# Phylogenetic Reconstruction \2017

*Practical 3 - Comp Gen 2017*

*Assistants: Daniel Morgan and Mateusz Kaduk*



## PURPOSE

In this practical you will learn how to investigate the evolutionary relations between your genomes, *ie* how similar genes are to one another. For this, you will need ubiquitous gene(s); 16S rRNA might be a good idea. For those of your genomes that are complete (*ie* bacteria and archaea), use BLAST to find homologs to the 16S rRNA *E. coli* gene. If you get several hits, take the best one from each genome. We will further align the sequences using Kalign and reconstruct the phylogenetic tree using Belvu.

## OBJECTIVES

- Make yourself familiar with Kalign and Belvu
- http://msa.sbc.su.se/cgi-bin/msa.cgi
- http://sonnhammer.sbc.su.se/Belvu.html
- http://etetoolkit.org/treeview/ (a web tool for viewing trees)

## MATERIALS NEEDED

1. Genome sequences in FASTA format
2. Nucleotide-Nucleotide BLAST+ 2.2.31 (verify by *blastn -h that BLAST **plus** version above 2.2.28 is used*)

## ACTIVITY

*Perform the following steps in this order*

### DNA tree reconstruction

1. Format a blast database for your genomes , so you can search locally.
    a. What do the parameters mean?
    b. Run the program for each nucleotide dataset

```
makeblastdb -in <inputfile.fa> -dbtype nucl
```

2. It might be the easiest (but not necessary) to have all your genomes (genes) in one file, so you can make a single database for them (as long as entries in each file are labeled with genome they belong to).

```
cat genome_0 genome_1 ... > genomes_all
```

3. Use BLAST to query the database for the 16S rRNA file. Find the best hit in each genome as "actual" 16S rRNA and gather them as entries in a fasta file. Extract 16S sequence from BLAST results that you run against whole genome.
    a. What other parameters do you need and what do they mean ?
    b. Where do you find and what you need in the output ?

```
blastn -outfmt 5 -query <query file.fa> -db <database file.fa> -out <output file>
```

### Parsing options A or B

A. Write a simple biopython blast parser using the **NCBIXML module**
    - To obtain XML output from BLAST use the -m 5 parameter.

B. If this is not possible, we can provide you with a existing script to parse the blast output (**blastResultParser.py**). However, you should answer following questions in addition.
    - How does the script choose the single best blast hit in each genome in the database?
    - To what does a blast record correspond?

- What do we assume about the BLAST XML output?
- What does this script output?

## Align your sequences

4. You can use KALIGN on the resulting sequence file to make a multiple alignment of the homologs identified in the previous step.
    a. What parameters for gap penalties exist, and would any of them make sense to apply?

```
kalign <infile> <outfile>
```

## Perform tree reconstruction

5. Reconstruct using more than one method, *ie* distance based and parsimony.
    a. What are the advantages of each? What influenced your choice?
6. Learn about the various options of Belvu and how to create a tree from an alignment.
    a. Which distance correction method does Belvu use?
    b. How would you request a different distance correction.

```
belvu -o tree <fasta file>
```

7. An alternative way to look at the tree is to use tree *newick* viewer http://etetoolkit.org/treeview/

## Sequence bootstrapping

8. Your result is might merely be an artifact of sampling? To get a significance measure for the tree you can use bootstrapping. Explain *what bootstrapping is*.
9. Construct a consensus tree with bootstrap support values from your alignment.
    a. What does N mean?
    b. What N do you choose and what consequences does that choice have?

```
belvu -b N
```

## Sanity Check (always a good idea)

10. To test the effects of poor sampling (or database errors, or recombination), make a copy of your original alignment.
11. For each sequence, paste a copy of itself after it (*ie* ATTCGT->ATTCGTATTCGT).
12. For one or more pairs of sequences, swap the second half of the sequence.
    a. This would match the situation where half of the sequence has been mislabelled, or where half of the sequence has evolved in a way that involves recombination.
13. Perform tree reconstruction as before on these sequences.
    a. What tree do you get?
14. Perform bootstrap-based analysis.

a. Can you deduce from the bootstrap scores which taxa you shuffled the second half of the sequence for?