Final Assignment Report

Javier Lanillos[1], Revant Gupta[2], Siddharth Tomar[3]

Abstract
This report summarizes the work performed for the final project, which is mainly divided into sections organized as well as the three principal tasks assigned are described. Every section describes the details of their specific goals as well as the results and relevant discussions. The first section shows and describes the results from calculating the frequency of nucleotide, dinucleotide, amino acid and diamino acid contents. The second part of this report focuses in the ORF Finder predictor, which first summarizes the basis and general rules by which the predictor has been built, then it explains how the developed algorithm works and validates the results from comparison to other tools like GLIMMER for the prokaryotic genomes and GeneScan for the eukaryotic genome. The last section reports the third goal of the project, calculating distances between the different genomes under certain chosen criteria and showing the corresponding results.

Keywords
Comparative — Genomics — Phylogeny

[1] *javi.lanillos@gmail.com*
[2] *aron0093@gmail.com*
[3] *tomar@kth.se*

Contents

Introduction

The initial dataset available to accomplish the assignment consists of five genomes, whose different species and kingdoms have been identified and summarized together with the nucleic acid content information in the Table 1

1. DNA and protein statistics

We evaluate the nucleotide, dinucleotide, amino acid and diamino acid frequencies along with the GC content per genome. We also comment on some peculiar patterns we noticed as well some inferences that were drawn from the plots.

The input genome files contain one of the strands of the DNA genome for each species and we counted the frequency of nucleotides present at the input genome file (shown in Fig 1.). It is observable that the number of Guanine and Cytosine nucleotides is almost equal to each other as the same happens with Adenine and Thymine. This means that a single strand of these genomes contains more or less the same A,T content and G,C content though the levels for A,T and G,C are different. These two observations were explained by Chargaff through his parity laws[1].

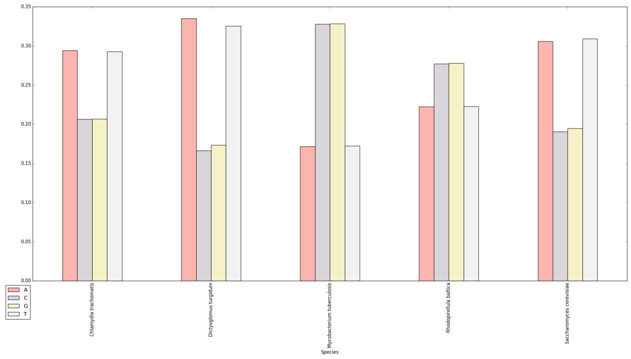

Figure 1. Nucleotide frequencies per genome

Figure 2 represents the dinucleotide frequencies counted in the genomes. For Chlamydia trachomatis, Dictyoglomus turgidum and Saccharomyces cerevisiae, the combinations including only Adenines and Thymines are the most common

File ID	Organism	Nucleic Acid Type	Length (bp)	Kingdom/Type
05.fa.txt	Chlamydia trachomatis	DNA circular	1042588	BCT
08.fa.txt	Dictyoglomus turgidum	DNA circular	1855560	BCT
14.fa.txt	Mycobacterium tuberculosis	DNA circular	4365724	BCT
16.fa.txt	Rhodopirellula baltica	DNA linear	7149689	BCT
25.fa.txt	Saccharomyces Cerevisiae	DNA linear	576874 (1chr)	PLN

Table 1. Summary of genomes utilized in the project

(AA, TT AT, TA), whereas Mycobacterium tuberculosis and Rhodopirellula baltica show higher frequency of dinucleotides containing only Cytosine and Guanine (CG and GC).

These results match consistently with the GC content of each organism, as it can be seen in Figure 3.

