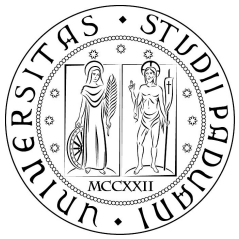

**UNIVERSITA DEGLI STUDI DI PADOVA**

DIPARTIMENTO DI INGEGNERIA

*CORSO DI LAUREA IN INGEGNERIA DELL'INFORMAZIONE*

# A tool for reconstructing phylogenies from the composition of protein motifs

by

Vasiar Allaj

Relatore Dr. Cinzia Pizzi Correlatore: Dr. Fabio Cunial

ANNO ACCADEMICO 2012/2013

## Abstract

The aim of this work was the development of a tool for phylogenetic analysis. In particular, the tool implements an alignment free approach that consider biological signals as vector units. For this reason we called it TBP as for Trees from Biologically significant Patterns. The architecture of the tool is explained in details. Some preliminary experiments hint that some evolutionary signal might be indeed encoded with presence/absence of biologically significant patterns. If this should be confirmed, then it might lead to some new biological insight.

# Contents

# List of Figures

# Chapter 1

# Introduction

Bioinformatics is an interdisciplinary field that develops methods for analysing biological data. One of the major activities in bioinformatics is to develop software tools to generate useful biological knowledge.

The first experiments in Bioinformatics date back to 1970s, when Elvin A. Kabat and Margaret Oakley Dayhoff, among others, were working on biological sequence analysis. With the fast growing of machines computability in the following years, and the increase data analysis in biology, bioinformatics started growing faster. Major research areas include: Sequence analysis, Genome annotation, Analysis of gene expression, Computational evolutionary biology etc.

Phylogeny is the science of inferring evolutionary insight for a group of organisms. To represent a phylogeny, a phylogenetic tree or cladogram is used. A phylogenetic tree is usually derived from the aligned sequences of common protein, RNA, rRNA, whole genome based etc. Even if alignment-based techniques offer good results in terms of reconstructing phylogenetic trees, they lack in defining good dissimilarity where there is no conserved contiguity between genomes [16]. Conserved contiguity is a fundamental hypothesis in sequence alignment.

To overcome this problem, alignment-free sequence comparison methods have been introduced. They take no conserved contiguity hypothesis in consideration. This al-

lows alignment-free methods, which represents sequences in terms of the substrings compoments, to compare very different genomes, and define a good distance between them. They are generally fast, althought not as precise as alignment sequence methods.

The purpose of this thesis was to develop a software to build phylogenetics trees from biologically significant signals, implementing an alignment free tecnhique based on biological signals databases[8]. Some preliminary, encouraging, tests were done to try to validate the hypothesis of whether there is a significant relationship between biologically significant patterns and phylogenetic relationships. Being at his first developmental stage, the Tree from Biologically significant Patterns (TBP) tool relies on some well known, although not always efficient, bionformatics software, and it has a limited set of options. However, the modular design of its architecture, will allow in the future to plug-in faster, state-of-the-art algorithms to speed-up the processing time, and to expand the set of data and options.

The thesis is organised as follows. Chapter 2 will introduce to the main concepts and definitions in phylogenetic analysis. In Chapter 3 the architecture and implementation details of the proposed software will be presented. Next, some preliminary experimental results will be shown in Chapter 4. Finally, conclusions will be drawn and future work discussed in the last chapter.

# Chapter 2

# Introduction to Phylogenetic Analysis

Bioinformatics is a field where people from different backgrounds, such as: biology, computer science and information technology, work together towards a common aim: understanding the mechanisms underlying biological processes. The following section will give an introduction to the main concepts and definitions in phylogenetics, while the next section will concentrate on a specific kind of approach for this analysis: alignment free techniques.

## 2.1 Basic Terminology

Phylogenetics, is the science of phylogeny that includes also taxonomy. Taxonomy is the science of naming and classifying the diversity of organisms. Phylogenetics is based on molecular sequencing data and morphological data matrices[4]. Molecular data tries to determine the precise order of nucleotides within a DNA molecule, where nucleotides are one of the four bases of DNA: Adenine, Guanine, Cytosine and Thymine. On the other hand morphology data deals with the study of the form, structure and features of organisms. It is one of the oldest and well studied methods used to classify organisms, and to get evolutionary insight.

Phylogenetic analyses tries to infers or estimate evolutionary relationships between

species. The basic idea behind cladistics is that members of a group or clade share a common evolutionary history and are more related to each other than to members of another group by sharing a unique feature that is not present in distant ancestors. These features may be found in: DNA, RNA, morphological features etc.
There are three basic assumptions in cladistics [4]:

- Any group of organisms is related by descent from a common ancestor (fundamental tenet of evolutionary theory).
- There is a bifurcating pattern of cladogenesis. This assumption is controversial.
- Change in characteristics occurs in lineages over time. This is necessary condition for cladistics to work.

The relationships from a cladistics analysis are most common represented by a phylogenetic tree. A number of terms frequently used in phylogenetic analysis are [4]:

- Clade - is a monophyletic taxon. Clades are groups of organisms or genes that include the most recent common ancestor of all of its members and all of the descendants of that most recent common ancestor.
- Taxon - is any named group of organisms but not necessarily a clade.
- Branch - correspond to divergence. It means, species from closer branches correspond to more related ones. In general the branch lengths are not shown since it has no biological meaning.
- Node - is a bifurcating branch point.

Here follows some definitions that will be used in the next chapters.

**Definition: Ultrametric tree**

Let $D$ be a symmetric $n$ by $n$ matrix of real numbers. An *ultrametrictree* for $D$ is a rooted tree $T$ with the following properties:

1. $T$ contains $n$ leaves, each labeled by a unique row of $D$.
2. Each internal node of $T$ is labeled by one entry from $D$ and has at least two children.
3. Along any path from the root to a leaf, the numbers labeling internal nodes *strictly decrease*
4. For any two leaves $i$, $j$ of $T$, $D(i,j)$ is the label of the least common ancestor of $i$ and $j$ in $T$.

**Definition: Additive tree**

Let $D$ be a symmetric $n$ by $n$ matrix where the numbers of the diagonal are all zero and the off-diagonal numbers are all strictly positive. Let $T$ be an edge weighted tree with at least $n$ nodes, where $n$ distinct nodes of $T$ are labeled with the rows of $D$. Tree $T$ is called an *additive tree* for matrix $D$ if, for every pair of *labeled* nodes $(i,j)$, the path from node $i$ to node $j$ has total weight (or distance) exactly $D(i,j)$

Both Ultrametric tree and is Additive tree definitions are important because it will allow us to convert a general tree into a binary tree in Chapter 3.

**Definition: Newick tree format**[1]

Conventions: Items in {.} may appear zero or more times. Items in [.] are optional, they may appear once or not at all. All other punctuation marks (colon, semicolon, parentheses, comma and single quote) are required parts of the format.

- $tree$ ==> $descendant\_list[root\_label][:branch\_length]$;
- $descendant\_list$ ==> $(subtree\{,subtree\})$
- $subtree$ ==> $descendant\_list[internal\_node\_label][:branch\_length]$
  ==> $leaf\_label[:branch\_length]$

[1]This definitions was taken from site: $http://evolution.genetics.washington.edu/phylip/newick\_doc.html$

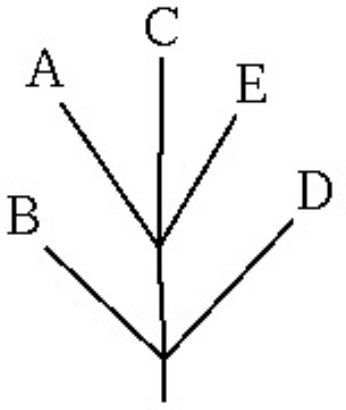


Figure 2.1 – The tree of the newick tree format: (B,(A,C,E),D);

- $root\_label ==> label$
- $internal\_node\_label ==> label$
- $leaf\_label ==> label$
- $label ==> unquoted\_label$
  $==> quoted\_label$
- $unquoted\_label ==> string\_of\_printing\_characters$
- $quoted\_label ==>' string\_of\_printing\_characters'$
- $branch\_length ==> signed\_number$
  $==> unsigned\_number$

Newick tree format is one of the most used tree format in Bioinformatics. Since it follows a natural language it makes it easy to be interpreted without drawing the tree with a software. For example the tree in Figure 2.1 is (B,(A,C,E),D);

## 2.2 Phylogeny and alignment-free techniques

In the last decades bioinformatics has known the fastest growing period as "big data" to analyse became available through high-throughput technologies. In fact, since the

first draft of the human genome was relised, new techniques have been introduced in genome sequencing and have made available huge amount of data. Nowadays DNA sequencing has become easier and orders of magnitude faster [14]. The genome data have pushed, among others, the phylogenetic analysis forward.

### 2.2.1 Main steps of phylogenetic analysis

Well-known steps in phylogenetic analysis are taken when reconstructing a phylogenetic tree. These steps are:

1. Determine a measure for sequence similarity
2. Compute pairwise distance of all the sequences
3. Build the tree
4. Evaluate the goodness of the tree

Among these steps, probably the most fundamental is to define a good distance measure between species, as species that have short distance will be grouped closed in the tree, while species with a greater distance will be apart. A number of studies have been done for the problem of finding a good definition for sequence similarity, and the whole genomes distance computations can be categorised in [9]:

1. frequencies of common words or motifs
2. presence or absence of shared homologous genes
3. gene order along the chromosomes
4. assembly of several gene trees

Methods used in text search and related fields were soon introduced to phylogenetic analysis to deal with the problem of computing the sequence similarity. Initially,

these methods rely on first aligning reference homologous sequences and then deriving a score for the alignment of individual units, typically the logarithm of the odds ratio [16]. A well known tool to establish phylogeny based in sequence alignment is BLAST (Basic local alignment search tool) [1]. It is widely used because it is fast and can be used for other purposes as well: identifying species, locating domains, DNA mapping etc. Alignment-based similarity measure, although precise, are quite slow and do not scale well when thousands of genomes need to be compared. Alignment free methods were initially used as pre-selection filters for alignment based methods [16]. Nowadays new techniques are developed, and alignment-free sequence comparison methods are becoming more appelaing even on their own.

