**Comparative genomics final report**

*by Shuhan Xu, Fuqi Xu, Milda Valiukonytė*

In this project, we have performed an analysis of five genomes, including their genetic composition (GC content, nucleotide, dinucleotide and diaminoacid frequencies), prediction of open reading frames and calculation of distances between genomes to understand the evolutionary relationships between species. We wrote three Python scripts for this purpose, named GC\_content.py, ORF\_predictor.py and distance.py (see attachments).

The five genomes included four prokaryotes and one eukaryote: *Escherichia coli* of *Enterobacteriaceae* bacterial family (genome length 5443340 bp), *Streptomyces coelicolor* of *Streptomycetaceae* family (9054847 bp), *Rubrobacter xylanophilus* of *Rubrobacteraceae* family (3225748 bp), *Spiribacter curvatus* of *Ectothiorhodospiraceae* family (1926631 bp), and *Saccharomyces cerevisiae* from *Saccharomycetaceae* yeastfamily (1531933 bp).

The genetic composition analysis has shown that there is a great variation among the five genomes in terms of their GC content, nucleotide, dinucleotide and diaminoacid frequencies. The GC content was calculated using the formula:

.

The highest GC content was found in *Streptomyces coelicolor*,while *Saccharomyces cerevisiae* had the lowest GC content. (Fig 1.)

**Fig 1.** GC content among genomes

Dinucleotide frequencies were calculated using formula:

The results have shown that bacterial species are rich in CC, CG, GC, GG dinucleotides, which is in line with the GC content results. Overall, we have observed great differences between dinucleotide frequencies among species which can indicate that dinucleotide frequency pattern is species-specific (Fig 2). Similar patterns have been observed in nucleotide and amino acid frequency distributions (Fig 3 and Fig 4).

**Fig 2.** Nucleotide frequencies among genomes

**Fig 3.** dinucleotide frequencies among genomes

**Fig 4.** Amino acid frequencies among genomes

**ORF Prediction**

Next, we used our own written script to predict open reading frames (ORFs) in our genomes. For the prediction, we made the following basic assumptions. First, An ORF starts with start codon (TAC) and ends with stop codons (ATT, ACT, ATC). Second, the minimum length of genes in prokaryotes is 200bp, 300bp in eukaryotes (excluding the stop codons). Third, in the same reading frame, if there are overlapping genes sharing the same stop codons. the longest ORF is the one we are looking for. Fourth, in different reading frames, the maximum overlapping length is 60bp. Fifth, in different reading frames, if maximum overlapping length is greater than 60bp, we keep the longer ORFs. Table 1 summarizes the ORF predictions across genomes.

Table : The ORF predictions across genomes

|  |  |  |
| --- | --- | --- |
| **Genome ID** | **Species** | **Predicted genes** |
| 9 | *Escherichia coli* | 5079 |
| 17 | *Streptomyces coelicolor* | 7172 |
| 24 | *Saccharomyces cerevisiae* | 724 |
| 49 | *Rubrobacter xylanophilus* | 2778 |
| 51 | *Spiribacter curvatus* | 1758 |

**Evaluation**

To evaluate the performance of Fumish predictor, we compared it with the state of art predictor, GLIMMER and assumed the prediction of GLIMMER are the real genes. Moreover, we compared the translated ORFs of Fumish with their proteome in Uniprot.

1) Comparison with GLIMMER

We compared the prediction accuracy of GLIMMER and Fumish at nucleotide level and compared the gene size distribution of them. We defined nucleotides predicted both in GLIMMER and our own predictor as true positive (TP), nucleotides that appears only in our predictor instead of GLIMMER as false positive (FP), nucleotides that presents in GLIMMER instead of our own predictor as true negative (TN) and nucleotides that don’t show up in both predictions as false negative (FN).

The following formulas were used to compare:

The prediction accuracy of *E.coli* is 0.922 and the lowest prediction accuracy is in yeast with AC = 0.529. The average length Fumish predicted is shorter in all prokaryote genomes. Table 2 summerizes the result.