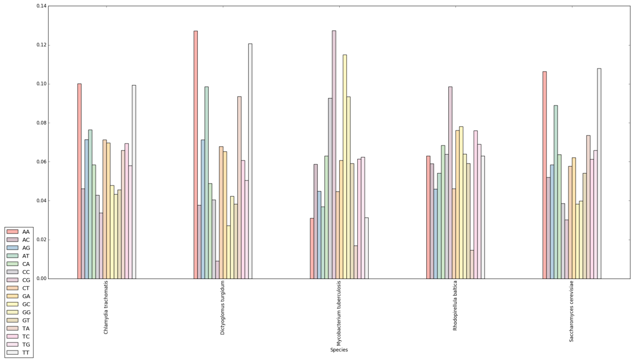

Figure 2. Dinucleotide frequencies per genome

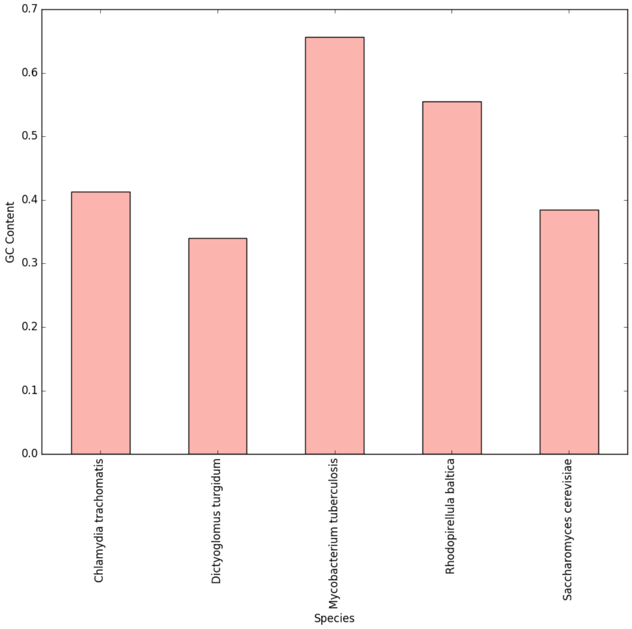

Figure 3. GC Content per genome

We can see that the two highest GC content correspond to R. baltica and M.Tuberculosis which actually contain the largest genomes among the others. As well as the highest GC content, we show later that these two has the highest number of predicted orfs/largest proteome. This could be an indicator of the link between GC content and coding sequences. However, as explained later, our ORF Finder predicts many false positives due to its strategy to find possible ORFs.

We have calculated the amino acid and diamino acid content of these genomes for each different reading frame. Each of the species show a characteristic and invariant pattern, regardless of the reading frame followed.

The amino acid frequencies are similar in all reading frames per species. Our understanding is that given the unique frequencies of nucleotides for each genome, the nucleotide distribution of each genome varies from each other. However, shifting the reading frame itself is analogous to the removal of a nucleotide at the beginning and shifting it to the end (for circular genomes). The shifting of one nucleotide in a genome containing hundreds of thousands of nucleotides does not impact the overall distribution of nucleotides. This of course leads to a near-identical distribution of codons in each reading frame, hence similar amino acid frequencies.

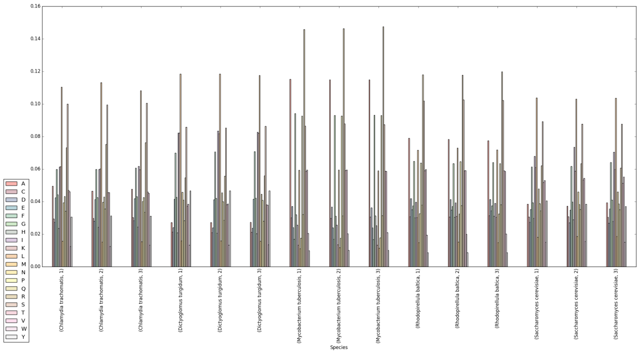

Figure 4. Amino acid frequencies per genome

Chlamydia trachomatis, Dictyoglomus turgidum and Saccharomyces cerevisiae share a common feature which are the two most common residues found: Leucine and Serine. Also, for both Mycobacterium tuberculosis and Rhodopirellula baltica, the two most common residues found are Alanine and Arginine.

