University of São Paulo
Institute of Mathematics and Statistics
Bachelor of Computer Science

# An approach based on hierarchical clustering for phyloproteomic analysis of snake venoms measured by mass spectrometry

Guilherme Costa Veira

Undergraduate Thesis

mac 499 — Capstone Project

Program: Computer Science
Advisor: Marcelo da Silva Reis

São Paulo
January 30, 2021

# Acknowledgment

First of all, I want to thank my parents and the rest of my family, who always gave me the love and support I needed. This accomplishment would not have been possible without them. I would also like to thank my friends who motivated me during this journey. Finally, I express my gratitude to my advisor, Marcelo Silva Reis, for all his assistance and dedicated involvement.

# Abstract

Snakes use their venom for both self-defense and killing prey. These substances are complex protein mixtures, usually studied using mass spectrometry (MS)-based proteomics. In recent studies, evidence has been shown that the proteomic profiles of snakes of the genus *Bothrops* correlate with the phylogenetic tree of these same organisms. However, the overrepresentation of some species in databases used for protein identification after the MS experiments introduced bias to these results. In order to mitigate this problem, a previous work proposed the usage of MS raw data, represented as retention time by mass-to-charge matrices, for the construction of phyloproteomic trees using Bayesian inference. However, the Bayesian inference of phyloproteomic trees is a computationally intense method that limits raw data partitioning exploitation. This project developed a methodology that uses the same MS raw data to generate phyloproteomic trees using hierarchical clustering, a less computationally intensive technique. To this end, the raw data is uniformly partitioned and mapped to one-dimensional vectors that were later used for clustering. The resulting dendrograms are validated with statistical bootstrapping. We show that the hierarchical clustering was able to yield similar results to previous works and that the use of raw MS data is a viable technique for phyloproteomic analysis.

**Keywords:** Phyloproteomic Analysis, Phylogeny, Snake Venoms, *Bothrops*, Mass Spectrometry, Hierarchical Clustering, Bootstrapping.

# Resumo

Serpentes utilizam seus venenos tanto para autodefesa quanto para obtenção de presas. Essas substâncias tratam-se na verdade de complexas misturas proteicas, que normalmente são estudadas através de técnicas como a proteômica baseada em espectrometria de massas (EM). Em trabalhos recentes, foram mostrados indícios de que os perfis proteômicos de serpentes do gênero *Bothrops* se correlacionam com a árvore filogenética desses mesmos organismos. Todavia, a superrepresentação de algumas espécies no banco de dados utilizado para identificação de proteínas após o ensaio de EM introduziu viés nesses resultados. Para mitigar isso, foi proposto em outro trabalho o uso de dados brutos de EM para construção de árvores filoproteômicas por inferência Bayesiana, utilizando como base matrizes de tempo de eluição por massa/carga. Todavia, a inferência Bayesiana de árvores filoproteômicas é um método computacionalmente intenso que limita a exploração de particionamento dos dados brutos. Neste projeto, desenvolvemos uma metodologia que utiliza os mesmos dados brutos de EM para gerar árvores filoproteômicas utilizando agrupamento hierárquico, uma técnica menos custosa computacionalmente. Para este fim, os dados brutos são particionados uniformemente e mapeados para vetores unidimensionais que posteriormente foram utilizados para construir hierarquias de grupos. Os dendrogramas gerados são validados com bootstrapping estatístico. Mostramos que o agrupamento hierárquico foi capaz de produzir resultados similares aos de trabalhos anteriores e que o uso de dados brutos de EM é uma técnica viável para análise filoproteômica.

**Palavras-chave:** Análise Filoproteômica, Filogenia, Venenos de Serpentes, *Bothrops*, Espectrometria de Massas, Agrupamento Hierárquico, Bootstrapping.

# Contents

# Appendices

# Chapter 1

# Introduction

Snake venoms are a complex mixture of peptides and proteins that play a fundamental role in the survival of venomous snakes. Their venoms are used both to immobilize and kill prey and predators. Protein sets, also called proteomes, are studied in the field of knowledge called proteomics. Mass spectrometry (MS) is a relevant analytical tool used in proteomics since it allows the identification of venom compositions, enabling the study of how the venom compounds affect poisoned organisms. MS fragments and ionizes samples by applying chemical and physical processes and subsequently measuring the mass-to-charge ratio of ions. Proteins can then be identified through database queries that associate an ion detection pattern to a protein. In recent studies, MS has been used to show evidence that the venom proteomic profile of species of the *Bothrops* genus correlates with the phylogenetic classification obtained through mitochondrial DNA (mtDNA) combined with morphological characters (Andrade-Silva, Zelanis, *et al.*, 2016, Raposo, 2018); a similar result, though with a smaller correlation, was also verified when compared venom glycan profiles of the same species against their phylogenetic classification (Andrade-Silva, Ashline, *et al.*, 2018).

However, Andrade-Silva, Zelanis, *et al.*, 2016, used techniques that limited an in-depth investigation of the experiments generated in these studies. For instance, hierarchical clustering was applied over proteins identified by MS with the aid of a database to generate cladograms. Nevertheless, some snake venoms are underrepresented in protein databases, which inevitably biased the results to some degree. To mitigate this problem, Raposo, 2018, applied *de novo* peptide sequencing to avoid the usage of a protein database. Bayesian inference of phyloproteomic trees was used instead of hierarchical clustering. Despite showing promising results, this methodology was limited by the number of false positive peptides candidates.

Maciel, 2019, initiated a novel alternative methodology that uses a partitioning of MS raw data to infer phyloproteomic trees using Bayesian inference without identifying proteins or peptides. This approach eliminates potential biases caused by molecular identification steps. Despite showing consistent results with previous works, Bayesian inference is a computationally expensive technique, which has limited the exploration of new data partitioning techniques that might yield better results. Therefore, it is an open problem the development of computationally cheaper approaches for this problem, which

would allow researchers to perform phyloproteomic analyses by comparing MS raw data and protein database identification phyloproteomic trees using the same tree generation technique.

## 1.1 Objectives

This thesis aims to develop a methodology to generate phyloproteomic trees from mass spectrometry raw data using hierarchical clustering, an approach that is both easy to interpret and computationally cheaper.

Specific goals of this work are the application of the developed methodology into seven venoms from species of the *Bothrops* genus measured by mass spectrometry, and also the validation of the resulting phyloproteomic trees with statistical bootstrapping.

## 1.2 Structure Overview

The remainder of this thesis is organized in the following manner:

- **Chapter 2** (Literature Review) provides the theoretical background required to understand this work. Key biological concepts, mass spectrometry-based proteomics, hierarchical clustering, and statistical bootstrapping applied to cluster validation are presented;
- **Chapter 3** (Materials and Methods) is a detailed description of all steps required to reproduce the pipeline proposed, from venom extraction until the generation of phyloproteomic trees. ;
- **Chapter 4** (Results) summarizes essential results obtained in a descriptive manner;
- **Chapter 5** (Discussion) discuss the findings of this work from both a technical and biological perspective;
- **Chapter 6** (Conclusion) sums up the content of this work, exposing important contributions and the next challenges of this line of research.
- **Appendix A** lists parameters used in the software used for processing LC-MS raw data.
- **Appendix B** presents a brief discussion of software improvements implemented in this work from a software engineering perspective.

# Chapter 2

# Literature Review

In this chapter, we review the main concepts used throughout this thesis. In Section 2.1, we present some biological concepts such as evolutionary trees, Central Dogma, mitochondrial DNA, and snake venoms. In Section 2.2, we review some properties of the mass spectrometry tecnique. Finally, in Sections 2.3 and 2.4, we present some fundamentals on, respectively, hierarchical clustering and its validation (bootstrapping).

## 2.1 Biological Concepts

The biological concepts presented in Sections 2.1.1–2.1.3 are based in Page and E. Holmes, 2009.

### 2.1.1 Evolutionary Trees

There is evidence that all current life on Earth has a common descent. The investigation of processes that gave rise to the great variety of life on Earth today is the main interest of evolutionary biology, a biology subfield. One of its goals is to reconstruct the "tree of life", a model that expresses how all life is related by common ancestry.

A tree is a mathematical structure typically chosen to model the history of evolution. It is a set of nodes connected by edges where terminal nodes, or leaves, represent species that can be either existing or extinct. Internal nodes may represent hypothetical ancestors of their child nodes. Consequently, the root node represents the hypothetical ancestor of all organisms in the tree.

There are various types of treelike evolutionary diagrams with different interpretations associated with them. Unfortunately, literature refers to these diagrams by different names. The evolutionary tree terminology used in this text is defined below:

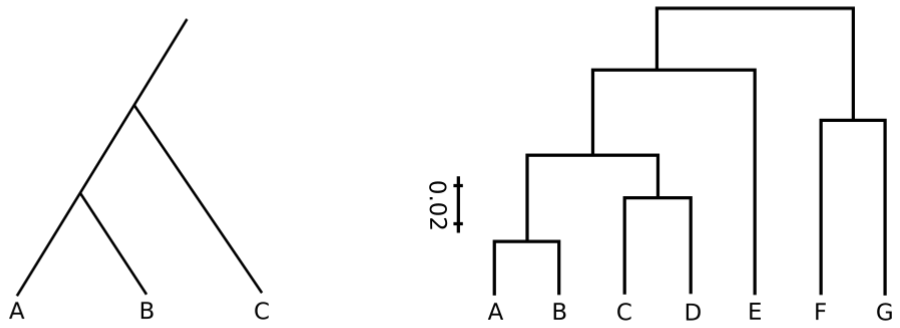
**Dendrogram:** a generic diagram representing a tree.

**Cladogram:** simple branching diagram where each branch indicates a shared common ancestry between terminal nodes. For example, Figure 2.1(a) shows that species A and B share a more recent common ancestor than they do with C.