### 2.2.2 Alignment-free techniques

To define dissimilarities, alignemnt-free techniques characterise a sequence in terms of its substring composition. A vector is assigned to each sequence to describe the presence/absence or frequency of the features that have been chosen as descriptors. Next, a distance between vectors is defined, and pairwise distance are computed between each pair of sequence in the input set. The computed values will fill a distance matrix that will subsequently be used to build a phylogenetic tree. While most alignment-free techniques relies on structural features (i.e. substrings of a fixed/variable length of the sequences), an interesting alternative would be to define the features in terms of biological functional elements [8]. The tool developed in this thesis is the first step towards an in-depth phylogenetic analysis using this approach based on biologically significan patterns as feature vectors to try to unveil possible evolutionary signals described by relevant biological patterns.

An overview of typical solution to the main steps of phylogenetic analysis provided by alignment-free techniques is given below.

**Measures of similarity**

Some of the most common distances in alignment free method are:

1. MSM (Maximum Significant Match) - is a word that it is present on two DNA sequences, which cannot be expended by chance and which is not expected to occur by chance[9].

2. ACS (Average Common Substring) - is based on the longest common words between two sequences[19]

3. Compression distance - considers the smallest size of program permitting to generate a sequence

4. The $k$-word distance - considers vectors of the word matches in the genome

The biological pattern approach described in [8], which is at the basis of the tool presented here, is close to $k$-word distance, more details are given about the different distances that can be found in literature. Suppose that we have built the following vectors $X = (X_1, X_2...X_n)$ and $Y = (Y_1, Y_2...Y_n)$ from pattern matches of two seqeunces, the distance between these vectors may be defined as[16]:

- method based on word frequencies, where words of fixed k-length (also variable) are searched for matches in the proteome. For instance:

    1. Euclidean distance, defined as:

$$d(X, Y) = \sqrt{\sum_{i=1}^{n} (X_i - Y_i)^2} \tag{2.1}$$

    2. Jaccard distance: let $A$ and $B$ be two sample sets, the Jaccard distance between sets $A$ and $B$ is defined as follows:

$$J(A, B) = 1 - \frac{|A \cap B|}{|A \cup B|} \tag{2.2}$$

3. Chi-square distance is defined as:

$$d(X,Y) = \frac{\sum_{i=1}^{n} \frac{(X_i - Y_i)^2}{X_i + Y_i}}{2} \tag{2.3}$$

- Weighted Euclidean distance is defined as:

$$d(X,Y) = \sqrt{\sum_{i=1}^{n} c_i (X_i - Y_i)^2} \tag{2.4}$$

- method based on correlation structure, where the correlation between frequencies of pattern matches is taken in consideration. For instance:

1. Linear Correlation Coefficient is defined as:

$$d(X,Y) = \frac{n \sum_{i=1}^{n} f_i^X \cdot f_i^Y - \sum_{i=1}^{n} f_i^X \cdot \sum_{i=1}^{n} f_i^Y}{\sqrt{n \sum_{i=1}^{n} (f_i^X)^2 - (\sum_{i=1}^{n} f_i^X)^2)} \cdot \sqrt{n \sum_{i=1}^{n} (f_i^Y)^2 - (\sum_{i=1}^{n} f_i^Y)^2)}} \tag{2.5}$$

$$f_i = \frac{X_i}{\sum_{j=1}^{n} X_{i,j}} \tag{2.6}$$

- method based on covariance, where the covariance between number of pattern matches is considered, for instace: Mahalanobis distance is defined as:

$$d(X,Y) = (X - Y)^T \cdot S^{-1} \cdot (X - Y) \tag{2.7}$$

where:

- $S = [s_{ij}]$ represents the covariance matrix of motif matches
- the $X^T$ represents the transpose vector of $X$

- Kullback-Leibler[2] (KL) discrepancy, measures relative entropy between two discrete probability distributions $f^X$ and $f^Y$ and is defined as:

$$KL(X,Y) = \sum_{i=1}^{n} f_i^X log\left(\frac{f_i^X}{f_i^Y}\right) \tag{2.8}$$

- Angel metrics is defined as:

$$d(X,Y) = -ln\left(\frac{1+\cos\left(\frac{X^T \cdot Y}{\|X\| \cdot \|Y\|}\right)}{2}\right) \tag{2.9}$$

**Computing pairwise distances and Building the tree**

The pairwise distance between vectors is needed because in this way we can define distance between species. Once the pairwise distances are defined we can build the matrix distance. Distance matrix $M[i,j]$, is a matrix where each cell $c_{ij}$ in it, contains the pairwise distance between row $i$ and column $j$. We have to build the distance matrix because we can use Neighbor-Joining (NJ) method [12] implemented in PHYLIP pacakage [11] to build the tree in Newick format.

**Finding a good reference**

In order to evaluate the goodness of an approach a reference tree is used in phylogenetic tree reconstruction. This tree is built by defining a simple evolutionary model[9] or it is compared to other methods that are known for the good results they offer. In this thesis we used as a reference tree the tree available from NCBI taxonomy (site: http://www.ncbi.nlm.nih.gov/taxonomy). This tree is built considering both methods in phylogenies: molecular sequencing data and morphological data. For this reason the tree result is very good and commonly used as a reference tree.

[2]it is not a metric distance

# Chapter 3

# Tree from Biologically significant Patterns (TBP) tool

A basic version of the biological pattern approach to phylogenetic reconstruction, described in the previous chapter, has been implemented in a tool. This tool, that we will refer to as TBP (Tree from Biologically-significant Patterns), has been developed on the Unix philosophy: building short, simple, clear, modular, and extendable code that can be easily maintained. This philosophy offers a lot of flexibility in programming large scale software. It makes the implementation process easier and debugging quicker. TBP software has a friendly user-interface. The user only needs to define the scientific names of the species as defined at NCBI Taxonomy database [17] [3], the tree building options in *Neighbor-Joining* method[12] and *drawtree* in PHYLIP package [11]. The tool will then compute the phylogenetic tree based on the distance between vectors that describe the species in terms of biological patterns, according to the chosen distance. In order to test the goodness of the result, the tool will also build a tree for the same set of species, based on the NCBI taxonomy, which can be consider as a reference.

In the following sections we will first give a short overview of the TBP architecture, then we will describe the steps of its pipeline in more details.

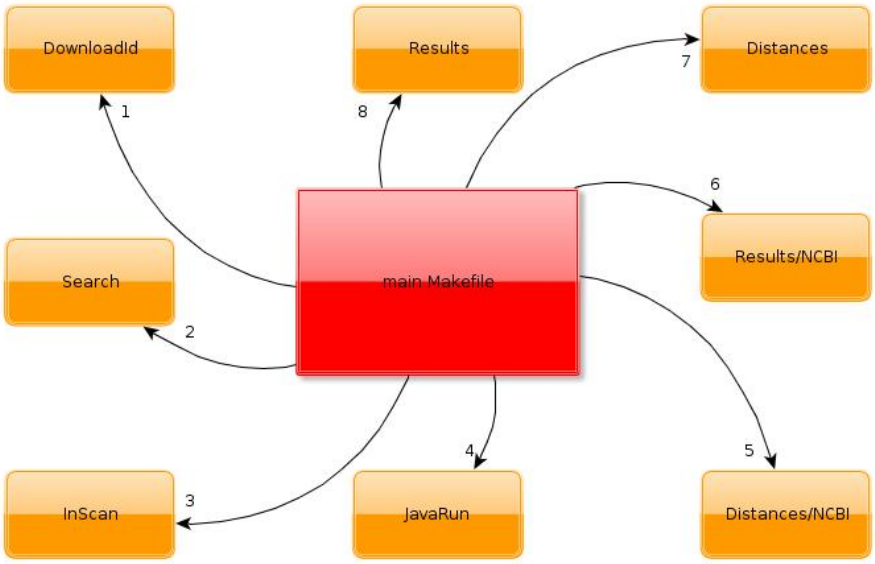


Figure 3.1 – Data flow diagram based on the main *Makefile*

## 3.1 TBP architecture

As a first step, the user should specify the (sub)set of species for which he/she wants to build the phylogenetic tree. The online or offline option must be indicated. Then the user needs to set some parameters: which distance should be used and whether the computation of the reference tree must be performed online or offline. Once all parameters are entered, the distance matrices for both the pattern-based approach and the reference-tree are built. Then they are given in input to the PHYLIP package[11], that will use the neighbor joining method [12] to build the trees. To visualize the results, the drawtree inteface of PHYLIP will be used.

The data flow of the software is managed by a main *Makefile* that calls in a specific order a series of other *sub-makefiles*. Figure 3.1 shows the flow-chart diagram that explains how the data flow runs.

The numbers close to each rectangle refer to the call order made by the main *Makefile*. The names in the rectangles are the original names of the folders in the TBP's package.

TBP software makes use of other software and/or packages. The user should install the required software before running TBP. These are:

1. Java (site: http://www.java.com/)
2. Perl (site: http://www.perl.org/)
3. Python (site: http://www.python.org/)
4. PHYLIP (site: http://evolution.genetics.washington.edu/phylip.html)

*Makefiles* were used to resolve the problem of putting in a single pipeline packages that are written in different programming languages.

## 3.2 Built-in datasets

In this section we overview the databases that need to be available to run TBP.

### 3.2.1 Proteomes

In the long term, TBP aims at allowing phylogenetic analysis for large sets of proteomes available at the NCBI site. However, in this very first implementation, only bacteria proteomes were used. The proteomes were downloaded from NCBI genome database (site: ftp: //ftp.ncbi. nlm.nih.gov/ genomes/ Bacteria/). The file *all.faa* contains the whole proteomes of all the bacteria sequenced so far. However, since there might have been several assembly projects for a single species, in the corresponding folder there might be repetition of data, or presence of plasmids, etc. This introduced the problem of how to select only the proteome files.

To overcome this problem we used the *all.Glimmer3* folder from NCBI genome database. The first line of each file in *all.Glimmer3* contains information about the NCBI Accession number and the kind of file it is: proteome, plasmids etc. Using

this information only the sequences that had a whole proteome were considered. A database of taxid corresponding NCBI Accession number is kept for fast mapping.

### 3.2.2 NCBI Taxonomy

For comparison purposes, beside the tree based on biologically significant patterns, the tool will also generate the phylogenetic tree based on the NCBI taxonomy database (ftp:// ftp.ncbi.nih.gov/ pub/ taxonomy/), for the same set of species. For this reason, in the offline option, the following files from *taxdump* folder are used:

1. nodes.dmp, from this databse TBP is able to get for each taxid the corresponding parent taxid

2. names.dmp, this database is used for matching names of the species with their corresponding taxid

### 3.2.3 PROSITE

The database chosen for this first implementation of TBP is PROSITE [18]. In particular, the tool uses the following data and tool avalable from the PROSITE web site (http://prosite.expasy.org/).

