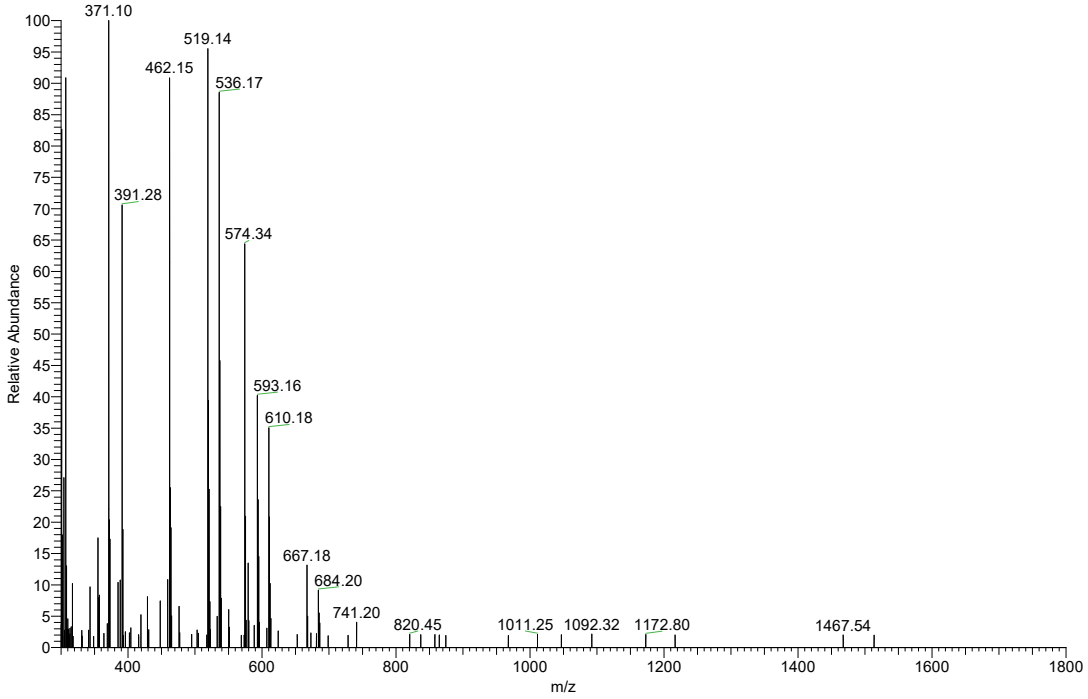


Figura 2.3: ***Exemplo de um espectro de massas.*** *Extraído de um experimento de espectrometria de massas feito com uma amostra de veneno da serpente B. jararaca.*

a identificação pode ser prejudicada se há uma subrepresentação no banco de dados de sequências protéicas de determinadas espécies. É o que ocorre, por exemplo, no caso das serpentes do gênero *Bothrops*, onde a *B. jararaca* possui muito mais dados nos bancos do que as demais espécies, o que gera vieses indesejáveis. O sequenciamento *de novo*, por sua vez, utiliza diretamente as informações contidas nos dados brutos, o que resulta num maior número de peptídeos identificados; porém, nessa estratégia são gerados muitos peptídeos falsos positivos, introduzindo ruídos que potencialmente interferem negativamente nos resultados.

2.2.2 Cromatografia líquida associada à espectrometria de massas

Na maioria das vezes os experimentos proteômicos são feitos a partir de amostras complexas, com uma variedade muito grande de proteínas em diferentes concentrações. Uma técnica muito utilizada para separar as proteínas e obter uma amostra mais simples é a cromatografia líquida. Quando associada à espectrometria de massas, essa técnica é chamada de LC-MS (abreviado do inglês *liquid chromatography-mass spectrometry*). Uma análise completa de LC-MS é chamada de **corrida**.

Na cromatografia líquida, a mistura a ser analisada é inserida em um solvente líquido,

que é bombeado até uma coluna preenchida com um material adsorvente. Os componentes presentes na mistura passam com diferentes velocidades pela coluna, com base na interação de cada um deles com o adsorvente, e o tempo levado para percorrê-la é chamado de **tempo de retenção**, ou **tempo de eluição** (RT, do inglês *retention time*). Ou seja, diferentes tempos de eluição caracterizam componentes diferentes. Após separados, os componentes são levados até o espectrômetro de massas, onde serão individualmente analisados.

Em um sistema LC-MS, os dados obtidos ao final do experimento possuem, além dos valores de massa/carga e intensidade dos íons, informações sobre os tempos de retenção. Essa nova dimensão aumenta ainda mais a capacidade de identificação de peptídeos e proteínas em uma amostra.

2.2.3 Fragmentação de proteínas

Proteínas são moléculas muito grandes para ionização e voo no espectrômetro. Por conta disso, é necessário que as proteínas da amostra passem por dois processos de fragmentação. O primeiro deles é a digestão por tripsina, uma serino-proteinase. Tripsina é uma enzima que cliva (i.e., corta) proteínas seletivamente na carboxila (i.e., região C-terminal) dos aminoácidos lisina e arginina, quando estes não estão ligados a uma prolina na região C-terminal. O resultado da digestão por tripsina é um conjunto de pequenas sequências de aminoácidos conhecidas como peptídeos. Esse conjunto é então fracionado por métodos tal como a cromatografia líquida e injetado no espectrômetro, onde os peptídeos passam pelo segundo processo de fragmentação, dessa vez por colisão com um gás. Por fim, os fragmentos resultantes são ionizados pela fonte de ionização, de acordo com o que foi explicado acima.

2.2.4 Mapas de intensidade de íons

As corridas LC-MS são comumente representadas por matrizes bidimensionais, nas quais as intensidades de detecção dos íons são apresentadas como função dos valores de massa/carga e do tempo de retenção. A Figura 1.1 contém dois exemplos dessa representação.

Neste trabalho, essas matrizes serão referidas como **mapas de intensidade de íons**, ou simplesmente **mapas de íons**, que são a tradução direta de *ion intensity maps* e *ion maps*, respectivamente, terminologia utilizada pelo programa de análise de dados de LC-MS Progenesis QI (Nonlinear Dynamics, Newcastle, Reino Unido) para se referir a essa estrutura de dados.

2.3 Inferência Bayesiana de árvores evolutivas

A inferência Bayesiana é uma estratégia muito utilizada na construção de árvores evolutivas, apresentando algumas vantagens em relação a outros métodos, como a possibilidade de incorporar informações já conhecidas, a facilidade de interpretação dos resultados e otimizações computacionais.

De forma geral, a inferência Bayesiana é utilizada para aumentar a informação *a priori* de uma quantidade de interesse θ, dada pela distribuição de probabilidade $p(\theta)$, a partir da observação de uma outra quantidade x relacionada a θ. O Teorema de Bayes é a ferramenta empregada para atualizar tal informação, obtendo assim a distribuição *a posteriori* $p(\theta \mid x)$, de modo que

$$p(\theta \mid x) = \frac{p(\theta, x)}{p(x)} = \frac{p(x \mid \theta)\ p(\theta)}{p(x)} = \frac{p(x \mid \theta)\ p(\theta)}{\int p(\theta, x) d\theta}. \tag{2.3}$$

Aqui, $p(x \mid \theta)$ é a função de verossimilhança, que descreve a probabilidade de x em função de diferentes valores de θ, e $p(x)$ é a probabilidade total de x, integrada sobre o conjunto de valores de θ.

Aplicada ao problema de inferir a filogenia de um grupo de organismos, pode-se reescrever a Equação 2.3 como

$$f(\tau, \upsilon, \theta \mid X) = \frac{f(\tau, \upsilon, \theta)\, f(X \mid \tau, \upsilon, \theta)}{f(X)} \tag{2.4}$$

em que τ é a topologia da árvore, υ é um vetor com os comprimentos dos ramos da árvore, θ é um vetor de parâmetros do modelo de substituição e X é uma matriz com dados morfológicos, de nucleotídeos e/ou de aminoácidos de cada organismo.

Os modelos de substituição descrevem hipóteses de como as mutações acontecem nas sequências de dados (nucleotídeos, aminoácidos etc.) ao longo do tempo. O ajuste dos parâmetros de um modelo é importante para determinar corretamente as distâncias entre os organismos analisados e inferir uma história evolutiva que seja coerente.

Desse modo, a distribuição *a posteriori* $f(\tau, \upsilon, \theta \mid X)$ é utilizada para decidir qual árvore, definida pelos valores dos parâmetros τ, υ e θ, é a mais provável, baseando-se nos dados da matriz X. Entretanto, não é possível computar a Equação 2.4 analiticamente. Logo, para obter uma estimativa da distribuição *a posteriori*, são empregados métodos de Monte Carlo com cadeias de Markov (MCMC, do inglês *Markov chain Monte Carlo*).

2.3.1 Monte Carlo com cadeias de Markov

Os métodos MCMC consistem na amostragem aleatória de uma distribuição de probabilidade que é difícil de ser amostrada diretamente, e são muito utilizados nos casos em que a distribuição alvo é multidimensional. Para isso, são construídas cadeias de Markov que, quando simuladas por um número suficiente de iterações, alcançam uma distribuição estacionária que aproxima a distribuição desejada.

Um método MCMC bastante importante na inferência de árvores evolutivas é o algoritmo de Metropolis-Hastings. Para simplificar a notação, vamos denotar (τ, v, θ) por Ψ. O algoritmo funciona da seguinte maneira:

(1) Seja Ψ_0 o estado inicial da cadeia de Markov, escolhido arbitrariamente.

(2) Sendo Ψ_k o estado atual na iteração k, proponha um novo estado Ψ' com probabilidade $q(\Psi' \mid \Psi_k)$, chamada distribuição de proposta.

