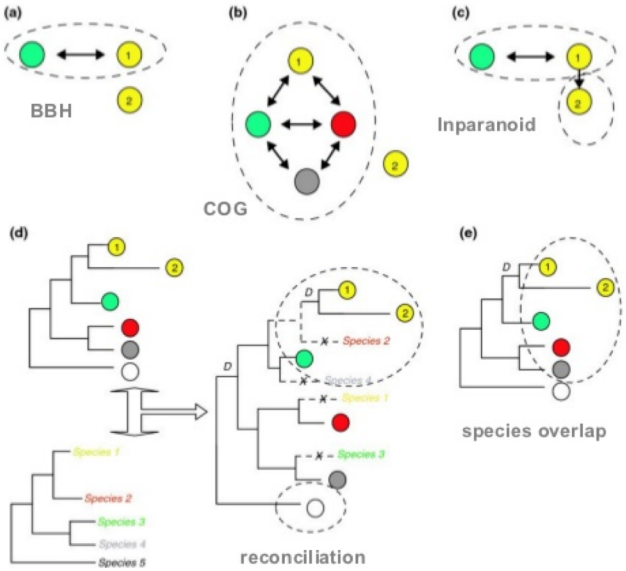
Orthology and paralogy prediction lab

Overview methodologies



Gabaldon, 2008

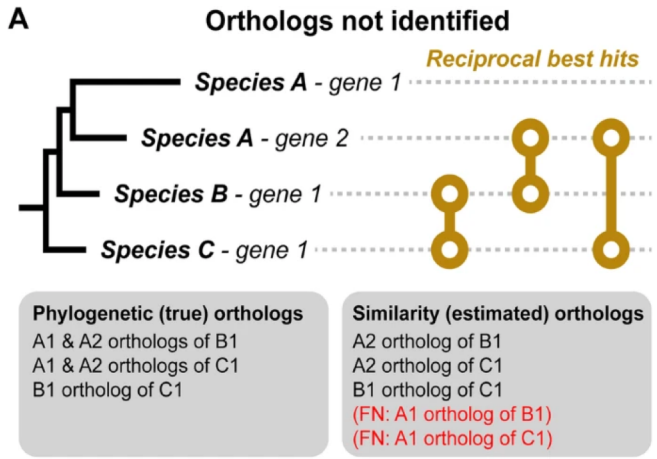
Outline

- Reminder
- Considerations before running an analysis
- Presentation of the dataset
- Approaches we will see to predict orthology and paralogy:
  - Best Reciprocal Hits
  - Orthofinder
  - Phylogenetic based orthology / paralogy predictions
  - Website pre-calculated predictions

https://drive.google.com/file/d/1-qRxfuGJoYw8xOtMPAJEdtp1t0kQXjQ3/view?usp=sharing

Reminders

Homologs: Sequences that descend from a common ancestor.
Orthologs: Sequences that come from a speciation event.
Paralogs: Sequences that come from a duplication event.



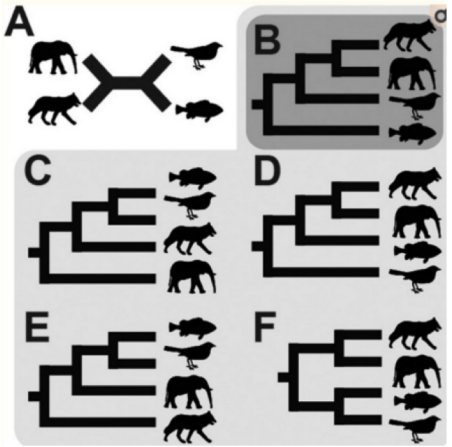
Orthogroups: Group of orthologous genes that can contain inparalogs

First considerations: What do you need to think about before starting.

