**Final project report**

1. Summary of your five genomes incl families, species, chromosomes, genome length.

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **File name** | **Family** | **Species** | **Chromosomes** | **Genome length** | **Organism type** |
| 16.fa.txt | Planctomycetaceae | *R. baltica* SH1 | ? | 7149689 | Prokaryotic |
| 20.fa.txt | Thermotogaceae | *T. maritima* strain Tma100 | Complete Genome | 1869610 | Prokaryotic |
| 29.fa.txt | Saccharomycetaceae | *S.cerevisiae* S288C | IX | 439888 | Eukaryotic |
| 44.fa.txt | Leuconostocaceae | *L. gelidum* JB7 | Complete Genome | 1893500 | Prokaryotic |
| 47.fa.txt | Neisseriaceae | *N. meningitidis* alpha710 | Complete Genome | 2242948 | Prokaryotic |

2. The results of applying your three Python scripts to your five genomes:

1. 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.

The python script stat\_tool\_1.py was used to evaluate both GC, nucleotide, and dinucleotide frequencies for the 5 genomes.

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| genome: 16.fa.txt  GC-content: 0.5548  #A = 1589939/7149689  #C = 1981614/7149689  #T = 1592923/7149689  #G = 1985204/7149689  #N = 9/7149689  #AA = 359096/7149689  #AC = 421691/7149689  #AT = 387766/7149689  #AG = 329767/7149689  #CA = 489129/7149689  #CC = 388521/7149689  #CT = 330792/7149689  #CG = 704454/7149689  #TA = 105255/7149689  #TC = 543628/7149689  #TT = 359547/7149689  #TG = 493028/7149689  #GA = 544841/7149689  #GC = 559056/7149689  #GT = 423352/7149689  #GG = 389509/7149689 | genome: 20.fa.txt  GC-content: 0.4625  #A = 504109/1869610  #C = 426011/1869610  #T = 500833/1869610  #G = 438657/1869610  #N = 0/1869610  #AA = 119138/1869610  #AC = 100392/1869610  #AT = 112580/1869610  #AG = 129961/1869610  #CA = 110613/1869610  #CC = 80193/1869610  #CT = 127257/1869610  #CG = 92039/1869610  #TA = 67219/1869610  #TC = 160752/1869610  #TT = 117679/1869610  #TG = 113839/1869610  #GA = 165101/1869610  #GC = 68766/1869610  #GT = 101972/1869610  #GG = 85445/1869610 | genome: 29.fa.txt  GC-content: 0.3890  #A = 134339/439888  #C = 85465/439888  #T = 134423/439888  #G = 85661/439888  #N = 0/439888  #AA = 34423/439888  #AC = 23399/439888  #AT = 38074/439888  #AG = 26070/439888  #CA = 28585/439888  #CC = 14956/439888  #CT = 25992/439888  #CG = 13236/439888  #TA = 31581/439888  #TC = 27250/439888  #TT = 34410/439888  #TG = 28797/439888  #GA = 27377/439888  #GC = 17163/439888  #GT = 23563/439888  #GG = 14961/439888 | genome: 44.fa.txt  GC-content: 0.3668  #A = 600998/1893500  #C = 346142/1893500  #T = 598020/1893500  #G = 348339/1893500  #N = 0/1893500  #AA = 159766/1893500  #AC = 105514/1893500  #AT = 190601/1893500  #AG = 88325/1893500  #CA = 138243/1893500  #CC = 54062/1893500  #CT = 87166/1893500  #CG = 58970/1893500  #TA = 149211/1893500  #TC = 94947/1893500  #TT = 159111/1893500  #TG = 138835/1893500  #GA = 96986/1893500  #GC = 83918/1893500  #GT = 105227/1893500  #GG = 54283/1893500 | genome: 47.fa.txt  GC-content: 0.5169  #A = 540971/2242948  #C = 577262/2242948  #T = 542592/2242948  #G = 582122/2242948  #N = 0/2242948  #AA = 136187/2242948  #AC = 116880/2242948  #AT = 136676/2242948  #AG = 97600/2242948  #CA = 140876/2242948  #CC = 121349/2242948  #CT = 96573/2242948  #CG = 195774/2242948  #TA = 83819/2242948  #TC = 125051/2242948  #TT = 136887/2242948  #TG = 143111/2242948  #GA = 126462/2242948  #GC = 191292/2242948  #GT = 118731/2242948  #GG = 122793/2242948 |