**Phylogram:** this diagram contains more information than cladograms. The branch lengths are associated with some metric that represents some evolutionary change metric. Figure 2.1(b) is an example of a phylogram.

A node that branches in more than two edges is a polytomy. There are two possible interpretations. One of them, hard polytomy, is a simultaneous divergence between all descents. The other, soft polytomy, happens when there is uncertainty about speciation order due to a lack of evidence available. Since hard polytomies are unlikely to occur, this text treats all polytomies as soft polytomies.

A phylogenetic tree is an evolutionary tree inferred from mitochondrial DNA and morphological characters. Similarly, a phyloproteomic tree is a dendrogram obtained from proteomic analysis.




**(a)** **(b)**

**Figure 2.1:** ***Evolutionary trees****. (a) A cladogram example containing three organisms. (b) A phylogram containing seven organisms. The branch lengths are proportional to the amount of change undergone by the organisms over time.*

### 2.1.2 Proteins, DNA, RNA and the Central Dogma of Molecular Biology

Proteins are long-chain molecules made of bonded amino acid strings. They play an essential role in the biological world, performing a wide variety of functions in living things. For instance, proteins can act as biological catalysts for chemical reactions, identify and neutralize infectious agents, and compose multiple body structures. Snake venoms are composed mostly of proteins and peptides (strings with up to 50 amino acids).

The production of proteins, also known as protein synthesis, happens within the cells of living beings. A molecule named deoxyribonucleic acid (DNA) encodes the instructions for assembling proteins. Offspring inherits DNA from the parent generation. The DNA molecule consists of two long chains of nucleotides organized in a double-stranded helix form. An encoded instruction in the DNA for building a particular protein is known as a gene.

The process of protein synthesis occurs in two steps:

**Transcription:** A gene encoded in the DNA is copied into a molecule called messenger ribonucleic acid (mRNA). This molecule acts as a messenger, carrying genetic information from the DNA to the cell structure responsible for assembling proteins, the ribosome.

**Translation:** the process by which the instructions in the mRNA molecule are decoded and turned into proteins. The ribosome reads the sequence encoded in the mRNA and turns each nucleotide triplet (i.e., codon) into an amino acid. As a result, a protein is formed.

This whole process is also known as Central Dogma of Molecular Biology. The relationship between Central Dogma and snake venom production is depicted in Figure 2.2.

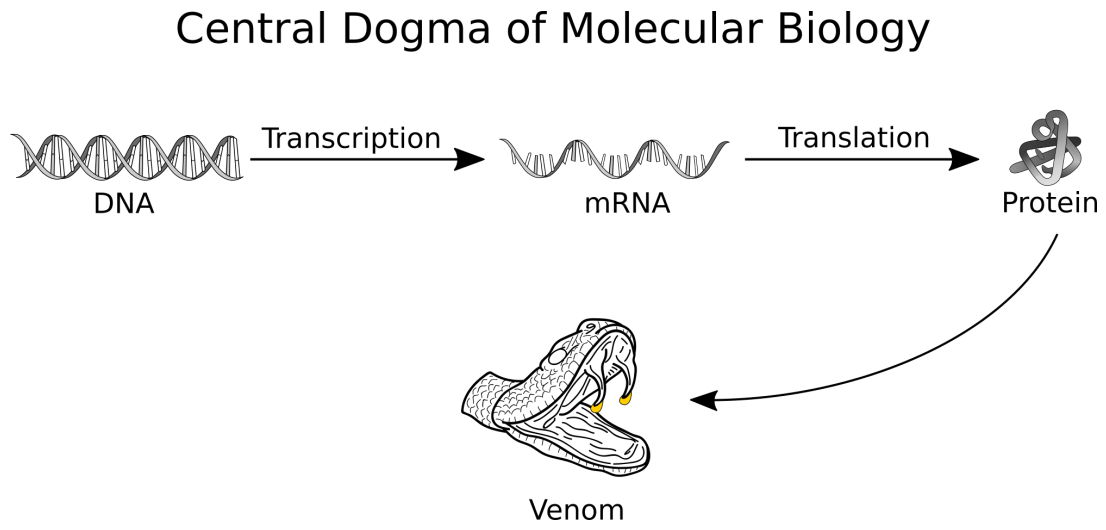



**Figure 2.2:** ***Illustration of the central dogma of molecular biology applied to snake venom protein synthesis****: DNA, mRNA and protein figures, by Philippe Hupé, are licensed under CC BY-SA 3.0. Snake image was designed by Vecteezy.com.*

### 2.1.3 Mitochondrial DNA

Cells usually are classified into two groups: eukaryotic and prokaryotic. The main distinction between these two groups is the absence of membrane-bound specialized subunits in prokaryotic cells. These subunits are known as organelles.

In a typical eukaryotic cell, one can find organelles called mitochondria. These structures are known to be primarily responsible for cellular respiration, a process in which glucose is broken and converted into adenosine triphosphate (ATP), the molecule that provides energy to the cell.

Although most of the cell's genetic material is found in an organelle called nucleus, a mitochondrion has its own DNA, referred to as mitochondrial (mtDNA). In plants and animals, mtDNA does not appear to be subjected to recombination during reproduction since mtDNA is inherited directly from the female parent, except for a few extraordinary cases in some species. Also, there is evidence that most mtDNA regions evolve much faster than nuclear DNA, which tends to be very informative in evolutionary research of similar

organisms. All in all, the lack of recombination due to maternal inheritance and the rapidly evolving genetic material make mtDNA analysis a vital tool in phylogenetic studies.

Cytochrome b (Cyt b) and NADH dehydrogenase subunit 4 (ND4) are examples of gene sequences found in mtDNA that are widely used as genetic markers due to their variability. Fenwick *et al.*, 2009 combined Cyt b and ND4 with morphological characters to construct phylogenetic trees of snake species of the *Bothrops* genus (Figure 2.3).

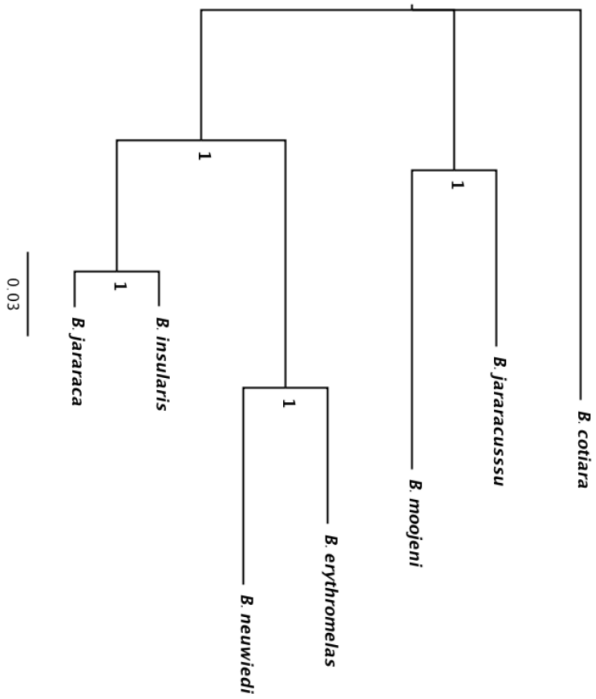



**Figure 2.3:** ***Phylogram of seven Bothrops species:*** *This phylogenetic tree was obtained through Bayesian inference over Cyt b and ND4 data. The numbers over the nodes are posterior probabilities of each branch. The segment next to the tree indicates the number of substitutions per site. Figure extracted from Raposo, 2018.*

### 2.1.4 Snake Venoms

Snake venoms are composed of a complex peptide and protein mixture. They play a fundamental role in the survival of venomous species since it is vital for hunting and self-defense. *Bothrops* snakes venom proteome contains more than a hundred different proteins that damage physiological functions of prey. Previous works (Andrade-Silva, Zelanis, *et al.*, 2016, Andrade-Silva, Ashline, *et al.*, 2018, Raposo, 2018, Maciel, 2019) presented evidence that *Bothrops* phyloproteomic trees generated from venom proteome correlate with phylogenetic trees.

## 2.2 Mass Spectrometry

Mass spectrometry (MS) is a sensitive analytical technique for measuring the mass-to-charge ratio ($m/z$) of ionized molecules in a particular sample. These measurements can be used for several purposes, such as identifying an unknown substance, quantifying known compounds, and investigating molecular structures. This section was based on the articles of Aebersold and Mann, 2003 and Colinge and Bennett, 2007. Refer to them for detailed information.

A mass spectrometer, the device used to perform an MS procedure, will produce a mass spectrum, a plot of relative ion intensity detection (or relative abundance) versus mass-to-charge ratio. There is a wide variety of different mass spectrometer designs. Choosing a mass spectrometer to run an analysis normally depends on the application, cost, and what kind of information one wishes to obtain. Nevertheless, all mass spectrometers work based on the same principles. According to the Lorentz force law (Equation 2.1), an electromagnetic force $\boldsymbol{F}$ applied on an object is dependent on its ionic charge $q$, where $\boldsymbol{E}$ is an electric field and $\boldsymbol{v} \times \boldsymbol{B}$ is the vector cross product between the magnetic field and the ion's velocity:

$$\boldsymbol{F} = q(\boldsymbol{E} + \boldsymbol{v} \times \boldsymbol{B}). \tag{2.1}$$

Additionally, from Newton's second law of motion (equation 2.2), it is implied that the net force $\boldsymbol{F}$ applied to a body with acceleration $\boldsymbol{a}$ is dependent on its mass $m$:

$$\boldsymbol{F} = m\boldsymbol{a}. \tag{2.2}$$

Thus, by combining Equations 2.1 and 2.2, one can conclude that it is possible to measure $m/z$ by applying an electric or magnetic field (or both) on an accelerating ion:

$$\frac{m}{q} = \frac{\boldsymbol{E} + \boldsymbol{v} \times \boldsymbol{B}}{\boldsymbol{a}}. \tag{2.3}$$

Consequently, mass spectrometers are generally composed of the same three basic components: an ion source, a mass analyzer, and an ion detector. In Figure 2.4, we present a general mass spectrometer model.

