Comparative Genomics 2018

Practical 2: Gene Prediction

Group 9
Martínez Hernández, Marina
Pérez Gómez, Fernando

Summary
In this practice we made use of two powerful gene prediction tools: Glimmer and GENSCAN. In the first part, we used Glimmer and the five genomes that were given to us to predict the genes contained in such genomes and then obtain the protein sequences that these genes were coding for. We also plotted the histograms of the gene size distribution for each genome, which show how the eukaryote genome contains more genes with a bigger size. In the second part, we focused on this eukaryote genome and used the GENSCAN tool. With this, it takes a very short time to directly obtain the nucleotide and protein sequences of the genome, as well as to obtain a graphical output of how the exons of this genome are distributed along the DNA strand. Overall, we gained a first insight into how gene prediction can be computationally done.

Key questions to answer
1. Glimmer
2. GENSCAN

Attachments
Activity 1.9: 04.fa.txt.pfa, 16.fa.txt.pfa, 18.fa.txt.pfa, 34.fa.txt.pfa and 49.fa.txt.pfa.
Activity 2.1: proteins34.out, proteins34.txt and nucleotides34.txt.

EXERCISE 1 - Glimmer

Please explain the parameters you are using to run Glimmer.

-n: when used with the long-orfs tool, this parameter is added so that the output file that is generated does not contain data regarding the settings of the program that is used, but the coordinates of the ORFs.
-t: when used with the long-orfs tool, this parameter is added to set a threshold based on the entropy distance score. In this case, those genes with a score less than 1.15 will not be taken into account.
-r: when used with the build-icm tool, the r or reverse parameter is added to specify that the ICM will be built with the reverse strings of those that have been input.
-o50: when used with the glimmer3 tool, this parameter establishes a maximum overlap length. In this case, this length is set equal to 50.
-g110: when used with the glimmer3 tool, this parameter establishes the minimum gene length. In this case, this length is set equal to 110 nucleotides.

For the 5 genomes that have been provided:

1.1. Find long ORF from genome.

With this code, the ORFs coordinates for each of the 5 genomes are generated.

1.2. Extract long ORF.

This makes it possible to get the nucleotide sequences of each ORF that has been previously generated.

1.3. Prepare training set.

This creates an interpolated context model (ICM) which is basically a probability model of the coding sequences.

1.4. Start Glimmer.

The glimmer3 tool is run.

1.5. Long ORFs are provided to construct the training set, what other two sources of sequences can be used instead of or in addition to long ORFs?

The sequences can also be obtained from genes that are known in that genome or from genes that have a very high similarity in other strains.

1.6. 6. Is Glimmer suitable for all genomes? Why?

No. It is only suitable for prokaryotic genomes. Because eukaryotic genomes are more complex than prokaryotic genomes: they contain introns and this makes it more challenging for the program to find the ORFs, since it is necessary to take into account that some regions will not be coding for genes.

