Comparative Genomics 2018

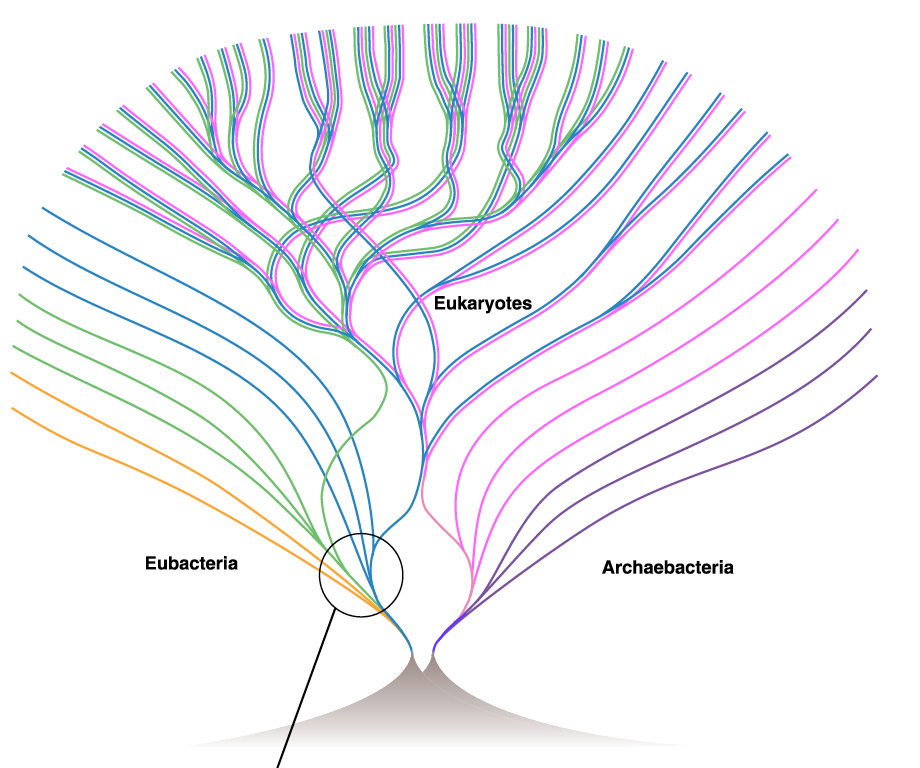
Practical 3: Phylogenetic Reconstruction

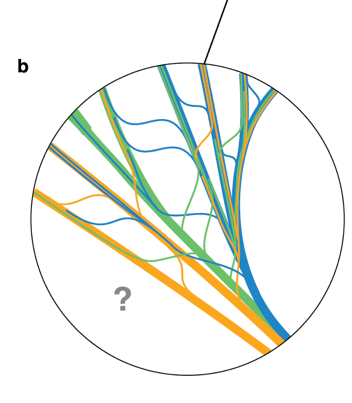
Group number: 6

Group members: Kyle Kimler, Kajetan Juszczak

**Summary**

Phylogenetics has come a long way since the days of Darwin’s first tree of life. Since the dawn of molecular biology, the field of phylogenetics has evolvedfrom comparing cytochrome and ferrodoxin amino acid sequences to sequencing 16S rRNA, small subunit of the ribosome conserved across all the domains of life. Phylogenetic reconstruction is interesting to us as a species not only from an existential, philosophical viewpoint but also from a medical viewpoint as advances in sequencing technologies allow us to reconstruct evolution of the code that informs the systemic differences between us and lower animals. Even as early as the ‘90s we found that the tree of life is very convoluted, with organisms even from different domains apparently having contributed to one another’s genomes after their original divergence, but also contributing to organismal metabolism in symbiosis. This, and the fact that oftentimes the fossil record is just not good enough for time-resolved phylogeny, makes phylogenetic reconstruction a modern computational problem.