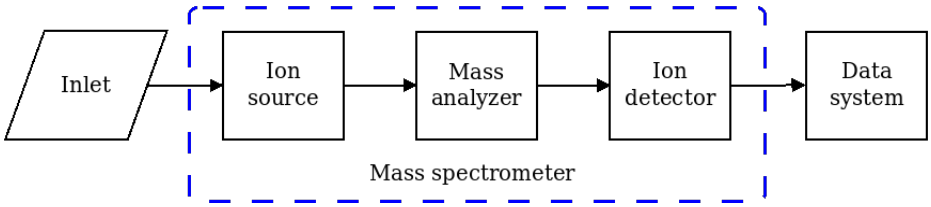



**Figure 2.4:** ***General mass spectrometer model.***

The mass spectrometer model works as the following: Initially, the injected sample is ionized by an ion source. A simple example of an ion source is electron ionization (EI), in which the sample is subjected to an electron beam that will eventually produce electrically charged compounds (i.e., ions). These ions are accelerated and travel to a mass analyzer, where they are separated according to their $m/z$. There are several different types of mass

analyzers, an example being a magnetic sector mass analyzer where separation is achieved by exposing ions to a magnetic field produced by an external magnet, causing a different deflection angle on ions with different *m/z* values. Finally, ions reach the detector where they are counted, and their *m/z* values are registered. Figure 2.5 illustrates a schematic view of a magnetic sector mass spectrometer.

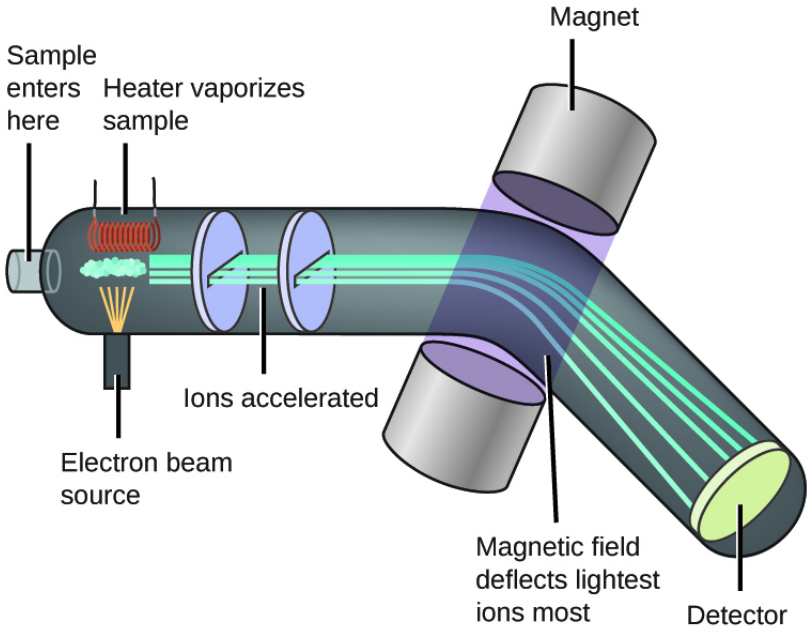



**Figure 2.5:** ***A schematic view of a magnetic sector mass spectrometer performing an analysis:*** *Figure by* Flowers et al., 2015 *is licensed under* CC BY 4.0.

Mass analyzers are a crucial component in MS as they directly impact critical parameters used to measure a mass spectrometer performance such as resolution, precision, mass accuracy, and abundance sensitivity. As mentioned earlier, there are different types of mass analyzers where each of them has its own advantages and disadvantages. Considering this, it is possible to couple two or more mass analyzers in tandem and benefit from their respective strengths, therefore producing a thorough analysis. This MS technique is known as tandem mass spectrometry (MS/MS or $MS^2$), and it consists of two stages. The first stage (MS1) selectively isolates ionized molecules by their *m/z*. Then the selected ions are fragmented in a process called collision-induced dissociation (CID). In the second stage (MS2), the *m/z* values of the fragments are calculated, making it possible to study a large molecule's composition in a complex mixture.

### 2.2.1 Mass Spectrometry-Based Proteomics

Proteomics is the large-scale study of proteomes in a biological system, focusing on protein identification, quantification, expression, interaction, and dynamics. The proteomics field has dramatically benefited from mass spectrometry, especially MS/MS. It has allowed an in-depth study of biomolecules structure and both proteins and peptides identification and quantification.

### 2.2.2 Protein and Peptide Identification

There are several approaches to MS-based protein and peptide identification. A common technique uses proteolytic enzymes (e.g., trypsin) to digest proteins into peptides. An MS analysis is performed over the peptides. They are individually identified by applying a score function to their fragmentation spectrum and using the score to find the corresponding peptide from a database. Therefore, by identifying the peptides present in a sample, it would be theoretically possible to apply reverse engineering to identify the proteins they used to compose. However, assigning a score rule for protein identification is still an open problem due to peptides that are shared by many proteins. Thus, there are many options available with their own pros and cons. For instance, a popular approach to this problem is merely summing the highest score for each peptide identified and comparing it to the protein scores previously calculated.

### 2.2.3 Liquid Chromatography-Mass Spectrometry

Protein samples usually are complex mixtures containing several other proteins and peptides. For this reason, MS/MS is often combined with liquid chromatography (LC), a method for physically separating a sample into individual components in order to examine them in simpler samples or selectively choose what will be analyzed by the mass spectrometer. This combination is known as liquid chromatography-mass spectrometry (LC-MS) and is done by coupling an LC system to a mass spectrometer. A complete LC-MS analysis is referred to as LC-MS run.

In the LC phase, the sample is soaked in a liquid solvent. The solution is pumped through a column filled with an adsorbent material. The molecules in the mixture are separated by their retention time (RT), i.e., the time they remain traveling in the column, as different compounds in the solution move with different velocities.

Afterward, the separated molecules are one by one sent to the mass spectrometer to have their $m/z$ measured. The outcome of an LC-MS run has three dimensions: ion intensity detection, mass-to-charge ratio, and retention time. An ion intensity map, or ion map, is a bidimensional representation of an LC-MS run, where the ion intensity detection is a function of $m/z$ values and retention times. The plot in Figure 2.6 is an example of an ion intensity map representation. This terminology is used by MS Progenesis QI software, and by Maciel, 2019. The additional sample separation provided by LC enriches the analysis that could be carried out from an LC-MS run, reducing spectral interference and increasing the retrieval of structural information about the sample.

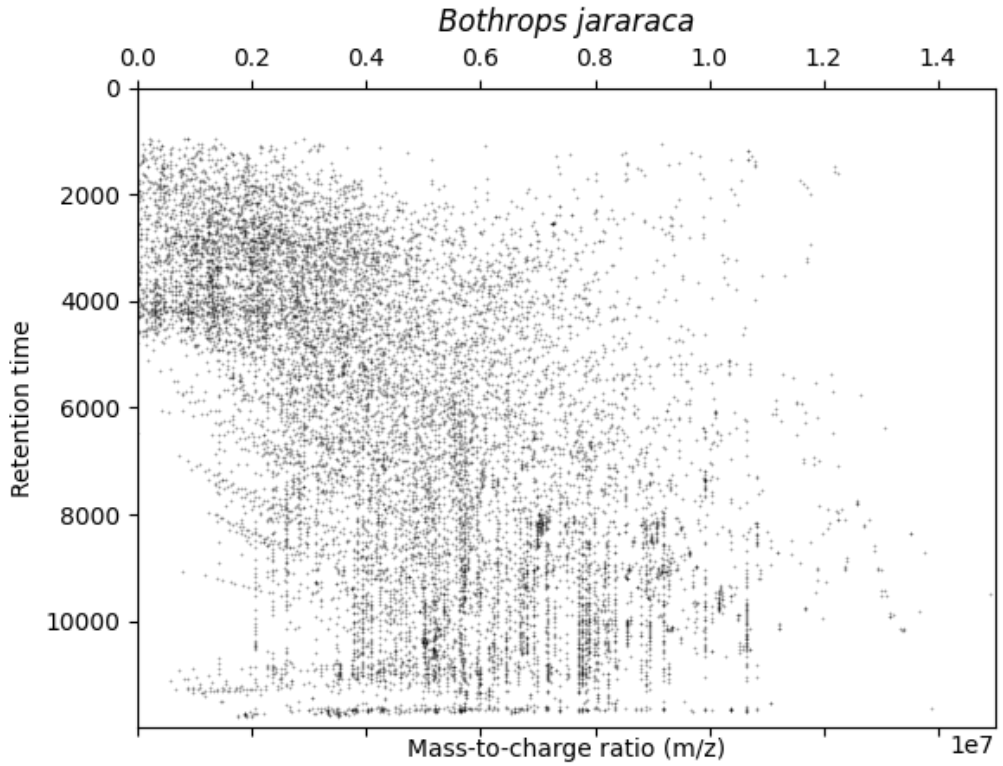



**Figure 2.6:** ***Ion intensity map:*** *a visual representation of raw data measured by LC-MS from Bothrops jararaca snakes venoms.*

## 2.3 Hierarchical Clustering

Hierarchical clustering is a common method used to cluster elements based on similarity. Its search strategy consists of building a binary tree in which the leaves are the initial elements, and the root represents a cluster that contains all elements. The final level of the tree is an ordered list of elements that can be easily visualized in diagrams that can represent trees, called dendrograms (Gentleman, 2008).

