Introduction to Bioinformatics - Project 2

Władysław Olejnik

December 28, 2022

1 Introduction

Protein clustering and phylogeny are essential tools in the field of bioinformatics, which deals with the analysis and interpretation of biological data. Protein clustering is the process of grouping proteins based on their similarities and differences, while phylogeny studies the evolutionary relationships between different organisms.

This project aimed to analyze eight proteins from different organisms, cluster the sequences, build phylogenetic trees and draw conclusions based on the results obtained.

2 Organisms and proteins

As soon as I saw phylogenetic trees for the first time (when I was still in middle school), I was impressed by what conclusions science is now leading us to. When selecting organisms for the task, I chose similar species and took the risk of analyzing genetically close organisms. I thought it would be interesting to analyze how the methods used would work for primates. The organisms I chose were:

1. Homo Sapiens (eng. Human, pol. Człowiek rozumny)
2. Callithrix jacchus (eng. Common marmoset, pol. Uistiti białoucha)
3. Groilla gorilla gorilla (eng. Western lowland gorilla, pol. Goryl nizinny)
4. Lemur catta (eng. Ring-tailed lemur, pol. Lemur katta)
5. Macaca thibetana thibetana (eng. Tibetan macaque, pol. Makak tybetański)
6. Pan troglodytes (eng. Chimpanzee, pol. Szympans zwyczajny)
7. Papio anubis (eng. Olive baboon, pol. Pawian oliwkowy)
8. Pongo abelii (eng. Sumatran orangutan, pol. Orangutan sumatrzański)
9. Saimiri boliviensis boliviensis (eng. Black-capped squirrel monkey)
10. Sapajus apella (eng. Tufted capuchin, pol. Kapucynka czubata)

As all primates are genetically close organisms, I decided to choose ten of them rather than eight. Then I decided to choose important proteins found in Human organisms:

1. Albumin: Albumin is a protein produced by the liver and is the most abundant protein in human blood plasma. It plays a crucial role in maintaining the osmotic pressure of blood and transporting various substances, such as hormones, enzymes, and medications, throughout the body.
2. Collagen: Collagen is a protein found in connective tissue, such as skin, tendons, and ligaments. It provides strength and support to these tissues and helps to maintain their structural integrity.
3. Insulin: Insulin is a hormone produced by the pancreas that regulates blood sugar levels. It helps transport glucose from the bloodstream into cells, which can be used for energy.

4. Keratin: Keratin is a protein found in the outer layers of skin, hair, and nails. It provides strength and protection to these structures.

5. Lactase: Lactase is an enzyme that breaks down lactose, a sugar found in milk and dairy products. It is crucial for the digestion of lactose in people who are lactose intolerant.

6. Myoglobin: Myoglobin is a protein found in muscle tissue that binds and stores oxygen, helping to provide oxygen to muscle cells during periods of activity.

7. Myosin: Myosin is a protein in muscle tissue responsible for muscle contraction. It works in conjunction with another protein called actin to move.

8. Sucrase-isomaltase: Sucrase-isomaltase is an enzyme that breaks down complex sugars, such as sucrose and maltose, into simpler sugars that the body can easily absorb. It is found in the small intestine and plays a vital role in the digestion of carbohydrates.

The above gives a final total of 80 different protein sequences.

3 Task solution

3.1 Dataset

At the very beginning, I proceeded to download protein sequences for selected organisms from the NCBI database. I later supplemented them with human sequences and saved them in the proteins/raw/... folder. Under the names of specific proteins. All sequences can be found in this folder in the file all_prot.txt. I used this file to create a local database using local BLAST. The database is located in the proteins/database/... folder. Using the database, I used the local BLAST MSA algorithm to confirm that I had the correct proteins - the results matched those of the NCBI database (which is not surprising since I also used the BLAST algorithm on the NCBI website).

3.2 Clustering

I chose CD-HIT as the algorithm for clustering because, among the algorithms I knew, it seemed to me that this one would return the most interesting results (clusters that differ from the correct groups of proteins). Perhaps scientifically, this is not a very appropriate approach, while I was keen not to analyze two identical sequence splits as an exercise.

CD-HIT algorithm is a fast and efficient way to cluster protein sequences based on their sequence identity. It is highly configurable, allowing users to specify the identity cutoff and other parameters to control the sensitivity and specificity of the clustering process.

The CD-HIT algorithm works by performing the following steps:

1. Input: The input to CD-HIT consists of a set of protein sequences that need to be clustered.

2. Preprocessing: Before the actual clustering process begins, CD-HIT performs some preprocessing steps on the input sequences. This can include filtering out low-complexity or short sequences and reducing the length of the sequences by removing low-scoring segments.

3. Sorting: CD-HIT sorts the sequences based on their lengths, with the shortest sequences being placed first. This helps ensure that shorter sequences are compared to longer sequences first, saving time and computational resources.