**Key Questions to Answer**

Ex.1 Finding homologs

1. -in input file name, we enter our fasta filename in here

-dbtype type of molecules in database in our case nucl for nucleotides

1. -
2. Blast results
   1. <Hit\_num>1</Hit\_num> - indicate best hit
   2. <Hsp\_hseq>sequence here </Hsp\_hseq>
      1. we can use max\_hsps flag to specify how many High-scoring pairs we want to get.
      2. We can specify what the output directory should be, I expect to see XLM file with best hits.

Ex.2 Parsing

B:

1. calling method .hsps with index 0 will give the best hit
2. Blast record correspond to BLAST output
3. that it contains all the hits we get while running BLAST with the flags we we specified. As default It goes from best to worst hit
4. output is a best hit

Ex.3

1. 217.00000000 gap open penalty

39.40000153 gap extension

292.60000610 terminal gap penalty

283.00000000 bonus

1. If the penalties are very high it means that the region is possibly extremely conserved.

Ex.4

1. belvu
   1. b -Scoredist distance correction – used as default

j - Jukes-Cantor distance correction

k -Kimura distance correction

s -Storm & Sonnhammer distance correction

r -uncorrected distances

* 1. We used UPGMA and neighbor-joining and for distance correction Kimura and Scoredist. There were no visible differences between different correction methods. Between neighbour-joining and UPGMA there is a clear difference because neighbour-joining method creates an unrooted tree while a UPGMA tree is rooted.

1. RaxML
   1. [check!](http://etetoolkit.org/treeview/?treeid=9a49b40c1db891e02180b0ac2f80fb88&algid=ce443ed1e53858bf4e11d1e069c7a927)
   2. link to tree : <http://etetoolkit.org/treeview/treeid=9a49b40c1db891e02180b0ac2f80fb88&algid=ce443ed1e53858bf4e11d1e069c7a927>["](http://etetoolkit.org/treeview/?treeid=9a49b40c1db891e02180b0ac2f80fb88&algid=ce443ed1e53858bf4e11d1e069c7a927)
   3. -f a (f – select algorithm)it performs rapid bootstrap analysis and look for trees with highest score

-x 54321 set the number – random seed and start rapid bootstrapping

-N 100 number of alternative runs starting from different starting trees

-T 4 number of threads to run

-p 12345 specify a number of random seed for the parsimony inferences

-m PROTCATBLOSUM62 uses the substitution matrix BLOSUM62 for amino-acids to model.

-s input

-n output

* 1. maximum-likelihood tree-generation methods typically generate a huge number of tree topologies and use probability-of-mutation scores for each basepair change, while distance based methods just calculate exact similarity, and are typically used to generate a single tree topology. Distance based methods are hugely limited by the fact that they only consider one topology, even in the case where they use similarity distances corrected by both aa-substitution matrices *and* normal mutation (Kimura) or other evolutionary model score corrections. While maximum likelihood tree generation may seem a viable solution to this problem, it takes a lot of computation as possible tree topologies increase exponentially with number of nodes, so usually they are reduced in possibility, which might make them miss large-scale genetic events like the genome duplications and large copy events that often occur in plants.

**Ex. 5**

1. Bootstrapping is a method of evaluating phylogenetic trees very similar to cross-training in machine learning, where subsets of the larger dataset are taken to build variations of the tree and the confidence in each split is calculated based on prevalence of that split in the set of many trees created by subset sampling tree construction. Typically this is applied because it is a very simple and effective way to test confidence in tree topology! There is another variation on bootstrapping that tests the confidence of branch lengths as well.

**2.1 and 2.2**. N is the number of bootstrap trees to make! The higher the N, the more confident we are in our tree. We chose N=100, gives a 2 decimal fraction, easy to convert to percentage.

**References**

Martin, W. (1999). Mosaic bacterial chromosomes: a challenge en route to a tree of genomes. *BioEssays*, 21(2), pp.99-104. (Figure in summary)

Zvelebil and Baum. (2007ish) “Understanding Bioinformatics”