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Proteome: 16.fa.txt.pfa  Amino acid frequency:  #A = 201732/2136223  #C = 23629/2136223  #D = 137859/2136223  #E = 133128/2136223  #F = 77611/2136223  #G = 162099/2136223  #H = 47555/2136223  #I = 107341/2136223  #K = 73760/2136223  #L = 201366/2136223  #M = 49888/2136223  #N = 72372/2136223  #P = 111032/2136223  #Q = 86631/2136223  #R = 140785/2136223  #S = 151835/2136223  #T = 124754/2136223  #V = 153409/2136223  #W = 32759/2136223  #Y = 46674/2136223 | Proteome: 22.fa.txt.pfa  Amino acid frequency:  #A = 34721/591793  #C = 4060/591793  #D = 29409/591793  #E = 52921/591793  #F = 30824/591793  #G = 40986/591793  #H = 9341/591793  #I = 42568/591793  #K = 44977/591793  #L = 59554/591793  #M = 13521/591793  #N = 21384/591793  #P = 23554/591793  #Q = 11932/591793  #R = 32590/591793  #S = 33353/591793  #T = 26779/591793  #V = 51533/591793  #W = 6553/591793  #Y = 21233/591793 | Proteome: 29.fa.txt.pfa  Amino acid frequency:  #A = 6050/107024  #C = 1372/107024  #D = 6057/107024  #E = 7014/107024  #F = 4895/107024  #G = 5233/107024  #H = 2149/107024  #I = 6782/107024  #K = 7751/107024  #L = 9990/107024  #M = 2171/107024  #N = 6564/107024  #P = 4721/107024  #Q = 4427/107024  #R = 4670/107024  #S = 9846/107024  #T = 6705/107024  #V = 5992/107024  #W = 1099/107024  #Y = 3536/107024 | Proteome: 44.fa.txt.pfa  Amino acid frequency:  #A = 43737/543724  #C = 1137/543724  #D = 30949/543724  #E = 28192/543724  #F = 24271/543724  #G = 35566/543724  #H = 11531/543724  #I = 43306/543724  #K = 32761/543724  #L = 53355/543724  #M = 14905/543724  #N = 28408/543724  #P = 18236/543724  #Q = 25038/543724  #R = 20988/543724  #S = 33163/543724  #T = 34858/543724  #V = 39122/543724  #W = 5733/543724  #Y = 18468/543724 | Proteome: 47.fa.txt.pfa  Amino acid frequency:  #A = 63433/629939  #C = 6710/629939  #D = 32866/629939  #E = 38300/629939  #F = 26326/629939  #G = 48429/629939  #H = 14054/629939  #I = 36567/629939  #K = 35954/629939  #L = 61782/629939  #M = 15093/629939  #N = 25907/629939  #P = 26906/629939  #Q = 25467/629939  #R = 35348/629939  #S = 35174/629939  #T = 33063/629939  #V = 42120/629939  #W = 7469/629939  #Y = 18971/629939 |

* 1. Present the formulas you used for the GC content and frequency calculation.

The formula used for GC content was (seq.count(“C”) + seq.count(“G”))/len(seq). The formula used for frequency calculation was seq.count(nucl)/len(seq), with nucl being an iterative element for a defined set of nucleotides/ dinucleotides.

* 1. If applicable use charts for the results you obtained.

1. 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?

The basic assumption of an Open Reading Frame (ORF), is that it is a stretch of DNA with a start codon (specify seq) on the 5’ end, and a stop codon on the 3’ end, which has the possibility to encode a protein coding sequence (1). Therefore, our ORF finder searches for sequences flanked by a start codon and stop codon, across all 6 reading frames; 3 per strand, forward and reverse.

Talk about algorithm? (specified for lengths >300 for prokaryotes and 100-1500 for eukaryotes).

Talk about removal of overlap.

ii. How many ORFs per genome did you predict?

To report numbers with overlap removed!

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..

Choose performance measure wisely -> Matthew’s Correlation Coefficient (MCC).

Gene counts from reference (NCBI BLAST and taxonomy):

16.fa.txt - 7406

20.fa.txt – 5806

29.fa.txt – 232

44.fa.txt – 1967

47.fa.txt - 2104

iv. Which additional improvements would be possible to increase the accuracy of your ORF predictor?

The first improvement that comes to mind will involve improving the specificity of the ORF predictor, by utilising more information available in the literature involving transcription and translation. This could include requiring that ORFs have upstream promoter sequences like the Pribnow box (TATAAT) (2), or Shine-Delgarno sequence (AGGAGG) (3). We briefly tried implementing these specifications, which resulted in a drastic reduction in the number of ORFs found, much lower than the reference. The suspected cause is the script being overly specific with the upstream length that the promoter sequences should be located at. Given more time a script could have been developed to allow for a larger range of such upstream lengths.

Another commonly used method to increase the accuracy of the finder is the Hidden Markov Model (HMM) architecture, found in the hugely popular GENSCAN (4), which overcomes the complexity of splicing in eukaryotic genomes.

1. The distance matrix for your genomes computed using your third script with the output of your first script.
   1. Which distance method did you use and why?

The distance method implemented was based on the pure distance between the various GC percentage values of the gene sets.

* 1. Construct and present a phylogenetic tree based on the calculated distance matrix. Which tree building method did you use and why?

The tree was constructed using Belvu, with the UPGMA method??

* 1. Compare shortly this tree with the trees you created during the various practicals.

References:

1. Zvelebil, M., and Baum, J. (2008). Understanding bioinformatics (New York: Garland Science).
2. Pribnow, D. (1975). Nucleotide sequence of an RNA polymerase binding site at an early T7 promoter. Proceedings Of The National Academy Of Sciences *72*, 784-788.
3. Malys, N. Mol Biol Rep (2012) 39: 33. https://doi.org/10.1007/s11033-011-0707-4.
4. Burge, C. and Karlin, S. (1997) Prediction of complete gene structures in human genomic DNA. *J. Mol. Biol.***268,**78-94.