4. Clustering: The actual clustering process begins with the first sequence in the sorted list. This sequence is compared to every other sequence in the list, and the sequence identity (percentage of identical amino acids) is calculated for each comparison. If the sequence identity is above a certain threshold (called the "identity cutoff"), the two sequences are considered part of the same cluster. The process is then repeated for each subsequent sequence in the list.

5. Output: Once the clustering process is complete, CD-HIT produces an output file that contains the clusters and the sequences that belong to each cluster.

The results obtained can be found in the proteins/output folder. The algorithm returned 11 clusters, three of which have only one sequence.

I then divided all the sequences into the designated clusters.

3.3 Phylogenetics

I started by determining the distance matrix for each group, cluster, and for all sequences using Clustal Omega to draw phylogenetic trees.

Then, using the Bio.Phylo module, I generated phylogenetic trees using the UPGMA algorithm.

The UPGMA (Unweighted Pair Group Method with Arithmetic Mean) algorithm is a method for constructing phylogenetic trees. It works by performing the following steps:

1. Input: The input to UPGMA consists of a matrix of pairwise distances between the taxa (organisms or groups of organisms) that are being analyzed. The distances can be calculated using various methods, such as percent identity or the number of substitutions per site.

2. Clustering: UPGMA begins by forming clusters of taxa based on their pairwise distances. It starts by selecting the two taxa with the smallest distance and creating a cluster. It then calculates the average distance between the cluster and every other taxon and chooses the taxon with the smallest average distance to add to the cluster. This process is repeated until all taxa are in a single cluster.

3. Tree construction: As the taxa are added to the clusters, UPGMA constructs a tree by connecting them with branches. The length of the branches is equal to the distance between the taxa.

4. Output: Once the tree construction is complete, UPGMA produces an output file that contains the tree, with the taxa arranged in a branching structure that reflects their evolutionary relationships. The output can be in various formats, such as Newick or Nexus.

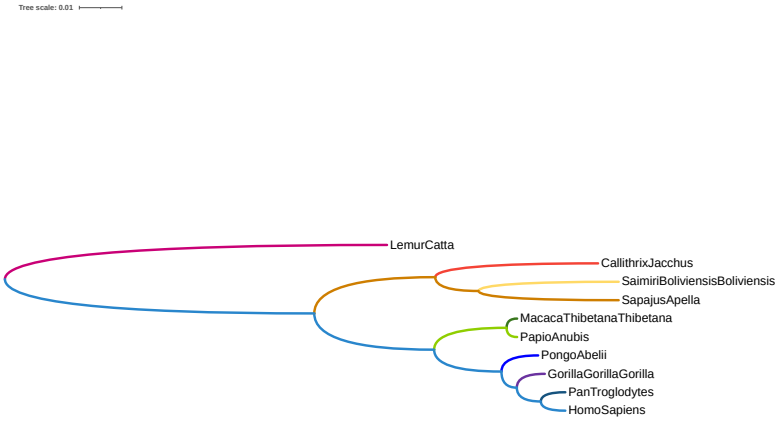
In this way, I obtained trees for each group, each cluster, and the set of all sequences. I then created consensus trees from the trees of each group and each cluster. I saved the three output trees in the folder trees/... . The code for this part of the task is available in the file clustering-phylogeny.ipynb

Then, using tools available on the Internet, I proceeded to edit the resulting trees visually.

4 Results and conclusion

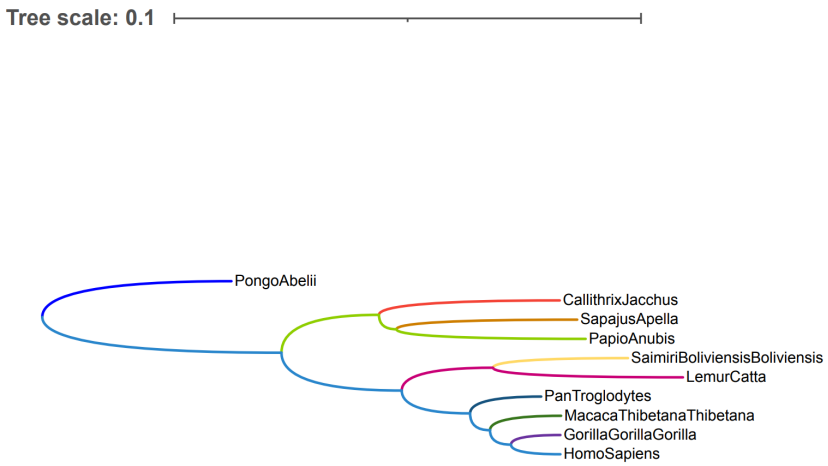
At this point, it is worth mentioning what science says about primate evolution. From the materials I've found, it appears that the similarity follows in this order: Homo Sapiens - Pan - Gorilla - Pongo - Papio - Macaca - Callithrix - Saimiri - Sapajus - Lemur.

Below is the consensus tree obtained from each group of proteins.



As you can see, the tree is almost perfect.