(3) Calcule a probabilidade r de aceitar o estado Ψ', tal que

$$r = \min\left[1, \frac{f(X \mid \Psi')}{f(X \mid \Psi_k)} \cdot \frac{f(\Psi')}{f(\Psi_k)} \cdot \frac{q(\Psi_k \mid \Psi')}{q(\Psi' \mid \Psi_k)}\right]. \tag{2.5}$$

(4) Gere um número aleatório u, uniformemente distribuído no intervalo $(0, 1)$. Se $u < r$, então Ψ' é aceito e então $\Psi_{k+1} = \Psi'$. Caso contrário, $\Psi_{k+1} = \Psi_k$, ou seja, o estado se mantém o mesmo.

(5) Se k ainda não atingiu o número (geralmente grande) de iterações necessárias para a cadeia convergir, volte para o passo 2.

(6) Devolva a coleção de estados aceitos.

Se a cadeia de Markov foi corretamente construída, tendo-se escolhido uma distribuição de proposta $q(\Psi' \mid \Psi_k)$ adequada, então a coleção de estados aceitos devolvida ao fim do processo é uma distribuição estacionária que aproxima a distribuição *a posteriori* $f(\Psi \mid X)$. A Figura 2.4 ilustra de maneira bem intuitiva o funcionamento desse algoritmo.

Para mais detalhes acerca da inferência Bayesiana de árvores evolutivas, ver Huelsenbeck e Ronquist (2001), Ronquist e Huelsenbeck (2003) e Altekar *et al.* (2004). A dedução formal do algoritmo de Metropolis-Hastings é explicada minuciosamente em sua página na Wikipédia.

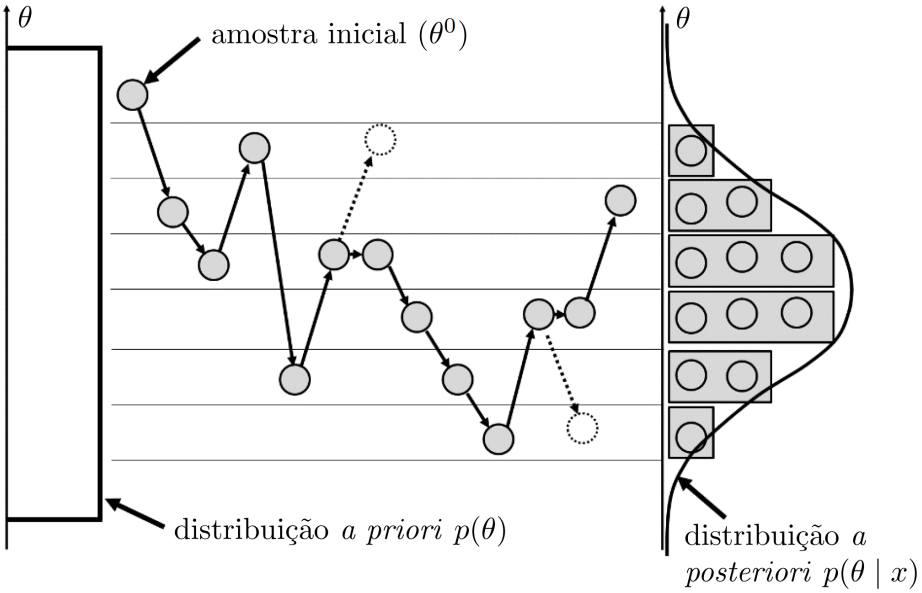


Figura 2.4: ***Exemplo ilustrativo do funcionamento do algoritmo de Metropolis-Hastings.*** *Os círculos cinzas e as setas preenchidas representam as transições de estados que de fato ocorreram na cadeia de Markov. Já os círculos e setas pontilhadas representam transições que não foram aceitas. Nesse exemplo a distribuição a priori é uma uniforme, que é uma escolha comum quando não se tem muita informação sobre a distribuição real. Figura adaptada de Lee et al. (2015).*

2.4 Teste CADM

O teste CADM (do inglês *congruence among distance matrices*) é um método que estima a congruência entre um número arbitrário de matrizes de distância. A hipótese nula é de que todas as matrizes são incongruentes entre si. Se essa hipótese for rejeitada, testes *a posteriori* podem ser aplicados para se obter mais informações acerca da congruência ou não entre certas matrizes.

Aplicado ao contexto de árvores evolutivas, é possível avaliar a congruência entre diversas árvores representadas por matrizes de distância, sendo duas árvores congruentes se descrevem uma história evolutiva semelhante, e incongruentes caso contrário. A semelhança entre duas árvores pode ser visualizada principalmente por suas topologias, mas também pelo comprimento dos ramos.

Dadas as matrizes de distâncias, o teste segue da seguinte forma:

(1) Transforme a parte superior (ou inferior) à diagonal principal de cada matriz em um vetor (assumindo que todas as matrizes são simétricas). Cada vetor será uma linha de uma tabela T, que será usada nos próximos passos.

(2) Para cada linha de T, transforme seus valores de acordo com uma relação de ordem (*ranking*), crescente ou decrescente, de tal modo que os valores passem a ser números

de 1 a n, sendo n o número de elementos em cada linha. Informalmente, podemos dizer que duas matrizes são congruentes se seus *rankings* são parecidos.

(3) Calcule W, o coeficiente de concordância de Kendall entre as linhas de T. Transforme W na estatística χ^2 de Friedman, obtendo assim χ^2_{ref}, que servirá de referência para o teste. O valor de W é limitado entre 0 (não congruência) e 1 (total congruência).

(4) Permute cada matriz de distâncias e faça os mesmos cálculos do item anterior, obtendo χ^{2*}. Faça o passo 1 novamente, dessa vez para as matrizes permutadas.

(a) Para o teste global, todas as matrizes são permutadas aleatoriamente e independentemente. Nesse caso a hipótese nula é de que todas as matrizes são incongruentes entre si.

(b) Para os testes *a posteriori*, uma matriz de distâncias é permutada por vez, de modo que a hipótese nula seja a incongruência da matriz escolhida em relação a todas as outras.

(5) Repita várias vezes o passo anterior, até que seja obtida uma boa estimação da distribuição da estatística χ^2. Adicione o valor de referência χ^2_{ref} à distribuição.

(6) Por fim, calcule a probabilidade permutacional P como sendo a proporção dos valores χ^{2*} que são maiores ou iguais a χ^2_{ref}. Assim, há matrizes congruentes se P for suficientemente pequeno (menor que um valor α predefinido), ou seja, se χ^2_{ref} é maior ou igual à maioria dos valores da distribuição obtida através das permutações.

Para uma descrição mais detalhada do teste CADM, incluindo os cálculos dos coeficientes utilizados e outras considerações, consulte Legendre e Lapointe (2004).

2.5 Operações sobre matrizes

Antes de prosseguir para o próximo capítulo, é necessário definir alguns termos que serão utilizados ao longo do texto. Mais especificamente, apresentaremos algumas definições de operações sobre matrizes, uma vez que utilizaremos esse tipo de estrutura de dados para armazenar as informações obtidas nos experimentos de LC-MS. As definições a seguir são baseadas no trabalho de Vaquero *et al.* (2005).

Podemos definir uma **matriz** como uma função de E em L, sendo E um subconjunto finito de $\mathbb{N} \times \mathbb{N}$ e L um subconjunto de $\mathbb{R}$. O conjunto de todas as matrizes será denotado por L^E. Uma **janela** é qualquer subconjunto finito W de E, e o número de pontos presentes nela é denotado por $|W|$.

Uma **configuração** é uma função de W em L, e pode ser escrita como uma matriz M com domínio restrito a W, denotada por $M{\restriction}_W$. O espaço de todas as configurações possíveis de W em L será denotado por L^W. Podemos também obter uma configuração $M{\restriction}_{W_t}$ transladando uma janela $W = \{w_1, w_2, \dots, w_n\}$ por $t \in E$, em que $n = |W|$ e W_t é a janela transladada, dada por $W_t = \{w_1 + t, w_2 + t, \dots, w_n + t\}$.

Por fim, um **operador de redução** é definido como uma função $\psi : L^W \longrightarrow L$, que mapeia configurações de W em L a elementos de L.

Capítulo 3

Metodologia

Nesse capítulo será apresentado o fluxo do trabalho de maneira geral, desde a obtenção dos dados brutos provenientes dos experimentos de LC-MS, até a geração e comparação das árvores filoproteômicas construídas a partir desses dados. Ao decorrer do capítulo também serão expostos os recursos externos utilizados e os scripts desenvolvidos.