- Prosite pattern, they are 1308 protein motifs (Septeber 2013) that are already published in the literature

- prosite_scan.pl, is a tool provided by Prosite to scan protein motifs in a protein[6]

## 3.3 Input validation

The first step in the data flow is the call made by the main *Makefile* to the *submakefile* in the *DownloadId* folder. This is where the data introduced by the user are elaborated. The steps are:

- Check if the names of the species that have been given in input are correct. In case a name is not correct, a warning message will be displayed, and this name will not be included in the following steps.
- The filtered names are then converted to their corresponding taxids of the NCBI taxonomy database (site: http://www.ncbi.nlm.nih.gov/taxonomy).
- The taxids will be converted in NC Accessions numbers. This file will be used in searching for the right proteomes in the following steps.

## 3.4 Building the tree from biological signals

### 3.4.1 Prosite scanning

By using a list of all NC accession numbers, TBP can search in *all.faa* folder for the proteome files. The *find* Unix command is used to take care of this step, and will copy the file in a single folder. This is going to be the "basket" from where the *prosite_scan.pl*[6] will take files to scan for the 1308 Prosite patterns. Since searching for patterns matches in different proteomes is an independent task, the user can run the process faster by scanning the proteomes simultaneously. To do this, one has to run the *makefile* with the *-j* option followed by a number. This value indicates the number of jobs you wish to be run simultaneously by *makefile*. This will take advantages of multi-core processors. If one wishes not to limit the number of simultaneous jobs that *makefile* can do, then, the *-j* option must not be followed by a number.

Each *protein* in the proteome is scanned and a vector of 1296 cells is built. Twelve common patterns of Prosite are not included in the vector cells. This is, because they are short and commonly found in protein sequences, so counting them would introduce noise. The cells of the vector contain the number of matches for every pattern. To calculate a vector of a proteome these steps are followed:

1. For each protein $p$ in the proteome $Pm$ the vector of pattern matches $v_p$ for $p$ is built

2. $v_p$ of the proteome $Pm$ are summed up to give a single vector of pattern matches

Note that the order in which proteins in a proteome are scanned is not important. In some recent articles such as [2] the proteins are ordered within the proteome before searching for protein motifs. In fact they do also considers the matches between two consecutive proteins. TBP does not find matches between consecutive proteins, since we believe that finding pattern matches between proteins has no biological meaning.

### 3.4.2 Computing proteome distances

In phylogeny reconstruction several distance measures have been used for frequencies of common motifs, as defined by Guyon[9] (explained in Chapter 2).

TBP was tested with three such distances: Euclidean distance, Euclidean distance of normalised vectors and Jaccard distance. While Euclidean distance calculate the pairwise distance between vectors based on the number of occurrences, the Euclidena distance of normalised vectors, calculates the pairwise distance based on the frequencies of the patterns. On the other hand the Jaccard distance calculate the pairwise distance based on the presence/absence of the patterns. Each of these distances suggest a different biological insight.

As explained in the Chapter 2, once we have the pairwise distance between all vectors, the distance matrix $M$ can be built, so that $M[i, j]$ holds the distance between species $i$ and species $j$.

### 3.4.3 Building the tree

The distance matrix $M$ is given in input to the PHYLIP package that will build a binary phylogenetic tree based on the Neighbor Join algorithm.

## 3.5 Building the reference NCBI tree

In order to test the goodness of the tree built on the biologically significant signals, we need a reference tree to compare too. For this purpose we chose as a ground-truth the NCBI taxonomy, and derive a tree from it. In particular, we will follow two steps: we will first build a generalized tree on the available taxonomy restricted to the input set of species specified by the user. Then we will transform that tree in a binary tree to allow for comparison with the tree generated on biological signals.

### 3.5.1 Building the generalized tree

Two options are offered to calculate the reference tree of a small or of a large set of species:

1. Online option. A script in Perl provided from iTOL will use batch access mode to make a call to iTOL sever(site: http://itol.embl.de/)[13] to calculate the NCBI tree from the the taxids given in input. This option was introduced because it is faster than the other option for a small set of species in input. Since the iTOL has already uploaded the data that are needed to calculate the trees in their server.

2. Offline option. The file *nodes.dmp* is first uploaded in the memory and then the NCBI tree is calculated. This option is provided because the software can run faster locally when experiments with a large list of species are required.

The output will be a phylogenetic tree, for the specified set of species, in Newick format and based on the NCBI taxonomy.

### 3.5.2 Building the binary tree

The NCBI tree can be considered as an additive tree. This allows to define the distance between two nodes as the number of nodes in the path from one node to the

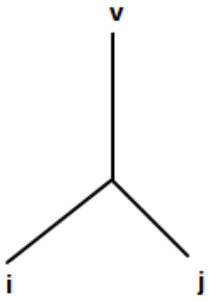


Figure 3.2 – A schematic representation of distances in tree a $T$ where $i$ and $j$ are the descendent of the node $v$

other.

The matrix $D$ induced from the NCBI tree can be built from all the pairwise distances. It is possible to convert an additive tree into a ultrametric tree [5] using the following:

**Lemma.** Let $T_1$ be the additive tree for matrix $D_1$, as defined in *Background* chapter, and $T_2$ be the ultrametric tree for matrix $D_2$. Without knowing $T_1$ or $T_2$ explicitly, we can deduce that $D_2 = m_v + (D_2(i,j) - D(v,i) - D(v,j)) \div 2$, where:

1. $i$ and $j$ are the descendent from the node $v$
2. $m_v$ is the maximum distance between two nodes of the trees
3. $v$ is one of the two nodes which distance is $m_v$

This Lemma suggests that we do not need calculate the tree, but we can work directly with the matrices. Hence, we will use it to obtain a distance matrix of a

binary tree to be given in input to the PHYLIP package.

## 3.6 Observations

- All the calculations were made using taxids and not specie's names. The reason for this choice is that *node.dmp* file of NCBI provide information about a given node and its parent based on taxids only. Moreover, not only *node.dmp* in the offline mode uses taxids, also the batch access mode script that makes a call to iTOL server must contain only the taxids of the species. Having a phylogenetic tree with taxids in its nodes makes it hard to read. A standard approach to solve this problem is to introduce name's abbreviations.

- Neighbor-Joining (NJ) method [12] implemented in PHYLIP pacakage [11] is used to calculate the Newick tree format from the distance matrix given in input. The output will be in Newick tree format thus including branch lengths. We used the drawtree in PHYLIP to draw the tree. In the output tree we did not include the branch length, since it is believed that they have no evolutionary insight in this kind of experiment.

# Chapter 4

# Testing

At the end of the computation two trees will be presented to the user. The NCBI tree converted in binary and the tree obtained with the proposed method. This is the first version of the software, and for this tool only some preliminary tests have been done. Complete assessment requires a fair amount of experiments, and comparison with the state-of-the-art, which will be the subject of a separate study.

## 4.1 Experimental Results

TBP is relatively slow. It takes about 20 minutes, in a PC of 2.00 GHz (x2) dual-core processor, to build both phylogenetic trees, the one based on the proposed method and the other one from NCBI. The bottleneck is the scanning with the prosite tool [6] which takes more than 80% of the running time. In the future better solutions could be plug-in the tool to speed up this step[7].

We run the experiments for three sets of species of different size. As pairwise distance between vectors that represent proteomes we tried Euclidean distance of pattern frequencies and Jaccard distance. We do not present the results of the simple Euclidean distance here because they were really poor.

For each experiment the following trees are shown: NCBI tree, NCBI binary tree,

Euclidean distance of normalised vectors tree and Jaccard tree. Trees have colour dots close to species names in order to make easier the comparison between the trees. The group of species that have the same colour, are the ones that, in the NCBI tree, are children of the same parent. This means, they share a close common ancestor.

### 4.1.1 First test

For the first test a set of 33 species were randomly chosen. The NCBI tree is shown in Figure 4.1 and the NCBI binary tree is shown in Figure 4.2. One can appreciate the power of converting an Additive tree into an Ultrametric tree by following the colors of the nodes associated in Figure 4.1 and Figure 4.2. The tree calculated with the biologically significant pattern based method, using Euclidean distance of normalised vectors is shown in Figure 4.3 and the tree calculated with the Jaccard distance is shown in Figure 4.4.

We first use Euclidean distance of normalised vectors to measure the distance between proteomes. This is a well studied studied method in reconstructing phylogenetic trees from the whole-genomic approach. Euclidean distance approach behave badly when it tries to compare proteomes of very similar species[16]. This, is confirmed by our results as well. Pink and yellow dots in Figure 4.3 are an example that confirms this statement.

We also tried the Jaccard distance. This distance could manage to "fix" the bad nodes (in the Euclidean distance) from very similar proteomes. But, it also kept the relatively good positions of the other nodes in it. The pink and yellow dots that were far in Figure 4.3 now are close in Figure 4.4. Not only the dots of the same colour share the same closest ancestor, but they also share ancestor that is relatively far. For instance, the dots in pink, yellow and brown are relatively close to each other in Figure 4.4 as they are in Figure 4.1.

