Comparative Genomics 2018

Practical 3: Phylogenetic Reconstruction

Assistants: Deniz Seçilmiş, Stefanie Friedrich, Miguel Castresana

All forms of plagiarism are forbidden, and if detected it will result in a lower grade.

In this practical, you are going to learn how to investigate the evolutionary relations among your genomes, i.e., how similar the genes are to another one. In order to make this investigation simpler, using ubiquitous gene(s) probably would help and 16S rRNA is suggested here as a suitable choice. You will use KALIGN to align your sequences and perform phylogenetic tree reconstruction using Belvu.

Suggestion: Before you start, make yourself familiar with Kalign and Belvu. You can use the following links.

- http://msa.sbc.su.se/cgi-bin/msa.cgi
- http://sonnhammer.org/Belvu.html
- http://etetoolkit.org/treeview/ (a web tool for viewing trees)
- Use ”module add” to activate these packages

Exercise 1 - Finding homologs

For those of your genomes that are complete (i.e., bacteria and archaea), use BLAST to find homologs to the 16S rRNA *E. coli* gene. In case you get several hits, take the best one from each genome.

DNA Tree Reconstruction

1. In order to be able to search locally, format a BLAST database for your genomes.
   1.1. Explain the parameters. What do they mean?
   1.2. Run the program for each nucleotide dataset.

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

2. Gather all your genomes in one file in order to have a single database.

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

3. Use BLAST to query the database for the 16S rRNA file.
   3.1. Find the best hit in each genome as “actual” 16S rRNA, and gather them as entries in a FASTA file.
   3.2. Extract 16S sequence from BLAST results that you run against the whole genome.

3.2.1. What other parameters do you need? Explain their meanings.

3.2.2. Where do you find the output? What do you expect to see in it?

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

Exercise 2 - Parsing (**Choose A or B**)
Here, you will choose one of the following options for parsing the BLAST output.

A. Write a simple biopython BLAST parser using the NCBIXML module. In order to obtain XML output from BLAST, use the -m 5 parameter.

B. A ready script (blastResultParser.py) is provided to you in order to parse the BLAST output in this option. You will discuss the following questions in detail:

   a. How does the script choose the single best BLAST hit in each genome in the database?
   b. What does a BLAST record correspond to?
   c. What is assumed about the BLAST XML output?
   d. What does this script output?

Exercise 3 - KALIGN

1. Use KALIGN on the resulting sequence file to make a multiple alignment of the homologs identified in the previous step.
   1.1. What are the gap penalties?
   1.2. Does any of the gap penalties make any sense to apply? Discuss why.

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

Exercise 4 - Tree building

1. Investigate the various options of Belvu and how to create a tree from an alignment.
   1.1. Which distance correction methods does Belvu use?
   1.2. Use two different distance corrections in combination with two tree building methods. How does this affect the tree?
2. Build a maximum likelihood tree with RAxML
   2.1. First make sure all gaps in the alignment are denoted with "-"
   2.2. Run e.g. raxmlHPC-PTHREADS-AVX -f a -x 54321 -N 100 -T 4 -p 12345 -m PROTCATBLOSUM62 -s input -n output
   2.3. Explain what the options do.
   2.4. Explain briefly what the main difference is between distance-based and maximum likelihood methods. What are their advantages/disadvantages?

Here is an online tool to view the tree: ETE tree viewer. You need to save the tree as a text file first.

Exercise 5 - Sequence Bootstrapping

1. What is bootstrapping? What is the reason to apply bootstrapping?
2. Construct a tree with bootstrap support values with Belvu from your alignment.
   2.1. What does the option N mean?

2.2. What value of N do you choose and what consequences does that choice have?

```
belvu -b N
```