Comparative Genomics course, HT07, 3: Phylogenetic reconstruction

Lab instructions here. Questions can be directed to Kristoffer Forslund, krifo at-sign sbc dot su dot se.

Part 1 - DNA tree reconstruction

- You should investigate how your chosen genomes are related. For this, you need some gene which is present in all of them. 16SRNA might be a good idea. **For those of your genomes that are complete (eg. the bacteria and archaea),** use BLAST to find homologs to the 16SRNA E. coli gene (provided by your instructor). If you get several hits, take the best one from each genome.
  - More specifically **/afs/pdc.kth.se/home/k/krifo/Public/16S_ecoli.fasta** is the 16S sequence from E. coli.
  - Format a blast database for your genomes, so you can search them locally. This is done using the formatdb program in the blast package, with the right parameters.
  - **/afs/pdc.kth.se/home/e/esjolund/space/Public/software/blast/i386_co5/blast-2.2.16/bin/formatdb -p F -i <input file>** (or just formatdb if you are able to do module add blast) will run the program for a nucleotide dataset. What do these parameters mean?
  - Probably, it is easiest to put your genomes one after another (by c**at genome_0 genome_1 ... > genomes_all** or something) then make a single database of them.
  - Once that is done, you can use blast to query the database for the 16SRNA file. Take the best hit in each genome as the “actual” 16SRNA gene for that genome. Take out their sequences – preferably automatically rather than by hand – and gather them as entries in a fasta file.
  - One way to run it is **/afs/pdc.kth.se/home/e/esjolund/space/Public/software/blast/i386_co5/blast-2.2.16/bin/blastall -p blastn -d <database name> -i <query file>** (or just blastall if you can add the module). What do these parameters mean? Also, where do you find what you need in the output?
  - To parse the result from the run above, it is easiest if you use the **-m 7** parameter to get XML output, and then write a simple biopython blast parser using the NCBIXML module. The perfect way of solving this exercise would be to write such a script, and if you attempt it, we will help you.
  - If this is not possible, we can provide you with a preexisting script to parse the blast output. This will be considered an acceptable, although not excellent, solution to the problem as long as you answer the associated questions concerning that program in your report. If you decide to use this shortcut, tell us and we will supply you with the script, **blastResultParser.py**.
    - 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. How you do it is up to you, KALIGN may be a good idea.
  - That is, you can use KALIGN on the resulting sequence file to align it.
  - One way to do that is **/afs/pdc.kth.se/home/e/erison/Public/bin/kalign/2.03/kalign**

<infile> <outfile> (just kalign if you can add the module). What parameters exist, and would any of them make sense to apply?

- Perform tree reconstruction on your sequences. Ideally, do so using more than one method; for instance, distance based and parsimony.
  - One simple way is to supply the alignment to belvu using the proper parameters and requesting tree output. This can be done as **/afs/pdc.kth.se/home/e/erison/Public/bin/belvu/2.34/belvu -o tree <fasta file>**. What do these parameters mean? Which distance correction method does the program use? How would you request a different distance correction?
- One way to look at the tree is to use http://www.proweb.org/treeviewer/ and paste it in as “User-supplied Newick Tree”.

Part 2 - sequence bootstrapping

- What if your result is merely an artifact of sampling? The tree needs some form of significance measure. Using the phylogenetic software suite of your choice, construct a bootstrap consensus tree with bootstrap support values from the alignment you previously constructed.
  - The most recent version of belvu has a parameter to do a bootstrap analysis.
  - This is done by adding a **-b N** flag to the command to run belvu. What does N mean? What N do you choose and what consequences does that choice have?

Part 3 - being mean to the analysis software

- To test the effects of poor sampling (or database errors, or recombination) hands on, make a copy of your original alignment.
  - This can be done in any text editor.
- For each sequence, paste a copy of itself after it (such as ATTCGT->ATTCGTATTCGT).
- For one or more pairs of sequences, swap the second half of the sequence. 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.
- Perform tree reconstruction as before on these sequences. What tree do you get?
- Perform bootstrap-based analysis. Can you deduce from the bootstrap scores which taxa you shuffled the second half of the sequence for?