The tree of elements can be built using two different approaches:

**Divisive:** there is one initial cluster containing every element, and the tree is built from top-down by dividing the elements into clusters recursively.

**Agglomerative:** each element is assigned to its own cluster, and the tree is built from bottom-up by grouping the leaves into clusters recursively.

The well known agglomerative hierarchical clustering is described in Algorithm 1, adapted from Duda *et al.*, 1973. It requires some strategy in order to measure how different two clusters are. These strategies are called linkage methods, and they generally are a function of some dissimilarity measure between two pairs of elements. The most trivial choices for a dissimilarity measure are distance metrics such as the Euclidean and Manhattan distance.

**Algorithm 1:** Agglomerative hierarchical clustering

```
Assign every element to its own singleton cluster
while number of clusters > 1 do
    Merge the two most similar clusters
end while
```

The most common linkage methods are listed below:

**Single linkage:** the distance between two clusters is the distance of the nearest pair of elements between the two clusters.

**Complete linkage:** the distance between two clusters is the distance of the farthest pair of elements between the two clusters.

**Average linkage:** the distance between two clusters is the pairwise average distance between the elements of the two clusters.

## 2.4 Cluster Validation with Bootstrapping

Hierarchical clustering methods results depend on the choice of dissimilarity metrics and the chosen linkage strategy. In high dimensional datasets, slight changes in the dataset often yield very different outcomes. A natural question that arises is whether the resulting clusters are just an artifact of the chosen cluster algorithm or a meaningful representation of an underlying structure in the data. Additionally, the identified clusters are also susceptible to statistical sampling error and natural sampling variability. Consequently, there are chances that the formed clusters may not reflect the true hypothesis. Therefore,

a methodology is required to measure cluster stability and assess the reliability of the hypothesis obtained from the cluster analysis (Zumel, 2015, Suzuki and Shimodaira, 2004).

Ideally, to test the stability of clusters in a particular clustering method, one would need to apply new samples from the data generating process to the algorithm and verify how the results change. However, when there is no possibility of obtaining new samples, a smart idea would be generating new samples by randomly sampling elements of the available data. This idea is known as statistical bootstrapping (S. Holmes and Huber, 2019, Suzuki and Shimodaira, 2006).

P-values for each cluster can be estimated from the bootstrapped samples. Hence, cluster accuracy can be measured. Moreover, hypothesis tests can be applied to each cluster. If the cluster estimated p-value is less than some $\alpha$, then this cluster is rejected with $\alpha$ significance level (Suzuki and Shimodaira, 2004).

Two examples of bootstrapping resampling p-values are listed below:

**Bootstrap probability (BP):** the frequency that a cluster appears in the bootstrap replicates.

**Approximately unbiased (AU):** an approximated p-value calculated by multiscale bootstrap resampling (Efron *et al.*, 1996, Shimodaira, 2002, Shimodaira, 2004), a bootstrap technique where the data size of bootstrap samples are intentionally modified. AU p-value is proved to be less biased than BP value.

In Figure 2.7, we provide an example of a phylogram generated by hierarchical clustering and assessed using bootstrapping.

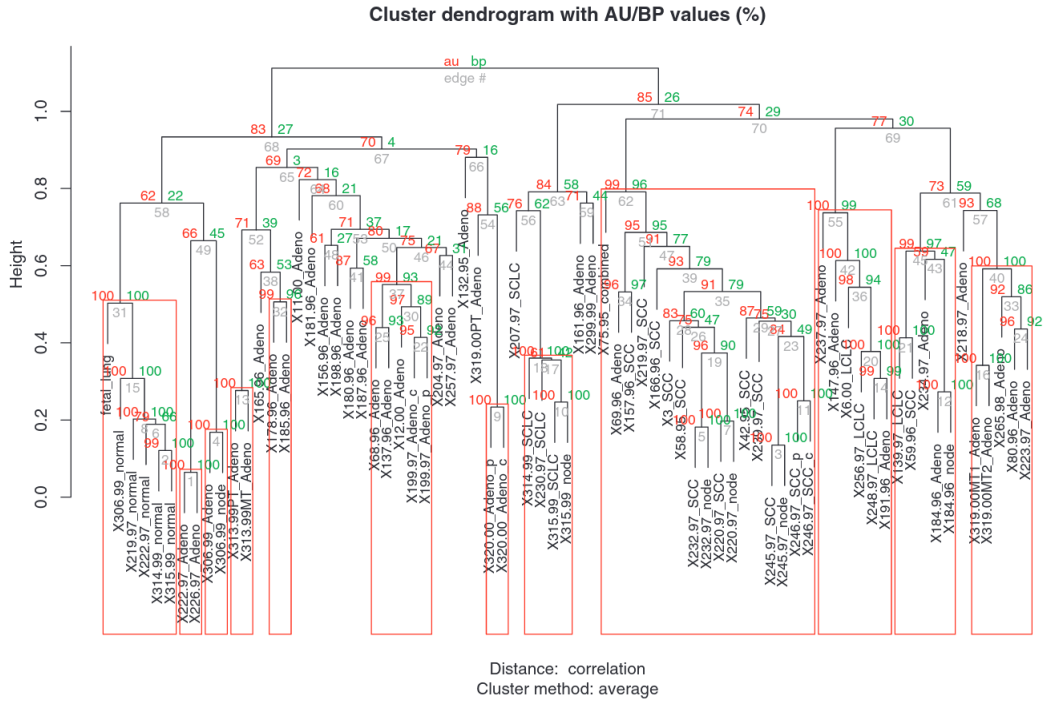



**Figure 2.7:** ***Hierarchical clustering of 73 lung tumors validated with bootstrapping****: values over the branches are AU (left, red) and BP (right, green) p-values, image from* Suzuki and Shimodaira, 2006.

# Chapter 3

# Materials and Methods

This chapter describes the methodology developed to generate phyloproteomic trees from LC-MS runs raw data. Furthermore, it includes details about the automation software implemented for this work as well as external resources used. Figure 3.1 is a flowchart where the full pipeline process can be visualized, from venom extraction to the generation of phyloproteomic trees. The full pipeline incorporates previous work on the subject. This work is a direct continuation of Maciel, 2019, which was based in Raposo, 2018 and Andrade-Silva, Zelanis, *et al.*, 2016.

The pipeline implementation is available under the GNU General Public License in the repository below:

https://github.com/GuilhermeVieira/MITE.

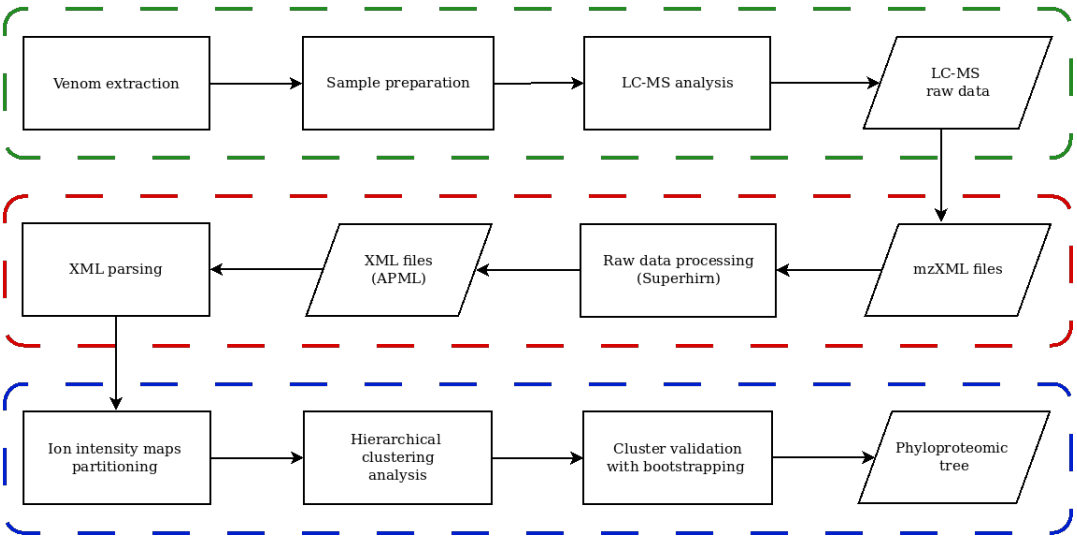



**Figure 3.1:** ***Pipeline flowchart****: rectangles show a process, and parallelograms are input or output data. The steps under the green and red dashed rectangles are described in Andrade-Silva, Zelanis, et al., 2016 and Maciel, 2019. The processes implemented in this work are under the blue dashed rectangles.*

## 3.1 LC-MS Data Acquisition

Andrade-Silva, Zelanis, *et al.*, 2016 describe venom extraction, sample preparation, and LC-MS analysis, conducted in two independent experiments. The highlighted green area in Figure 3.1 represents these steps. Pooled venom samples were extracted from seven species of *Bothrops* genus (*B. cotiara*, *B. insularis*, *B. jararaca*, *B. moojeni*, *B.neuwiedi*, *B. jararacussu*, and *B. erythromelas)*, where each venom pool was composed of extractions from at least ten specimens. Also, samples were subjected to trypsin digestion. The resulting peptide mixtures were injected into an EASY II-nanoLC system coupled to an LTQ-Orbitrap Velos mass spectrometer, both Thermo Fisher Scientific instruments. An LC-MS run was performed for each experiment, where the output is a RAW file, a proprietary file format. In total, fourteen files were generated, two for each species. For detailed information about these processes, see Andrade-Silva, Zelanis, *et al.*, 2016.

