Comparative Genomics 2018

Practical 4: Phylogenomics

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 will learn how to identify simple orthologs for a complete genome using the best BLAST hit. You will further align these ortholog sequences to find the metagene sequence and reconstruct a phylogenetic tree using Belvu.

Exercise 1 - Orthology Search
Select one of your prokaryotic genomes as the reference genome. For this prokaryotic reference genome, you will search for the best hit orthologs in the remaining prokaryotic genomes. Use BLAST for this purpose.

1. For all of your prokaryotic genomes (including the reference genome), find multi FASTA files with proteins from one of the previous practicals.
   The multi FASTA file contains all protein sequences inferred from your genomes and will be called as the proteome file.

2. Configure BLAST
   Copy the configuration file to be able to use the BLAST commands.

   ```
   cp /afs/pdc.kth.se/home/a/arnee/.ncbirc ~/
   ```

3. Create a BLAST database for your proteomes.
   In the previous practicals, you used makeblastdb to create a database for blastn. Repeat the same procedure for all of your selected proteomes individually (amino acid type).

4. Perform a *blastp* search against your reference proteome.
   `blastp -outfmt 5 -query <ref-proteome.fa> -db <query-proteome.fa> -out <output file>`
   (Added -m 5 parameter returns an XML output which is easy to parse in biopython)

   4.1. Repeat this process for all of your query proteomes and create the output XML files.

5. Modify your script from Practical 3 to parse the XML output. The input will be like:
`<xml output> <tag for reference genome> <tag for target genome>`

   5.1. Reference and target genome tag parameters are used to assign the genome name for the query and target sequences

   `<tag for reference genome> <query protein id in reference genome>`

   `<tag for target genome> <best hit protein id in target genome>`

   5.2. The output for the script should be in one line with the following four columns:
   `human proteinI chicken proteinXV`

   `human proteinI mouse proteinXX`

   `human proteinII chicken proteinIX`

6. Combine the best hits into one cluster file
   6.1. Use the output of your BLAST parser as the input. Each query can have the best hits in different species.
   6.2. Group them in one line as in the example below, in which each line represents a cluster of orthologs:
   `human_proteinI chicken_proteinXV mouse_proteinXX ...`

7. Select 10 different ortholog clusters such that all of your prokaryotic genomes are included and acquire the corresponding protein sequences from the multi FASTA files.
   7.1. The result should be a number of multi FASTA files, one for each ortholog cluster.
   7.2. The multi FASTA file should include all sequences appearing in the cluster of orthologs (including the query sequence).

Exercise 2 - Metagene Approach

1. Make a multiple alignment of each of your multi FASTA cluster files in order to have one alignment for each cluster of orthologous genes.
   1.1. Concatenate the alignments into a single, long metagene. The end result should be a single FASTA file with a sequence for each bacterial genome consisting of the aligned genes from each genome.

1.2. Perform the tree reconstruction using Belvu. What does the tree look like?

1.3. Perform sequence bootstrapping on the metagene. Evaluate the quality of the reconstruction.

Exercise 3 - Consensus Reconstruction

1. Perform a tree reconstruction using Belvu for each individual alignment.
   1.1. Obtain 10 different trees, one for each ortholog cluster, in Newick format.
   1.2. How precise are those trees? Discuss.
   1.3. Point a few specific genes or classes of the genes that cause disagreement.
   1.4. Discuss the reasons of this disagreement.

2. Construct a consensus tree from the gene trees using Phylip example.
   For combining the tree, it is important that you use the species names as protein identifier in each tree.
   2.1. Use Phylip consense to construct a consensus tree;
      2.1.1. Put the tree files together into a file containing a list of the trees.

   Phylip should be preinstalled

   `phylip consense`

      2.1.2. Name this file intree.
      2.1.3. Phylip will read the file named intree in the present directory, ask you a few questions and then give you the consensus tree. Describe this process.