- Species selection, specially outgroups
- Filtering of isoforms
- Fasta headers
- Computational resources

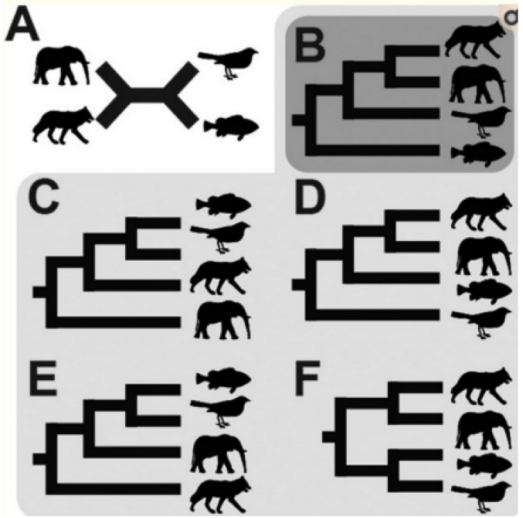
Species selection, specially outgroups

- How many species should we use?
- Genomes? Transcriptomes?
- Outgroups? How many?

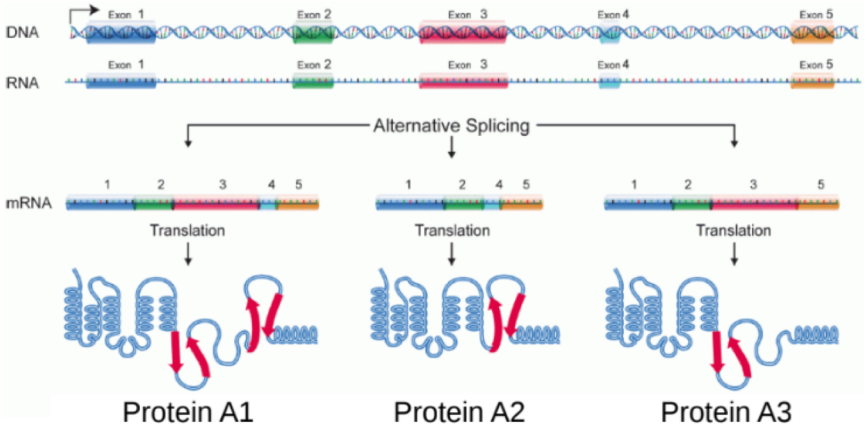


Outgroups

- When building a species tree, it is very important to use an outgroup in order to give directionality to the tree.
- Outgroups will also be necessary to root gene trees and perform orthology and paralogy predictions.
- If possible add at least two outgroups.

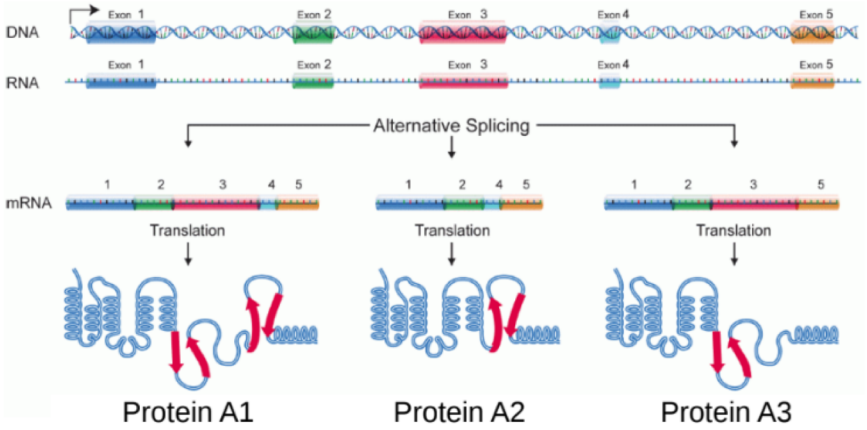


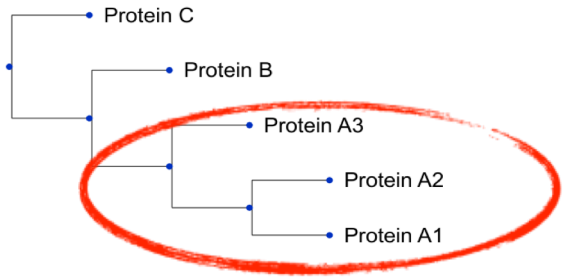
Isoforms



Should we add isoforms in our analysis?