## 3.2 Raw Data Processing

Raw data was processed using SuperHirn, according to the pipeline described in Maciel, 2019. SuperHirn is an open-source program that implements a set of tools to process high-resolution LC-MS data. Prof. Ruedi Aebersold group originally developed this software at the Institute of Molecular Systems Biology (ETHZ, Switzerland). SuperHirn is programmed in C++ and is available for Unix platforms. Maciel, 2019 modified the SuperHirn to add extra functionality required to do phyloproteomic analysis.

For the time being, SuperHirn only supports the mzXML open file format, introduced in Pedrioli *et al.*, 2004. The necessary conversion from RAW files to mzXML was done using the msconvert program, available in ProteoWizard, an open-source set of tools for proteomics.

SuperHirn organizes its functionality into modules, divided into two groups: preprocessing and post-processing. Modules must be executed in order, and each one of them outputs the input for the next one. Moreover, every module output is an XML file defined by the Annotated Putative Markup Language (APML), an XML-based data format specified in Brusniak *et al.*, 2008.

The following modules were executed to acquire the seven final XML files:

**Preprocessing modules**:

1. MS1 feature extraction;
2. Build alignment tree;
3. Multiple LC-MS alignments.

**Post-processing modules**:

4. MasterMap intensity normalization;
5. Multiple alignments between runs.

Preprocessing modules perform MS1 feature extraction, build an alignment tree between the two runs of the same species, and combine them into a single MasterMap. The MasterMap is a file that contains all the necessary information to perform further analysis. Post-processing modules provide data analysis or other data processing tools that might be used depending on the analysis intention. The post-processing modules used in this work normalizes intensity values between all runs that compose a MasterMap, and align all MasterMaps values. The last module was implemented by Maciel, 2019.

SuperHirn supports parameter calibration to be adaptable to multiple mass spectrometers devices. The modified SuperHirn parameters are listed in Appendix A, and they are the same parameters used by Maciel, 2019. For detailed algorithmic information about SuperHirn, see Mueller *et al.*, 2007. SuperHirn manual serves as a quick reference to SuperHirn installation, configuration, and execution. More information about SuperHirn modified version is detailed in Maciel, 2019. This modified version source code is accessible via the link below:

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

An automation module was implemented in this work to handle all SuperHirn-related tasks to facilitate new analysis. Details about this automation and other general improvements in the pipeline are discussed in Appendix B.

## 3.3 XML Parsing

The resulting aligned XML MasterMaps were loaded into the Python 3 pipeline automation software developed for this work. The scripts for parsing and loading the ion maps were adapted from Maciel, 2019. The pipeline code was refactored in order to decouple it from the methodology previously implemented, and improve extensibility to facilitate the implementation of other phyloproteomic analysis methods.

The XML MasterMaps were read by an XMLReader class, which uses the ElementTree API to parse XML data efficiently. RT and $m/z$ values are values with two and four decimal places, ranging from 0 to 120 and 300 to 1,800, respectively. Their values were discretized by multiplication by powers of ten to use these values as matrix indices.

Next, each parsed MasterMap was loaded into an IonMap class. Due to the discretization process, ion maps ended up with gigantic dimensions (12,000 × 15,000,000) with approximately 12,000 nonzero points. Hence, the IonMap class was designed to store ion maps in the form of matrices where most elements are zero, namely sparse matrices. SciPy 2-D sparse matrix package was used in this class to build and store the ion maps. It provides multiple data structures to store and manipulate sparse matrices efficiently.

## 3.4 Ion Intensity Maps Partitioning

Hierarchical clustering algorithms build a hierarchy of clusters based on distance matrices. Most implementations support either a 1D condensed distance matrix or a 2D

array of observation vectors as inputs. In the latter option, the implementation receives a distance metric as a parameter and calculates the distance matrix.

A 2D array was built to be an input to the hierarchical clustering algorithm, where each row is an observation vector of an ion intensity map. Each index represents a coordinate $(x, y)$, and the value is the detected intensity. Thus, differences in intensity detection at every point in the matrix can be calculated and compared. However, applying a pairwise distance metric on every point would be problematic because most points would be compared to zero intensity detection since ion maps are naturally sparse.

In order to reduce data granularity, ion maps were partitioned into $d$ squared areas. Information about intensity detection was summarized by summing up intensity values of each point within an area. These partitioned areas were mapped to a 1D vector representing the entire ion map, where the indices correspond to an area, and the value is the sum of intensity values. If ion maps are partitioned in the same number of areas, a pairwise comparison between intensity values of areas from different ion maps can be made when building a distance matrix between ion maps.

In this study experiments, the number of partitions $d$ ranged between 1,250 and 1,250,000. Figure 3.2 illustrates a toy example of an ion intensity map partitioned in 42 areas and transformed into a 1D vector.

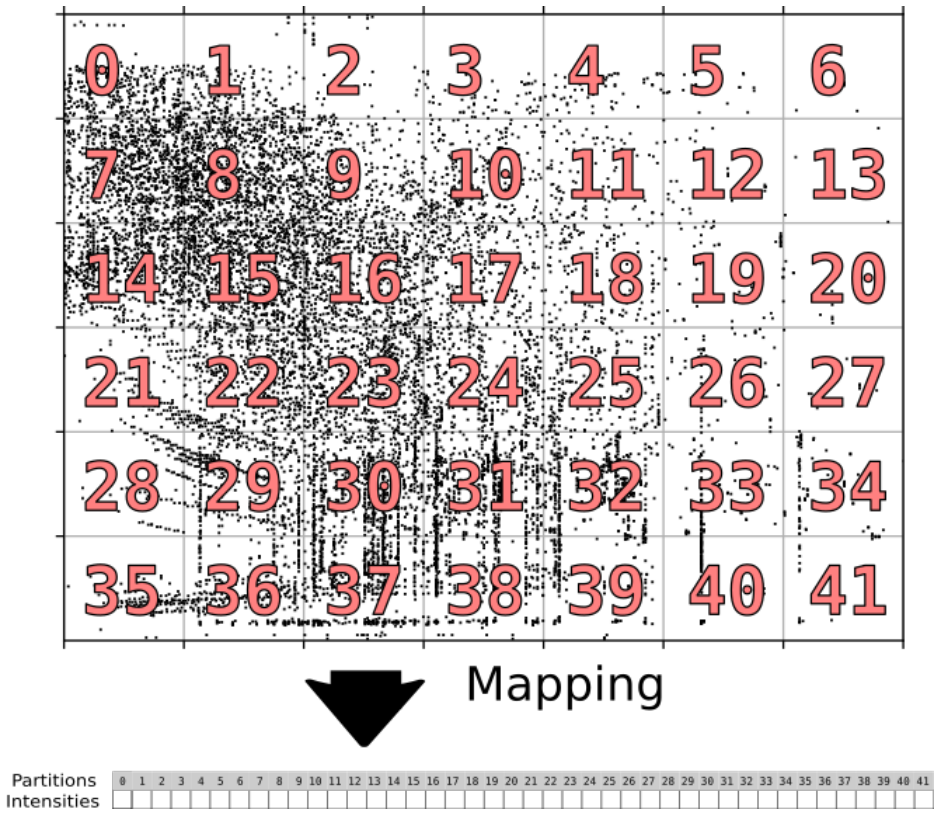



**Figure 3.2:** ***A toy example of ion intensity maps partitioning****: ion maps were partitioned in $d = 42$ areas and mapped to a 1D vector. Array indices correspond to the areas numbered in red in the ion map. Intensities values are the sum of intensities detected in the corresponding area.*

## 3.5 Hierarchical Clustering Analysis and Bootstrapping

Hierarchical clustering analyses were performed using the pvclust R library with varying linkage methods and dissimilarity metrics. This library was chosen due to its capability of validating the clusterings with multiscale bootstrapping resampling. However, pvclust uses the dist and hclust R libraries to calculate distance matrices and build hierarchical clusterings, respectively. The high-level interface rpy2 was used to convert Python objects to R and make R functions available from Python code. In the previous chapter, Figure 2.7 depicts a plot produced by pvclust that contains p-values for each of the clusters formed.

The number of bootstrap samples created by pvclust is determined by the *nboot* parameter. It was set to $nboot = 1000$ in every analysis performed since Suzuki and Shimodaira, 2006 recommends generating at least 1000 for statistical significance. Single, complete, and average linkage methods were used. The distances used as dissimilarity metrics are listed in Table 3.1.

| **Distance** | **Formula** | **Parameters** |
|---|---|---|
| Euclidean | $\|\|u - v\|\|_2$ | $u$ : n-dimensional vector<br>$v$: n-dimensional vector |
| Manhattan | $\sum_i \|u_i - v_i\|$ | |
| Canberra | $\sum_i \frac{\|u_i - v_i\|}{\|u_i\| + \|v_i\|}$ | |
| Cosine | $1 - \frac{u \cdot v}{\|\|u\|\|_2 \|\|v\|\|_2}$ | |
| Jaccard | $\frac{c_{TF} + c_{FT}}{c_{TT} + c_{FT} + c_{TF}}$<br>where:<br>$c_{ij}$ is the number of occurrences of<br>$u[k] = i$ and $v[k] = j$ for $k < n$ | $u$ : n-dimensional boolean vector<br>$v$: n-dimensional boolean vector |

**Table 3.1:** ***Dissimilarity metrics used in this work:*** *Adapted from The SciPy community, 2020.*

# Chapter 4

# Results

Multiple dendrograms were generated using the methodology described in Chapter 3 with varying parameters. This chapter presents a summary of the most relevant phyloproteomic trees generated by this methodology. All results are presented using the average linkage method since different linkage methods did not significantly affect the topology of dendrogram.