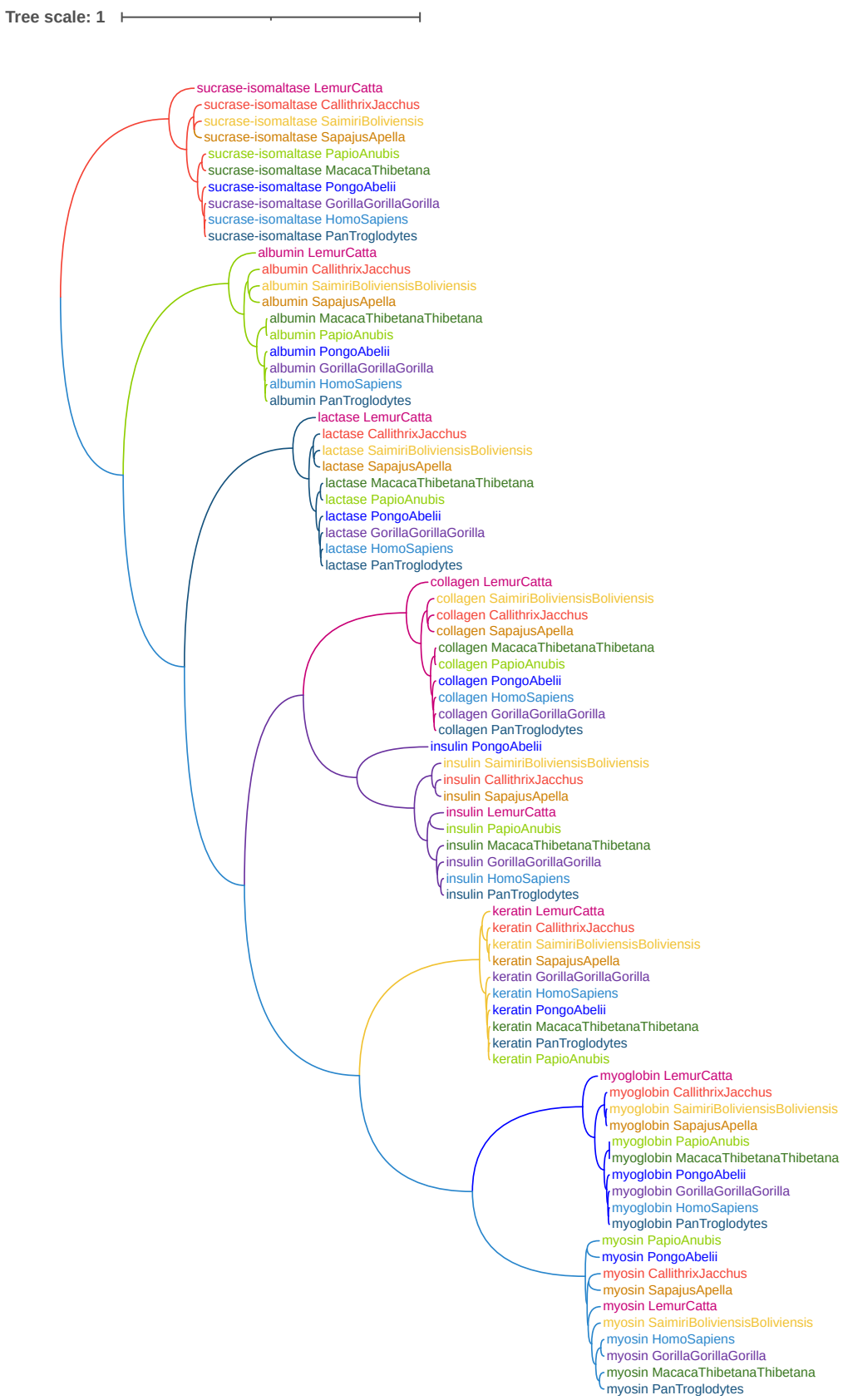
However, this cannot be said of the tree formed from the clusters I designated (the edge colors correspond to the same organisms as before):



In the case of clusters, the tree represents the kinship closest to humans quite well. In addition, the number of organisms in each leaf cluster also matches the results achieved in the previous approach - the trees are visually similar.

At this point, I cannot say whether this indicates a complete failure of the clustering method. Unfortunately, in my approach, the choice of closely related organisms, an algorithm whose clusters did not coincide with protein groups, and another layer of errors and simplifications with the tree-generating algorithm result far from the truth. Note, however, that this is a study based only on eight selected protein sequences - this may be too small a dataset to classify similar organisms at a good enough level effectively. Therefore, the method based on groups of proteins seems better. Another critical issue is whether we have a priori or a posteriori knowledge. The method based on groups of proteins was possible mainly because it matched precisely how the data was selected for the task. This situation is not achievable in every experiment. On the other hand, methods based on clustering can spot something we could not predict by relying on purely mathematical classification.

The last tree presented is the tree created from all the sequences used in the task. The colors of the labels correspond to the colors of the organisms used in the previous trees.



At first glance, you can see that the segments mostly correspond to the correct relationship of organisms. In addition, the shape of each segment corresponding to a particular protein is similar to consensus trees.