As a standard approach in reconstruction phylogenetic trees the Robinson and Foulds (RF) distance [10] of the trees calculated in this experiment are given. They

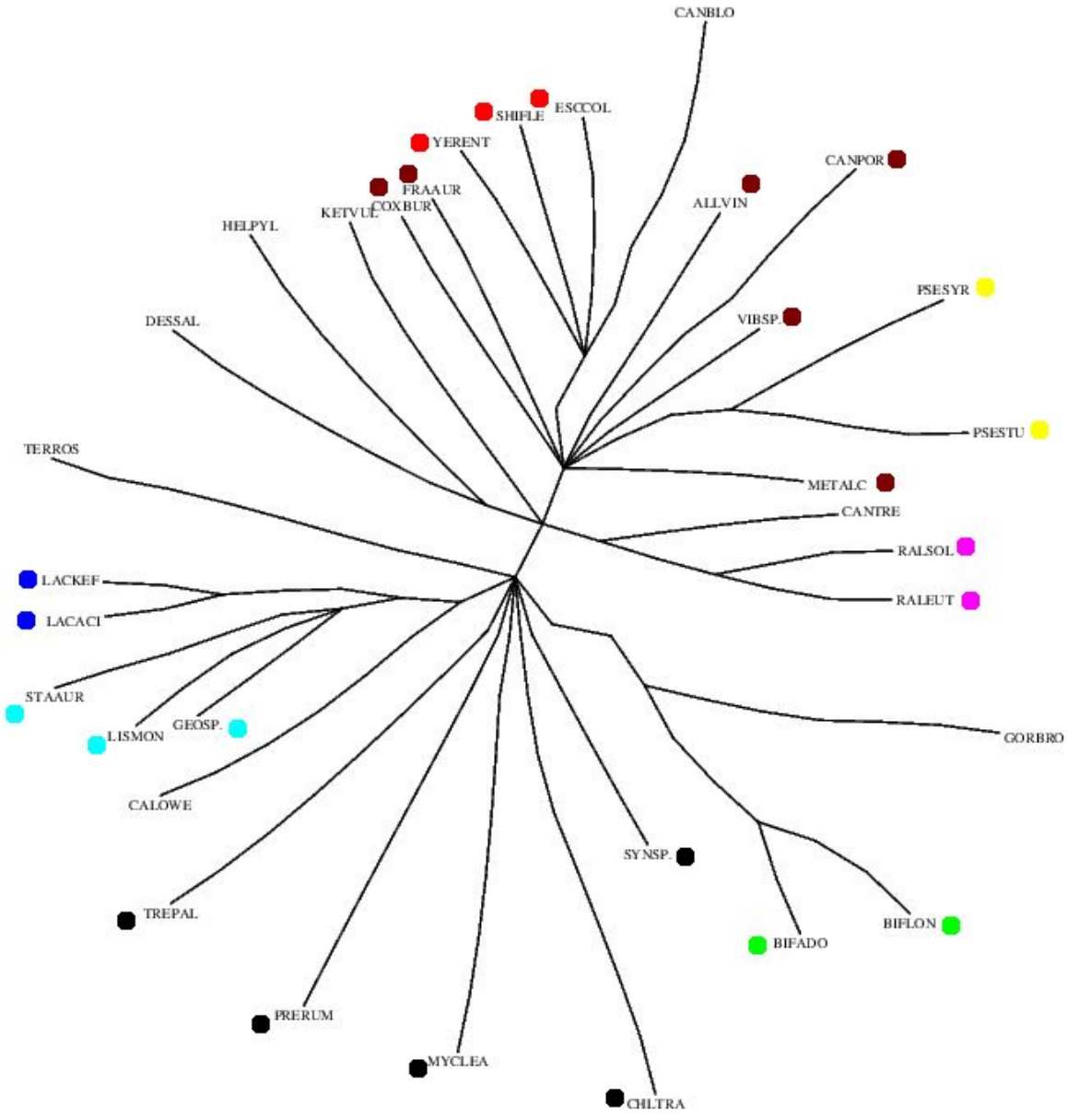


Figure 4.1 – The NCBI tree of the first test

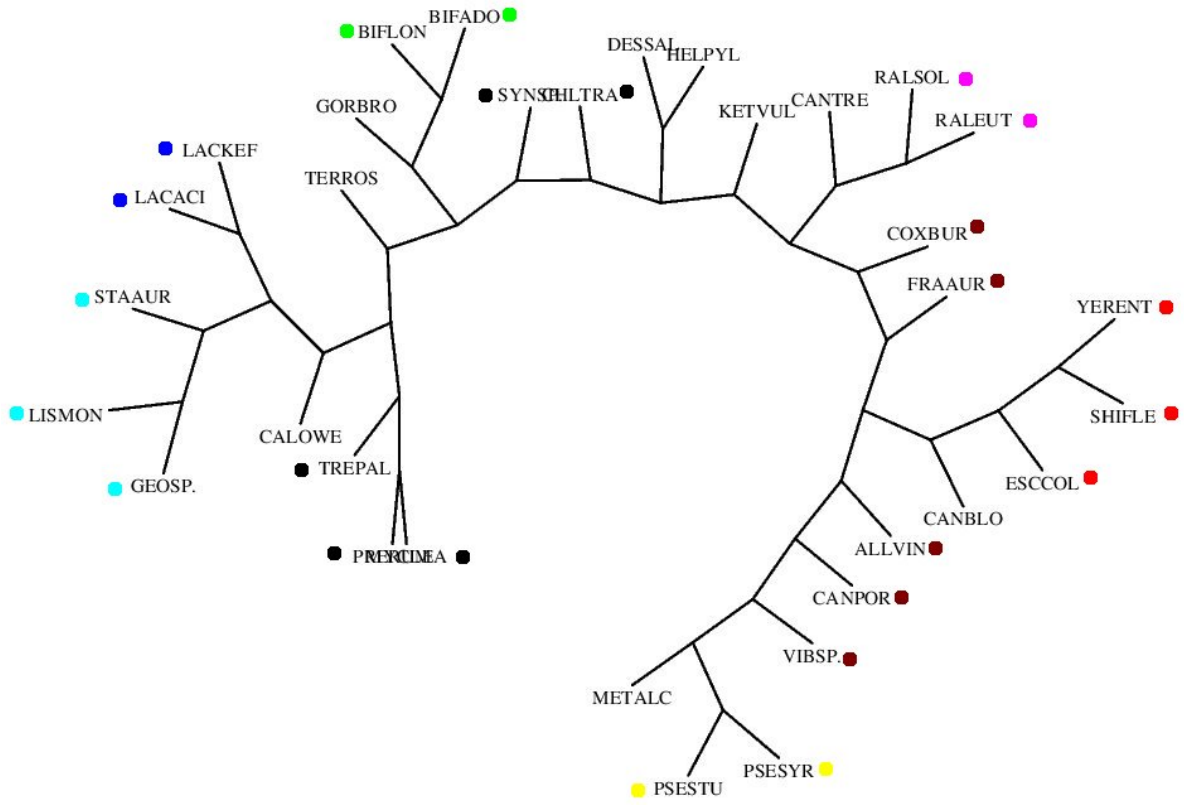


Figure 4.2 – The NCBI binary tree of the first test

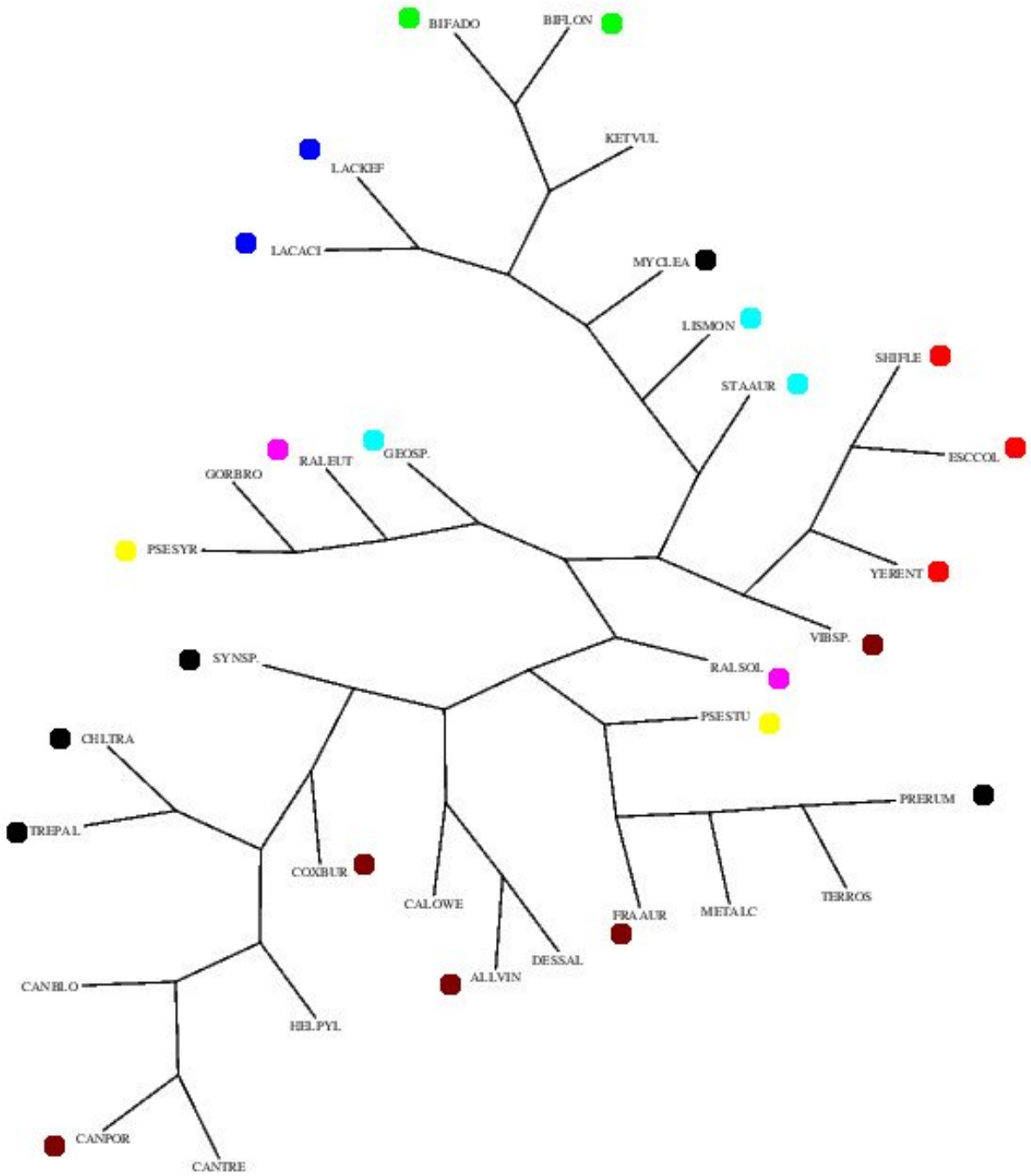


Figure 4.3 – TBP tree calculated by using the Euclidean distance of normalised vectors of the first test