Dendrograms generated using Canberra, Cosine, and Jaccard distances qualitatively converged to the same topology after partitioning by a particular number of areas. A representative plot of this topology can be seen in Figure 4.1, which is a Canberra distance dendrogram partitioned in $d$ = 20,000. BP and AU p-values in this figure reveal high robustness in all clusters formed.

Nevertheless, dendrograms generated by $L_k$ Norm distances (Manhattan and Euclidean) are consistent with a ladder-like pattern, no matter how many areas they were partitioned. Figure 4.2 exemplifies the dendrogram pattern found when applying $L_k$ Norm distances.

After partitioning in $d$ = 625,000 and $d$ = 180,000, Canberra and Jaccard distance dendrograms respectively converged to the same ladder-like pattern of $L_k$ Norm distances. In contrast, Cosine distance dendrograms did not converge to the ladder-like pattern in the experiments, maintaining the representative topology when partitioned in a high number of areas (Figure 4.6b).

Among Canberra, Cosine, and Jaccard dendrograms, Canberra distance ones were the most consistent in terms of topology formed. Their topology did not change when partitioning from $d$ = 1,250 until $d$ = 625,000.

Before stabilizing in the topology shown in Figure 4.4b, Jaccard distance dendrograms changed intermittently to the one observed in Figure 4.4a. BP and AU p-values indicate that the formed clusters were not robust. Meanwhile, before converging to the representative topology, Cosine distance formed robust dendrograms with another topology shown in Figure 4.6a.

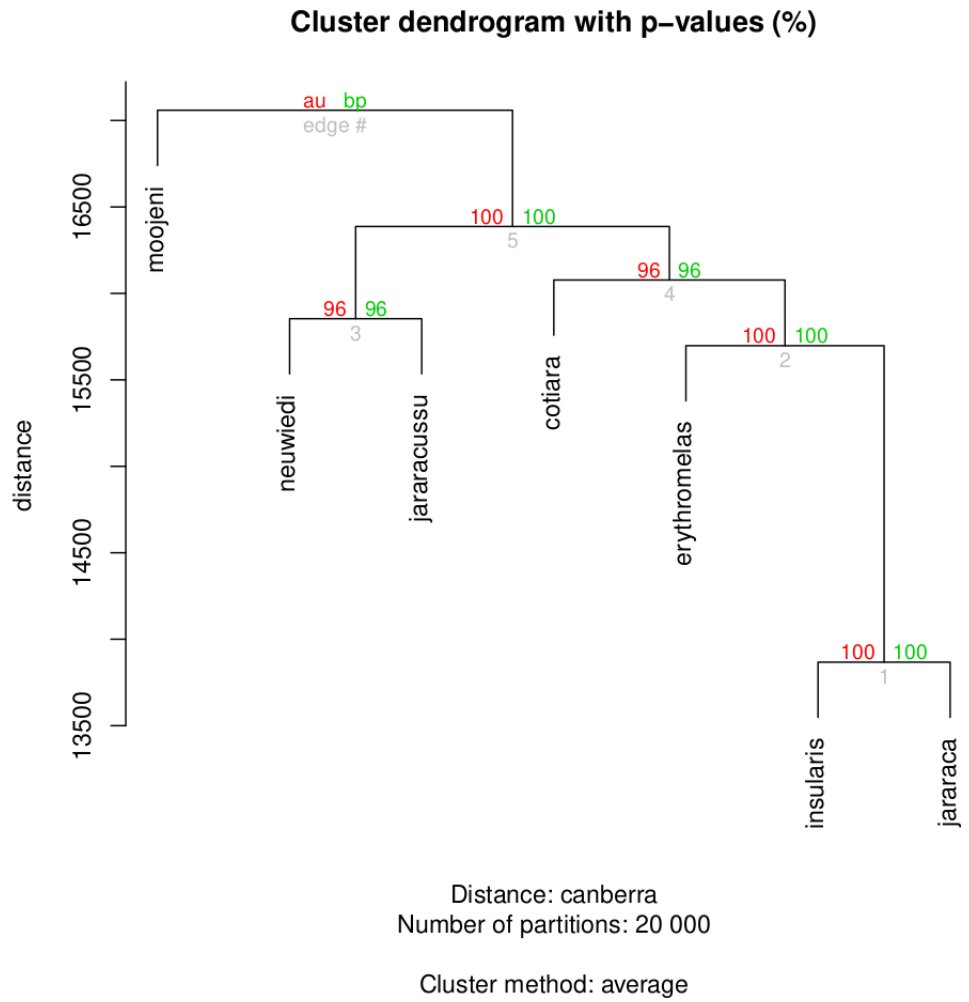



**Figure 4.1:** ***Hierarchical clustering dendrogram of venoms from seven species of the Bothrops genus validated with bootstrapping****: the venom ion map was partitioned in d = 20,000 squared areas. Canberra distance and average linkage method was used to cluster the venom profiles. Values over the branches are AU (left, red) and BP (right, green) p-values.*

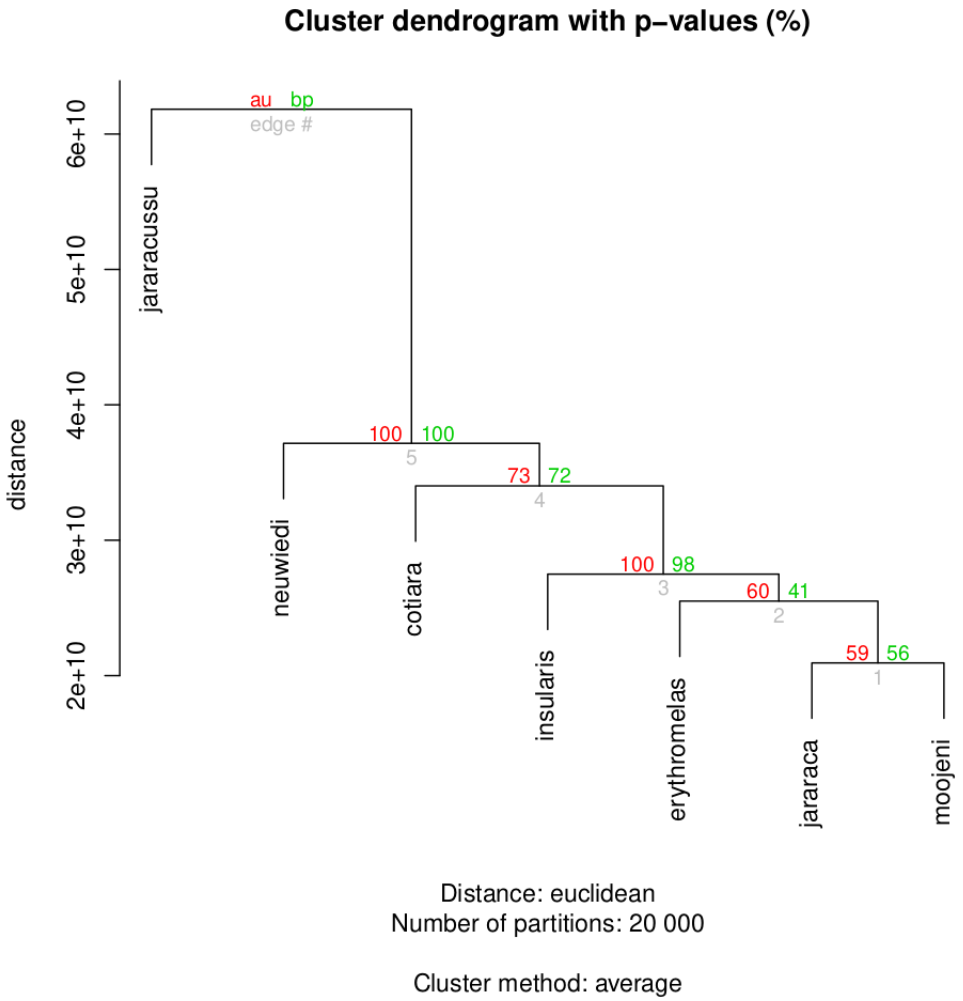



**Figure 4.2:** ***Hierarchical clustering dendrogram of venoms from seven species of the Bothrops genus validated with bootstrapping****: the venom ion map was partitioned in d = 20,000 squared areas. Euclidean distance and average linkage method was used to cluster the venom profiles. Values over the branches are AU (left, red) and BP (right, green) p-values.*