Finally, Figure 5. shows the diamino acid content for the five species at the three reading frames in the forward strand direction. Each of the thin columns represents one of the possible combinations found between two amino acids. We have not shown a legend to illustrate what means each column, but we do prefer to highlight the pattern repetition between the three patterns that each species have. So, from this plot, we see that the three reading frames contain fairly the same diamino acid content. This result is very peculiar, although it seems to be correct after checking the calculations twice.

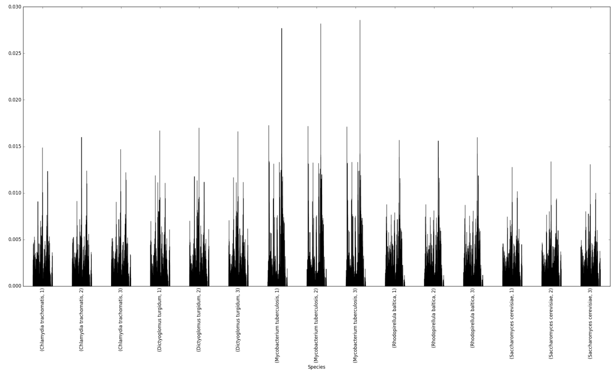

Figure 5. DiAmino acid frequencies per genome

2. ORF generator

2.1 Background

By definition, and Open Reading Frame (ORF) is considered as a part of a reading frame that has the potential to be translated into one or more proteins, thus, including a coding DNA sequence corresponding to a gene or some genes[2]. We have made use of several properties to elaborate an algorithm able to identify potential ORFs along our input genomes:

- An ORF does not contain stop codons and it is finished by the first STOP codon found downstream the initiation codon (usually UAA, UAG or UGA).

- The length of an ORF will be greater than 100 bp at least. The length distribution of the genes has been calculated among most of the available genomes and species[3].

Not only the definition of an ORF is enough to accept it as so, but there are also some other main elements present genes within an ORF to undergo transcription of genes. These elements are specific for each species although there are some consensus among them. Also, there exist important differences between the configuration and presence of this elements in Prokaryotic versus Eukaryotic organisms:

- The identification of Ribosomal Binding Sites (RBS) in Eukaryotes is used to determine the site of translation initiation in an unannotated sequence, called N-terminal prediction. This is especially useful when multiple start codons are situated around the potential start site of the protein coding sequence[4].

- Furthermore, there are sequences associated with translation that can be used to identify regions of the genome that may contain potential proteins. For eukaryotes we used one such sequence. The Kozak consensus sequence appears on eukaryotic mRNA as (gcc)gccRccAUGG plays a role in the initiation of the translation process[5].

- Shine-Dalgarno (SD): is a ribosomal binding site in bacterial and archaeal messenger RNA, generally located around 8 bases upstream of the start codon AUG[6].

- The Pribnow box is the sequence TATAAT of six nucleotides and plays an important role of a promoter site on DNA for transcription to occur in Prokaryotes like the bacteria[7].

- The TATA-box in Eukaryotes, as part of the promoter region with a consensus sequence TATAAA[2] We have also considered the CAAT-box promoter sequence[8].

It is well known that each species has characteristic pattern of use of synonymous codon, different patterns of use of codons in strongly versus weakly expressed genes and organisms with high GC content have a bias towards G and C in the third codon position. However we have in the current version of the ORF predictor not included this constraint. We feel that deciding a cutoff or score based on GC content of an ORF requires deeper investigation.

2.2 Algorithm

As described before, we have worked with four Prokaryotic genomes and one Eukaryotic genome. Eukaryotic genomes contain a wider complexity due to genome length and higher different content such as exonic and intronic regions and different transcription elements. Thus, Prokaryotic genomes seem to be easier to predict ORFs, for example, they show no modification from the DNA through the mRNA to give the final protein, so that, it is quite suitable to think of finding the longest ORFs and constraint the results above a minimum length.