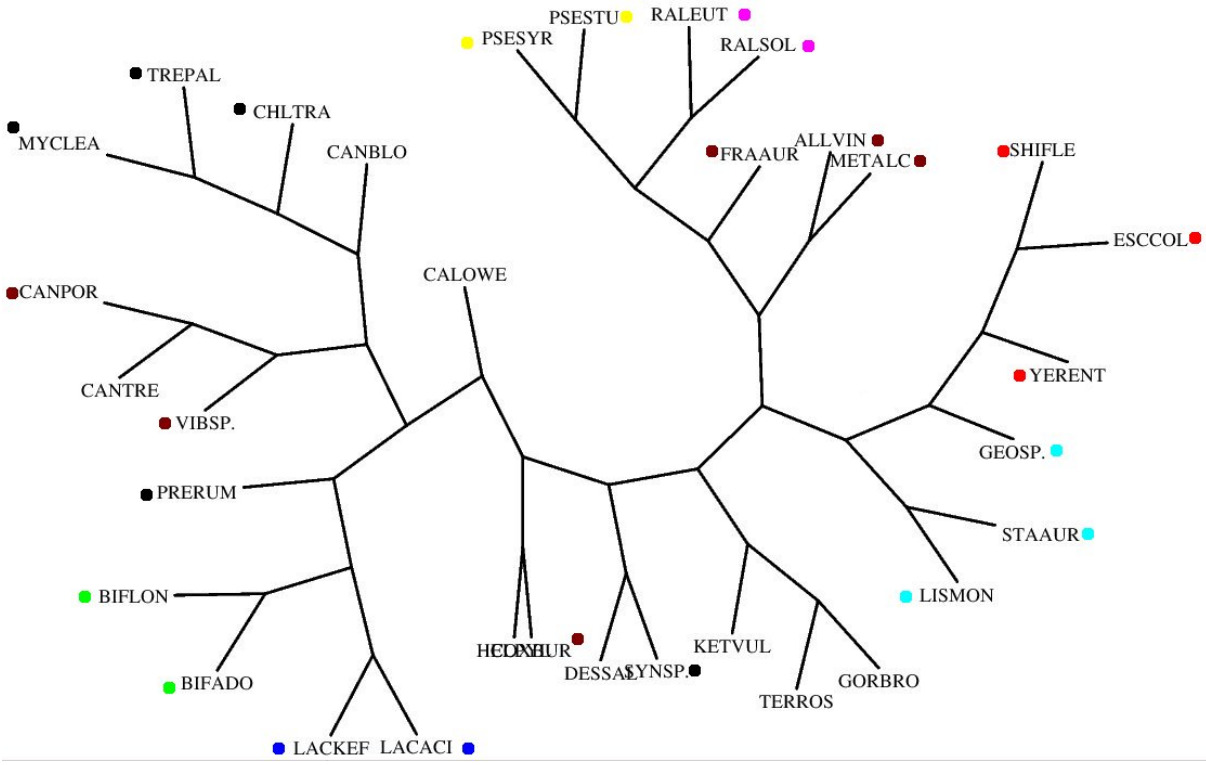


Figure 4.4 – TBP tree calculated by using the Jaccard distance of the first test

were calculated using *treeidst* in PHYLIP package. The distance between binary NCBI tree and Euclidean distance is 54. On the other hand better distance, as expected, is given from *treedist* while comparing binary NCBI tree with the Euclidean distance of normalised vectors. The distance is 50.

The RF (Robinson and Foulds) distance can show which one of the methods is better, but it has no immediate statistical interpretation. We cannot say whether a larger distance is significantly larger than a smaller one.[1] The RF distance can range from 0 to twice the number of internal branches, so that for n species it can be as large as 2n-6 (for 3 species or more).

### 4.1.2 Second test

For the second test a set of 46 species were randomly chosen. As in the section above the following trees are given: NCBI tree in Figure 4.5; NCBI binary tree in Figure 4.6; Euclidean distance of normalised vectors in Figure 4.7 and the Jaccard distance tree in Figure 4.8.

In this experiment the same statements as above are confirmed. The Euclidean distance of normalized vectors in Figure 4.7 lack in grouping together species that have the same common ancestor in Figure 4.5. For instance, the dots in pink and blue. Better results are achieved for with the Jaccard distance in Figure 4.8.

The RF distance between binary NCBI tree and Euclidean distance of normalised vectors is 72, and the distance between binary NCBI tree and Jaccard distance is 66. Once again the Jaccard distance was confirmed as a better distance.