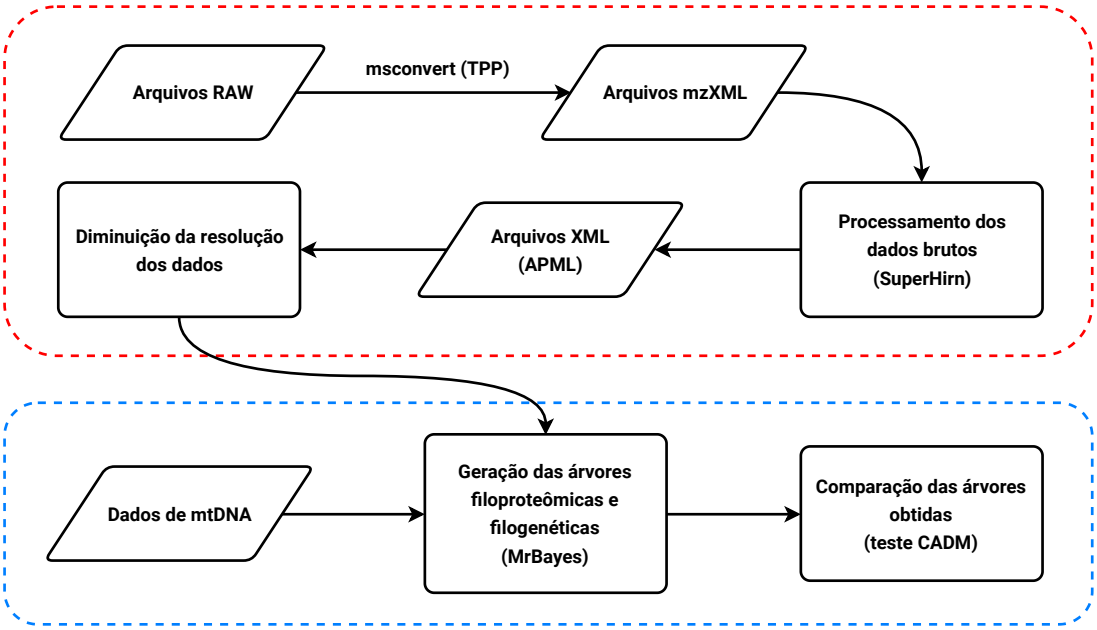


Figura 3.1: ***Fluxograma do encadeamento de processos.*** *O retângulos representam os processos e os paralelogramos representam os dados utilizados ao longo do encadeamento. A parte envolta pela linha tracejada azul foi implementada em* Raposo *(2018), enquanto que a envolta pela linha tracejada vermelha é uma contribuição deste trabalho.*

É importante deixar claro que este trabalho é uma continuação de Raposo (2018) e, portanto, parte do *pipeline* desenvolvido contém partes que já foram implementadas anteriormente. A Figura 3.1 apresenta um fluxograma que descreve o encadeamento de

processos desenvolvido para este trabalho, além de destacar o que é novo e o que já havia sido feito. O *pipeline* foi implementado na linguagem de programação Python (versão 3) e encontra-se disponível, de forma livre e gratuita, no seguinte repositório:

https://github.com/msreis/MITE.

Nas próximas seções, exploraremos cada etapa do encadeamento de processos apresentado acima.

3.1 Processamento dos dados brutos

O espectrômetro de massa utilizado nos ensaios foi o LTQ Orbitrap Velos MS, da Thermo Fisher Scientific. Os dados brutos gerados por esses ensaios foram obtidos em arquivos no formato proprietário RAW, cada um contendo dados de uma corrida do experimento. Foram analisados os venenos de sete espécies de serpentes do gênero *Bothrops*, e para cada uma foram feitas duas corridas, totalizando assim quatorze arquivos.

Para a extração e o processamento dos dados desejados desses arquivos, foi utilizada a ferramenta SuperHirn, de código aberto, desenvolvida pelo grupo do Prof. Ruedi Aebersold do Institute of Molecular Systems Biology (ETHZ, Switzerland). O *software* foi programado em C++ e é compatível com plataformas Unix. Entretanto, o SuperHirn não suporta arquivos no formato RAW, somente no formato aberto mzXML (descrito em Pedrioli *et al.* (2004)). A fim de realizar a conversão de um formato para o outro, foi usado o *software* `msconvert`, embutido em uma coleção de ferramentas integradas para proteômica baseada em espectrometria de massas, a Trans-Proteomic Pipeline (TPP).

As funcionalidades do SuperHirn são separadas em módulos, formando dois grupos: os módulos de pré-processamento e os de pós-processamento dos dados. Os de pré-processamento estão listados abaixo (devendo ser executados exatamente nessa ordem), e cada um será detalhado em seguida:

1. Extração das *features* MS1;
2. Análise de similaridade entre as corridas e construção da topologia de alinhamento;
3. Alinhamento múltiplo entre as corridas e construção do *MasterMap*.

Cada módulo gera arquivos que são utilizados pelo módulo seguinte. O primeiro constrói um arquivo XML para cada corrida analisada, contendo informações sobre as *features* MS1, que são definidas pela massa/carga, tempo de retenção e carga de uma aglomeração de íons detectados com valores de m/z muito próximos.

O segundo módulo usa essas informações para construir uma topologia hierárquica de alinhamento baseada na similaridade entre as corridas. Essa topologia é uma árvore binária em que a proximidade de dois nós, que representam as corridas, indica a similaridade entre eles. O alinhamento é necessário para que a dimensão do tempo de retenção possa ser levada em conta nos processos de análise dos dados, pois podem haver flutuações desses valores entre diferentes corridas. Tais flutuações são atribuídas ao processo de separação dos peptídeos pela cromatografia líquida.

A topologia de alinhamento serve então de entrada para o último módulo de pré-processamento, que efetua o alinhamento múltiplo e a fusão entre as corridas em um único arquivo XML, denominado *MasterMap*. Isso é feito tomando, sequencialmente, as duas corridas mais semelhantes, alinhando-as e fundindo-as em uma nova corrida, até que sobre somente uma corrida na topologia, que é o *MasterMap*. O processo de alinhamento utiliza uma implementação modificada do método *accurate mass retention time pairs* (AMRT) (Silva *et al.*, 2005). A ideia é identificar *features* comuns entre as duas corridas e normalizar seus tempos de retenção, corrigindo as flutuações mencionadas no parágrafo anterior. A fusão das corridas é composta pelas *features* comuns e pelas que só aparecem em uma delas. Nesse trabalho foi construído um *MasterMap* para cada espécie.

Todos os módulos de pós-processamento partem do princípio de que o *MasterMap* já esteja construído. Os utilizados nesse trabalho foram os seguintes:

4. Normalização das intensidades das *features* MS1;

5. Alinhamento entre corridas, tomando uma delas como referência.

O módulo 4 tem como função normalizar as intensidades das *features* MS1 de cada corrida contida no *MasterMap*. Assim como o alinhamento entre diferentes corridas, essa normalização é essencial para corrigir variações dos valores de intensidade obtidos em diferentes experimentos, causadas por diferenças nos volumes das amostras ou por variações no processo de ionização de peptídeos, por exemplo. Isso é feito utilizando uma versão modificada do método *central tendency normalization* (Yang *et al.*, 2002). O processo de normalização segue a estrutura da topologia de alinhamento construída no módulo 2, começando sempre com as duas corridas mais próximas. Ao fim, é obtida uma média ponderada dos valores das intensidades de cada *feature* MS1, usada para normalizar esses valores. Para mais detalhes acerca da implementação ver Mueller *et al.* (2007).

Já o módulo 5 foi implementado especificamente para a realização deste trabalho, ou seja, não está presente nas versões originais do SuperHirn. Ele realiza um alinhamento do tempo de retenção entre as corridas, assim como no módulo 3, mas tomando uma delas como referência (parâmetro dado pelo usuário). A saída compreende os arquivos XML

das corridas alinhadas, com os valores modificados pelo alinhamento. O que motivou a implementação desse módulo foi a necessidade de alinhar os *MasterMaps* de cada espécie entre si, e receber como saída os *MasterMaps* alinhados, algo que não é possível utilizando os módulos do programa original. A Figura 3.2 ilustra como os módulos foram utilizados para gerar os arquivos finais.

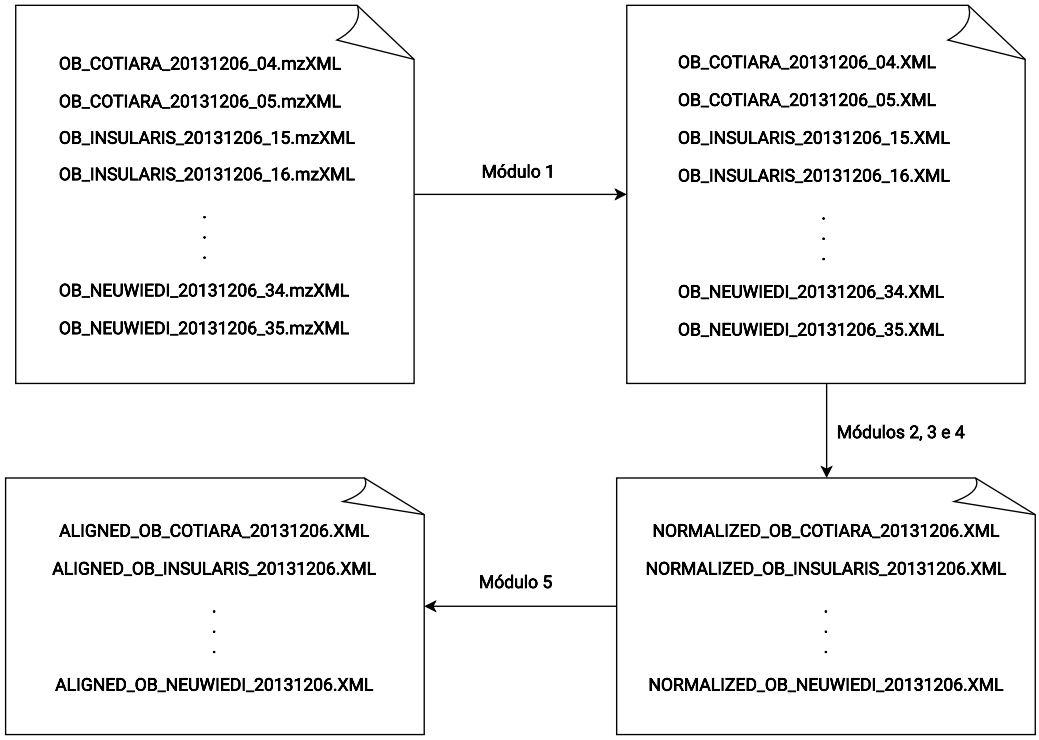


Figura 3.2: ***Diagrama da utilização dos módulos do SuperHirn.*** *Cada retângulo contém a lista de arquivos gerados pelos módulos indicados nas setas. Note que após os módulos 2, 3 e 4 o número de arquivos ficou pela metade. Isso acontece porque, no módulo 3, as duas corridas de cada espécie são fundidas em um só arquivo.*

Todos arquivos XML gerados durante a execução desses módulos são baseados no formato APML (Annotated Putative Markup Language), especificado em Brusniak *et al.* (2008). Para mais detalhes acerca da implementação das funcionalidades do SuperHirn, ver Mueller *et al.* (2007). O Apêndice B detalha um pouco mais as modificações feitas no programa original.