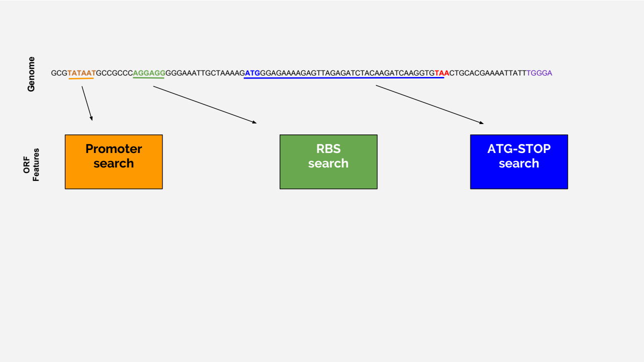
Genome
GCGTATAATGCCGCCCAGGAGGGGGAAATTGCTAAAAGATGGGAGAAAAGAGTTAGAGATCTACAAGATCAAGGTGTAACTGCACGAAAATTATTTGGGA
ORF Features
Promoter search
RBS search
ATG-STOP search

Figure 6. Features used to predict ORFs

The first step of our ORF Finder consists of finding long (¿100 bases) sequences starting by ATG and ending at the most proximal STOP codon downstream, By default, only segments longer than 100 bases are allowed and the other are

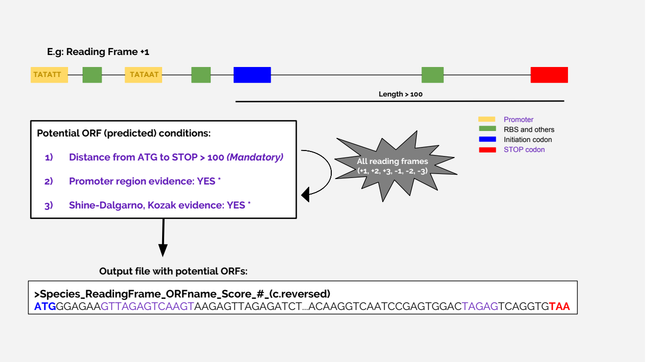
E.g: Reading Frame +1
TATATT
TATAAT
Length > 100
Promoter
RBS and others
Initiation codon
STOP codon
Potential ORF (predicted) conditions:
1) Distance from ATG to STOP > 100 (Mandatory)
2) Promoter region evidence: YES *
3) Shine-Dalgarno, Kozak evidence: YES *
All reading frames (+1, +2, +3, -1, -2, -3)
Output file with potential ORFs:
>Species_ReadingFrame_ORFname_Score_#_(c.reversed)
ATGGGAGAAGTTAGAGTCAAGTAAGAGTTAGAGATCT...ACAAGGTCAATCCGAGTGGACTAGAGTCAGGTGTAA

Figure 7. Constraints used to predict ORFs

discarded but the cut-off length can be customized as desired.

The next step of the algorithm consists of looking for more pieces of evidence that can support that our queried ORF could be accepted as a potential ORF. Each of those sequences longer than the cut-off individually undergo a search of the characteristic genetic elements such as RBS and promoter regions. These elements are searched within certain specific ranges upstream the ATG position, and for the case of Prokaryotes, sometimes also downstream:

As reported in the literature, there are not fixed promoter and other sequences but it is a consensus[9].

Thus, this algorithm accepts perfect matches with the consensus sequences and also aligns others that do not match perfectly, providing all of them for a score. For example, if during the search of a promoter sequence like TATAAT is hit upstream the ATG codon, the algorithm will check if the first four letters ('TATA') match with and will try to align the rest, providing with a lower score than the perfect matches to the consensus sequences. Finally, the best scored sequence (closer to the consensus region and better aligned) will be finally chosen among some others, if exist. The same approach for RBS and others is performed as for the promoter sequences. This algorithm also takes into account deals with possible invertible RBS sequences. Finally, each ATG-STOP sequence longer than 100 nucleotides by default which contained any or both of evidences (RBS or a promoter sequence), will be and included into the final FASTA file of potential ORFs