Table : The comparision of ORF predictions between Fumish and GLIMMER

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| **Species** | **Predicted ORFs** | **Average length in GLIMMER(bp)** | **Average length in Fumish(bp)** | | **Sensitivity** | **Specificity** | **AC** |
| Escherichia coli | 6329 | 888 | | 773 | 0.95 | 0.917 | 0.922 |
| Streptomyces coelicolor | 8175 | 939 | | 954 | 0.59 | 0.61 | 0.529 |
| Saccharomyces cerevisiae ch4 | 1604 | 1235 | | 782 | 0.97 | 0.88 | 0.91 |
| Rubrobacter xylanophilus | 3167 | 879 | | 869 | 0.623 | 0.671 | 0.586 |
| Spiribacter curvatus | 1955 | 973 | | 897 | 0.841 | 0.869 | 0.829 |

We also plotted the gene size distribution of both GLIMMER and Fumish. Our predictor predicts more short genes (gene size <500bp) than GLIMMER. The prediction agrees with GLIMMER in longer, especially genes that longer than 2000bps.The gene length distribution of *E.coli* and *S.coelicolor* of Fumish are very similar with the predictions of GLIMMER, which agrees with the approximate correlation coefficient value.

A screenshot of a cell phone

Description generated with very high confidenceA screenshot of a cell phone

Description generated with high confidenceA screenshot of a cell phone

Description generated with very high confidenceA screenshot of a cell phone

Description generated with high confidenceA screenshot of a cell phone

Description generated with very high confidence

Figure :The gene length distribution of GLIMMER(left) and Fumish(right). 09 is *E.coli*, 17 is *S.coelicolor*, 24 is *S.cerevisiae*, 49 is *R.xylanophilus*, 51 is *S.curvatus*

2) Comparison with Uniprot proteome

To better evaluate the performance of Fumish, we translated our predicted genes and compared them with the proteome in Uniprot. To identify the proteins that were correctly predicted, we performed two BLASTp tests. Firstly, we used the Uniprot proteome as database and the predicted proteins as query to blast. Hits with e-value less than 0.001 were selected as the predicted proteins that appears in the Uniprot proteome. Then, we used the Uniprot proteins as queries and the predicted proteome as the dataset and ran blast again. Proteins that show up in both blast tests are considered as true positive predictions.

*True positive: Proteins that appears in both the real proteome and the predicted proteome.*

*False positive: Proteins that appears only in the predicted proteome instead of the Uniprot proteome.*

*False negative: Proteins that appears only in the Uniprot proteome instead of the predicted proteome.*

The sensitivity and specificity are calculated using formulas above. F1 score measures the overall prediction accuracy.

Table 3 summarizes the results.

Table 3

In order to increase the accuracy of our prediction, the following improvements could be made:

1) Adding promoter information

Adding promoter information filters small ORFs and selects the overlapping ORFs more precisely. Pribnow boxes are used as the promoter in prokaryotes. The Pribnow box is located at 10bp upstream the transcriptional start site [1] and the transcrition start site is 20 to 40 nucleotides upsteam the start codon[2]. We can search for the Pribnow box and select the start codons that are 30bp-50bp downstream the Pribnow box. In eukaryotes, TATA boxes information can be used in the same way as the Pribnow box.

In our *E.coli* genome, there are only around 700 Pribnow boxes(TATAAT)

2) The minimum gene length

The average gene length is 991bp and there are around 10% of the genes that shorter than 300bp in E.coli. (ref) So we selected 200bp as our minimum gene length threshold. While the minimum gene length varies among different species and each species should be treated differently.