[1]site: http://evolution.genetics.washington.edu/phylip/doc/treedist.html

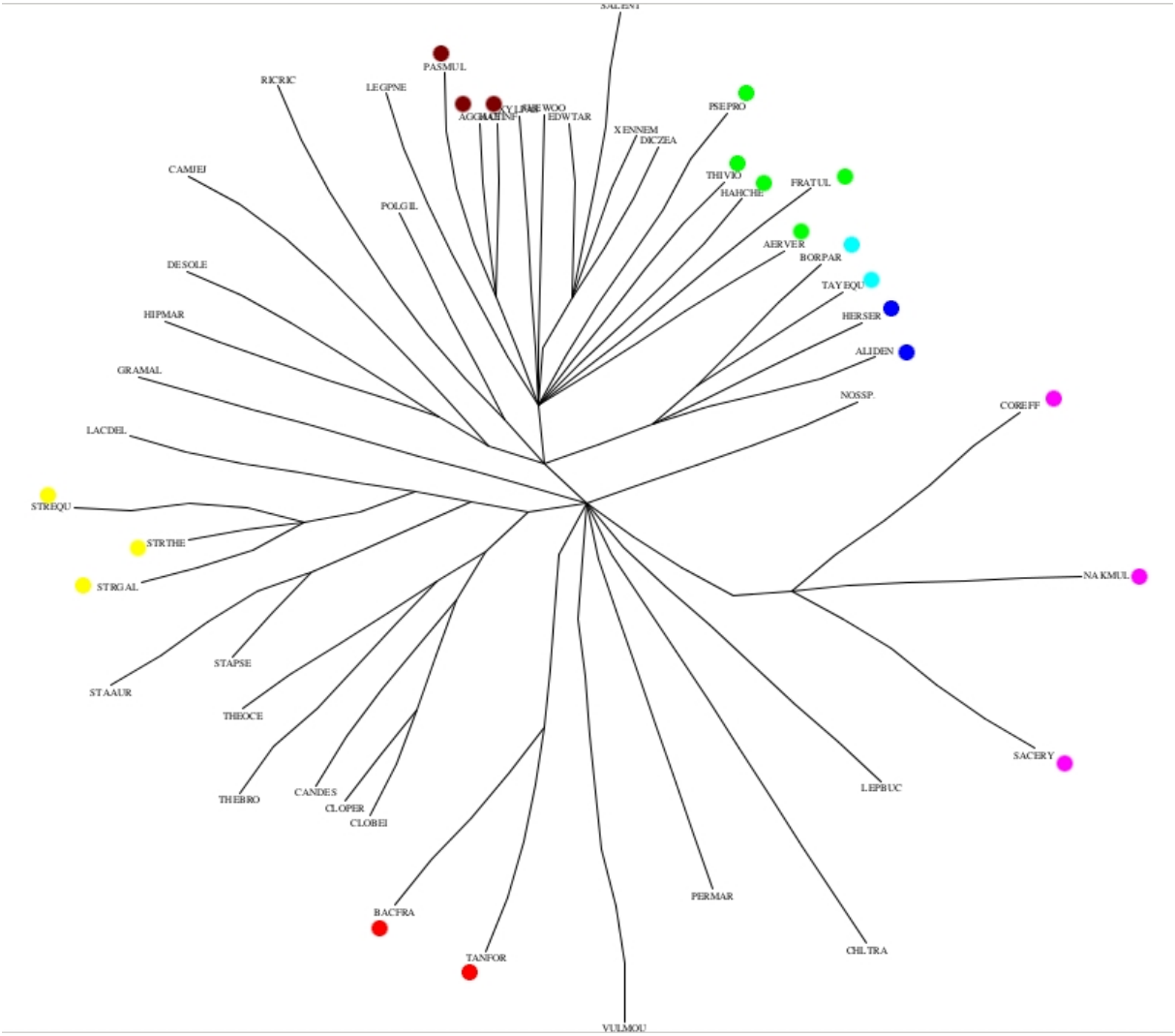


Figure 4.5 – The NCBI tree of the second test

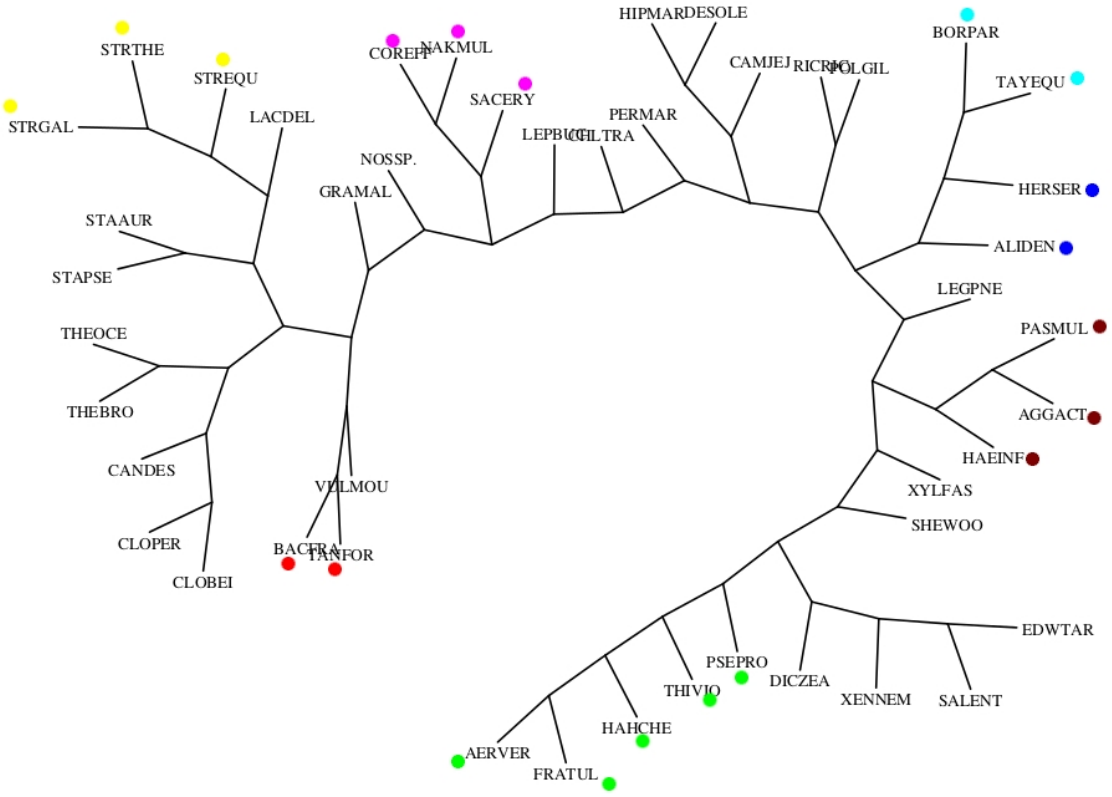


Figure 4.6 – The NCBI binary tree of the second test

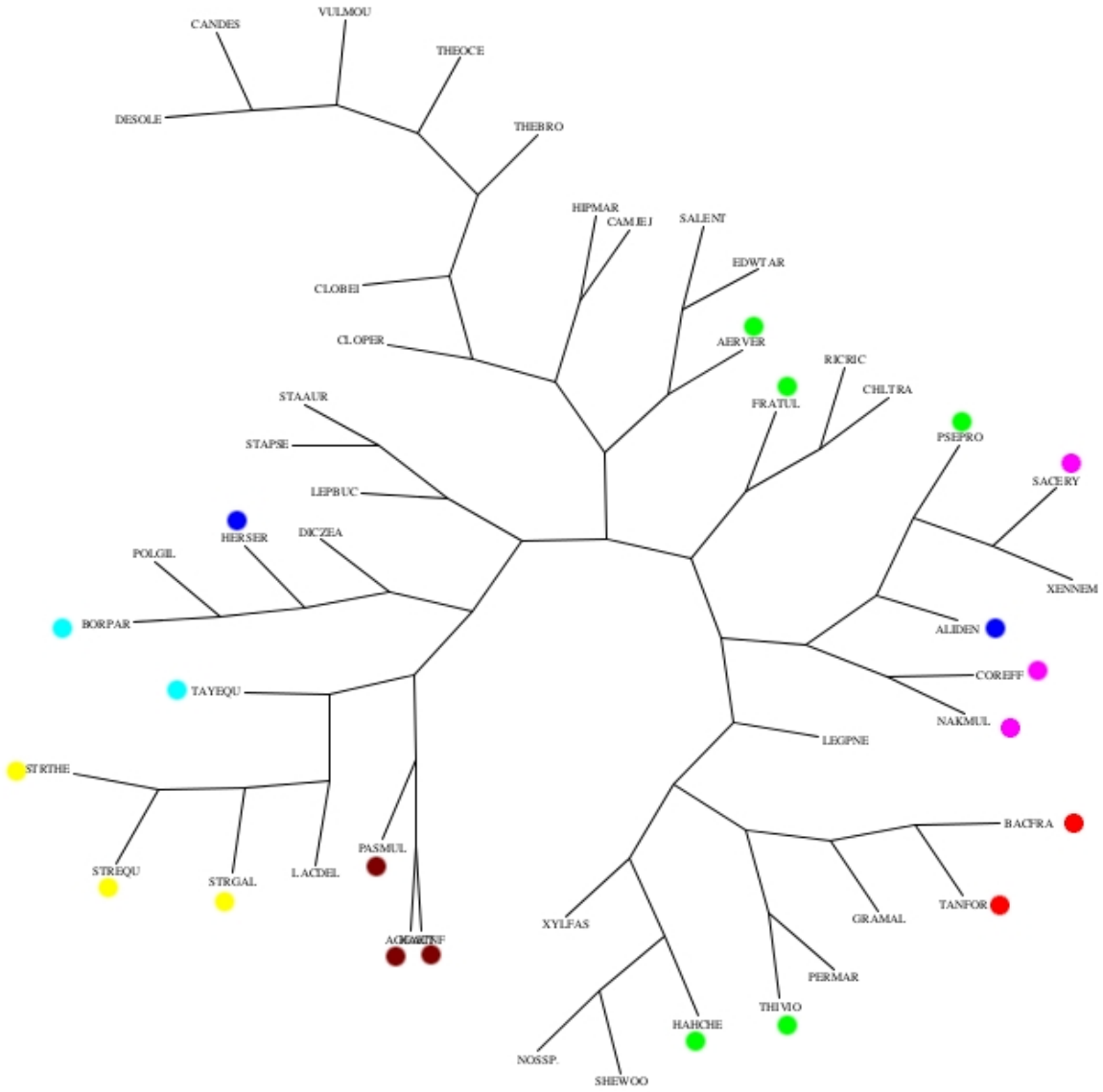


Figure 4.7 – TBP tree calculated by using the Euclidean distance of normalised vectors of the second test

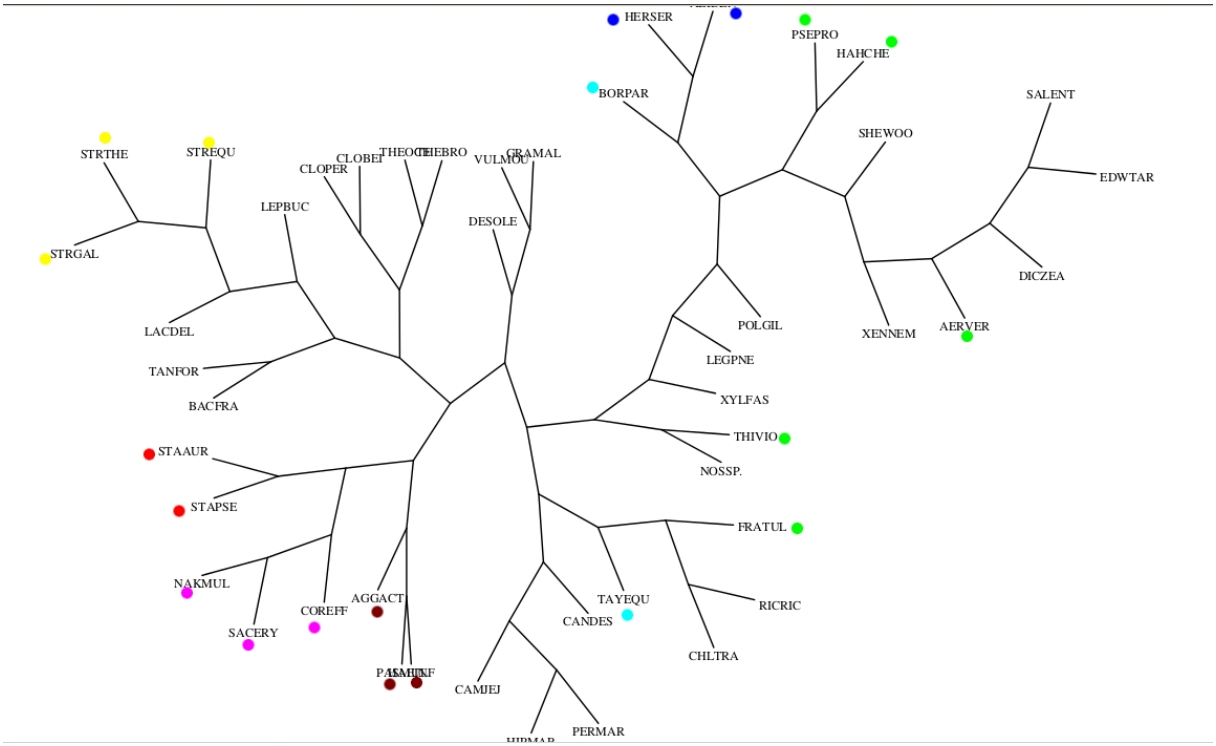


Figure 4.8 – TBP tree calculated by using the Jaccard distance of the second test

### 4.1.3 Third test

A set of 66 species were randomly chosen for the third test. This is the biggest set presented in this thesis, because it is difficult to visualise the tree for trees bigger than that set. As in the section above the following trees are given: NCBI tree in Figure 4.9; NCBI binary tree in Figure 4.10; Euclidean distance of normalised vectors in Figure 4.11 and the Jaccard distance tree in Figure 4.12.

This experiment, having a bigger set of species, shows how good can be the tested distances in grouping together species that differ from a far ancestor. Only a part of the reference tree was used. The species chosen in Figure 4.10 are associated with a colour dot.

One can easily notice the difference between Figure 4.11 and Figure 4.12. While the tree in Figure 4.11 does not group together the chosen species, the tree in Figure 4.12 does. For instance, the dots in red and black are far away from the other species