	Consensus Sequence name	Sequence
Prokaryote	Pribnow sequence	TATAAT (TSS - 10')
		TTGACA (TSS - 35')
	Shine-Dalgarno	AGGAGG (ATG - 8')
Eukaryote	TATA-box	TATAAA (TSS - 35')
	Kozak sequence	gcc(G/T)ccAUGG

Table 2. Summary of features used to predict ORFs

2.3 Results and validation

Organism	Score 1	Score 2	Score 3
Chlamydia trachomatis	220	9	9116
Dictyoglomus turgidum	180	20	14688
Mycobacterium tuberculosis	184	44	58186
Rhodopirellula baltica	48	35	107602
Saccharomyces Cerevisiae	27	4390	0

Table 3. Predicted ORFs per score class

In order to validate our results and compare them to other algorithms, we have translated both our ORF predictions and GLIMMER/GeneScan into proteins and ran a local BLAST search against the corresponding reference proteome, so that it will provide us an idea of how precise are each of them.

The table (4) to summarize the number of true positive translated ORFs (true predicted proteins, verified by BLAST hit) of our ORF Finder and Glimmer/GeneScan (prokaryote/eukaryote) found against the total number of ORFs predicted (true and false positives) We used blastp to compare the orfs predicted by Glimmer/GENSCAN to our predictions. For each genome the proteome was obtained from Uniprot. We then ran blastp with the species proteome as the query database and the predictions by each method as the reference database. The set of homologues that we obtained, with e-value set at 0.0001 and taking the first hit, was considered to be correct predictions.

As we can see from the table 4, Glimmer and GENSCAN perform very well with high sensitivity and specificity. We can also see that our predictions are much higher in number than the actual proteomes. We attribute this largely due to the separate prediction of nested genes as orfs. While Glimmer and GENSCAN also analyse nested genes they do not predict them as separate orfs, during translation then, they do not translate every single nested gene. Our method, however does not predict every start/stop codon pair and therefore does have some selectivity. Additionally our method has high sensitivity though this can be attributed to the relatively high number of predictions. Apart from the features used for ORF prediction, both GENSCAN and Glimmer are HMM and IMM based predictors that have the advantage of getting trained on patterns not detectable or implementable through a constraint based method like ours. Machine learning represents the most powerful technique to analyse hidden patterns in data. However, these methods also faced similar challenges that they have overcome over the years. Some of these challenges include handling nested, overlapping and also partial genes[10].

The primary drawback of our method remains the generation of false positives. We propose to improve our method by incorporating a BLAST search for predicted ORFs and using homologues as evidence. However, the large coverage that has been achieved makes possible to reliably search un-

Organism	Number of proteins at the Reference Proteome	Our ORF Finder (BLAST hit/total pred)	Glimmer or Genescan (BLAST hit/total pred)
Chlamydia trachomatis	895	879/9345	872/956
Dictyoglomus turgidum	1743	1723/14888	1731/1863
Mycobacterium tuberculosis	3993	3783/58414	3771/4183
Rhodopirellula baltica	7271	5739/107685	4933/6414
Saccharomyces Cerevisiae	328	273/4417	206/206

Table 4. Results and comparisons of the ORF predictor vs Glimmer/GENSCAN per genome

Feature	GENSCAN	Glimmer	ORF generator
ORF length	Yes	Yes	Yes
Promoter/TSS	Yes	Yes	Yes
RBS	Yes	Yes	Yes
GC comp	Yes	No	No
Splice signals	Yes	N.A.	No
Homology	Yes	Yes	No

Table 5. Features used for ORF prediction per method

known sequences and find homologues if the predictions are valid[11].

3. Distance Matrix and trees

3.1 Input

As required, the statistics generated from the first program were used to calculate distance matrix between given five species;

- GC Content
- Nucleotide Frequency
- DiNucleotide Frequency
- AminoAcid Frequency(In all three reading frames)
- DiAminoAcid Frequency(In all three reading frames)

Above frequencies were used as an input to create a larger vector for each respective genome, resulting vector size of 1560 data points.