Isoforms


DNA
RNA
Exon 1
Exon 2
Exon 3
Exon 4
Exon 5
Alternative Splicing
mRNA
1 2 3 4 5
1 2 4 5
1 2 3 5
Translation
Translation
Translation
Protein A1
Protein A2
Protein A3


Protein C
Protein B
Protein A3
Protein A2
Protein A1

Headers

Fasta files contain headers that can be complicated. At first it will not bother you, but the downstream analysis can become much more complicated.

>sp|D2H788|RN182_AILME E3 ubiquitin-protein ligase RNF182 OS=Ailuropoda melanoleuca OX=9646 GN=RNF182 PE=3 SV=1

This is a typical Uniprot header.

Do you think it's a good idea to use it as such?

Computational resources

There are many ways to calculate orthology relationships, and some are more computationally expensive than others.

Homology search: **Blast** is the tool by default, yet **Diamond** is much faster when the database is big.

Computational resources

Orthology prediction: Tree based orthology prediction is more accurate, yet similarity based methods are faster.

Species selection: More species give more resolution, yet everything becomes more computationally expensive.

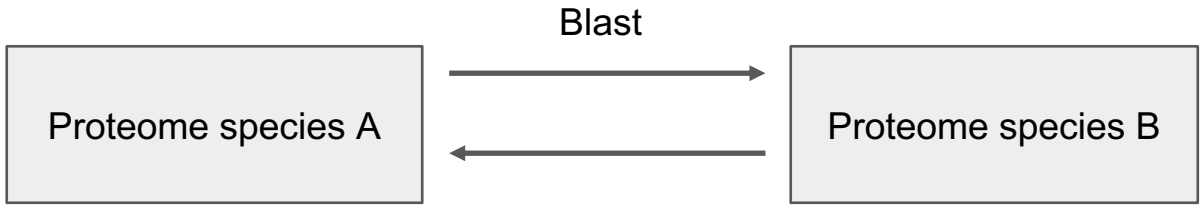
Before running an analysis always consider what you need and if you have the resources to get it.

Exercise 1: Best Reciprocal Hits

It’s the most simple of the methods and is based on the idea that if two sequences from different species are each other’s best hit then they are likely orthologs.

Advantages: It’s fast and simple to understand.

Disadvantages: Will ignore complex evolutionary scenarios. Will be confused by differential gene loss.