3) The maximum overlapping length

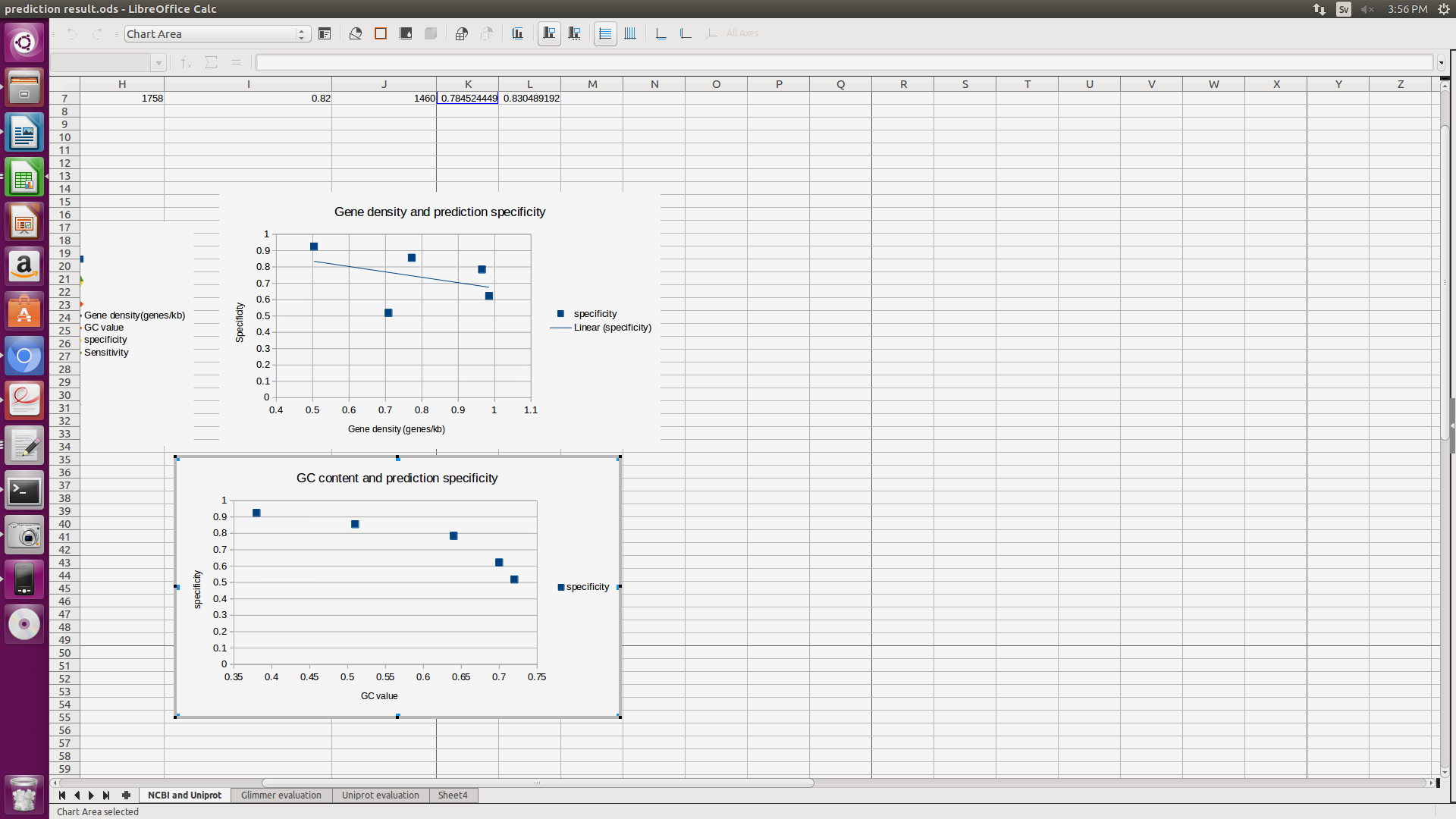
Overlaps in different reading frames that are longer than 60bp are forbidden in our predictor. In this case, we only select the longest ORF in the overlapping genes. However, the maximum overlapping length and the selection of overlapping distance depends on the species and should be treated differently in different species.

4) Gene length distribution

Our predictor selects the longest gene among all overlapping genes(genes that shares the same ORFs). However, the longest gene is not always the most likely ORF. Instead, we can find information of the length distribution of ORFs and calculates the probability of each gene lengthin the genome and select the most likely ORF.

