Bioinformatics
Clustering and Phylogeny

Hassan Jaber

Contents

Functional Documentation

1. **Dataset description:**

a. For our project, I chose the following interesting human proteins:

- **Apolipoprotein A1**: it's a protein that in humans is encoded by the APOA1 gene. As the major component of HDL particles, it has a specific role in lipid metabolism. The protein, as a component of HDL particles, enables efflux of fat molecules by accepting fats from within cells (including macrophages within the walls of arteries which have become overloaded with ingested fats from oxidized LDL particles) for transport (in the water outside cells) elsewhere, including back to LDL particles or to the liver for excretion.

- **Kallikrein-11**: it's a protein that in humans is encoded by the KLK11 gene. Kallikreins are a subgroup of serine proteases having diverse physiological functions. Growing evidence suggests that many kallikreins are implicated in carcinogenesis and some have potential as novel cancer and other disease biomarkers. This gene is one of the fifteen kallikrein subfamily members located in a cluster on chromosome 19. Alternate splicing of this gene results in two transcript variants encoding two different isoforms which are differentially expressed.

- **Adenosine deaminase**: (also known as adenosine aminohydrolase, or ADA) is a protein involved in purine metabolism. It is needed for the breakdown of adenosine from food and for the turnover of nucleic acids in tissues. Its primary function in humans is the development and maintenance of the immune system. However, the full physiological role of ADA is not yet completely understood.

- **BCL-6**: (B-cell lymphoma 6) is a protein that in humans is encoded by the BCL6 gene. BCL6 is a master transcription factor for regulation of T follicular helper cells (TFH cells) proliferation. BCL6 has three evolutionary conserved structural domains. The interaction of these domains with corepressors allows for germinal center development and leads to B cell proliferation. The deletion of BCL6 is known to lead to failure to germinal center formation in the follicles of the lymph nodes, preventing B cells from undergoing somatic hypermutation. Mutations in BCL6 can lead to B cell lymphomas because it promotes unchecked B cell growth. Clinically, BCL6 can be used to diagnose B cell lymphomas and is shown to be upregulated in a number of cancers.

- **Thymosin beta-4**: it's a protein that in humans is encoded by the TMSB4X gene. Recommended INN (International Nonproprietary Name) for thymosin beta-4 is 'timbetasin', as published by the World Health Organization (WHO).
This gene encodes an actin sequestering protein which plays a role in regulation of actin polymerization. The protein is also involved in cell proliferation, migration, and differentiation. This gene escapes X inactivation and has a homolog on chromosome Y.

- **Calsyntenin 1**: it's a protein that in humans is encoded by the CLSTN1 gene. Mutations in this gene have been shown associated to pathogenic mechanisms of Alzheimer's disease.

- **Collagen, type I, alpha 1**: also known as alpha-1 type I collagen, is a protein that in humans is encoded by the COL1A1 gene. COL1A1 encodes the major component of type I collagen, the fibrillar collagen found in most connective tissues, including cartilage.
Collagen is a protein that strengthens and supports many tissues in the body, including cartilage, bone, tendon, skin and the white part of the eye (sclera).

- **Keratin**: it's one of a family of structural fibrous proteins also known as scleroproteins. Alpha-keratin (α-keratin) is a type of keratin found in vertebrates. It is the key structural material making up scales, hair, nails, feathers, horns, claws, hooves, and the outer layer of skin among vertebrates. Keratin also protects epithelial cells from damage or stress. Excessive keratinization participates in fortification of certain tissues such as in horns of cattle and rhinos, and armadillos' osteoderm. Keratin comes in two types, the primitive, softer forms found in all vertebrates and harder, derived forms found only among sauropsids (reptiles and birds).

b. After choosing the above proteins, I run BLAST using https://www.ncbi.nlm.nih.gov/ for each proteins, then I downloaded the sequences for 7 different organism other than human for each protein.
The organisms that I chose are :

- Three Primates animals: Callithrix jacchus, Rhinopithecus roxellana, Theropithecus gelada.
- Four Seal family (Pinnipedia) related animals : Callorhinus ursinus, Mirounga angustirostris , Neomonachus schauinslandi, Phoca vitulina.


