**LIFE 4141**

**Practical – from sequence to phylogeny**

**Practical designed by Drs Mary J O’Connell and Vladimir Ovchinnikov**

**School of Life Sciences, University of Nottingham.**

**Aim:** To apply the skills required to generate a gene tree from an initial single query sequence.

**Learning outcomes:**

1. Perform command-line operations that form the basis of many comparative genomics and phylogenomics studies.
2. Perform a standalone BLASTp
3. Generate multiple sequence alignments from variety of methods and compare the results using specific custom made statistics.
4. Identify the most significant MSA fit for a gene family.
5. Generate a phylogenetic tree.
6. Considering scale-up and automation.

**Data:** On the server in “LIFE4141\_Practicals” you have been provided with two files: (1) all protein coding regions from a range of species, and (2) a query protein.

Please take a moment to examine these files now.

**Setting up conda**

**(skip this section if you have already set it up)**

You will first need to download and install Miniconda3-py38\_4.10.3-Linux-x86\_64.sh

from here:

<https://docs.conda.io/en/latest/miniconda.html#linux-installers>

move the sh script to your home directory on the server and install by running

sh <*filename.sh*>

You should see this or similar, and then select yes to initialise miniconda3.

Text

Description automatically generated

Download the **yaml** file, and all other files required for this practical, from the github page

git clone https://github.com/lldean18/LIFE4141-gene\_tree

Then create the conda environment, and activate it as follows:

conda env create --name gene\_tree --file=gene\_tree.yaml

conda activate gene\_tree

[This ensures you are using the applications we have installed in the central shared resource on the server]

**Our goal is to generate and interpret a phylogenetic tree for this gene family.**

Steps 1-7 make up your pipeline, i.e. the set of associated command line operations required to achieve your goal.

***Step 1:* Identify all homologs of the query protein (using the E-5).**

makeblastdb -dbtype prot -in *input.fasta* -parse\_seqids -out *blast\_db*

[formats the database for blast, have a look at the output files generated]

Text

Description automatically generated

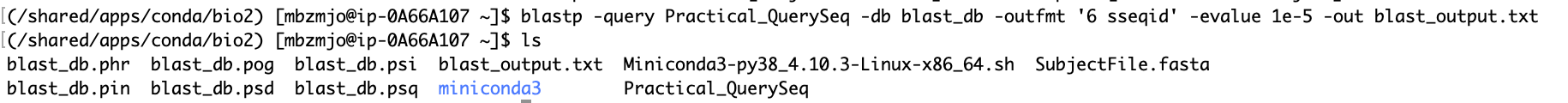
Observe the set of new files generated:



To perform blastp:

blastp -query *query.fasta* -db *blast\_db* -outfmt '6 sseqid' -evalue 1e-5 -out *blast\_output.txt*

[The second command line makes a file with subject IDs, the portion of the command that states <-outfmt '6 sseqid'> means that only subject IDs from blast results are shown]



Examine the output file.

***Step 2:* Extract from the subject database a single file that contains fastA formatted sequences of all homologs identified.**

sort *blast\_output.txt* | uniq > *file\_with\_IDs.txt*

[removes duplicates from the file with subject IDs… sometimes blast can have several hits for one query and one subject - when there is a big non-conserved region between two conserved ones]

blastdbcmd -db *blast\_db* -entry\_batch *file\_with\_IDs.txt* -outfmt "%f" -out *extracted\_proteins.fasta*

[extracts sequences (in fasta format) from blast database using filtered blast results].

Text

Description automatically generated

***Step 3:* Align the sequences using 2 of the most highly used multiple sequence alignment methods MUSCLE (Edgar, 2004), and MAFFT (Katoh et al, 2002).**

muscle -align *extracted\_proteins.fasta* -output *muscle\_alignment.fasta*

and

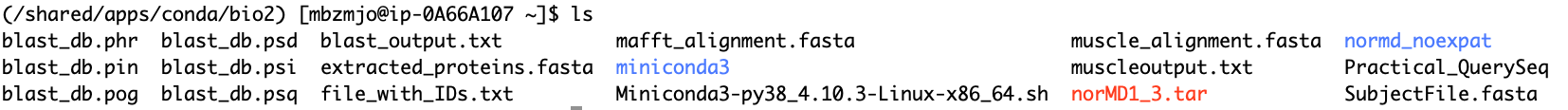
mafft --auto *extracted\_proteins.fasta* > *mafft\_alignment.fasta*

[generates MSAs using default parameters for your data, in your practice sessions feel free to play with these settings for both MUSCLE and MAFFT, their manuals are readily available online].