How to run a local blast

Fasta file with the queries (can be one or many)

```
>ASPCL_0074_00795
MSKSGGSSFIPDYASYNWTGAPSNYEQFTTNEIGGDSRTENLNKWFQSGDQAYIIVASAM
VMVMIPGLGFLYSGLARRKSALSMIWACMASFSVVTFQWYFWGYSLAFSPTATNGYIGNL
RNFGLMNTLADPSPGSPLIPNLLYAFYQMQFCGVTAAIVMGAVAERGRLLPAMVFTFIWA
TIVYCPMACWVWNVNGWAYNYGVMDYAGGGPVEIGSGMAALAYSMVLGRRQERMMLNFRP
HNVSLILLGTVFLWFGWLGFNGGSAFGANIRATMACWNTNLTAAFGAITWVLLDWRLARK
WSMVGWCSGTISGLVAATPASGYLSPWASVILGIVTGIVCNYATKVKYWIRIDDSMDVFA
EHGVAGIVGLLFNALFGDDAIVGLDGVNTGSGVGGWVIHNYKQLYIQVAFIVASCAYSFV
VSAIIAYAINAIPGLKLRASEEAELLGMDDDQLGEFAYDYVEVRRDYLAWTPQKHDQLED
GHEIPHAARYGIGEHSEMLEGQTPVRIHSQGCSEGDSGIQELKIAPAPRQVAEHHAPAPS
APQTNGQPAPQQIEEKQESST
>ASPCL_0089_08960
MASITSVSLLSEAEFGCEHLASILAQDGNTANQFKSSFIKTHKALAPQRIATEDLAKQKG
GLRPASLLRPKYTCLTCSEACVPAKRETHTGETGHQFYMESRGRALFCQSCRDLVYDHDL
ERLRSSSQNGTVQEIRRRRFSENSSDEQYVKINANKRPCAKQGVRGLYNLGQTCYLNVIL
QTLLHDPILNTYFLGSGHQSHDCNAPDCIGCAVAEAFADFNSSDKAEGFAALNLLLASWR
ASPTLAGYHQQDAHEYYQFLVDKLHASTEGHVEDHDQACSCFFHKTFYGKLRSSVTCDKC
GNVTKTEDPMVDLSLDVQVQAKKRAMGGGVGPSATPTLNGCLESFTSPEKLMAGVYNCSG
CGNTPQKATKQLRIKKLPAILCMQLKRFEHSLAVSEKVEGRVDFPLSINMLPYTTNPNSK
VDKSKYIYDLSSAVVHKGKLDAGHYYVYCKQGDQWVLFNDDQVTAVAEADVLNADAYLLF
YSLRTFGSGLQ
>ASPCL_0074_01073
MRHSFISRCAFISCLLGSSFHAHAQSGTCSNTQPCTSGCCSNSGHCGFGPDFCGSDACVS
TCDAVAECGEYAAVNGTRCPLNVCCSPYGFCGTTELFCGTGCQSGCEAVNKPSCSGTSSD
AIYMGYYEGWNPTQRMCDILLPQDINVTPWTHLYYAFAGIDSTDSTITTTNPNDEEYWRQ
FTALKQKKPSLKTYISVGGWDLGGKVFSDMVKFPGTRQSFITSAIAMMKQYGFDGIDIDW
EYPAAEDRGGVEGDTANLVKFLAEMRDAIGNDFGLTATLPSSYWYMKGFDIVSMAKYVDY
FNFMAYDIHGTWDGTTSNSSSPDVNPHTNLTEISAGLDLLWRNSIDPSKVLLGLGFYGRS
FTLADPSCNTPGCPFYTKNNSTGGGVAGECTVTSGILSDYEINRILEQYNVNVEYDATAG
VNWMTWNSNQWVSYDNARTLRQKADFANGKCLAGLFSWAVDLGGPGTLTNPNDLTASDFS
MAGASTDGGDDGSGIVYVGQDIFGPSPTVSAIAPVSLIFPPFVLPTPTVITPDPYPTSLE
VAWPTVLPVVTSGTTTYTTTITRTIVNTTLEVSPITTTALHFWGWNLTNGVNSTSGPLII
SLDIPDTTIEIGPVPGVTRTPTPRVVKIPPWPWVTTTGGIEPTVHFIQGNPPSPTCTANC
GHKCYSFCDGPCLVDCGSDSGSSSGFLDPEDSDPPSVGKCVGPDCKNGKCTGTLCVQKGC
TGDDCESGICLGSHCTPTGCTGSDCDDGHCAGSHCQDHGCVGSECNSGSGTCWGLSCLSW
GCIGLDCSGSSFTCSGPLCHVVSCSGPKCSEGICTGSGCQSEDGDCQSSEADVCTEWITS
TLVTPASTYSTSTITSHCSTITACSAQATTSTSTVGSSGLVEGTVSDVYFSPTANSNLAA
SADAYWSTFWSQFEGASPTTISPTTTPPTTTSPTTTTTAPSTTNIPNSFMIFKYEVHTVY
FDTSETYSYSYWYGDYYSYMQDVTSDNVCTNSYKVIGPVDANAGDPFPASLRSFNLPAQYT
GCTYSGSTDSVGSVSCGSNQFSCSKIDGYDGKTPTYDCGKTTADGGSTYIYHYYFEAIQC
SITY
```