Callithrix jacchus


Rhinopithecus r.


Theropithecus gelada

Callorhinus ursinus

Mirounga angustirostris

Neomonachus schauinslandi

Phoca vitulina

I thought that it would be interesting to compare the human proteins with two different families of animals, so we can see the evolution and the similarities between humans, primates and pinnipedias.

2. Local Blast:

After downloading all the sequences for each protein chosen for the 8 organisms(7 organisms + human). I did a multiple sequences alignments(MSA) between sequences of different organisms for each protein.
For MSA I used an application called ClustalW, downloaded from http://www.clustal.org/download/current/.

ClustalW take as input a file in fasta format consisting of 3 or more protein (or nucleotides) sequences and align them, then it outputs a fasta format(or other formats, you can choose in the application) of the aligned sequences.
For example: if we use "*adenosine_BLAST_sequences.txt"* from the project folder, which contains the protein sequences of adenosine from different organism, as an input in ClustalW application, we would get a fasta file "adenosine_msa_output" that contains the alignment of those sequences.

After finished doing MSA for all the protein sequences, I decided to edited their files ("adenosine_msa_output" ➔ "adenosine_msa_output_edited").
I added the name of the organism in the header for each sequence in each protein group so we can be more clear in the future when we build trees.
For example in the aligned sequences files we would have:

>[Homo_sapiens] NP_066932.1
----------MSDKPDMAEIEKFDKSKLKKTETQEKNPLPSKETIEQEKQAG
ES----------

instead of:

>NP_066932.1
----------MSDKPDMAEIEKFDKSKLKKTETQEKNPLPSKETIEQEKQAG
ES----------

This way it would be clearer to know which organism each protein sequence belongs to, in each group of proteins.

3. Clustering:

a. Since we have all the files that contains aligned sequences from different organisms for each protein, we can use those files as an input for clustering, this way we would have clusters(groups) of sequences for each protein.

For clustering method I chose to do it using Biopython Cluster Library.

We can use from Cluster library a function that is caller **kcluster**.
This function performs k-means clustering on the values in data, and returns the cluster assignments, the within-cluster sum of distances of the optimal k-means clustering solution, and the number of times the optimal solution was found.

a.1. Input of kcluster():

It has the following keyword arguments:

Bio.Cluster.kcluster(*data*, *nclusters=2*, *mask=None*, *weight=None*, *transpose=False* , *npass=1*, *method='a'*, *dist='e'*, *initialid=None*).
The description of each keyword argument can be found in:
https://biopython.org/docs/1.75/api/Bio.Cluster.html#Bio.Cluster.kcluster

We are only interested in the following keywords:

- data: It should be a 2 Dimensional array of arrays representing the sequences. But after reading the MSA protein sequences files using AlignIO.read() and append each sequence to dataset of lists, we would get a list of strings containing the aligned sequences. So we have to convert it into a 2 Dimensional array of integers so that the argument data for kcluster() function would be correct. We can do that using a function called fromstring() from numpy library:
https://numpy.org/doc/stable/reference/generated/numpy.fromstring.html

We can see the code for the explanation above in *"Clustering_Phylogeny.ipynb"* in the Project folder.

- nclusters: It's the number of clusters, the default value is 2, there is no correct value for nclusters as along as its greater than 1 and less than the number of sequences we are using as input , to choose the right nclusters we should study our dataset, in our dataset since I chose two families for organism(Primates and Seals) other than human, I chose nclusters = 2, so we are going to leave the

default value.