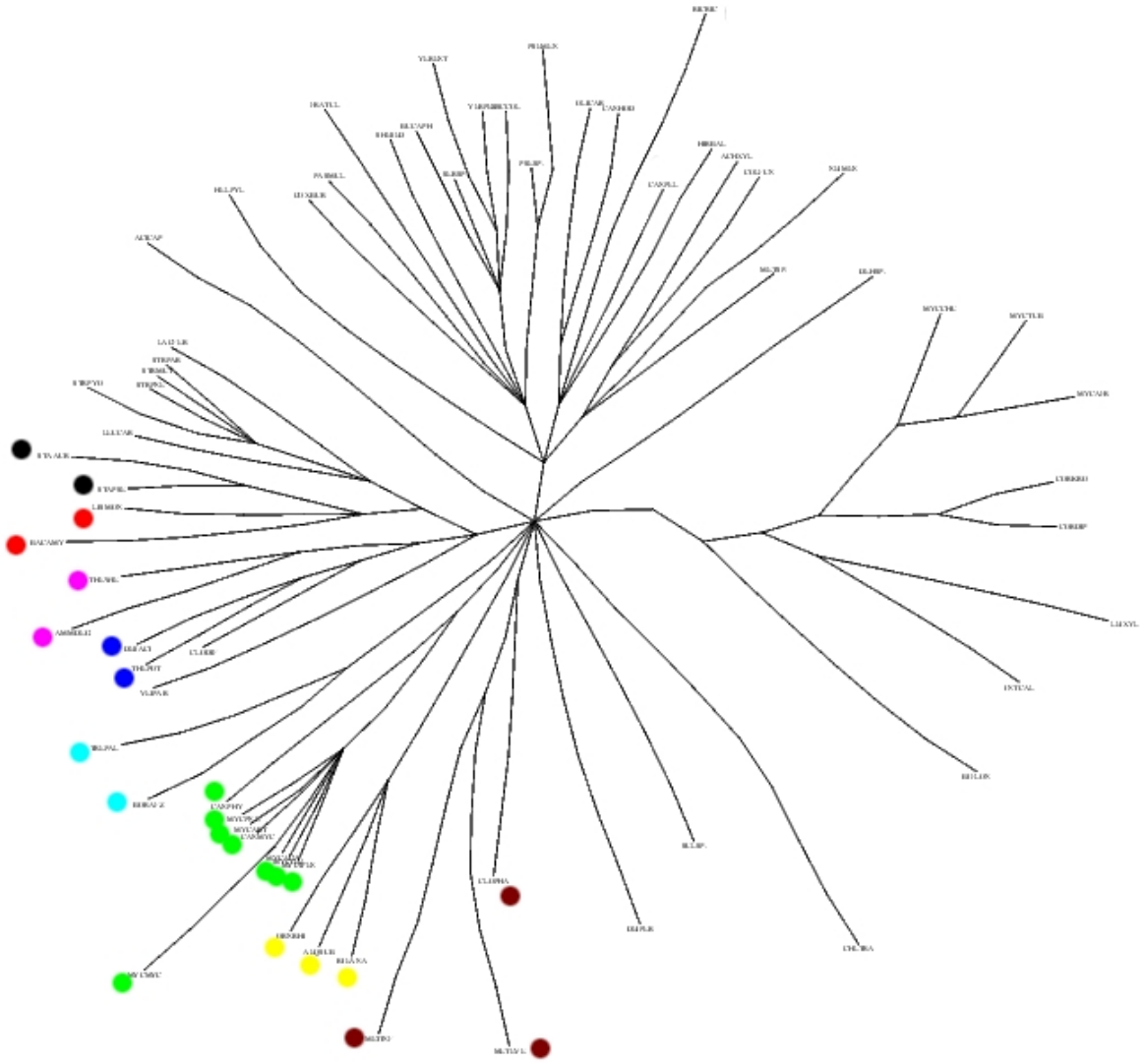

Figure 4.9 – The NCBI tree of the third test

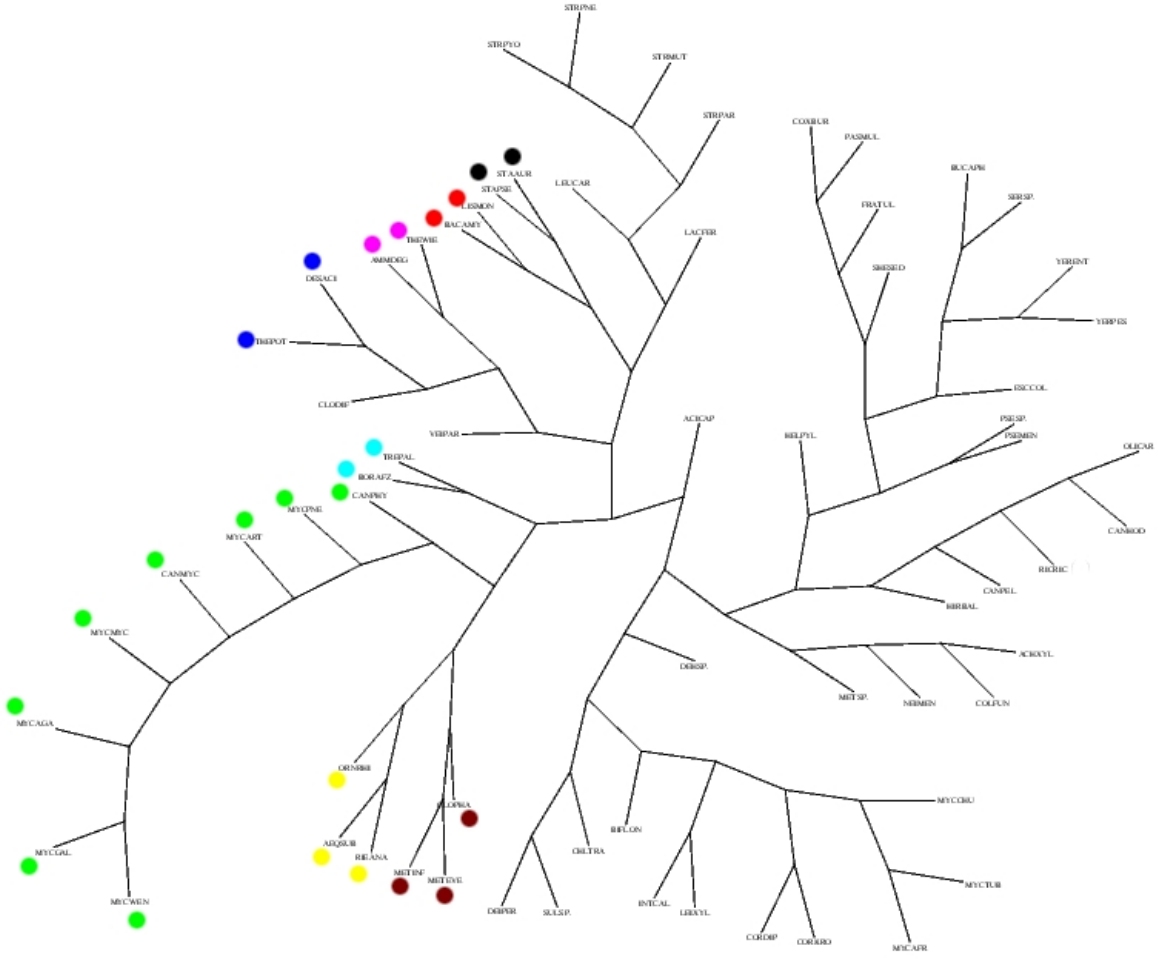


Figure 4.10 – The NCBI binary tree of the third test

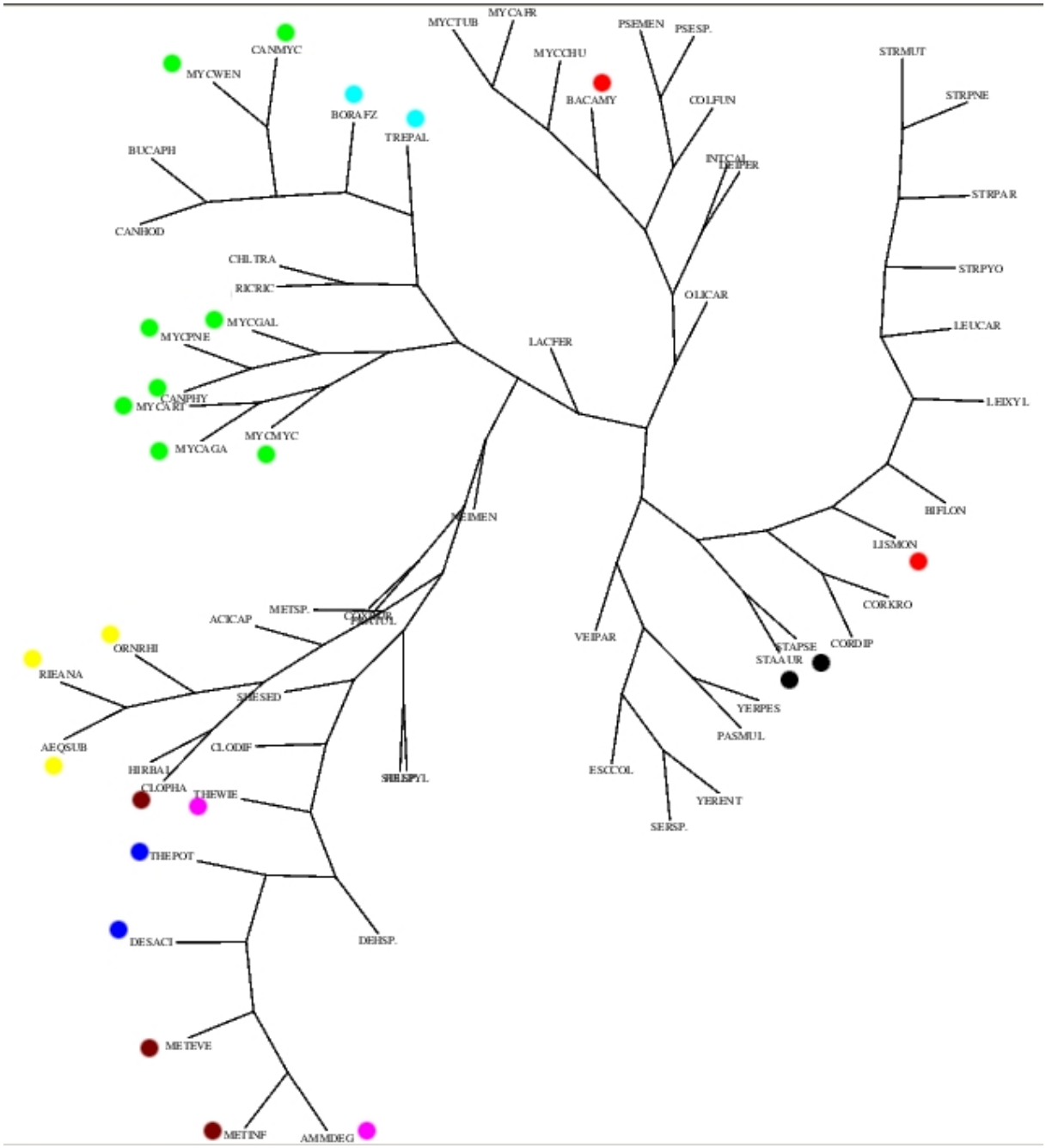


Figure 4.11 – TBP tree calculated by using the Euclidean distance of normalised vectors of the third test

