Comparative Genomics 2018

Final Project

All forms of plagiarism are forbidden, and if detected it will result in a lower grade.

PURPOSE

You will piece together each practical into an overall story and present this as your final project. However, instead of using given tools or databases for a comparative genome analysis, you will analyse your genomes with your own written scripts. The analysis will contain the calculation of nucleotide frequency and GC content, the prediction of possible ORFs in your genomes, a calculation of distance matrices to find the evolutionary relationship between the genomes, and a visualisation of this relationship as a tree.

Additional to your presentation you will write and submit a report containing a summary of your presented results. Please also attach your three scripts in three separate files to your final project report.

PRESENTATION OF YOUR FINAL PROJECT

The presentations will be held in the computer lab; you can show your programs and results directly via the projector. Simple PowerPoint/OpenOffice-Impress presentations should be prefered where applicable. Each presentation should take about 15 minutes.

Your presentation should include:

1. Summary of your five genomes incl families, species, chromosomes, genome length.
2. The results of applying your three Python scripts to your five genomes:
    a. The GC, nucleotide and dinucleotide frequencies for the five genomes & amino acid and diamino acid frequencies for your predicted proteins (output of your ORF finder) using your statistics tool
        i. Present the formulas you used for the GC content and frequency calculation
        ii. If applicable use charts for the results you obtained
    b. The predicted ORFs in your five genomes by your ORF predictor (include some examples and some statistics, e.g.
        i. Which basic assumption on ORFs did you apply?
        ii. How many ORFs per genome did you predict?
        iii. Assess the accuracy of your ORF predictor by comparing its predictions to a reference. Specify the used reference, accuracy measure, criteria for defining true positives, false positives, false negatives, etc..
        iv. Which additional improvements would be possible to increase the accuracy of your ORF predictor?
    c. The distance matrix for your genomes computed using your third script with the output of your first script
        i. Which distance method did you use and why?
        ii. Construct and present a phylogenetic tree based on the calculated distance matrix. Which tree building method did you use and why?
        iii. Compare shortly this tree with the trees you created during the various practicals

YOUR PYTHON SCRIPTS

1. Statistics tool

The tools should calculate statistics for a genome (DNA) and a proteome.

- GC content and nucleotide (dinucleotide) frequency in your genomes
- amino acid (diamino acid) frequencies in your proteomes

For the GC content, decide whether or not to count undefined nucleotides (Ns) as part of the sequence for the purpose of computing the frequency.

For dinucleotides, both reading frames should be considered. That is, for the sequence

AGCCCAAGACACC

your results should be something like

#AG = 2/12

#GC = 1/12

#CC = 3/12

#CA = 2/12

#AA = 1/12

#GA = 1/12

#AC = 2/12

2. ORF finder

The ORF finder should predict Open Reading Frames (ORFs) in your five genomes.

The input should be a genome file in FASTA format; the output should also be a file in FASTA format with separate entries for each ORF gene sequences and unique names identifying these ORFs.

3. Distance matrix tool

The tool should compute the distance between two genomes from the DNA statistic above. The distance matrix is then used to create a species tree.

DISTANCE CALCULATION METHODS

There are many ways of defining distances between biological objects. In this assignment you should compute the distance with regards to various requirements.

A distance D (g1, g2) between two genomes g1 and g2 must satisfy two basic criteria:

D (g1, g1) = D (g2, g2) = 0

that is, everything will be at zero distance from itself, and

D (g1, g2) = D (g2, g1)

that is, distances are the same regardless of which direction you look at them from. These properties mean that a distance matrix will always

- have zero as the diagonal elements
- be symmetric, so that it is mirrored in the diagonal

Any function D (g1, g2) that fulfills these two conditions is a distance function.

Examples of a distance function could be the pure distance between GC-values

$$D = \sqrt{(GC\ of\ genome\ 1 - \ GC\ of\ genome\ 2)^{\ 2}}$$

or the distance between nucleotide frequencies as

$$D = \sqrt{(G_1 - G_2)^2 + (C_1 - C_2)^2 + (A_1 - A_2)^2 + (T_1 - T_2)^2}$$

where G_1 , for instance, is the G frequency of genome 1, and the others are named correspondingly.

The expression could be extended further to use other statistics such as dinucleotide frequencies, or the angle between the frequency lists, or you could add some scaling or normalizing to make the distances better suited for the analysis.

Python does square root of expression as **math.sqrt (expression), and square of expression as **expression**2**. To use **math.sqrt ()**, you must import **math**.