Para que todas essas etapas de processamento dos dados brutos fossem executadas com a precisão desejada, fez-se necessária a configuração dos parâmetros do SuperHirn, espelhando a configuração do espectrômetro de massa utilizado. A extração dos parâmetros de configuração do LTQ Orbitrap Velos foi feita com o *software* Xcalibur, da Thermo Fisher Scientific. No Apêndice A as configurações em questão são apresentadas com mais detalhes.

3.1.1 Representação dos dados na memória

De posse dos arquivos XML finais (os que foram gerados após a execução do módulo 5), o próximo passo foi representar esses dados como mapas de intensidade de íons. Com essa representação é possível explorar as propriedades de esparsidade e não homogeneidade das informações contidas nos mapas.

Foram desenvolvidas algumas classes e scripts para implementar essa representação. Um pacote utilizado que foi fundamental é o scipy.sparse, que implementa diversas maneiras de representar e fazer operações com matrizes esparsas de modo eficiente. Cada classe desse pacote dispõe de uma implementação que é eficiente para um uso específico. Por exemplo, as classes `coo_matrix` e `lil_matrix` foram empregadas na construção dos mapas de íons, enquanto que as classes `csr_matrix` e `csc_matrix` foram empregadas em situações nas quais houve necessidade de executar operações envolvendo eles. É possível fazer conversões entre as classes, em alguns casos em tempo linear.

Uma API que também foi muito útil na construção dos mapas é a ElementTree, que possui funções que auxiliam na interpretação dos arquivos XML.

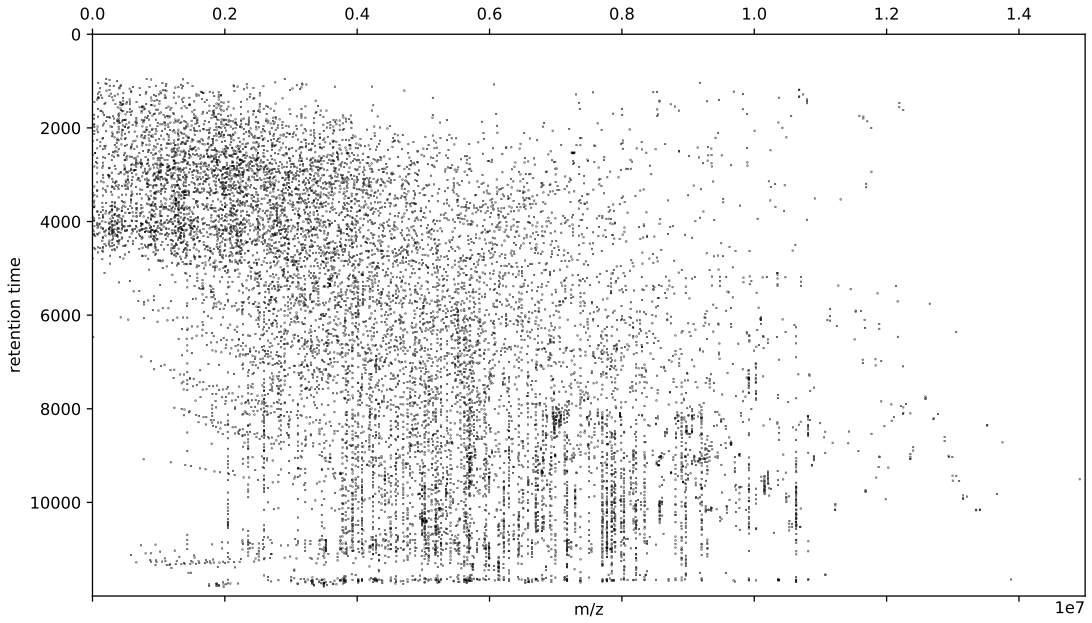


Figura 3.3: ***Exemplo de visualização de matriz construída a partir de arquivo XML (mapa de intensidade de íons).*** *Esse mapa foi construído a partir do arquivo XML contendo os dados proteômicos do veneno da espécie B. jararaca.*

Na construção dos mapas de íons (ver Figura 3.3), as linhas e colunas das matrizes foram representadas pelo tempo de retenção e pelas razões massa/carga, respectivamente. As intensidades de detecção dos íons são visualizadas por meio de pontos em escala de cinza. Nos arquivos XML, essas três propriedades têm valores decimais. Os valores de tempo de

retenção possuem valores de 0 a 120, com 2 casas decimais; os de massa/carga possuem valores de 300 a 1800, com 4 casas decimais. Ambos são discretizados multiplicando-os por potências de 10, obtendo assim números inteiros, que podem servir de índices nas matrizes. O modo como os valores de intensidade foram manipulados é discutido na próxima seção.

3.2 Diminuição da resolução dos dados

Os mapas de íons construídos a partir dos arquivos XML são representados na memória por matrizes cujas dimensões são muito grandes (pela necessidade de multiplicar suas dimensões reais por potências de 10). Para que a inferência das árvores filoproteômicas a partir dos dados contidos nos mapas de íons fosse computacionalmente viável, viu-se necessária a redução da dimensionalidade deles. Para isso, foi desenvolvido um algoritmo, esboçado no Programa 3.1.

Programa 3.1 Redução de dimensionalidade.

```
1   função reduz_dimensionalidade(M, m, n, w, h, tamanho_max)
2       antiga ← M
3       enquanto m * n > max_tamanho
4           m′ ← ⌈m/w⌉
5           n′ ← ⌈n/h⌉
6           nova ← matriz m′ × n′ vazia
7           W ← {(i, j) ∈ ℕ² | 0 ≤ i ≤ h − 1, 0 ≤ j ≤ w − 1}   ▷ Janela com dimensões w × h
8           para i de 0 até n passo h faça
9               para j de até m passo w faça
10                  W_t ← {w + (i, j) | w ∈ W, w + (i, j) ∈ M}   ▷ Janela transladada
11                  nova (i/h, j/w) = ψ(antiga⌈_{W_t})
12          m ← m′
13          n ← n′
14          antiga ← nova
15      devolva nova
16  fim
```

A função `reduz_dimensionalidade` recebe como parâmetros um mapa de íons M com dimensões $m \times n$, as dimensões $w \times h$ das janelas que serão consideradas na redução e o tamanho máximo $tamanho\_max$ que o mapa reduzido pode ter. Aqui, W é uma janela e $\psi : L^W \longrightarrow L$ é um operador de redução (Seção 2.5). A ideia do algoritmo se baseia na utilização do operador sobre configurações de M obtidas pelas translações da janela W. O objetivo do operador de redução é, de alguma forma, reduzir a informação contida em uma configuração.

O projeto de um operador depende de como a informação (nesse caso as intensidades dos íons) está representada. Os valores originais das intensidades são números reais, o que torna o processo de inferência das árvores inviável. As subseções a seguir apresentam duas maneiras distintas de discretizar esses valores, e consequentemente dois projetos diferentes de operadores de redução.

3.2.1 Discretização binária

Como uma primeira tentativa de fazer o *pipeline* funcionar, os valores das intensidades de detecção dos íons foram discretizados da seguinte maneira: se M é o mapa de íons e x e y são valores arbitrários de massa/carga e tempo de retenção, respectivamente, então a intensidade $M(x, y)$ é dada por

$$M(x, y) = \begin{cases} 1, & \text{se existe uma } \textit{feature} \text{ com } m/z = x \text{ e RT} = y \\ 0, & \text{caso contrário.} \end{cases} \tag{3.1}$$

Considerando um número real $f \in [0, 1]$, o operador de redução ψ é definido da seguinte maneira:

$$\psi(M\restriction_W) = \begin{cases} 1, & \text{se } \dfrac{1}{|W|} \displaystyle\sum_{w \in W} 1_{\{M(w)=1\}} \geq f \\ 0, & \text{caso contrário.} \end{cases} \tag{3.2}$$

Ou seja, $\psi(M\restriction_W)$ é 1 se a frequência de pontos $w \in W$ tais que $M(w) = 1$ é maior ou igual a f, e 0 caso contrário.