1.7. Make a histogram of predicted gene lengths for each genome in R.

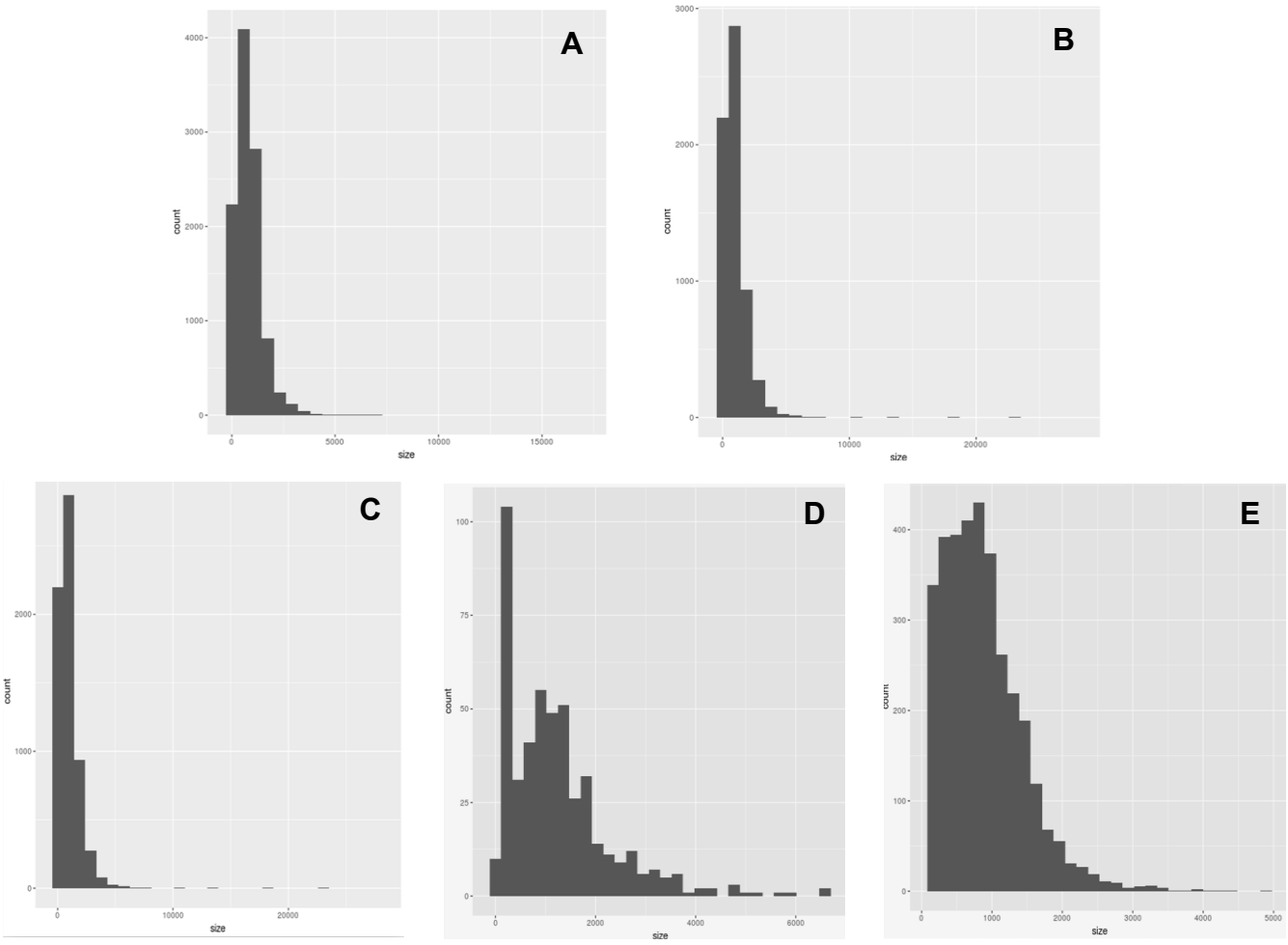

Figure 1. Distribution of predicted gene sizes in genome (A) 04.fa.txt, (B) 16.fa.txt, (C) 18.fa.txt, (D) 34.fa.txt and (E) 49.fa.txt.

1.8. Do all gene sizes follow the same distribution in all genomes?

No. The histogram of the predicted gene lengths in file 34.fa.txt does not follow the same distribution as the other ones. This is due to the fact that this is a eukaryotic genome and Glimmer is not very suitable to analyze this kind of genomes. Also, it makes sense that the eukaryotic genome histogram represents genes with a larger size than the prokaryotic histograms, since eukaryotic organisms have longer genes than prokaryotic and this can be seen in the figures above.

1.9. Extract the protein sequences from the predicted genes obtained. Use the script parseGlimmer.py.2 available in the script directory.

The files containing the protein sequences are attached in the e-mail with the names 04.fa.txt.pfa, 16.fa.txt.pfa, 18.fa.txt.pfa, 34.fa.txt.pfa and 49.fa.txt.pfa.

EXERCISE 2 - GENSCAN

Using the eukaryote genome (34.fa.txt):

2.1. From GENSCAN output, extract the amino acid and nucleotide sequences and make separate files for each.

The files are attached in the e-mail: proteins34.out and proteins34.txt are the files containing the amino acid sequences (same content and different format); nucleotides34.txt is the file containing the nucleotide sequences.

2.2. Create the PostScript (graphical) output, which is a diagram of the locations and DNA strand of all predicted exons/genes.

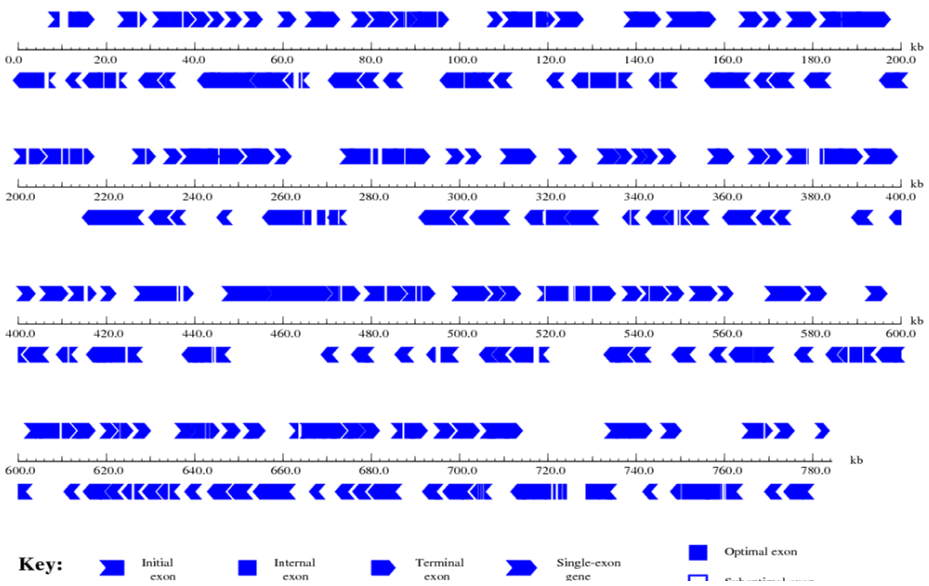

Figure 2. Diagram showing the location of the predicted exons.

2.3. Using BLAST and the nucleotide sequences extracted from GENSCAN output, tell me the protein names of the first two nucleotide sequences.

Using BLASTx to figure out the protein that the first two nucleotide sequences code for, we got:

- For the first sequence (>./34.fa.txt|GENSCAN_predicted_CDS_1|5589_bp), the corresponding protein name is Y' element ATP-dependent helicase protein 1 copy 6 [Saccharomyces cerevisiae S288C].

- For the second sequence (>./34.fa.txt|GENSCAN_predicted_CDS_2|1812_bp), the corresponding protein name is Cos1p [Saccharomyces cerevisiae S288C].