- npass: This argument represent the number of times the clustering algorithm is performed, and each time with a different initial condition.
I change its value to a large number (for example: 1000), this way we would have a more accurate result.

a.2. Output of kcluster():

The output of this function is a tuple that represent:

- Clusterid: array containing the id number of the cluster (0 or 1) to which each item was assigned.
For example if Clusterid = [1 1 1 1 1 0 0 0], that means that first 5 sequences are in a cluster and the last 3 sequences are in other cluster according to similarity.
Note: [1 1 1 1 1 0 0 0] is equivalent to [0 0 0 0 0 1 1 1], because we still have the same 2 clusters but with different ids.

-Error: the within-cluster sum of distances for the returned clustering solution.

-nfound: the number of times this solution was found.

After finishing clustering, I decided to write each cluster that contain one or more aligned sequences from each protein in a fasta format file to use it later to build trees for each one of the clusters. For example we can see in the folder and in the code that we created two fasta format files that contain the two clusters of adenosine, (adenosine_cluster1 and adenosine_cluster2)

b. Let's see if clusters of sequences of each protein if it correspond with the similarity of those sequences from BLAST:

Adenosine deaminase: After performing MSA then clustering for this protein on different organism we can see in the output in *"Clustering_Phylogeny.ipynb"* is [0 0 0 0 1 1 1 1] or [1 1 1 1 0 0 0 0] which means:

- First cluster contain the first four sequences from alignments with the corresponding proteins: (XP_032275931.1 , XP_045741765.1, XP_021544790.2, XP_025731180.1) which are relatively related to following the organisms: (Phoca vitulina, Mirounga angustirostris, Neomonachus schauinslandi, Callorhinus ursinus).

-Second cluster contain the rest of the sequences from alignments with the corresponding proteins: (NP_001308980.1, XP_030770406.1, XP_025255098.1, XP_035154576.1) which are relatively related to the organism: (Homo sapiens, Rhinopithecus roxellana, Theropithecus gelada, Callithrix jacchus)

We can observe from BLAST(online version) that the similarity between this protein sequences from the 8 different organism(including human) as follows:

Homo sapiens: 100%
Rhinopithecus roxellana: 97.94%

Theropithecus gelada: 96.17%
Callithrix jacchus: 93.49%

Neomonachus schauinslandi: 80.28%
Phoca vitulina: 79.83%
Mirounga angustirostris: 79.56%
Callorhinus ursinus: 79.28%

In conclusion, we can see that the clusters corresponds to the similarity between those protein sequences from different organisms from BLAST, and in the *"Clustering_Phylogeny.ipynb"* we can see the same result for all the remaining proteins.

4. Phylogenetics:

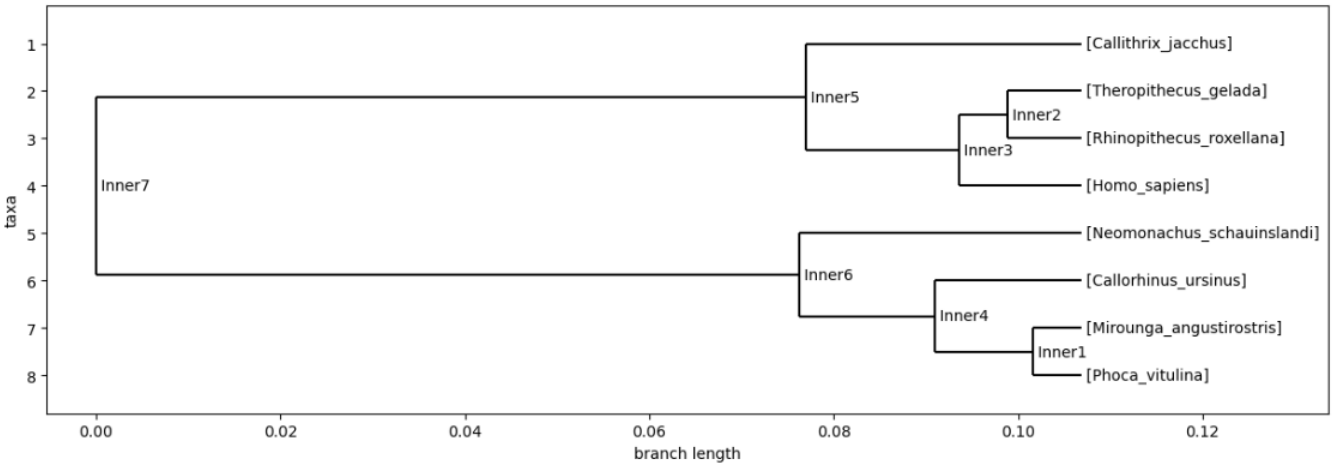
In our next step, we are going use Phylo.TreeConstruction from Biopython to draw Phylogenetic trees, for each group of protein after being aligned(MSA), for each cluster, and one last tree for all the downloaded sequences together.