3.1.1 Scaling

We normalized each vector by Z-score, i.e. each point in the vector is derived by subtracting the population mean from an individual data point and then dividing the difference by the standard deviation for whole vector.
The normalized value of e_i for vector E in the i^{th} column is calculated as:

$$Normalized(e_i) = \frac{e_i - \bar{E}}{std(E)} \tag{1}$$

where

$$std(E) = \sqrt{\frac{1}{(n-1)} \sum_{i=1}^{n} (e_i - \bar{E})^2} \tag{2}$$

and

$$\bar{E} = \frac{1}{n} \sum_{i=1}^{n} e_i \tag{3}$$

The main motivation for using this method of normalisation over more common MinMax scaling was the fact that we had outliers in our data, specially in case of DiAminoAcid frequency where some pairs didn't occur in genomes or their contribution was abysmally low. These outliers can affect MinMax scaling dramatically, considering that AminoAcid and DiAminoAcid frequencies contribute to the bulk of data used for distance calculation.

3.2 Distance calculation

For calculation of distances we used two distance metric, namely euclidean (eq. 4) and angular distance(eq. 5 and eq. 6). The reason for selecting these two distance metrics was the fact that our data was highly linear and with relatively less amount of separation between each datapoint for all genomes(including Yeast) and moreover since the data was normalised, we didn't have any concerns regarding different in distribution of data falling into extremes.

$$d(G_1, G_2) = d(G_2, G_1) = \sqrt{\sum_{i=1}^{n} (G_{1i} - G_{2i})} \tag{4}$$

$$Angular\ distance = 1 - Cosine\ simalirty \tag{5}$$

$$Cosine\ Similarity = cos(\theta) = \frac{\sum_{i=1}^{n} G1_i \cdot G2_i}{\sqrt{\sum_{i=1}^{n} G1_i^2}\sqrt{\sum_{i=1}^{n} G2_i^2}} \tag{6}$$

3.2.1 Distance matrices

For the analysis below, we have condensed the species name to a shorthand notation in table 6 for ease of representation. Matrices for Euclidean (10) and Angular distances can be found towards the end of this section (11).

Distance matrices calculated by both metrics follow similar trend, albeit with different scaling. Angular distance essentially is euclidean distance; Cosine similarity[1] corresponds

[1] Angular distance is 1 - cosine similarity

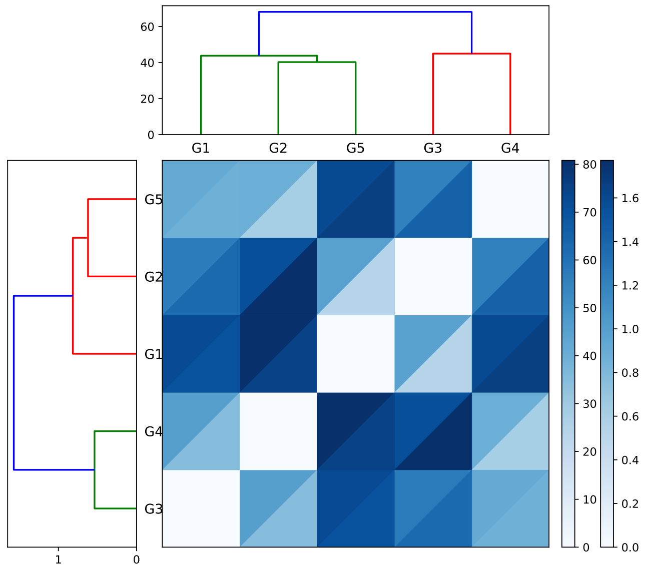

Figure 8. This figure illustrates the similarity between two distance metrics and the tree generated from them. The upper diagonal region shows distance from Euclidean metric, whereas the lower corner shows the distance from angular matrix. The reason to use identical color scale for both distances is to emphasize the fact that apart from magnitude, they are are similar. Tree on top is generated from euclidean distances and tree on left is generated using angular distance. Both trees were generated using UPGMA clustering.

to Euclidean distance after scaling each data point to unit length, and thus the correlation between two metrics. It is visible in the figure that these two distance metrics yield in a similar tree, not if identical, and the magnitude respective to each "distance"$[i_1 = i_2, j_1 = j_2]$ is similar, after accounted for the scaling. The resulting trees[Figure 8] also share similar branch lengths[2].