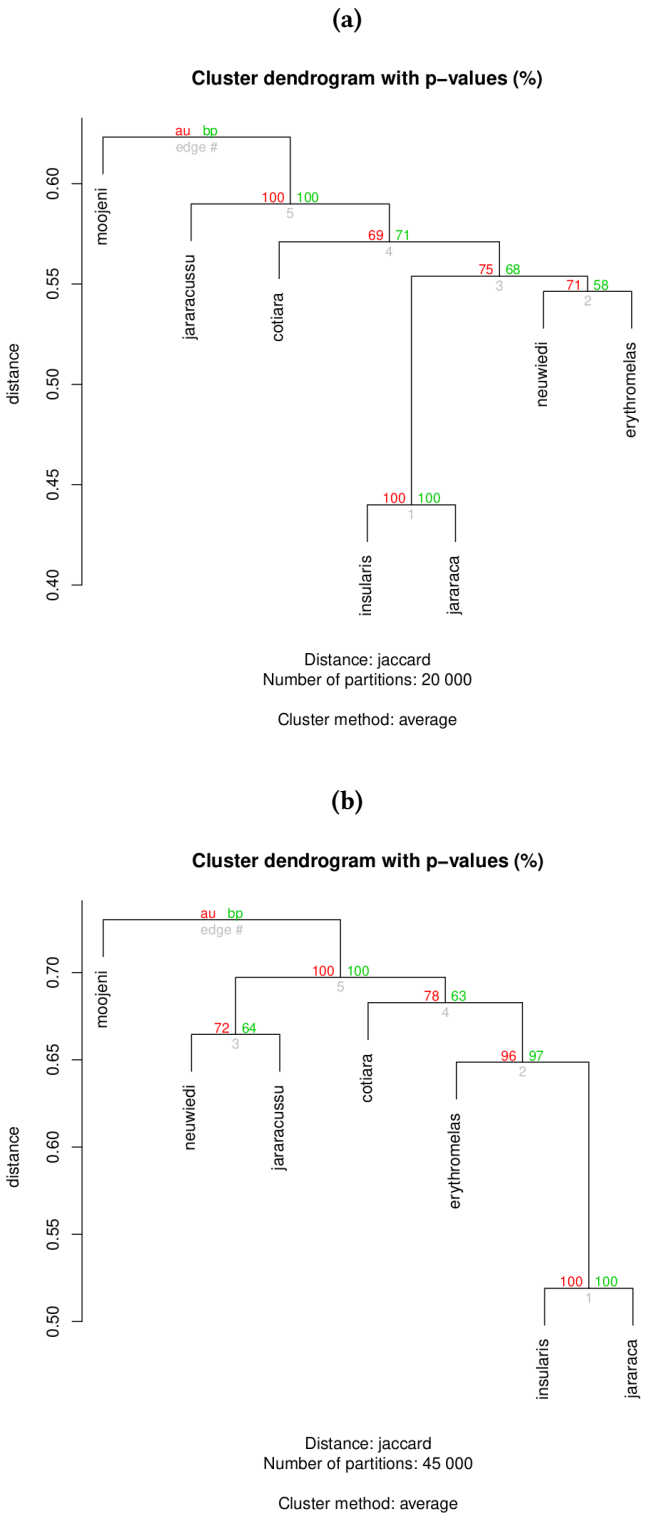



**Figure 4.3:** ***Hierarchical clustering dendrogram of venoms from seven species of the Bothrops genus validated with bootstrapping****: the venom ion map was partitioned in d = 20,000 (a) and in d = 45,000 squared areas. Jaccard distance and average linkage method was used to cluster the venom profiles. Values over the branches are AU (left, red) and BP (right, green) p-values.*

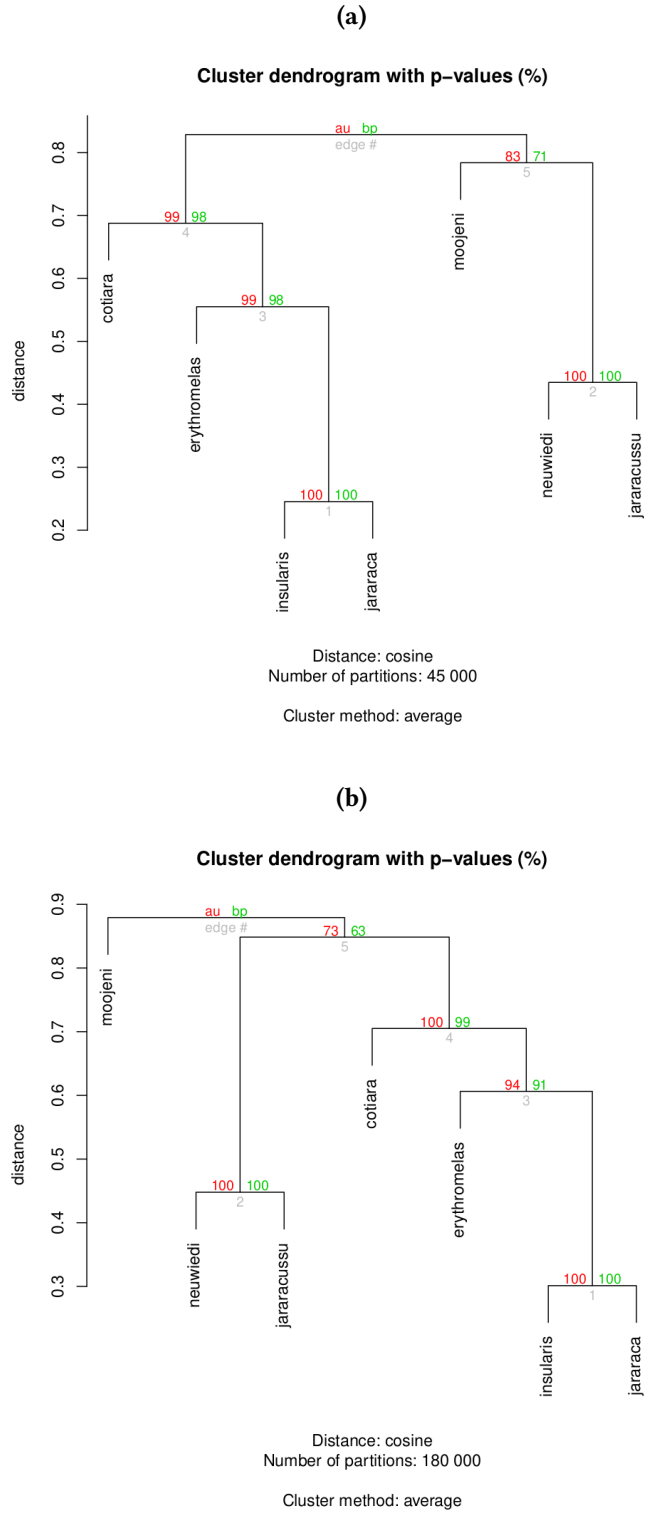



**Figure 4.5:** ***Hierarchical clustering dendrograms of venoms from seven species of the Bothrops genus validated with bootstrapping****: the venom ion map was partitioned in d = 45,000 (a) and in d = 180,000 (b) squared areas. Cosine distance and average linkage method was used to cluster the venom profiles. Values over the branches are AU (left, red) and BP (right, green) p-values.*

# Chapter 5

# Discussion

In this study, hierarchical clustering was utilized to generate phyloproteomic trees from raw data of venoms of seven *Bothrops* species measured by LC-MS. It is a direct continuation of Maciel, 2019, which inferred phyloproteomic trees over raw data from LC-MS using a Bayesian inference approach.

The predominant topology observed in all generated dendrograms (Figure 4.1) is similar to the reference phylogram inferred from mitochondrial DNA and morphological characters (Figure 2.3). The branching pattern witnessed is nearly the same, aside from *B. neuwiedi* and *B. moojeni*. Moreover, *B. cotiara* position in the tree is consistent with the polytomy in the reference phylogram. *B. neuwiedi* was clustered in agreement with the phylogenetic tree in some dendrograms when using Jaccard distance (Figure 4.4a). This distance metric measures the dissimilarity between two Boolean arrays. The arrays used for clustering store the sum of intensities detected in a given area of an ion map. As a consequence, Jaccard distance discards this sum and only counts as similarity if both areas had detections or both areas had no detection. Although the cluster formed with *B. erythromelas* was not robust, this result suggests that the venom profiles of these snakes are more similar when verifying if analogous areas had detections. Phyloproteomic trees where *B. neuwiedi* is fully consistent with the genetic phylogram were not reported in Maciel, 2019. However, Raposo, 2018 found trees where *B. neuwiedi* and *B. erythromelas* are clustered together when using binary peptidic data from whole venom analysis. The cladograms with this cluster generated by Raposo, 2018 had low posterior probability, which might be related to the low robustness of these clusters found in this work. This relationship is a compelling case for future investigation.

The discrepancies regarding *B. neuwiedi* have been a recurring theme in previous works. Raposo, 2018 raised two hypotheses to explain these discrepancies to the reference phylogram. *B. neuwiedi* has recently undergone a taxonomic revision where its former subspecies became different species. There is no report about how these species have contributed to the venom pool used in the LC-MS assays in Andrade-Silva, Zelanis, *et al.*, 2016. Thus, the first hypothesis is that the venom pool might contain individuals of different species, which may have impacted the consistency of the results. The second hypothesis concerns the pace of evolutionary pressure on the venom. In short periods, changes are more likely to occur in the venom composition if there are intense environmental

pressures on its performance. Hence, these protein profiles of snake venoms could have experienced significant alterations in their proteome in a relatively short time, which explains the fact that cladograms generated from venom profiles do not perfectly correlate with phylogenetic trees.

A ladder-like pattern was observed in the dendrograms generated by $L_k$ Norm distances (Euclidean and Manhattan). This pattern could also be visualized in trees formed using other distances when the number of ion map partitions increased, aside from Cosine distance. A possible explanation for these results is the argument stated in Beyer *et al.*, 1997 that under reasonable conditions, the difference between the nearest and the farthest data points approaches zero as the dimensionality of the problem grows. Hence, the nearest neighbor problem becomes ill-defined and, therefore, meaningless. There was empirical evidence that this was the case since BP and AU p-values indicate that these clusters are unstable. Euclidean and Manhattan distances are known to discriminate poorly in high dimensions. On the other hand, Cosine distance is widely used in high dimensionality problems, such as measuring document dissimilarity, and it is perceived to be a better discriminator in high dimensions. Before converging to the predominant topology mentioned earlier, the dendrograms generated by this distance (Figure 4.6a) were robust and topologically identical to some phyloproteomic trees generated by Maciel, 2019.

# Chapter 6

# Conclusion

In this work, a methodology was developed to generate phyloproteomic trees using hierarchical clustering from raw data measured by LC-MS experiments. This approach skips intermediate steps as protein and peptide identification that may introduce bias to the analysis. Furthermore, hierarchical clustering provides results that are simple to understand and easy to interpret.

Raw LC-MS data was partitioned into squared areas and mapped to 1D arrays. Hierarchical clustering analyses were employed over these arrays using different dissimilarity metrics and linkage methods. The phyloproteomic trees generated were validated with statistical bootstrapping.