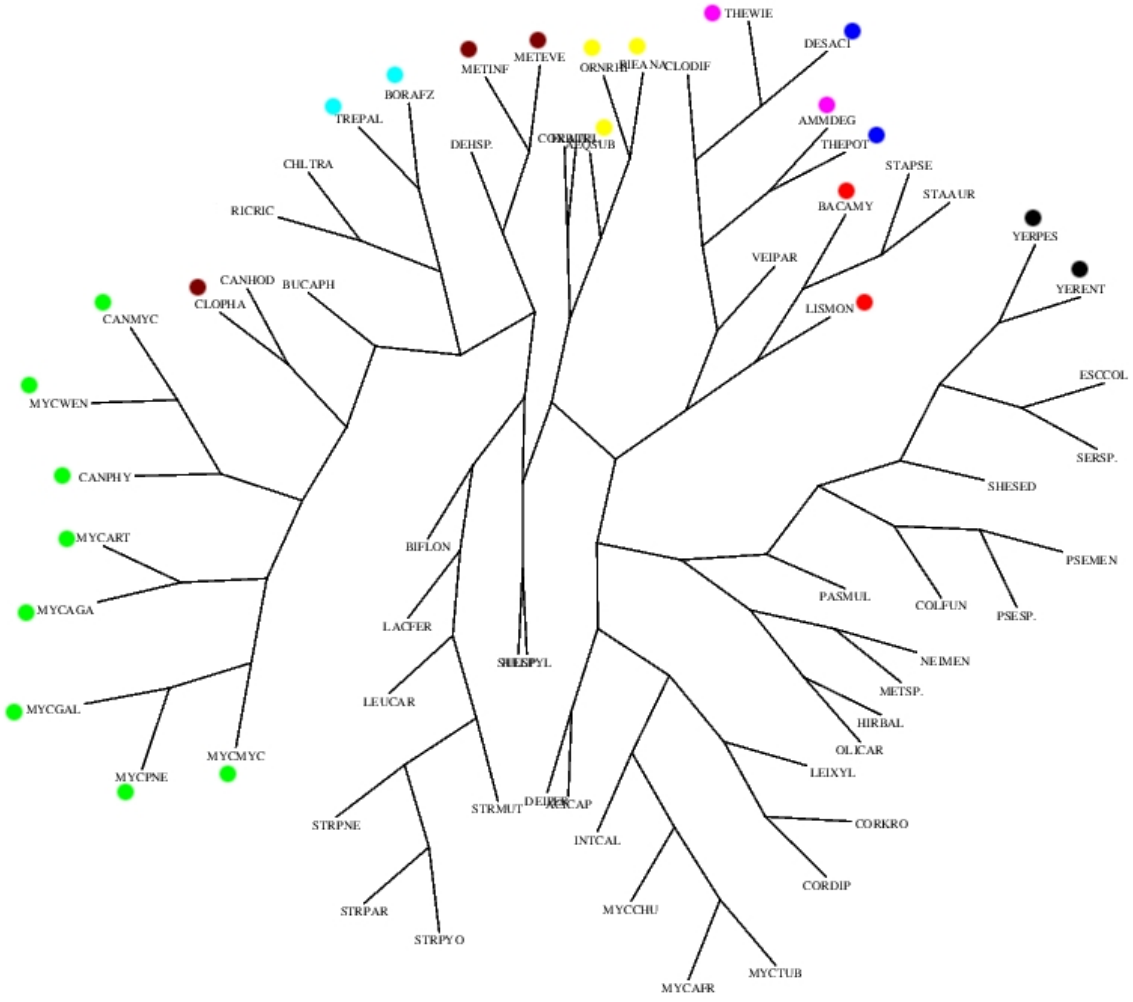


Figure 4.12 – TBP tree calculated by using the Jaccard distance of the third test

considered in Figure 4.11. This shows that Jaccard distance can give better results in grouping together species that differ from a far ancestor.

The RF distance between binary NCBI and Euclidean distance of weighted vectors is 100 and the distance between binary NCBI tree and Jaccard distance is 82.

## 4.2 Discussion

TBP is a software built on other software and data. This means that its accuracy depends strongly on the quality of the both mentioned. The software accuracy were tested before used, but the quality of the data (proteomes) were not. This may introduce a problem. TBP searches for proteomes in the *all.faa* files from the NCBI. This means that the number of matches strongly depend on the quality of the proteomes.

No filter or trigger for the minimum number of matches is used. This means that, every match that is found in the proteome is counted. This may introduce some noise. All experiments are done by using the option of *not* considering the common patterns in Prosite. This partially alleviated the problem. But still there is some noise in the data due to patterns that are found by chance. In case it happens for the case of Euclidean distance of normalised vectors, its influence will be lower given its lower number of false matches.

In the case of Jaccard distance they do in fact influence the distance. The vector used for the Jaccard distance is a binary vector that reveals the presence or absence of a pattern. Even in case there is a single false match, this is going to be counted as well. A propriety of dataset in the Jaccard distance that filter these false matches will give better results. This may be done by filtering the number of pattern matches while building proteome vectors, or by using a trigger value higher than one in the Jaccard distance.

Simple Euclidean distance was also tested with TBP. The results were poor. Given the fact that different species (bacteria in this case) have different length of proteomes,

this measure introduces a problems when defining the significance of a pattern in its species. For instance, 10 matches of a pattern in a relatively short proteome are more significant than 10 matches of the same pattern in a relatively long proteome. Weighting the number of matches by the length of the proteome gives more importance to a pattern in this proteome. More biological meaning can be gain from it.

TBP with the Jaccard distance offers a relatively good phylogenetic tree. It is very sensitive to very similar proteomes as Figure 4.4 shows. The dots with the same color found close to each other hints that the approach is a promising one. It also seems that the biologically significant pattern based method of TBP looses its accuracy for higher taxonomic groups. This is a common problem to the whole-gnomic approach to phylogenetic inference, as has been reported [15]. For instance, the distance can do very good to group together species sharing only one common ancestor, such as blue, light blue, green dots etc. (compare to Figure 4.1) But it does not do so well when species share a far common ancestor.

# Chapter 5

# Conclusions and future work

## 5.1 Conclusions

The first conclusion of this work is that, it seems possible to build good phylogenetic trees using biologically significant patterns, such as the patterns from Prosite database[18]. It be should kept in mind that once the vector of proteomes are calculated, the distance method is fundamental. Different distances lead to different biological insight.

For instance, defining Euclidean distance means, grouping species together only if they share the same number of biologically significant patterns. The results gave a poor phylogenetic tree. Hence, from our small set of experiments, it seems that there could be no relationship between phylogenies and the number of biologically significant patterns.

Defining the Euclidean distance of normalised vectors means, grouping similar species together based on the importance of their patterns in the proteome. Importance of a pattern here, is defined as a pattern that occur relatively more than others. If two species share the same important patterns, these species will be close in the phylogenetic tree. Since this distance gives a poor phylogenetic tree in output, this suggest that the *frequency* of the patterns do not give meaningful biological insight

to find good phylogenetic relationships between species.

On the other hand, defining a Jaccard distance between proteomes means, grouping similar species together if they share the same presence/absence of patterns in their proteome. Since the tree built with this distance is relatively good, this suggest that the presence/absence of the biological significant pattern may be related to phylogenies.

As it was made clear in the other chapters, the objective of the thesis was to implement a pipeline of analysis in a tool and do some preliminary tests, and that the complete assessment requires a fair amount of experiments that will be the subject of a separate study.

## 5.2 Future work

TBP has a basic pipeline. It was build in order to do some preliminary tests on an alignment-free method based on biologically significant patterns. Since the experiments suggest that relatively good phylogenetic trees can be build from this method, a series of improvements can be done.

TBP pipeline include a series of other software, this may introduce incompatibility or other issues related to installing the required software for the users. Therefore, implementing the whole software in one programming language will eliminate these issues and it will make the software run faster. Another option would be uploading the software in a server where users can use it online.

TBP is implemented to work on Unix environment but it would be better offering this software for Windows OS as well. Introducing a better interface with more options that the user can define, such as: more distances between vectors, comparing it with trees built from other software etc.

While the TBP software runs it also starts to upload in memory the files it needs. TBP upload these files for every run. This would make running the TBP a series of

times inconvenient. In order to faster the software it is fundamental to first upload all the package needed in the memory and then offer to the user the option to run the software.

While other pattern matching software used with proteomes takes in consideration only matches as symbos, TBP offers the chance to find biologically significant matches. This may offer new hypothesis about finding the relationship between a certain distance and its corresponding biological meaning. Different distances should be tested in order to, not only achieve to reconstruct good phylogenetic trees, but also to get more biological information from the results.

# Bibliography


[1] Miller W. Myers E. Lipman D. Altschul S., Gish W. Basic local alignment search tool. *Journal of Molecular Biology*, 215:403–410, 1990.

[2] Caglioti E. Benedetto D. and Chica C. Compressing proteomes: The relevance of medium range correlations. *EURASIP Journal on Bioinformatics and Systems Biology*, 2007, 2007.

[3] Lipman DJ. Ostell J. Sayers EW. Benson DA., Karsch-Mizrachi I. Genbank. *Nucleic Acids Research*, 37:D26–31, 2009.

[4] Fiona S. L. Brinkman and Detlef D. Leipe. Phylogenetic analysis. *Bioinformatics*, 2002.

[5] Gusfield D. Algorithms on strings, trees, and sequences. pages 466–469, 1997.

[6] Gattiker A. Bulliard V. Langendijk-Genevaux PS. Gasteiger E. Bairoch A. Hulo N. De Castro E., Sigrist CJA. Scanprosite: detection of prosite signature matches and prorule-associated functional and structural residues in proteins. *Nucleic Acids Research*, 34:W362–5, 2006.

[7] Szymon G. Emanuele G. and Esko U. Fast match-ing of transcription factor motifs using generalized position weight matrix models. *Journal of Computational Biology*, 20, 2013.

[8] Cunial F. Alignment free sequence comparison with biologically significant patterns. *personal communication.*

[9] Guyon F. and Guenoche A. Alignment free string distances for phylogeny. *Classification as a Tool for Research*, 2010.

[10] Robinson D. F. and Foulds L. R. Comparison of phylogenetic trees. *Mathematical Biosciences*, 53:131–147, 1981.

[11] Felsenstein J. Phylip (phylogeny inference package).

[12] Saitou N. and Nei M. The neighbor-joining method. *Molecular Biology and Evolution*, 4:406–425, 1987.

[13] Bork P. and Letunic I. Interactive tree of life v2: online annotation and display of phylogenetic trees made easy. *Nucleic Acids Research*, 39(suppl 2):W475–W478, 1987.

[14] Ahamadian A Pettersson E., Lundeberg J. Generations of sequencing technologies. *Genomics*, 93, 2008.

[15] Hao B. Qi J., Wang B. Whole proteome prokaryote phylogeny without sequence alignment: A k-string composition approach. *Molecular Evolution*, 58:1–11, 2004.

[16] Vinga S. and Almeida J. Alignment-free sequence comparison-a review. *Bioinformatics*, 19(4):513–523, 2003.

[17] Benson DA. Bryant SH. Canese K.-Chetvernin V. Church DM. DiCuccio M. Edgar R. Federhen S. Feolo M. Geer LY. Helmberg W. Kapustin Y. Landsman D. Lipman DJ. Madden TL. Maglott DR. Miller V. Mizrachi I. Ostell J. Pruitt KD. Schuler GD. Sequeira E. Sherry ST. Shumway M. Sirotkin K. Souvorov A. Starchenko G. Tatusova TA. Wagner L. Yaschenko E. Ye J. Sayers EW., Barrett T. Database resources of the national center for biotechnology information. *Nucleic Acids Research*, 37:D5–15, 2009.

[18] Cerutti L Cuche BA Hulo N. Bridge A. Bougueleret L. Xenarios I. Sigrist CJA., de Castro E. New and continuing developments at prosite. pages 1–4, 2012.

[19] Tuller T. Chor B. Ulitsky I., Burnstein D. Fast match-ing of transcription factor motifs using generalized position weight matrix models. *Journal of Computational Biology*, 13:336–350, 2006.