***Step 4:* Compare the statistical fit of the alignments to the data and select the best fit.**

[norMD is used to assess the quality of a MSA, a norMD score of >0.6 indicates a reliable alignment]

Note the input file name .. you will be doing this task for both of your alignments. First you will need to copy the compressed norMD file from the practical folder into your own directory, uncompress it and then inside that folder it makes, then run the “make” command.

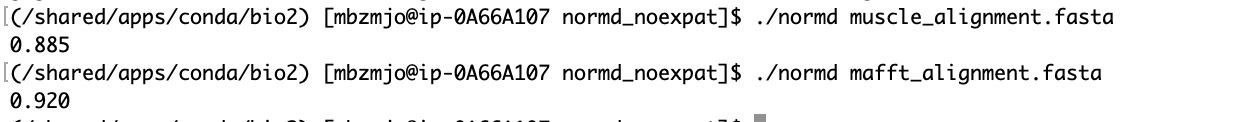


Then make a copy of each of your alignments in the normd\_noexpat folder.

Text

Description automatically generated with medium confidence

And go into that folder and run norMD on each alignment as follows:



./ normd *mafft\_alignment.fasta* > *mafft\_output.txt*

./normd *muscle\_alignment.fasta* > *muscle\_output.txt*

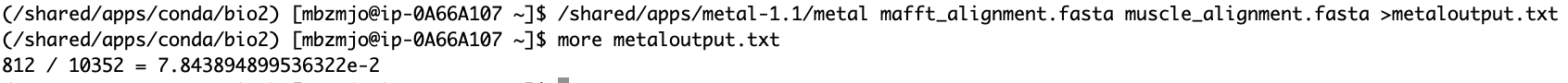
Return to the parent folder using *cd..* then run metal on your two alignments to quantify the level of difference between.

./metal *muscle\_alignment.fasta* *mafft\_alignment.fasta* > metal*output.txt*

[metAl (https://github.com/benb/MetAl) is used calculate a metric distance between alternative alignments of the same sequences. A comparison of these two aligners output is presented as XXX/YYY = 0.ZZZZ, where XXX is the score for first aligner, YYY is the score for second, and 0.ZZZZ is the difference.

e.g. If 0.ZZZZ = 0.149 for example then these two alignments would be 15% (i.e. significantly) different and based on the norMD score above you would choose the better fitting aligner for this dataset.

e.g. If 0.ZZZZ = 0.00003 then there is no significant difference between alignment methods, and provided norMD agrees they are good quality, then you randomly choose one].



***Step 5:* Generate a phylogenetic tree in IQTree (Nguyan L.-T., Schmidt HA, von Haeseler A, Minh BQ (2015) *Mol. Biol. Evol.*, 32:268-274).**

iqtree -s *mafft\_alignment.fasta* -b *100* -nt AUTO

iqtree -s *muscle\_alignment.fasta* -b *100* -nt AUTO

[run iqtree with input file called input.fasta, and 100 bootstraps, and in the background it is also running a modelFinder search to provide the best fitting substitution model for your data]

More options for running IQtree are found here <http://www.iqtree.org/doc/>



***Step 6:* Tree Visualisation**

To visualise your tree quickly just look at the end of the “X.iqtree” output file.

How might you do this?

Then if all looks well, you can download and install software on your own machine to visualise this in nicer format. To do this grab this file off the server and place on your local machine to visualise.

Use whatever file transfer protocol makes sense to you, perhaps your version of these:

SCP for file upload:

scp /Users/sian\_bray/Dropbox/Bray/001\_Teaching/000\_Bioinformatics\_Masters/000\_Final\_Materials/005\_Weeks\_1\_and\_2\_Assessment/Make\_Cell/protein.faa [sbzsmb@10.102.161.7](mailto:sbzsmb@10.102.161.7):.

SCP for file download:  
scp sbzsmb@10.102.161.7:/shared/Exam\_Submission.zip ~/

For Tree visualisation, I recommend the following :

ETE: <http://etetoolkit.org/treeview/>

Dendroscope3: <https://www.wsi.uni-tuebingen.de/lehrstuehle/algorithms-in-bioinformatics/software/dendroscope/>

FigTree: <https://github.com/rambaut/figtree/releases>

Or for a quick online solution : https://itol.embl.de/

***Step 7: Scale-up – discussion and implementation***

Today you had one query sequence and a small subject database. Let’s discuss and work out together how you might scale up to a larger number of queries, all of which needed to be passed through the same pipeline.

Problem solving: We had a repetitive task in our practical today, the NORMD task. If we had many many more query sequences then it would be really useful to be able to automate that process. Let’s work out how to do that now.

And then let’s run this solution for the two alignments you have in your directory from today.