The methodology was tested in pooled venom samples from seven species of the *Bothrops* genus. The findings reaffirm that cladograms formed from snake venom proteomes are correlated with phylogenetic trees of these species. Furthermore, results indicate that using raw data from LC-MS experiments to find patterns in proteomic data is a reliable alternative to peptide and protein identification.

However, some discrepancies between the topologies of phylogenetic and phyloproteomic trees have been recurring in previous works. These disparities require further investigation in future research. In this regard, a possibility is to apply this methodology into MS-data generated with new venom samples; this would be especially interesting for the case of *B. neuwiedi*, which had the most unstable topology in our computational experiments. Another possibility in this research line would be to apply our methodology into other proteomic data rather than snake venoms and benchmark it with other approaches.

# Appendix A

# SuperHirn Configurations

| Parameter | Default Value | Modified Value |
|---|---|---|
| MS1 retention time tolerance | 1.0 | 2.5 |
| MS1 m/z tolerance | 1.0 | 6 |
| RT end elution window | 180.0 | 120.0 |
| MS1 feature signal to noise threshold | 0.5 | 3.0 |
| MS1 feature intensity cutoff | 5000 | 200 |
| MS1 feature CHRG range min | 1 | 2 |
| MS1 feature CHRG range max | 5 | 9 |
| MS1 feature mz range min | 0 | 300 |
| MS1 feature mz range max | 2000 | 1800 |
| FT peak detect MS1 m/z tolerance | 10 | 6 |
| FT peak detect MS1 intensity min threshold | 1000 | 200 |
| Relative isotope mass precision | 10 | 6 |
| Activation of MS1 feature merging post processing | 1 | 0 |

**Table A.1:** ***SuperHirn parameters that have been modified from their default values:*** *Table extracted from Maciel, 2019.*

# Appendix B

# Software Improvements

This appendix discusses some improvements implemented in the pipeline from a software engineering perspective.

## B.1 Pipeline Containerization

The pipeline implemented in previous works requires processing steps in external software, such as SuperHirn, for raw data processing of LC-MS experiments, and MrBayes, for Bayesian inference of phyloproteomic and phylogenetic trees. However, the installation process of these currently demands a manual compilation for some computer environments. Also, the main pipeline developed uses several R and Python packages and libraries that must be individually installed. These complex software requirements add burden to the end-user and make the developed methodology hard to reproduce, which is crucial for science.

In order to solve this problem, a containerization solution was applied in this work using Docker. All software dependencies were packaged into a Docker image. Thus, one who wishes to reproduce the pipeline only needs a Docker version installed in their machine and this project image to run this project.

## B.2 SuperHirn Wrapper

Raw data processing with SuperHirn is composed of several specific steps described in subsection 3.2. The execution of these steps is monotonous and error-prone. An automation module was implemented in Python to encapsulate SupeHirn processing steps to the end-user. This wrapper calls SuperHirn modules and manages their input and output files. All these steps occur within the Docker container. In the end, the SuperHirn wrapper copies the final XML files to the project folder.

# References

[Aebersold and Mann 2003] Ruedi Aebersold and Matthias Mann. "Mass spectrometry-based proteomics". In: *Nature* 422.6928 (2003), pp. 198–207. DOI: 10.1038/nature01511. URL: https://www.nature.com/articles/nature01511 (cit. on p. 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". In: *Molecular & Cellular Proteomics* 17.7 (2018), pp. 1261–1284. ISSN: 1535-9476. DOI: 10.1074/mcp.RA118.000748. eprint: https://www.mcponline.org/content/17/7/1261.full.pdf. URL: https://www.mcponline.org/content/17/7/1261 (cit. on pp. 1, 6).

[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". In: *Journal of proteome research* 15.8 (2016), pp. 2658–2675 (cit. on pp. 1, 6, 15, 16, 27).

[Beyer *et al.* 1997] Kevin Beyer, Jonathan Goldstein, Raghu Ramakrishnan, and Uri Shaft. "When Is "Nearest Neighbor" Meaningful?" In: *ICDT 1999. LNCS* 1540 (Dec. 1997) (cit. on p. 28).

[Brusniak *et al.* 2008] Mi-Youn Brusniak *et al.* "Corra: Computational framework and tools for LC-MS discovery and targeted mass spectrometry-based proteomics". In: *BMC Bioinformatics* 9.1 (2008), p. 542. DOI: 10.1186/1471-2105-9-542 (cit. on p. 16).

[Colinge and Bennett 2007] Jacques Colinge and Keiryn Bennett. "Introduction to computational proteomics". In: *PLoS computational biology* 3.7 (2007), e114 (cit. on p. 7).

[Duda *et al.* 1973] Richard Duda, Peter Hart, and David Stork. "Unsupervised learning and clustering". In: *Pattern classification.* 1973 (cit. on p. 11).

[Efron *et al.* 1996] Bradley Efron, Elizabeth Halloran, and Susan Holmes. "Bootstrap confidence levels for phylogenetic trees". In: *Proceedings of the National Academy of Sciences* 93.14 (1996), pp. 7085–7090. ISSN: 0027-8424. DOI: 10.1073/pnas.93.14.7085. eprint: https://www.pnas.org/content/93/14/7085.full.pdf. URL: https://www.pnas.org/content/93/14/7085 (cit. on p. 12).

[Fenwick *et al.* 2009] Allyson Fenwick, Ronald Gutberlet, Jennafer Evans, and Christopher Parkinson. "Morphological and molecular evidence for phylogeny and classification of South American pitvipers, genera *Bothrops*, *Bothriopsis*, and *Bothrocophias* (Serpentes: Viperidae)". In: *Zoological Journal of the Linnean Society* 156.3 (2009), pp. 617–640 (cit. on p. 6).

[Flowers *et al.* 2015] Paul Flowers, Klaus Theopold, Richard Langley, and William Robinson. "2.3 Atomic Structure and Symbolism". In: *Chemistry*. OpenStax, 2015. URL: https://openstax.org/books/chemistry/pages/2-3-atomic-structure-and-symbolism (cit. on p. 8).

[Gentleman 2008] Robert Gentleman. "Cluster Analysis of Genomic Data". In: *Bioinformatics and computational biology solutions using R and bioconductor*. Springer, 2008 (cit. on p. 11).

[S. Holmes and Huber 2019] Susan Holmes and Wolfgang Huber. "Clustering". In: *Modern statistics for modern biology*. Cambridge University Press, 2019. URL: https://web.stanford.edu/class/bios221/book/Chap-Clustering.html (cit. on p. 12).

[Maciel 2019] Gustavo Maciel. *Um método baseado em multirresolução para análise filoproteômica de venenos de serpentes*. Tech. rep. Monografia de graduação em Computação. Instituto de Matemática e Estatística, Universidade de São Paulo, 2019 (cit. on pp. 1, 6, 9, 15–17, 27, 28, 31).

[Mueller *et al.* 2007] Lukas Mueller *et al.* "SuperHirn - A novel tool for high resolution LC-MS-based peptide/protein profiling". In: *Proteomics* 7 (Oct. 2007), pp. 3470–80. DOI: 10.1002/pmic.200700057 (cit. on p. 17).

[Page and E. Holmes 2009] Roderick Page and Edward Holmes. *Molecular evolution: a phylogenetic approach*. Blackwell Science, 2009 (cit. on p. 3).

[Pedrioli *et al.* 2004] Patrick Pedrioli *et al.* "A Common Open Representation of Mass Spectrometry Data and Its Application to Proteomics Research". In: *Nature Biotechnology* 22 (Nov. 2004), pp. 1459–1466. DOI: 10.1038/nbt1031 (cit. on p. 16).

[Raposo 2018] Victor Raposo. *Análise filogenética computacional de serpentes do gênero* Bothrops *a partir de proteomas de venenos*. Tech. rep. Monografia de graduação em Computação. Instituto de Matemática e Estatística, Universidade de São Paulo, 2018 (cit. on pp. 1, 6, 15, 27).

[Shimodaira 2002] Hidetoshi Shimodaira. "An Approximately Unbiased Test of Phylogenetic Tree Selection". In: *Systematic biology* 51 (July 2002), pp. 492–508. doi: 10.1080/10635150290069913 (cit. on p. 12).

[Shimodaira 2004] Hidetoshi Shimodaira. "Technical details of the multistep-multiscale bootstrap resampling". In: (Apr. 2004) (cit. on p. 12).

[Suzuki and Shimodaira 2004] Ryota Suzuki and Hidetoshi Shimodaira. "An Application of Multiscale Bootstrap Resampling to Hierarchical Clustering of Microarray Data: How Accurate are these Clusters?" In: *The Fifteenth International Conference on Genome Informatics* 2004 (Jan. 2004) (cit. on p. 12).

[Suzuki and Shimodaira 2006] Ryota Suzuki and Hidetoshi Shimodaira. "Pvclust: an R package for assessing the uncertainty in hierarchical clustering". In: *Bioinformatics* 22.12 (Apr. 2006), pp. 1540–1542. issn: 1367-4803. doi: 10.1093/bioinformatics/btl117. eprint: https://academic.oup.com/bioinformatics/article-pdf/22/12/1540/543117/btl117.pdf. url: https://doi.org/10.1093/bioinformatics/btl117 (cit. on pp. 12, 13, 19).

[The SciPy community 2020] The SciPy community. *Distance computations*. 2020. url: https://docs.scipy.org/doc/scipy/reference/spatial.distance.html (cit. on p. 19).

[Zumel 2015] Nina Zumel. *Bootstrap Evaluation of Clusters*. Sept. 2015. url: https://www.r-bloggers.com/2015/09/bootstrap-evaluation-of-clusters/ (cit. on p. 12).