3.2.2 Discretização por quartis

A discretização binária feita na subseção anterior possivelmente introduz ruídos nas análises, pois não leva em conta a magnitude das intensidades de detecção de íons, somente indica se houve ou não detecção. Como uma discretização mais refinada dos dados aumenta muito o tempo gasto para realizar a inferência das árvores, decidiu-se discretizar os valores de intensidade com base nos quartis Q_1, Q_2 e Q_3, que dividem o conjunto dos valores de forma que

$$\begin{cases} Q_1 : & \text{mediana entre o valor mais baixo do conjunto de dados e o } Q_2 \\ Q_2 : & \text{mediana do conjunto de dados} \\ Q_3 : & \text{mediana entre } Q_2 \text{ e o valor mais alto do conjunto de dados.} \end{cases} \tag{3.3}$$

Assim, sendo M o mapa de íons, x e y valores arbitrários de massa/carga e tempo de retenção, respectivamente, e i a intensidade real, então a intensidade discretizada $M(x, y)$ é dada por

$$M(x,y) = \begin{cases} 0, & \text{se não existe nenhuma } \textit{feature} \text{ com } m/z = x \text{ e RT} = y \\ 1, & \text{se } i \leq Q_1 \\ 2, & \text{se } Q_1 \leq i \leq Q_2 \\ 3, & \text{se } Q_2 \leq i \leq Q_3 \\ 4, & \text{se } i \geq Q_3. \end{cases} \tag{3.4}$$

Nesse caso, o operador de redução é dado por

$$\psi(M\restriction_W) = \left\lfloor \frac{\sum_{w \in W} M(w)}{\sum_{w \in W} 1_{\{M(w)>0\}}} \right\rceil. \tag{3.5}$$

Ou seja, ele simplesmente calcula a média arredondada dos valores de $M\restriction_W$ que são maiores que 0.

3.3 Geração das árvores filoproteômicas

Para gerar os cladogramas a partir dos dados genéticos e proteômicos, foi utilizado o MrBayes (Huelsenbeck e Ronquist, 2001), na versão 3.2.7. Esse programa de inferência Bayesiana usa métodos MCMC para estimar as probabilidades *a posteriori* das árvores, a partir dos dados, parâmetros e comandos contidos em arquivos do tipo NEXUS (Maddison *et al.*, 1997). Os dados são colocados em uma matriz, na qual cada linha pode ter uma sequência de nucleotídeos ou aminoácidos, dados morfológicos ou dados binários de uma espécie. Pode-se ter, também, uma mistura desses tipos de dados em um mesmo arquivo.

As análises foram feitas da mesma maneira que em Raposo (2018), utilizando os mesmos parâmetros e comandos, gerados pelo código que fora desenvolvido para escrever os arquivos do tipo NEXUS. Cada análise foi executada com 2×10^6 gerações, amostrando a cadeia a cada 100 delas, e descartando o primeiro quarto das amostras geradas (*burn in*).

Ao fim das análises, são gerados vários arquivos NEXUS que resumem os resultados obtidos. Dentre eles, é gerado um arquivo que descreve a árvore mais provável, que pode ser visualizada com o programa FigTree.

3.3.1 Formatação dos dados de entrada

Para escrever um arquivo NEXUS a partir dos dados contidos nos mapas de íons, é necessário transformar cada um deles em uma *string*, de modo que cada linha da matriz do arquivo NEXUS seja uma dessas *strings*. A princípio, isso foi feito simplesmente concatenando as linhas dos mapas de íons, que são sequências de números inteiros. Após a construção da matriz, as colunas nas quais todos os elementos são iguais foram retiradas, removendo assim informações redundantes e diminuindo o volume dos dados, sem interferir no resultado final.

3.4 Comparação das árvores obtidas

Para verificar a proximidade entre as árvores obtidas (filoproteômicas e filogenéticas), foi usado um pacote de análise filogenética e de evolução para R chamado ape, que fornece uma implementação do teste CADM. A fim de utilizá-lo no encadeamento de processos desenvolvido em Python, foi necessário utilizar a interface rpy2, que permite executar códigos escritos em R dentro de um programa em Python.

Um script que automatiza a utilização do teste CADM já havia sido desenvolvido em Raposo (2018), mas algumas adaptações foram necessárias para incluir as árvores obtidas pelo método desenvolvido nesse trabalho.

Os resultados dos testes são expressos pelos valores das estatísticas W de Kendall e χ^2 de Friedman, já mencionadas na Seção 2.4. Como χ^2 é calculada a partir de W, ambas são equivalentes.

3.5 Particionamento dos mapas de íons

Como visto na Subseção 3.3.1, cada linha da matriz de um arquivo NEXUS foi construída pela concatenação das linhas de um mapa de íons, considerando-o como um todo. Todavia, uma abordagem potencialmente mais eficaz, inspirada no trabalho de Vaquero *et al.* (2005) sobre operadores multirresolução, é particionar todos os mapas de íons da mesma maneira, obtendo regiões menores que contém dados correlacionados entre todos eles. Como dito na Subseção 2.2.1, os valores de massa/carga e abundância relativa, bem como os valores de tempo de retenção, podem ser utilizados para identificar as proteínas e peptídeos presentes em diversas amostras, e a partir daí podem ser geradas árvores filoproteômicas relacionando seus perfis proteicos. Assim, é intuitivo pensar que regiões específicas e comuns entre todos os mapas de íons agrupam informações correlacionadas e que essa ideia pode ser utilizada para construir árvores evolutivas mais coerentes.

Sendo $M : E \longrightarrow L$ um mapa de íons, podemos definir uma **partição** (ou **particionamento**) $\mathcal{P}$ de M como um conjunto de k janelas w_i, disjuntas, de M tais que

$$E = \bigsqcup_{i=1}^{k} w_i. \tag{3.6}$$

Ou seja, o domínio de M é a união disjunta dessas janelas. Cada elemento de uma partição será chamado de **parte**.

Assim, considerando uma partição $\mathcal{P}_{M_i} = \{p_1^{M_i}, p_2^{M_i}, \dots, p_k^{M_i}\}$ para cada mapa de íons $M_i : E_i \longrightarrow L_i$, com $i \in [1, m]$, sendo m a quantidade de mapas de íons, pode-se escrever um arquivo NEXUS da seguinte maneira:

(1) Para todo mapa de íons M_i, reduzir (Seção 3.2) e transformar cada configuração $M_i\restriction_{p_j^{M_i}}$, com $j \in [1, k]$, em uma *string*, e concatená-las, obtendo assim uma única *string* de tamanho $n = |E_i|$.

(2) Construir uma matriz D com dimensões $m \times n$, em que cada linha i corresponde à *string* associada a M_i.

(3) Remover as colunas de D nas quais todos os elementos são iguais.

A matriz D construída pelos passos descritos acima contém os dados principais de um arquivo NEXUS.

É importante que o particionamento seja feito antes de diminuir a resolução dos mapas de íons, para que os dados originais sejam particionados, pois com a redução de dimensionalidade há perda de informação. Além disso, o particionamento deve ser igual para todos os mapas de íons, pois é de interesse que as regiões selecionadas sejam comparadas entre os diferentes mapas. E daí surge, naturalmente, um problema de otimização, em que deseja-se encontrar a melhor forma possível de particionar os mapas de íons, sujeito à restrição de que o particionamento deve ser o mesmo para todos os mapas. Esse problema pode ser escrito como

$$\begin{aligned} &\underset{\mathcal{P}}{\text{maximize}} \quad c(\mathcal{P}) \\ &\text{sujeito a} \quad \mathcal{P}_{M_i} = \mathcal{P} \quad \forall M_i,\ i \in [1, m]. \end{aligned} \tag{3.7}$$

A Figura 3.4 ilustra, de forma geral, um encadeamento lógico para resolver esse problema.

Nesse caso, a função custo c é o resultado do teste CADM comparando a árvore gerada a partir do arquivo NEXUS escrito utilizando uma partição $\mathcal{P}'$, com uma árvore filogenética

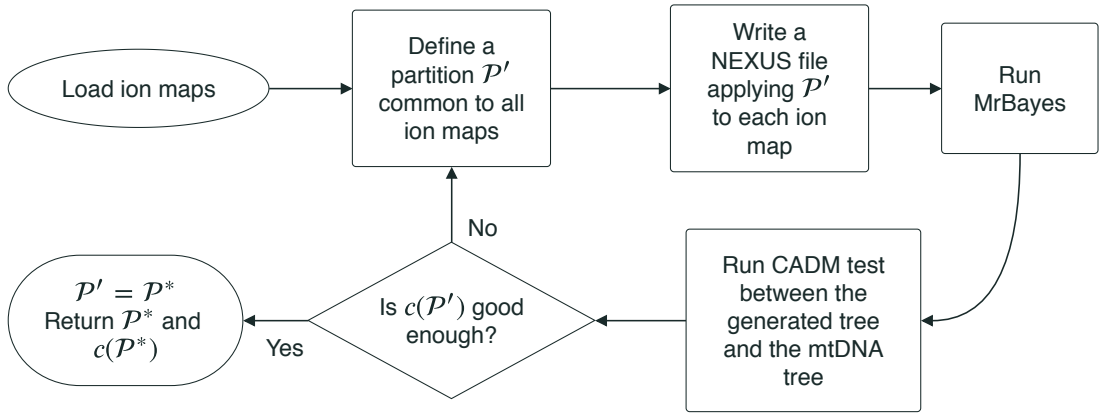


Figura 3.4: ***Ideia geral de um algoritmo para resolver o problema de otimização proposto.***

de referência. Quando for decidido que o melhor resultado encontrado for suficientemente bom, ou após um número determinado de iterações, o algoritmo devolve o particionamento ótimo $\mathcal{P}^*$ e o valor ótimo $c(\mathcal{P}^*)$.

Nos experimentos realizados, as partições foram geradas de maneira uniforme, de modo que todas as suas partes tivessem as mesmas dimensões. Isso foi feito simplesmente dividindo as linhas e as colunas dos mapas de íons por um número natural n, para todos os valores de n no intervalo $[1, 300]$. Os mapas de íons extraídos pelo SuperHirn possuíam dimensões 12000×15000000 ($RT \times m/z$), e consequentemente, as dimensões das partições variaram entre 12000×15000000 (para $n = 1$) e 40×50000 (para $n = 300$). O método de otimização só foi utilizado para mapas de íons binários, visto que os valores de intensidades discretizados por quartis aumentam consideravelmente o tempo computacional da inferência das árvores. A Figura 3.5 exibe um exemplo de mapa de íon particionado.