Relationship between Euclidean and angular distance(in this specific case, squared euclidean which penalizes more distant points):-

From law of cosines:

$$\text{Euclidean}\,(G_n, G_k)^2 = ||G_n||^2 + ||G_k||^2 - 2||G_n||||G_k||\,\text{CosSim}\,(G_n, G_k) \tag{7}$$

After "normalizing" data:

$$\text{Euclidean}\left(\frac{G_n}{||G_n||}, \frac{G_k}{||G_k||}\right)^2 = 2\,[1 - \text{CosSim}\,(G_n, G_k)] \tag{8}$$

And we know that $Angular\ distance = 1 - CosSim$

$$\text{Euclidean}\left(\frac{G_n}{||G_n||}, \frac{G_k}{||G_k||}\right)^2 = 2[Angular\ distance] \tag{9}$$

The above relations establish a direct correlation between Euclidean and angular distance and explains the reason why the distances and trees from both metrics are related.

3.3 Trees

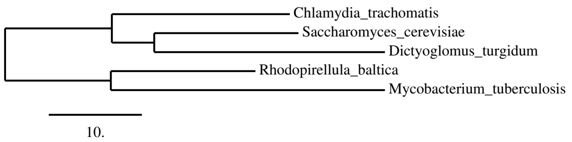

Figure 9. Tree calculated from the euclidean distance metric

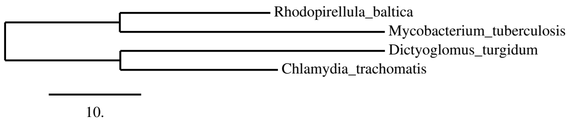

Figure 10. Tree calculated from the euclidean distance metric without Saccharomyces cerevisiae

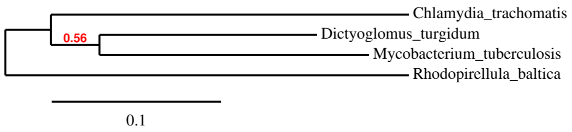

Figure 11. Tree from metagene with bootstrapping

It is apparent from above trees that there is significant disagreement between them. The differences can be attributed to two major factors, i.e. the tree construction method and the input data.

3.3.1 Tree construction algorithm

We constructed tree[Figure 11] for metagene using maximum likelihood algorithm, which by far is a better model for generating evolutionary inferences. The trees[Figure 9 and 10] which we calculated using distances from frequency cannot resolve evolutionary signals as well. More importantly, distance based tree calculation is as sensitive as the metric used for calculating the distances itself, and in our case we used Euclidean distance, which is one of simplest distance metric. We also used angular distance but the results were similar.

3.3.2 Data source

Another reason for differences in tree is the data used for construction. The distances in our method are based on the frequency of characters in genome, while the data used for generating other trees was based orthologous sequences which exhibit strong evolutionary correlation and it is comparatively easier to extract evolutionary inferences from multiple sequence alignment. Maximum likelihood uses these aligned positions for the evolutionary model and generates the tree, whereas character frequencies don't carry a similar degree of information. Similar to this, the tree[Figure 12] generated from gene order is significantly different too, mainly because

[2] Again, after accounting for magnitude

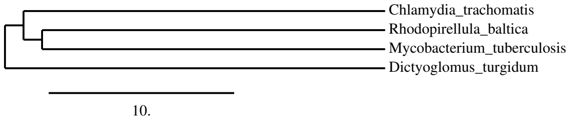

Figure 12. Tree from gene order analysis

bacterial genomes don’t share high level of homology, even in order of genes.