5) Different start codons

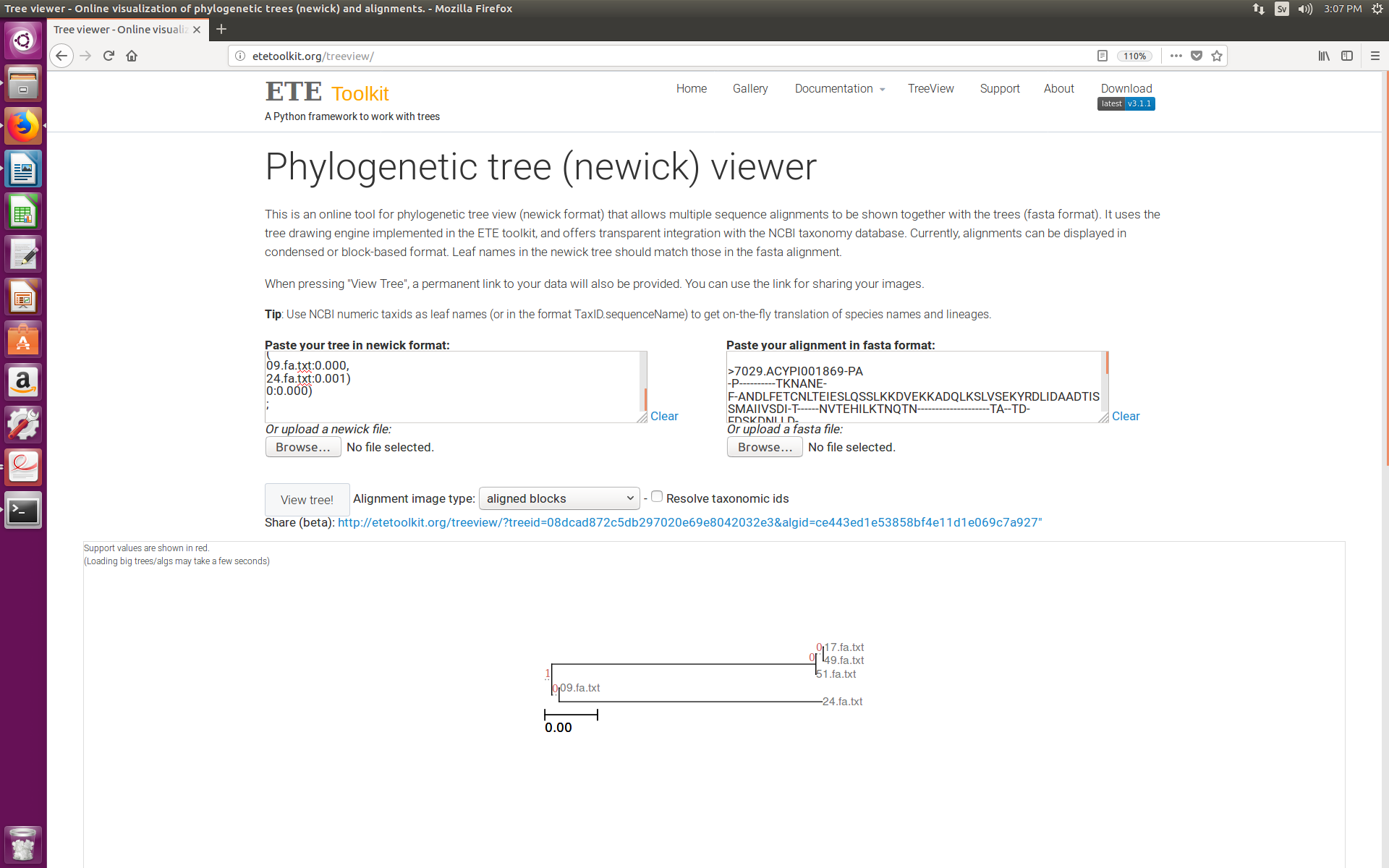
In prokaryotes, the start codon could be ATG, GTG and TTG. Our preditor only takes ATG as the start codon. However, some genes starts with Val instead of Met in the real genome and in the prediction of GLIMMER. We found that the prediction specificity decreases with the increase of GC content in our prediction (Fig 5.)



