For exercice 1 and 2 we provide a piece of DNA belonging to the prokaryotic species *Actinomyces johnsonii* named contigA.fasta.

Exercise 1.- Annotate a prokaryotic genome with a homology based model (blast)

Blast is probably one of the most useful and used tools that exist in genomics, and is used constantly by bioinformaticians and wet-lab scientists alike. It is normally implemented in websites such as NCBI or UniProt. The problem is that when you are working with a newly sequenced genome it will likely not be present in any website. While blast is not very useful as a genome annotation tool, we still will have a look on how to use it.

1.- The first step consists in converting our genome into a blast database. In order to do that you can use the following command:

formatdb -p F -i contigA.fasta

The -p F options indicates that our database contains a nucleotide sequence. For a protein database either put -p T or don't put anything.

2.- To perform a blast search we need at least one gene or protein to use as query. In this case we provide the proteome of a closely related species: *Actinomyces oris*. The proteome can be found in the file **protein_list.fasta**

3.- Now you can run the blast search using the following command:

blastall -p tblastn -i protein_list.fasta -d contigA.fasta -m 8 -b 1 -e 0.01 -o contigA.blast

Where:
-i indicates the list of proteins
-d indicates the genome database
-p is the kind of blast program we're going to use
-m 8 indicates the output format
-b 1 indicate we only want one result per protein
-e is the e-value threshold
-o is the output file

Open the file and have a look at the results, why do you think that this method is not useful for gene annotation? What do you think it can be useful for?

Exercise 2.- Annotate a prokaryotic genome using Glimmer3.

Glimmer3 is one of the most used gene annotation programs in prokaryotes. There are different versions for the annotation of eukaryotes, metagenomes and other, for more information about those you can check out this website: https://ccb.jhu.edu/software/glimmer/ In addition there is an on-line version that can be used: http://www.ncbi.nlm.nih.gov/genomes/MICROBES/glimmer_3.cgi

We are going to use glimmer to annotate the piece of DNA found in contigA.

Ab-initio programs use a parameters file to make gene predictions. The web version of Glimmer only

runs the program on standard prokaryotic parameters, ideally we want to give it a personalized set. In order to do that we first need to find a set of proteins to train the model. Glimmer allows us to make a first orf prediction and use that as input for the training program.

1.- Obtain the long ORFs from contigA:

tigr-glimmer long-orfs contigA.fasta contigA.longOrfs

Open the file and have a look at the results.

2.- Now you need to extract the nucleotide sequence from the predicted ORFs.

tigr-glimmer extract contigA.fasta contigA.longOrfs >contigA.longOrfs.fasta

As you see now the contigA.longOrfs.fasta file contains the sequences. You can count how many sequences you found by using this command:

grep -c “>” contigA.longOrfs.fasta

3.- Use the predicted proteins to create the parameter file that glimmer needs to work:

cat contigA.longOrfs.fasta | tigr-glimmer build-icm contigA.icm

4.- And finally make the glimmer prediction

tigr-glimmer glimmer3 contigA.fasta contigA.icm contigA.results

This command will output two files: contigA.results.detail and contigA.results.predict. The final coordinates for the prediction can be found in the .predict file. The .detail file shows all the genes that were considered whether they made it to the final annotation or not.

5.- Extract the ORFs obtained in the final prediction as shown above. How many genes have you predicted? Compare it to the ORF sequences.

6.- What would happen if we used a different species to build the parameters file? Use the fasta file containing 500 proteins of *Streptococcus suis* (seqs.Streptococcus.fasta) to obtain a second set of gene predictions. Compare the two of them.

Exercise 3.- Homology based annotation

contigB.fasta contains a fragment of chromosome VII of the fungus *Aspergillus nidulans* (also called *Emericella nidulans*). We are going to use it to have a look at a homology based annotation method. We are going to use exonerate (http://www.ebi.ac.uk/about/vertebrate-genomics/software/exonerate).

To run an annotation method based on homology we first need a set of proteins to search for. These are already provided in the file protein_list.fa

1.- Execute exonerate to check which options are available. (exonerate --help)

2.- Choose those options needed to run exonerate considering that the file contigB.fasta contains a genome.