Fasta file with the targets

↓ makeblastdb -in target_file.fasta -dbtype prot

Format the targets file into a blast database

↓ blastp -query query_file.fasta -db target_file.fasta -evalue 1e-5 -outfmt 6 -out results.blast -num_threads 4

Execute the blast search and output it into a file

How does the blast results file look like:

```
# BLASTP 2.6.0+
# Query: PENCH_0037_10162
# Database: ASPCL.fasta
# Fields: query acc.ver, subject acc.ver, % identity, alignment length, mismatches, gap opens, q. start, q. end, s. start, s. end, evalue, bit score
# 4 hits found
PENCH_0037_10162        ASPCL_0077_01917        59.836  122     45      3       1       121     1       119     1.69e-45        138
PENCH_0037_10162        ASPCL_0089_08902        57.258  124     38      4       1       121     1       112     8.06e-39        121
PENCH_0037_10162        ASPCL_0081_03871        52.041  98      38      2       20      117     39      127     1.88e-30        100
PENCH_0037_10162        ASPCL_0089_08749        40.816  98      47      3       19      116     390     476     4.45e-17        69.7
```

(this is with option -outfmt 7, -outfmt 6 is the same but without the headers)

Exercise 1:

1.- Download the orthology_lab.tar.gz file from the course website, uncompress it and go to exercise 1

2.- Build the blast databases for Human and Zebrafish

3.- Run the blast of Human against Zebrafish and the other way around

4.- Execute the program using get_BRH.py using python

This program needs 5 inputs:

- Fasta file 1
- Fasta file 2
- Blast results where Fasta file 1 was the query
- Blast results where Fasta file 2 was the query
- Output file name

5.- How many BRH do you have between Human and Zebrafish?

6.- How many unpaired proteins are left. If this was a complete proteome and you had the same values, what would be the explanation for this?

7.- Now take one of the pairs in the file of BRH and go back to the blast results files. Using grep, check out that the chosen pair is indeed the best reciprocal hit of each other.

8.- Protein DANRE_1539 does not have a best reciprocal hit. Looking at the blast files, can you figure out why not?

OrthoFinder

OrthoFinder is a fast, accurate and comprehensive pipeline for comparative genomics. It finds orthogroups and orthologs, infers rooted gene trees for all orthogroups and identifies all of the gene duplication events in those gene trees.