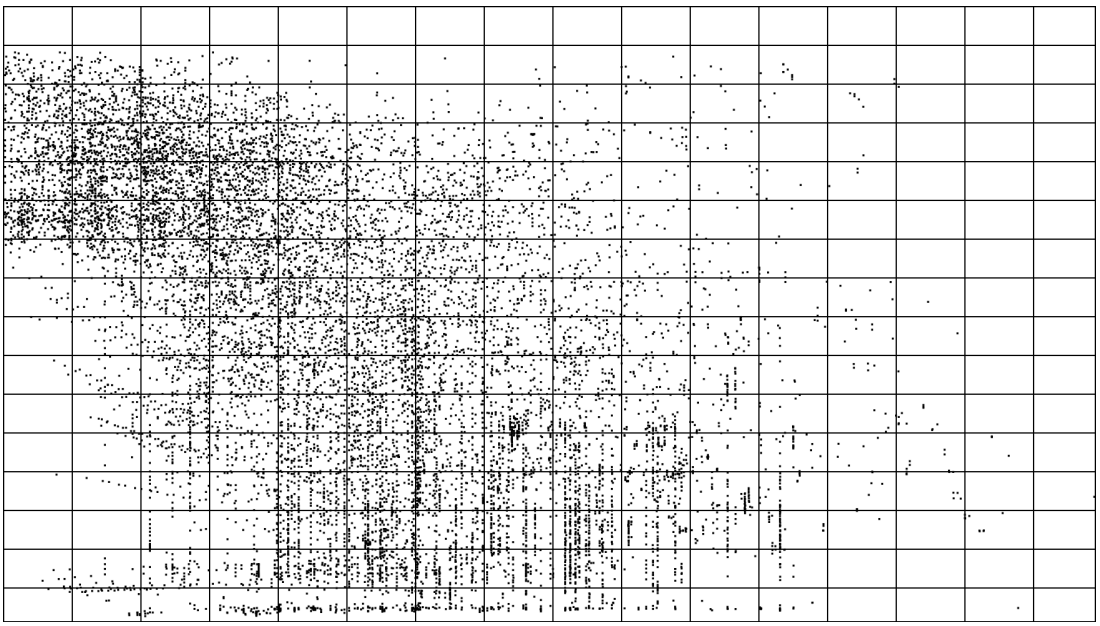


Figura 3.5: *Exemplo de visualização de um mapa de íon particionado de forma uniforme. Nesse exemplo,* $n = 16$.

Capítulo 4

Resultados

Neste capítulo serão apresentados os resultados obtidos pela aplicação do encadeamento de processos desenvolvido (detalhado no Capítulo 3) nos dados proteômicos brutos de sete espécies de serpentes do gênero *Bothrops*.

São exibidas as árvores evolutivas inferidas pelo MrBayes e os valores das estatísticas W de Kendall e χ^2 de Friedman, que constituem os resultados dos testes CADM.

Todas árvores obtidas usando a metodologia desenvolvida nesse trabalho foram comparadas com uma árvore filogenética de referência, gerada com dados dos genes mitocondriais ND4 e Cytb, que pode ser visualizada na Figura 4.1.

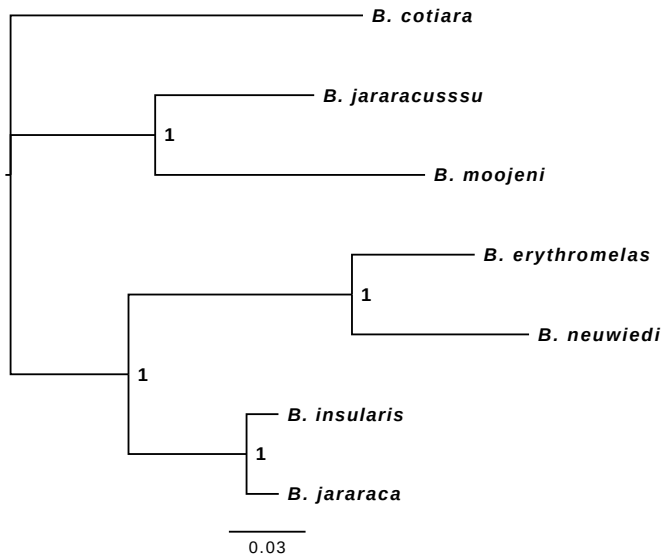
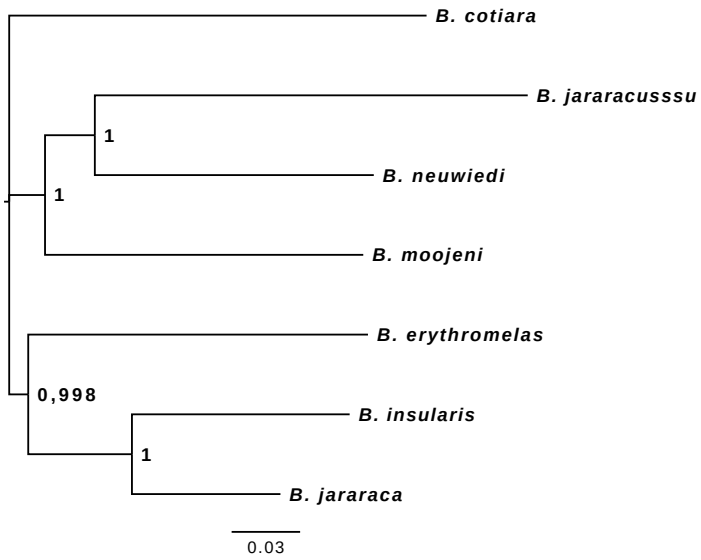


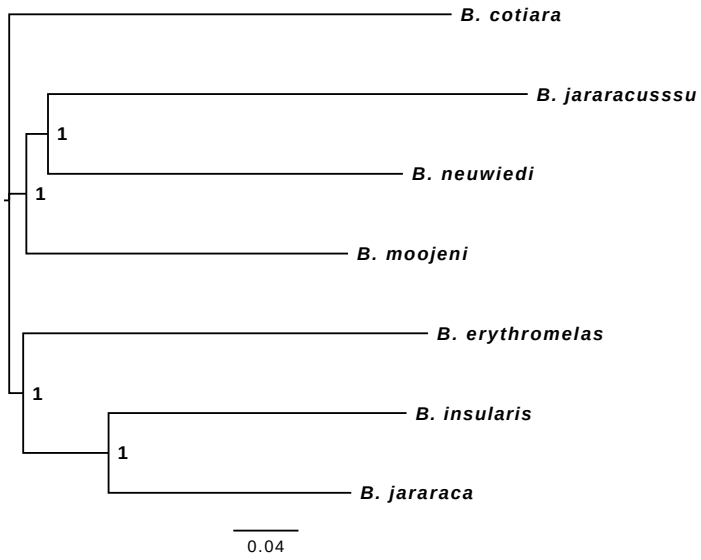
Figura 4.1: ***Árvore filogenética das sete espécies do gênero Bothrops estudadas.*** *Os números que rotulam os nós internos são as probabilidades a posteriori de cada ramificação. O segmento abaixo da árvore indica o número estimado de substituições por campo. Árvore extraída de* Raposo *(2018).*

4.1 Árvores obtidas pelo método *naive*

Primeiramente, os testes foram feitos utilizando o método mais simples desenvolvido (*naive*), que constrói os arquivos NEXUS diretamente dos mapas de íons, após a redução global de dimensionalidade. Foram geradas árvores (Figura 4.2) a partir dos mapas de íons binários e dos obtidos pela discretização por quartis apresentada na Subseção 3.2.2.



(a)



(b)

Figura 4.2: ***Árvores filoproteômicas das sete espécies do gênero Bothrops estudadas, obtidas pelo método naive.*** *A árvore da Figura 4.2a foi construída a partir da discretização binária dos valores de intensidades. Já a árvore da Figura 4.2b foi obtida para valores de intensidades no intervalo* $[0, 4]$*. Os números que rotulam os nós internos são as probabilidades a posteriori de cada ramificação. Os segmentos abaixo de cada árvore indicam o número estimado de substituições por campo.*

Note que quase não há diferença entre as duas árvores da Figura 4.2, somente uma pequena variação no comprimento dos ramos e na probabilidade *a posteriori* de um dos nós internos. Isso indica que talvez a discretização por quartis não influencie tanto na topologia das árvores.

Comparando-as com a árvore de referência, pôde-se observar uma discordância significativa nas topologias, apesar de ainda manter uma certa coerência. Essa diferença é reforçada pelos resultados obtidos ao aplicar o teste CADM, mostrados na Tabela 4.1. Como visto na Seção 2.4, o valor de W varia de 0 (não congruência) a 1 (total congruência). Sendo assim, o resultado $W = 0.4464048$ indica uma topologia não muito congruente.

Árvore	W	χ^2
Discretização binária	0.4464048	17.8561917
Discretização por quartis	0.4464048	17.8561917

Tabela 4.1: *Resultados dos testes CADM entre a árvore de referência (Figura 4.1) e as duas árvores da Figura 4.2.*

4.2 Árvores obtidas usando o método de particionamento

Nesta seção, serão mostradas algumas árvores inferidas pelo método de particionamento desenvolvido (Seção 3.5), assim como suas respectivas pontuações no teste CADM. Foram somente selecionadas as árvores em que as probabilidades *a posteriori* das ramificações se mostraram suficientemente próximas de 1.