**Fig 5.** GC content and prediction specificity

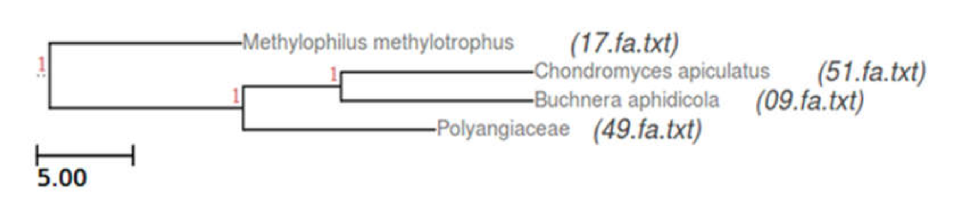
The higher the GC content is, the more likely the ORFs starts with GTG instead of ATG. We should also include other possible start codons in the predictor.

For the final part, evolutionary relationship analysis, we chose dinucleotide frequencies to calculate the distance between genomes and visualized the evolutionary relationship between species in a phylogenetic tree using neighbour-joining method (Fig 6). The motivation for chosing dinucleotide frequencies for distance calculation is that it has been shown that dinucleotide frequency distributions are quite species specific and are even considered a genomic signature (Karlin and Burge, 1995). Secondly, we have chosen this distance calculation method over methods using amino acid/diamino acid frequencies because it is not based on prediction.



**Fig 6.** Evolutionary relationship between species based on dinucleotide frequencies

The tree agrees with the consensus tree built using gene order to calculate distances between species (Fig 7.)



**Fig 7.** Evolutionary relationship between species based on gene order

**Python Scripts**

1. **Statistics tool**

Script for calculating GC content, nucleotide frequencies and dinucleotide frequencies: *nucl\_statistics.py*

Usage: python3 nucl\_statistics.py <genome1.fa.txt> <genome2.fa.txt> <genome3.fa.txt> …

Description: The script takes any number of genome fasta files and calculates the GC content, nucleotide frequencies and dinucleotide frequencies. The results are output to three .csv files ‘gc\_freq.csv’, ‘nucl\_freq.csv’ and ‘dinucl\_freq.csv’ in the working directory.

Script for calculating amino acids frequencies and diamino acids frequencies: *prot\_statistics.py*

Usage: python3 prot\_statistics.py <proteome1.fa.txt.pfa> <proteome2.fa.txt.pfa> <proteome3.fa.txt> …

Description: The script takes any number of proteome fasta files and calculates the amino acid frequencies and diamino acid frequencies. The results are output to three .csv files ‘amino\_freq.csv’ and ‘diamino\_freq.csv’ in the working directory.

1. **ORF finder**

Script for finding open reading frame: *predict\_orf.py*

Usage: python3 predict\_orf.py <genome.fa.txt>

Description: The script takes one genome fasta file and find ORFs that satisfy the conditions stated in the report above. The coordinates of the predicted ORFs are then saved to ‘genome.fa.txt.predict’ in the working directory.

The ‘genome.fa.txt.predict’ file has the same format as GLMMER’s .predict file. To retrieve the nucleotide or protein sequences, the script ‘parseGlimmer.py.2’ can be used.

python2 parseGlimmer.py.2 <genome.fa.txt> <genome.fa.txt.predict>

or

python2 parseGlimmer.py.2 <genome.fa.txt> <genome.fa.txt.predict> --translate

Script for evaluating the nucleotide level accuracy of our predicted ORFs against GLIMMER ORFs: *evaluate.py*

Usage: python3 evaluate.py <genome.fa.txt> <genome.glimmer.predict> <genome.fa.txt.predict>

Description: The script takes the genome fasta file, glimmer prediction file and the prediction file generated above and calculate sensitivity, specificity and approximate correlation coefficient based on nucleotide level accuracy in all six reading frames.

3. **Gene distribution Plotter**

Script for plotting the gene length distribution of two predictors: *plotGeneLength.py*

Usage: python3 plotGeneLength.py <predictionResult1> < predictionResult1>

Description: This scripts takes two gene prediction results and plots the gene distribution of the two predictions.

The input prediction result file should be written in GLIMMER output format.

<column0> <column1> <column2> <others>

Gene name Start codon position Stop codon position …