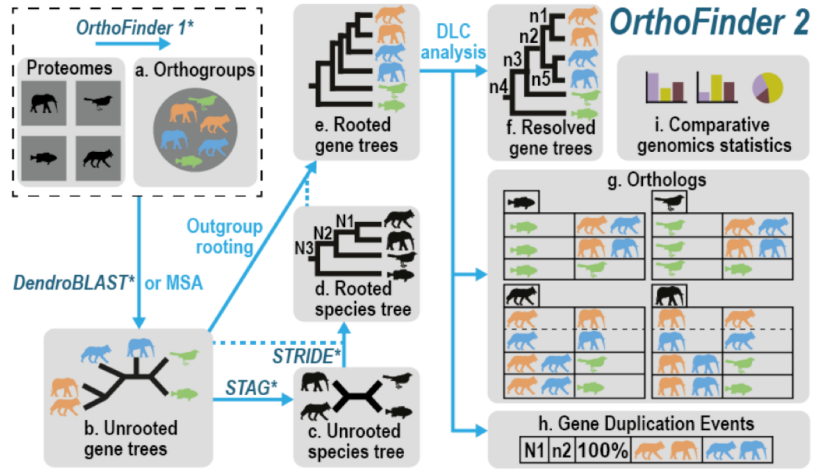


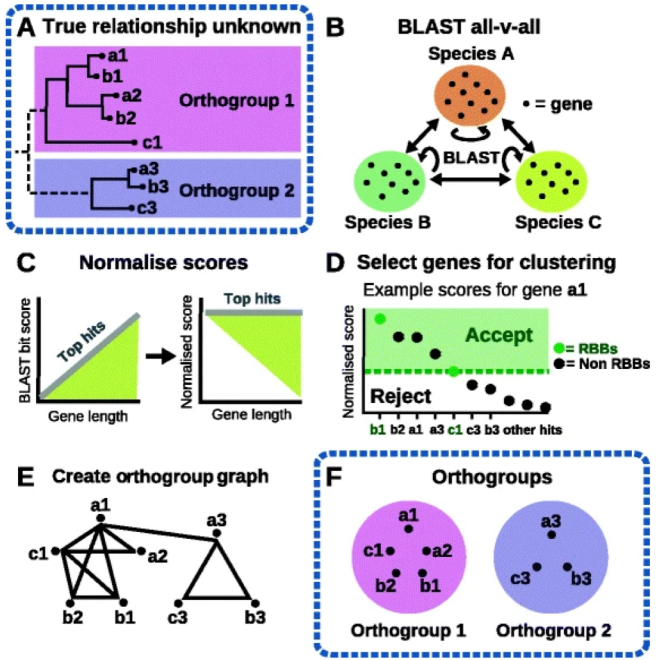
Figure 1: Automatic OrthoFinder analysis

OrthoFinder

Things that Orthofinder solves compared to other algorithms:

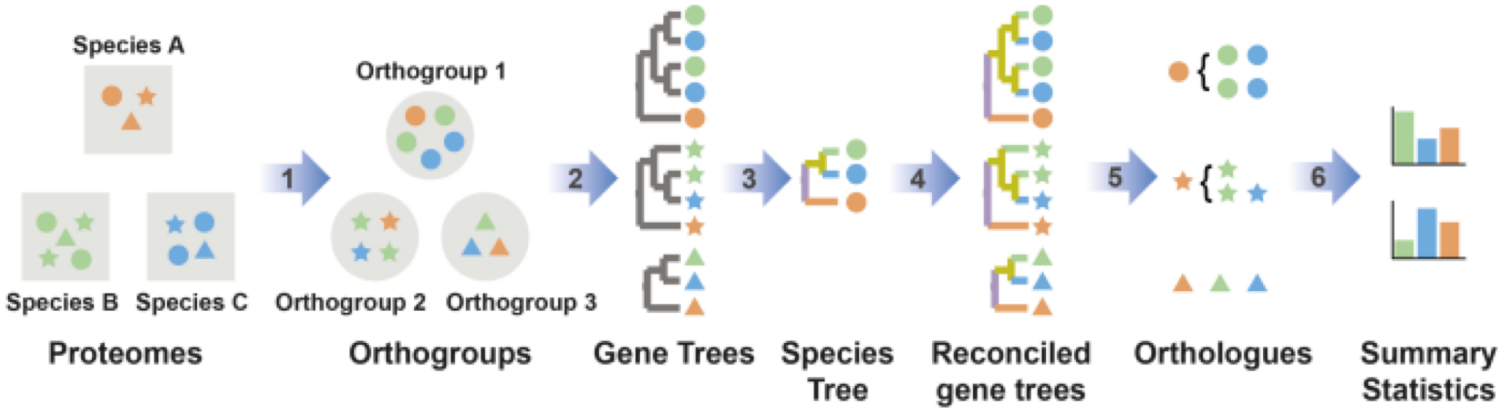
Bias towards gene length.

Bias towards distantly related species.