A Figura 4.3 apresenta duas árvores geradas que não tiveram o melhor resultado no CADM. Comparando a topologia da árvore da Figura 4.3a com a da Figura 4.2a, vemos que elas são bem parecidas. Isso se dá pelo fato de que, por ter sido utilizada uma partição com um número pequeno de partes, o tamanho dessas partes ficou muito grande, o que não é suficiente para melhorar o uso das informações. Já a árvore da Figura 4.3b, que foi particionada em mais partes, ficou com uma topologia mais próxima da referência. Essa melhora faz sentido, já que o tamanho das partes ficou menor. Esses resultados são quantificados pelos valores presentes na Tabela 4.2.

Finalmente, a árvore filoproteômica da Figura 4.4, que foi gerada utilizando uma partição com partes de tamanho menor, foi a que obteve melhor pontuação no teste CADM. Comparando sua topologia com a da Figura 4.3b, pode-se observar que a única diferença foi que a espécie *B. erythromelas* saiu da subárvore na qual se encontram as espécies *B. jararacussu*, *B. moojeni* e *B. neuwiedi*. Comparando-a com a árvore de referência, não

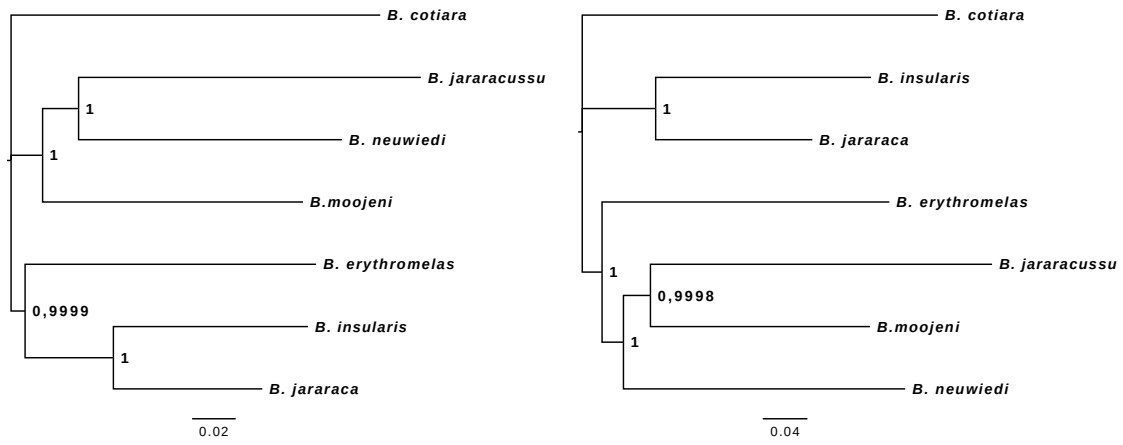


Figura 4.3: ***Árvores filoproteômicas das sete espécies do gênero Bothrops estudadas, obtidas pelo método de particionamento.*** *A árvore da Figura 4.3a foi a melhor obtida com n no intervalo* $[23, 33]$*, enquanto que a da Figura 4.3b foi a melhor com n no intervalo* $[111, 121]$*. Os números que rotulam os nós internos são as probabilidades a posteriori de cada ramificação. Os segmentos abaixo de cada árvore indicam o número estimado de substituições por campo.*

é muito intuitivo pensar que a topologia melhorou. Mas segundo o coeficiente W de Kendall obtido, mostrado na Tabela 4.2, pode-se afirmar que essa foi a árvore com melhor pontuação.

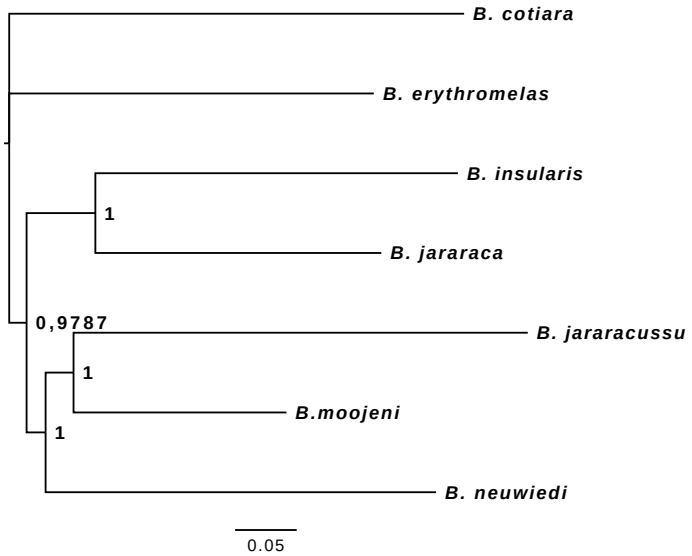


Figura 4.4: ***Melhor árvore filoproteômica obtida das sete espécies do gênero Bothrops estudadas.*** *Melhor árvore obtida com n no intervalo* $[271, 280]$ *e no geral. Os números que rotulam os nós internos são as probabilidades a posteriori de cada ramificação. O segmento abaixo da árvore indica o número estimado de substituições por campo.*

Árvore	W	χ^2
Partição com n em $[23, 33]$	0.4464048	17.8561917
Partição com n em $[111, 121]$	0.554261	22.170439
Partição com n em $[271, 280]$	0.6080151	24.3206054

Tabela 4.2: *Resultados dos testes CADM entre a árvore de referência (Figura 4.1) e as três árvores inferidas pelo método de particionamento.*

Capítulo 5

Conclusão

Neste trabalho, desenvolvemos um método para a análise de dados proteômicos brutos provenientes de experimentos de LC-MS, sem depender da identificação de proteínas e peptídeos, seja com o auxílio de bancos de dados, ou utilizando métodos como o sequenciamento *de novo*. Para tal, foi utilizada a ideia de particionamento dos dados de matrizes esparsas e com distribuição não-homogênea das informações, inspirada no uso de operadores multirresolução no contexto de processamento de imagens.

Essa metodologia foi aplicada a dados de venenos de serpentes de sete diferentes espécies do gênero *Bothrops*, e mostrou resultados significativos, que mais uma vez, confirmaram a relação entre o perfil proteômico dos venenos com a classificação filogenética das espécies estudadas. Todavia, como já visto anteriormente, houveram divergências quanto a classificação topológica das espécies *B. erythromelas* e *B. neuwiedi*, que não se mostrou congruente às árvores filogenéticas de referência, obtidas com o uso de dados de DNA mitocondrial.

Acreditamos que o método desenvolvido ainda tenha muito potencial para melhorias. Em uma eventual continuidade dessa linha de pesquisa, pode-se citar como ideia relevante, a aplicação de heurísticas na exploração do espaço de busca do problema de otimização apresentado, tais como algoritmos genéticos e *simulated annealing*. Técnicas como essa permitiriam que particionamentos subótimos dos dados brutos, comuns para todas as espécies, fossem encontrados de forma computacionalmente viável.

Outra possibilidade seria estender a estratégia utilizada neste trabalho para outros tipos de dados de venenos de serpentes, como as estruturas de N-glicanos de espécies do gênero *Bothrops* vistas em estudos anteriores.

Apêndice A

Configuração de *software* e equipamentos

Este apêndice relaciona as configurações utilizadas tanto no *software* de aquisição e processamento dos dados brutos (SuperHirn), quanto no espectrômetro de massa que gerou os arquivos RAW utilizados (Thermo Fisher Scientific LTQ Orbitrap Velos MS).

Para configurar corretamente os parâmetros do SuperHirn, foi preciso extrair as configurações do LTQ Orbitrap Velos. Para tal, o *software* Xcalibur, da Thermo Fisher Scientific, foi empregado. A Tabela A.1 lista os parâmetros de configuração mais relevantes que foram obtidos.

Parâmetro	**Valor usado**
Faixa de massa/carga	300 – 1800 m/z
Resolução (a 400 m/z)	30000
Massa perdida (*lock mass*)	445.120025
Modo de aquisição dos dados	profile
Janela de isolamento	3 m/z
Tempo de ativação	30 ms
Energia de colisão normalizada	35%
Tempo de exclusão dinâmica	90 s
Tamanho da lista de exclusão dinâmica	500
Duração	120 min

Tabela A.1: ***Alguns parâmetros de configuração do LTQ Orbitrap Velos.*** *Somente foram listados os parâmetros mais relevantes.*

Houve, então, a tentativa de relacionar os parâmetros do SuperHirn com os do es-

pectrômetro de massa. Alguns, como os intervalos de massa/carga e tempo de retenção estavam bem óbvios, mas para a grande maioria não foi possível estabelecer uma paridade com total certeza. Ao pesquisar por trabalhos que utilizaram o SuperHirn e o LTQ Orbitrap Velos, foram encontrados dois (Vasilj *et al.*, 2012; Resende, 2013) que configuraram um subconjunto dos parâmetros com os mesmos valores. Tais configurações também foram usadas neste trabalho. A lista dos parâmetros do SuperHirn que tiveram seus valores alterados é exibida na Tabela A.2.

Parâmetro	Valor usado	Valor padrão
MS1 retention time tolerance	2.5	1.0
MS1 m/z tolerance	6	10
RT and elution window	120	180
MS1 feature signal to noise threshold	3.0	0.5
MS1 feature intensity cutoff	200	5000
MS1 feature CHRG range min	2	1
MS1 feature CHRG range max	9	5
MS1 feature mz range min	300	0
MS1 feature mz range max	1800	2000
FT peak detect MS1 m/z tolerance	6	10
FT peak detect MS1 intensity min threshold	200	1000
Relative isotope mass precision	6	10
Activation of MS1 feature merging post processing	0	1

Tabela A.2: ***Alguns parâmetros de configuração do SuperHirn.*** *Aqui só estão listados os parâmetros que tiveram o valor alterado no arquivo de configuração. Os que não estão listados permaneceram com os valores padrão.*

Apêndice B

Modificações feitas no SuperHirn

Como já dito na Seção 3.1, foi necessário implementar novas funcionalidades no programa SuperHirn, utilizado para fazer a extração dos dados brutos contidos nos arquivos gerados pelos experimentos de LC-MS.