a. Trees for each group of protein:

First we read the aligned sequences using the class module *AlignIO* with its function *read()*, then we use the *DistanceCalculator()* function from Phylo to calculate the distances between the sequences, after that we use the function *DistanceTreeConstructor()* to construct using UPGMA.

UPGMA (Unweighted Pair Group Method with Arithmetic Mean) is an algorithm used in computational biology to create a phylogenetic tree. It is a method of clustering that is used to build a tree by successively merging clusters of similar items.

The figure below is an example of a tree of the protein **Adenosine deaminase:**

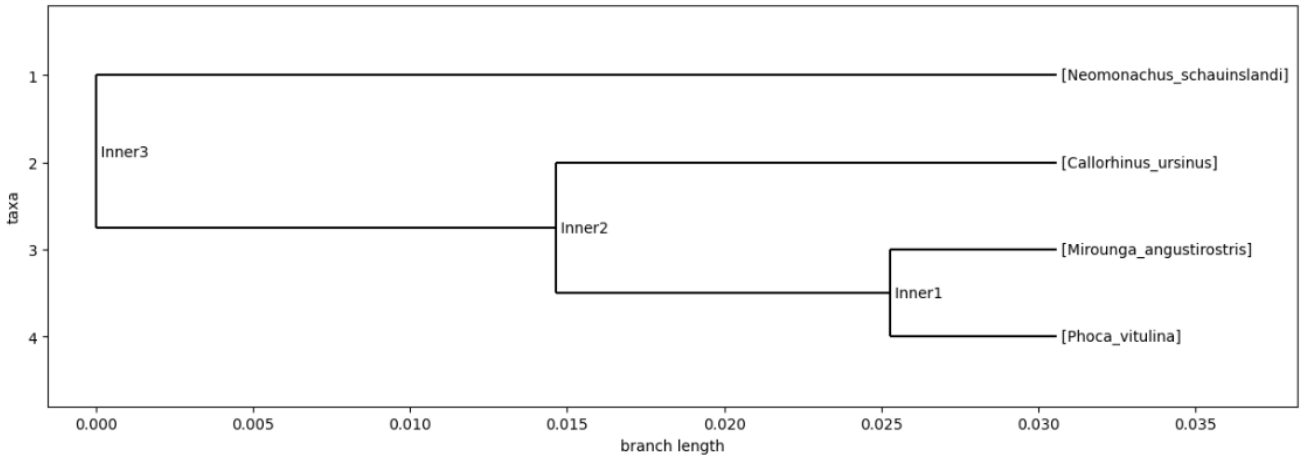


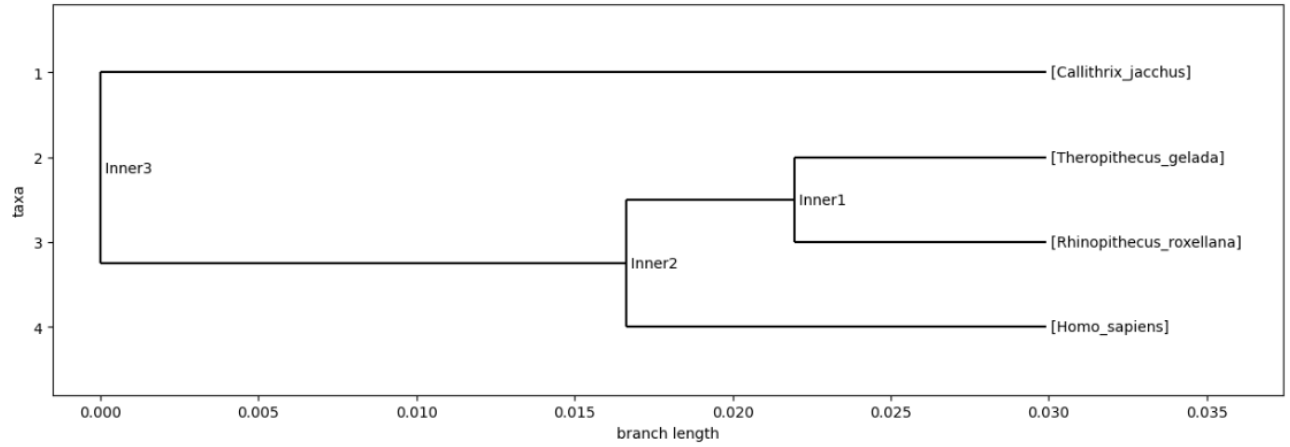
After constructing and drawing each tree, we are going to append it in a newick format file *("trees.dnd")*, after building all the trees and appending all of them in the file, this file will contains all the trees for all the proteins, we need this file so we can use it later to build a consensus tree.

b. Trees for each cluster:

For building trees of clusters we are going to use Phylo.TreeConstruction and UPGMA algorithm like above.

The figures below shows each cluster tree of the protein **Adenosine deaminase.**





c. Common tree for all the sequences:

For building one common tree for all the downloaded sequences, what I did is that I wrote all the sequences in one fasta format file, then I used ClustalW to apply multiple sequences alignment (MSA) on them, because it is not possible to build a phylogenetic tree for the sequences if those sequences were not aligned between each other and do not have the same length.

After that I used Phylo.TreeConstruction module and UPGMA algorithm to build the tree (The tree is showed below).

d. Consensus Trees:

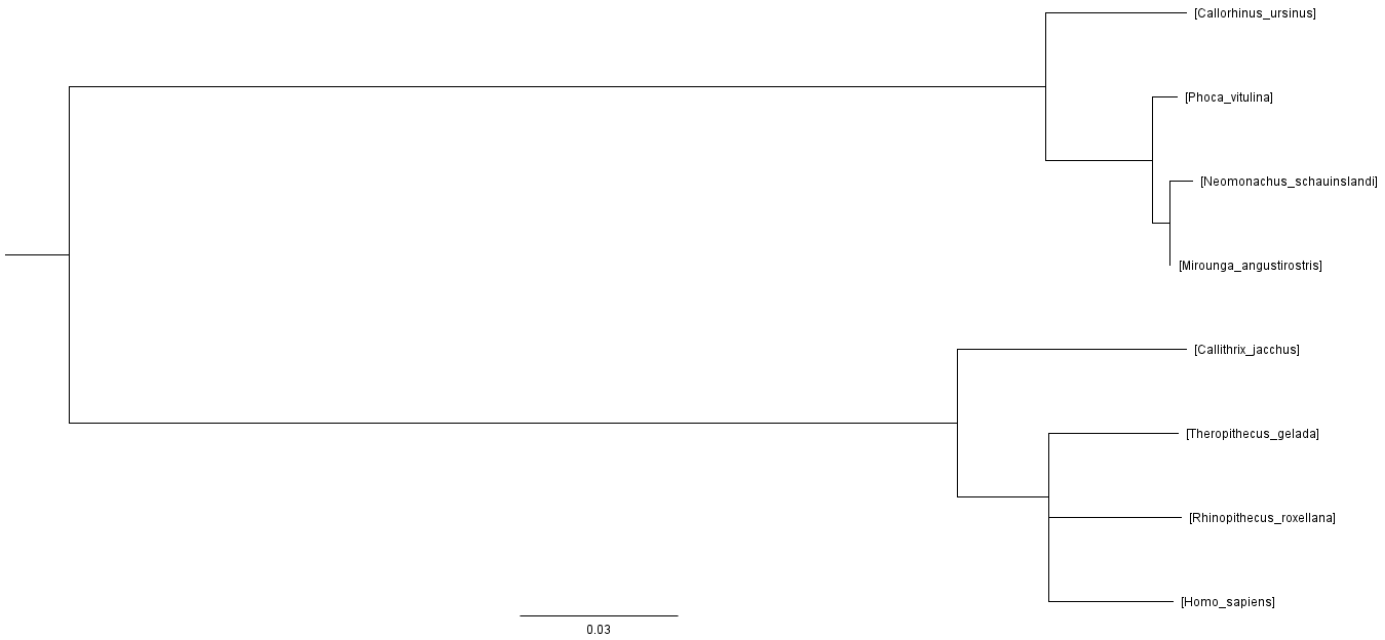
A consensus tree is a summary tree that represents the most widely supported relationships among the taxa (species or other groups) in a set of phylogenetic trees. It is typically constructed by combining the information from multiple phylogenetic trees that have been generated from the same set of taxa, using a process called consensus clustering.

The first consensus tree that I tried to build should be constructed from the trees of clusters, but that was not possible because tree should have the same

taxonomy(some clusters contain 2 sequences, other are 3 or more…) in order to build the consensus tree, therefore it is not possible to build that tree.

The second consensus tree that I created is constructed from the trees of each protein group, that are already in a newick format file that is called *"trees.dnd".* For building this tree I used a software called geneious, this software has a lot of functionality, one of them is that you can import a file that contain multiple trees (but they should have the same taxa, so that's why we edited the msa output files before.) and the software can build one consensus tree from those trees.

The tree below is the second consensus tree:



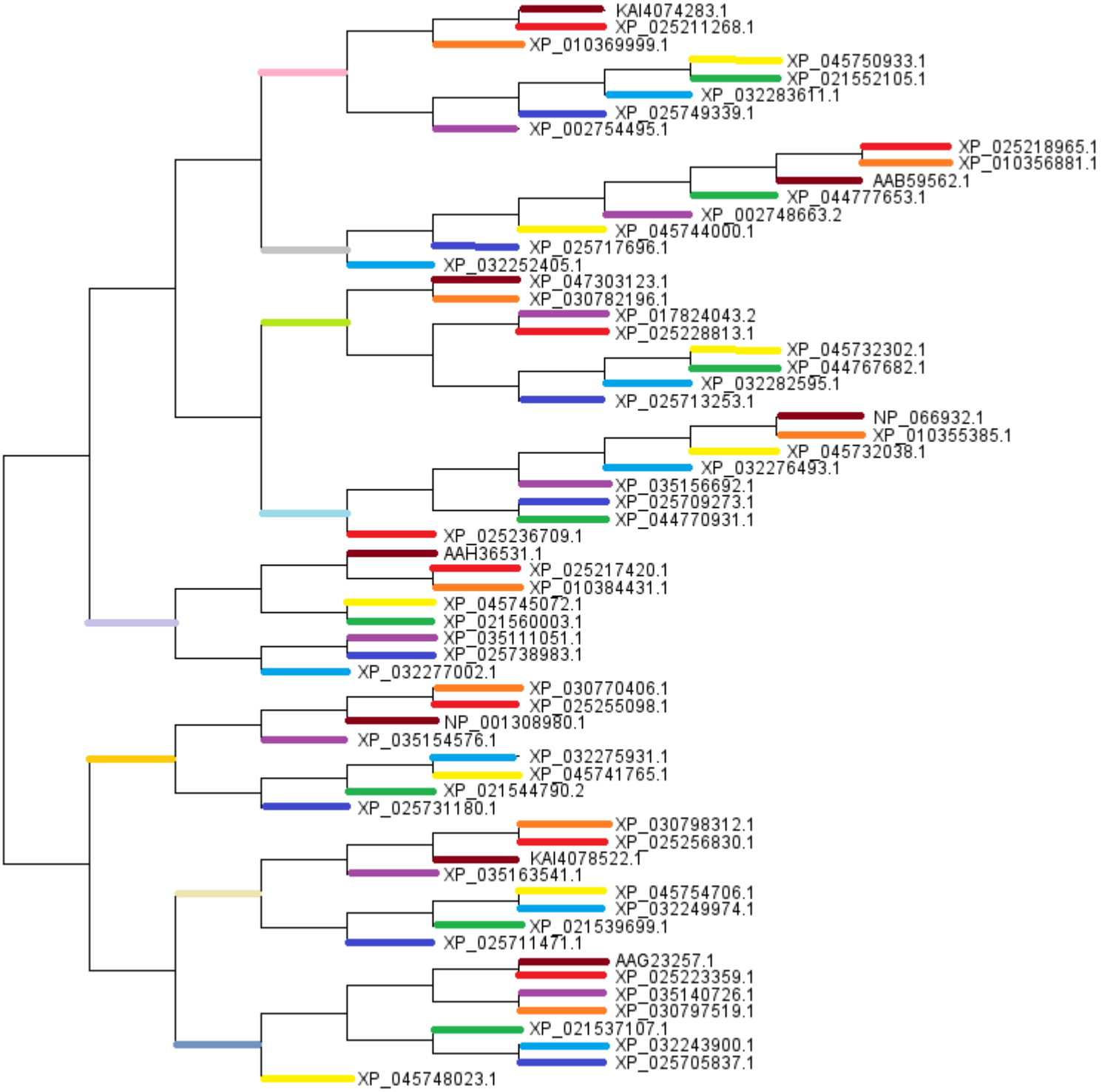
The Tree below is the common tree for all the sequences:
I colored the branches according to organisms:

Brown: Homosapiens
Red: Theropithecus gelada
Orange: Rhinopithecus roxellana
Yellow: Mirounga angustirostris
Green: Neomonachus schauinslandi
Blue: Phoca vitulina
Navy blue: Callorhinus ursinus
Purple: Callithrix jacchus

And I colored other branches according to the proteins:

Rose: Apolipoprotein A1
Gray: Keratin
Lime: BCL6
Light blue: Thymosin beta-4
Lavender: Collagen
Gold: Adenosine deaminase
Blure-gray: Kallikrein-11

Light yellow: Calsyntenin-1



5. Conslusion:

After building the consensus tree and the common tree, we can see obviously that the trees are not similar.

We can observe from the common tree that there is somehow for the most of the proteins, a relation between Homosapiens ,Theropithecus gelada, Rhinopithecus roxellana, and sometimes Callithrix jacchus. Which should be true since those organisms are primates, so they should have similarity between them and humans. And we can see somehow a relation between the seal family organisms as well.

But we can observe very obviously in the consensus tree that, there is a relation between the primates and human, and a relation between the seal family organisms.

Thus we can conclude that the consensus tree is the best approach to observe the evolution between organisms, because it is accurate and straight forward, unlike the common tree because it is not always accurate for all the clades (each clade describe a protein) and it is messy.
In our case we are only comparing 8 organisms between each other, so it would be very hard to read the common tree and observe it for relations if we are comparing thousands of organisms and thousands of proteins.