OrthoFinder

The pipeline goes from a set of proteomes to fully resolved gene trees and their orthologs and paralogs



OrthoFinder

Go to the folder Exercise 2 (cd exercise2). Now we have not only two small proteomes but 5. And we are going to run Orthofinder to extract the orthology relations between them.

We first will run orthofinder with default values, the only input will the proteome folder:

orthofinder.py -f proteomes/

If you have a look, you will see that there are 5 files in there, one for each species.

OrthoFinder

Once orthofinder has finished you will find the results in a folder called:

proteomes/OrthoFinder/Results_XXX/

Note that OrthoFinder uses the date to name the different runs plus an "_Number" if you execute it multiple times on the same day. To make sure we can keep track of the different runs we will create links to each folder:

ln -s proteomes/OrthoFinder/Results_Jan24 Results_default_parameters

OrthoFinder: The inflation parameter

One of the most important parameters is the -I option, this regulates the size of the orthogroups and as such has a direct impact on orthology and paralogy relationships.

Execute OrthoFinder with an inflation parameter of 3.0.

Note that OrthoFinder is modular so you can re-use part of the previously computed results (You cannot use the link we created for this). In this case we will use the homology search:

orthofinder.py -b proteomes/OrthoFinder/Results_Jan24/WorkingDirectory/ -I 3.0

ln -s proteomes/OrthoFinder/Results_Jan24/WorkingDirectory/OrthoFinder/Results_Jan24/Results_I30

OrthoFinder

You now should have two folders of results called Results_default_parameters and Results_I30. Within each result folder you will have numerous folder with different kind of information, check out the Orthogroups folder and answer these questions:

1.- Are there differences between the number of orthogroups of the two rounds?

2.- How many single copy orthogroups are there in each run?

3.- How many genes have been left unmapped in each case?

OrthoFinder

Remember that orthogroups can contain orthologs and inparalogs. If we want to distinguish between the two and obtain only orthology relationships we need to take this a step further and predict them. OrthoFinder does it using tree based methods, this means that first it needs to reconstruct a species tree. If this tree is not correct, it will affect your orthology inference.

Go to the species tree folder, get the newick of the species tree and check if it is correct in both cases (you can use phylo.io for that).

Note that the species tree needs to be correct and properly rooted!

OrthoFinder: Running with a custom species tree

If you are aware of the topology of your species tree, you can use it from the start with the -s parameter. In this case we will re-run the orthofinder run with the inflation parameter set to 3.0 as before but with the correct species tree. As the orthogroups are not defined by the species tree we will restart from that point on:

```
orthofinder.py -fg
proteomes/OrthoFinder/Results_Jan24/WorkingDirectory/OrthoFinder/Results_Jan24/
-s proteomes/OrthoFinder/Results_Jan24/Species_Tree/SpeciesTree_rooted.txt

ln -s
proteomes/OrthoFinder/Results_Jan24/WorkingDirectory/OrthoFinder/Results_Jan24
_1/ Results_I30_correct
```

Orthofinder

Now lets check the Orthology results found in folders Orthologues and Comparative_Genomics_Statistics. Explore the files found in these folders and try to answer these questions:

1.- How many orthologs did Orthofinder find using default parameters between Human and Mouse?

2.- Were there differences in the other two runs? (Results_I30 and Results_I30_correct). Did the change in species tree affect the prediction of orthologs?

3.- These results are depicted in a matrix, but this matrix is not symmetrical (Orthologs between Human and Mouse are not the same as orthologs between Mouse and Human). Can you think why?