Foi feito um *fork* do repositório oficial do SuperHirn, onde foram desenvolvidas as novas funcionalidades. Este projeto encontra-se sob Licensa Apache, versão 2.0, e pode ser acessado pelo link abaixo:

https://github.com/mergipe/SuperHirn.

Nas duas seções que se seguem, as modificações realizadas e as motivações para implementá-las serão explicadas.

B.1 Extração de corridas alinhadas

O SuperHirn realiza um alinhamento entre os tempos de retenção das corridas em análise para fundí-las em um único arquivo, chamado de *Master Map*. Neste trabalho, foram construídos um *MasterMap* para cada uma das sete espécies de serpentes, usando quatorze corridas obtidas de ensaios de LC-MS.

Porém, para executar a metodologia desenvolvida neste trabalho, o alinhamento entre os sete *MasterMaps* é essencial, visto que as flutuações nos valores de tempo de retenção podem influenciar negativamente nos resultados. O problema é que o alinhamento de corridas e a construção do *MasterMap* estão acoplados na versão original do SuperHirn, não sendo possível atualizar os arquivos XML das corridas após o alinhamento sem que as

corridas sejam fundidas.

Logo, foi criado um novo módulo do programa que, usando as funções base de alinhamento já implementadas na versão original, realiza o alinhamento entre diversas corridas, tomando uma delas, escolhida pelo usuário, para servir de referência. Para executar esse módulo, basta rodar o SuperHirn da seguinte maneira:

```
$ ./SuperHirnv03 -AR <FILENAME>
```

em que `<FILENAME>` indica o nome do arquivo XML com a corrida de referência. Para mais informações, veja o manual do SuperHirn.

B.2 Inclusão dos intervalos de RT e m/z nos arquivos XML

Como visto na Subseção 3.1.1, os mapas de íons foram representados na memória por matrizes esparsas (implementadas pelo pacote `scipy.sparse`). Para facilitar essa tarefa, viu-se necessário que todas as matrizes fossem criadas com as dimensões exatas, de acordo com os intervalos de tempo de retenção e massa/carga das respectivas corridas LC-MS. Entretanto, esses dados não são incluídos nos arquivos XML construídos pelo SuperHirn.

Para resolver este problema, foram incluídas algumas linhas simples de código na classe que cuida da escrita dos arquivos XML. Assim, é possível obter esses valores automaticamente com a leitura dos arquivos XML, sem que o usuário precise inseri-los manualmente como entrada do programa.

Referências

[Altekar *et al.* 2004] Gautam Altekar, Sandhya Dwarkadas, John P Huelsenbeck e Fredrik Ronquist. "Parallel metropolis coupled markov chain monte carlo for bayesian phylogenetic inference". Em: *Bioinformatics* 20.3 (2004), pgs. 407–415 (citado na pg. 13).

[Aebersold e Mann 2003] Ruedi Aebersold e Matthias Mann. "Mass spectrometry-based proteomics". Em: *Nature* 422.6928 (2003), pg. 198 (citado na pg. 9).

[Andrade-Silva, Zelanis *et al.* 2016] Débora Andrade-Silva, André Zelanis *et al.* "Proteomic and glycoproteomic profilings reveal that post-translational modifications of toxins contribute to venom phenotype in snakes". Em: *Journal of proteome research* 15.8 (2016), pgs. 2658–2675 (citado nas pgs. 1, 6, 7).

[Andrade-Silva, Ashline *et al.* 2018] Débora Andrade-Silva, David Ashline *et al.* "Structures of N-Glycans of *Bothrops* venoms revealed as molecular signatures that contribute to venom phenotype in viperid snakes". Em: *Molecular and Cellular Proteomics* (2018). In revision (citado nas pgs. 1, 7).

[Brusniak *et al.* 2008] Mi-Youn Brusniak *et al.* "Corra: computational framework and tools for lc-ms discovery and targeted mass spectrometry-based proteomics". Em: *BMC bioinformatics* 9.1 (2008), pg. 542 (citado na pg. 20).

[Colinge e Bennett 2007] Jacques Colinge e Keiryn L Bennett. "Introduction to computational proteomics". Em: *PLoS computational biology* 3.7 (2007), e114 (citado na pg. 9).

[Chiaradia *et al.* 2008] Mariza C Chiaradia, Carol H Collins, Isabel CSF Jardim *et al.* "O estado da arte da cromatografia associada à espectrometria de massas acoplada à espectrometria de massas na análise de compostos tóxicos em alimentos". Em: *Química nova* (2008) (citado na pg. 9).

[Fenwick *et al.* 2009] Allyson M Fenwick, Ronald L Gutberlet, Jennafer A Evans e Christopher L Parkinson. "Morphological and molecular evidence for phylogeny and classification of South American pitvipers, genera *Bothrops*, *Bothriopsis*, and *Bothrocophias* (Serpentes: Viperidae)". Em: *Zoological Journal of the Linnean Society* 156.3 (2009), pgs. 617–640 (citado na pg. 6).

[Fox e Serrano 2008] Jay W Fox e Solange MT Serrano. "Exploring snake venom proteomes: multifaceted analyses for complex toxin mixtures". Em: *Proteomics* 8.4 (2008), pgs. 909–920 (citado na pg. 2).

[Hassanin *et al.* 2013] Alexandre Hassanin, Junghwa An, Anne Ropiquet, Trung Thanh Nguyen e Arnaud Couloux. "Combining multiple autosomal introns for studying shallow phylogeny and taxonomy of laurasiatherian mammals: application to the tribe bovini (cetartiodactyla, bovidae)". Em: *Molecular Phylogenetics and Evolution* 66.3 (2013), pgs. 766–775 (citado na pg. 6).

[Huelsenbeck e Ronquist 2001] John P Huelsenbeck e Fredrik Ronquist. "MRBAYES: Bayesian inference of phylogenetic trees". Em: *Bioinformatics* 17.8 (2001), pgs. 754–755 (citado nas pgs. 13, 24).

[Legendre e Lapointe 2004] Pierre Legendre e François-Joseph Lapointe. "Assessing congruenceamong distance matrices: single-malt scotch whiskies revisited". Em: *Australian & New Zealand Journal of Statistics* 46.4 (2004), pgs. 615–629 (citado na pg. 15).

[Lee *et al.* 2015] Jaewook Lee, Woosuk Sung e Joo-Ho Choi. "Metamodel for efficient estimation of capacity-fade uncertainty in li-ion batteries for electric vehicles". Em: *Energies* 8.6 (2015), pgs. 5538–5554 (citado na pg. 14).

[Maddison *et al.* 1997] David R Maddison, David L Swofford e Wayne P Maddison. "Nexus: an extensible file format for systematic information". Em: *Systematic biology* 46.4 (1997), pgs. 590–621 (citado na pg. 24).

[Mueller *et al.* 2007] Lukas N Mueller *et al.* "Superhirn–a novel tool for high resolution lc-ms-based peptide/protein profiling". Em: *Proteomics* 7.19 (2007), pgs. 3470–3480 (citado nas pgs. 19, 20).

[Pedrioli *et al.* 2004] Patrick GA Pedrioli *et al.* "A common open representation of mass spectrometry data and its application to proteomics research". Em: *Nature biotechnology* 22.11 (2004), pg. 1459 (citado na pg. 18).

[Raposo 2018] Victor Wichmann Raposo. *Análise filogenética computacional de serpentes do gênero* Bothrops *a partir de proteomas de venenos*. Rel. técn. Monografia de graduação em Computação. Instituto de Matemática e Estatística, Universidade de São Paulo, 2018 (citado nas pgs. 1, 7, 17, 24, 25, 29).

[Resende 2013] Virginia Maria Ferreira Resende. "Análise proteônica de venenos de Apis mellifera baseada em espectrometria de massas: abordagem quantitativa label-free e identificação de fosforilação". Tese de dout. Universidade de São Paulo, 2013 (citado na pg. 38).

[Ronquist e Huelsenbeck 2003] Fredrik Ronquist e John P Huelsenbeck. "Mrbayes 3: bayesian phylogenetic inference under mixed models". Em: *Bioinformatics* 19.12 (2003), pgs. 1572–1574 (citado na pg. 13).

[Silva *et al.* 2005] Jeffrey C Silva *et al.* "Quantitative proteomic analysis by accurate mass retention time pairs". Em: *Analytical chemistry* 77.7 (2005), pgs. 2187–2200 (citado na pg. 19).

[Vasilj *et al.* 2012] Andrej Vasilj, Marc Gentzel, Elke Ueberham, Rolf Gebhardt e Andrej Shevchenko. "Tissue proteomics by one-dimensional gel electrophoresis combined with label-free protein quantification". Em: *Journal of proteome research* 11.7 (2012), pgs. 3680–3689 (citado na pg. 38).

[Vaquero *et al.* 2005] Daniel André Vaquero, Junior Barrera e R Hirata. "A maximum-likelihood approach for multiresolution w-operator design". Em: *XVIII Brazilian Symposium on Computer Graphics and Image Processing (SIBGRAPI'05)*. IEEE. 2005, pgs. 71–78 (citado nas pgs. 2, 15, 25).

[Yang *et al.* 2002] Yee Hwa Yang *et al.* "Normalization for cdna microarray data: a robust composite method addressing single and multiple slide systematic variation". Em: *Nucleic acids research* 30.4 (2002), e15–e15 (citado na pg. 19).