3.3.3 Anomaly and more

The only significant anomaly which is visible in the trees is association of yeast with bacteria. We were assuming it to be an outlier but considering the arguments above, it is possible that this inclusion is due to ”pseudo- random” nature of input data. Moreover, we used just one chromosome of yeast which does not reflect whole genome. One possible way to test this is by including a randomly generated genome with a characteristic distribution curve and see the association. This is one of the shortcoming of our method, considering we didn’t perform random sampling of data either, which could have given a statistically more reliable result to see if the data is associated with any type of phylogenetic signals.

3.3.4 Data

Table 6. Organisms and their respective notations

Organism	Code
Chlamydia trachomatis	G1
Dictyoglomus turgidum	G2
Mycobacterium tuberculosis	G3
Rhodopirellula baltica	G4
Saccharomyces cerevisiae	G5

Euclidean distance matrix:

$$\begin{bmatrix} & G1 & G2 & G3 & G4 & G5 \\ G1 & 0.000 & 45.704 & 72.048 & 56.824 & 41.904 \\ G2 & 45.704 & 0.000 & 80.796 & 71.226 & 40.268 \\ G3 & 72.048 & 80.796 & 0.000 & 44.970 & 72.461 \\ G4 & 56.824 & 71.226 & 44.970 & 0.000 & 55.514 \\ G5 & 41.904 & 40.268 & 72.461 & 55.514 & 0.000 \end{bmatrix} \tag{10}$$

Angular distance matrix:

$$\begin{bmatrix} & G1 & G2 & G3 & G4 & G5 \\ G1 & 0.000 & 0.761 & 1.537 & 1.378 & 0.869 \\ G2 & 0.761 & 0.000 & 1.646 & 1.771 & 0.621 \\ G3 & 1.537 & 1.646 & 0.000 & 0.537 & 1.666 \\ G4 & 1.378 & 1.771 & 0.537 & 0.000 & 1.427 \\ G5 & 0.869 & 0.621 & 1.666 & 1.427 & 0.000 \end{bmatrix} \tag{11}$$

4. GIT

Link to predicted ORFs:
`https://github.com/aron0093/KB8019/tree/master/data/ORFs`
Link to master branch:
`https://github.com/aron0093/KB8019/tree/master/`

References

[1] D. ELSON and E. CHARGAFF. On the desoxyribonucleic acid content of sea urchin gametes. *Experientia*, 8(4):143–145, Apr 1952.

[2] T.A. Brown. *Genomes 2*. Garland Science, 2002.

[3] J. Zhang. Protein-length distributions for the three domains of life. *Trends Genet.*, 16(3):107–109, Mar 2000.

[4] J. Shine and L. Dalgarno. Determinant of cistron specificity in bacterial ribosomes. *Nature*, 254(5495):34–38, Mar 1975.

[5] M. Kozak. An analysis of 5’-noncoding sequences from 699 vertebrate messenger RNAs. *Nucleic Acids Res.*, 15(20):8125–8148, Oct 1987.

[6] J. Shine and L. Dalgarno. The 3’-terminal sequence of Escherichia coli 16S ribosomal RNA: complementarity to nonsense triplets and ribosome binding sites. *Proc. Natl. Acad. Sci. U.S.A.*, 71(4):1342–1346, Apr 1974.

[7] D. Pribnow. Nucleotide sequence of an RNA polymerase binding site at an early T7 promoter. *Proc. Natl. Acad. Sci. U.S.A.*, 72(3):784–788, Mar 1975.

[8] D. P. Ramji and P. Foka. CCAAT/enhancer-binding proteins: structure, function and regulation. *Biochem. J.*, 365(Pt 3):561–575, Aug 2002.

[9] W. Shi and W. Zhou. Frequency distribution of TATA Box and extension sequences on human promoters. *BMC Bioinformatics*, 7 Suppl 4:S2, Dec 2006.

[10] C. Burge and S. Karlin. Prediction of complete gene structures in human genomic DNA. *J. Mol. Biol.*, 268(1):78–94, Apr 1997.

[11] A. L. Delcher, K. A. Bratke, E. C. Powers, and S. L. Salzberg. Identifying bacterial genes and endosymbiont DNA with Glimmer. *Bioinformatics*, 23(6):673–679, Mar 2007.