3.- Have a look at the results. How many proteins do you have?
Can you play a bit with the outputs in order to obtain a clearer picture of what is going on?

4.- Repeat the analysis asking that just the best protein is annotated.

The most useful commend will look like this:

exonerate -q protein_list.fa -t contigB.fasta -m p2g --showalignment False --showvulgar False --showtargetgff -n 1

Where:
-q is the query sequence in this case the protein list.fasta
-t is the target (contig in our case)
-m is the model we're going to apply, in this case it's a shortcut for protein to genome
-n 1 show only the best result for each protein
and the following commands just modify the output: don't show the alignment (--showalignment), don't show vulgar annotation (--showvulgar False), show the gff annotation of the results (--showtargetgff)

Exercise 4.- Annotate a eukaryotic genome with Augustus.

We are going to use the same piece of DNA to predict the proteins in and ab-initio program (Augustus).

1.- As a reminder, contigB.fasta contains a fragment of chromosome VII of the fungus *Aspergillus nidulans (Emericella nidulans)*. First check whether this species is already available in Augustus.

To know the list of species for which we have parameters predicted you can user the --species=help command.

2.- Once you verify that it is, you can check the different options that Augustus provides. We can run Augustus by typing this:

augustus [parameters] --species=aspergillus_nidulans contigB.fasta

Where parameters are any of the choices the program offers.

3.- Run Augustus to obtain proteins with a complete gene_model, with the output including the gff3 (check augustus for the proper command)

4.- In order to extract the protein sequences we can use a script provided by augustus called getAnnoFasta.pl that you can find in your working folder. Notice that this script is also able to provide the CDS if augustus has been called with the --codingseq=on option.

5.- What happens if you use a different species to define your parameters? Repeat the augustus prediction using the parameters of Aspergillus fumigatus and of tomato. In both cases extract the protein sequences using getAnnoFasta.pl and save them in separate files.

6.- Compare the three results, what do you observe?

Exercise 5.- Search BRH between the proteins predicted and *A. fumigatus*.

We will search BRH between two closely related genomes, *E. nidulans* and *A. fumigatus*. The first step is to perform the blast search.

1.- Format the two files provided: EMENI.fa and ASPFU.fa

2.- Now perform the two searches, ASPNI.fa to ASPFU.fa and ASPFU.fa to ASPNI.fa (use the -m8 option so that the blast can be easily parsed)

3.- Process the results with the script get_BRH.py.

python get_BRH.py -s1 ASPNI.fa -s2 ASPFU.fa -h1 ASPNI2ASPFU -h2 ASPFU2ASPNI -o BRH.ASPFU.txt

4.- Look at the results. How many BRH did you find?
5.- Now repeat the process with the yeast proteome. What did you find? Which species is better to annotate your genome?

Exercise 6.- Perform a MCL clustering between the proteins obtained in exercise 4 (EMENI.fa) and other Aspergillus proteins found in the file (Aspergillus.fa)

1.- You need the results of an all against all blast search, therefore you'll have to perform two blast searches as before or alternatively you can pool all the proteins in a single file and perform a blast search from this file to itself.

2.- Perform the mcl clustering.

mcl fileName --abc -o outfileName -I inflation_value

Where the inflation value affects the cluster granularity, basically it will make the clusters larger or smaller. The smaller the value the largest the groups will be. The value ranges from 1.2 to 5.0.

3.- Perform the clustering with different inflation values: 1.2, 2.0 and 5.0. Have a look at the results, are there differences?

Exercise 7: Reconstruct phylogenetic trees for each of the proteins predicted.

The phylogenetic reconstruction process can be performed in three steps: homology search, alignment reconstruction and phylogenetic reconstruction. For this exercise we assume that the homology search has already been performed and each of the proteins found in ASPNI is found in a different file and contains its sequence and the sequences of its homologs.

1.- Perform a multiple sequence alignment using muscle:

muscle -in fileName -out fileName.alg

2.- Trim the alignment using trimal

trimal -in fileName.alg -out fileName.trim -gt 0.1

Take note that trimal is able to change the format of the alignment which is very useful if we want to perform a phylogenetic tree with other programs such as PhyML or RAxML.

3.- Reconstruct phylogenetic tree using fasttree

fasttree fileName.trim >fileName.tree