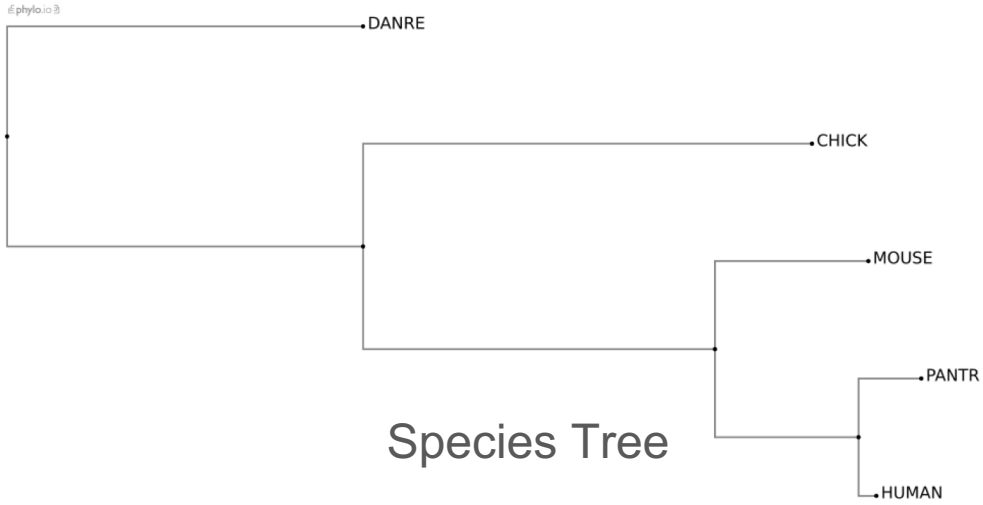
Orthofinder

We can also obtain some information regarding duplications. Duplications can be specific for a single species or they can involve multiple species. Go to the Gene_duplication_Events folder and answer the following questions:

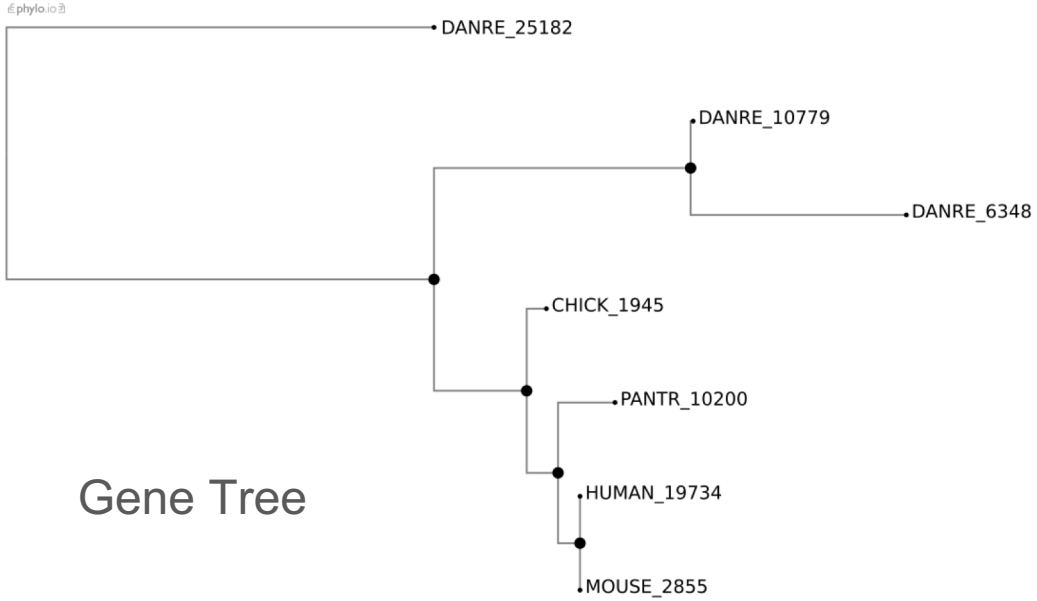
1.- How many duplications can you find in the run with the default parameters. Do they always involve one single species?

2.- How many species specific duplications for Mouse can be found using default parameters? Is this number affected by the inflation parameter?

Tree based methods



Species Tree



Gene Tree

Can you color the dots on the gene tree in red if there is a duplication event or in blue if there is a speciation event according to the species overlap algorithm or the reconciliation algorithm?