Faculdade de Ciências da Universidade de Lisboa

Master's in Data Science

Bioinformatics and Computational Biology

(2022/2023)

Project Report

"Genomic comparison and similarity analysis of

Escherichia coli and *Vibrio Cholerea* using Biopython"

Group 13

Ana Araújo (59457), Cláudia Afonso (36273), João Faia (47051)

I. Introduction

In this bioinformatics project, the main aim is to explore and analyze the genomes of two selected species using Biopython, an open-source collection of modules written in the Python programming language that facilitate the analysis and interpretation of biological data.[1] This project encompasses three main goals, namely (1) the selection of two species of interest and the retrieval of their respective genomes, (2) the sequence alignment of the two genomes and the calculation of similarity between them, and (3) the retrieval of research papers related with each of the chosen species. The file containing the code developed for the project is attached to this report with the name “Script_G13.py”.

II. Goal 1 - Selection of two species and retrieval of their genomes

For this project, we initially decided to focus on the genomes of model organisms, since these have been widely studied by the scientific community over many years. Thus, their genomes are fully sequenced and well-characterized, resulting in a vast amount of experimental data that has led to a proportional increase in the volume of published research. In this context, the two species chosen initially were *Drosophila melanogaster* and *Mus musculus*. Unfortunately, when proceeding to the following stage of the project where the goal is to perform sequence alignment between the genomes of the two chosen species, the following error related to the length of the sequences was obtained: “Length limit exceeded. Please reduce your query/subject sequence length to 10,000,000 letters or less”. Thus, as a viable alternative to circumvent this unexpected issue, we decided to choose species with smaller genomes. To this end, we decided to focus on bacterial genomes due to the generally smaller size of their genomes when compared to those of more complex organisms such as plants or animals.

Description of the *Escherichia coli* and *Vibrio cholerae* species

Along the same line of thought that led us to initially select genomes of model organisms, one of the chosen species at this stage of the project was *Escherichia coli*, commonly known as *E. coli*. Owing to its simplicity, low cost, ease of maintenance and well-characterized genetics, this Gram-negative bacterium is one of the most studied species on earth and has been heavily used in research institutes as a model organism to understand fundamental biological and biochemical processes. It is also widely employed in the pharmaceutical and biotechnology industries where it serves as a host to produce therapeutic proteins, vaccines, enzymes and biofuels. In addition, *E. coli* is also important for environmental monitoring as an indicator organism, particularly to assess water and food quality. Although it can be found in the gut microbiome of birds, reptiles and fish, as well as in soil, water, plants and food, *E. coli* primarily exists in nature as a member of the gut microbiome of mammals.[2,3] While most strains are typically harmless commensals that exist in the normal gut microbiota to establish a symbiotic relationship with their hosts and rarely induce disease, pathogenic variants categorized as diarrheagenic and extraintestinal also exist for *E. coli*.[4] Its genome was first sequenced in 1997 using the K-12 strain, which contains 4,401 genes encoding 116 RNAs and 4,285 proteins.[5,6]

To compare the *E. coli* genome with a second species, another Gram-negative bacterium capable of colonizing the gastrointestinal tract of animals was considered. For this purpose, the human pathogen *Vibrio cholerae* (*V. cholerae*) was chosen as an ideal candidate. This microorganism causes the severe diarrheal disease known as cholera, infecting approximately 3 million people worldwide and leading to 95 000 deaths annually.[7] Although infection can occur through person-to-person close contact and through food, *V. cholerae* is primarily transmitted through the fecal-oral route via contaminated drinking water.[8] Thus, cholera mostly plagues developing nations and regions with inadequate sanitation, with a disease burden that is highest in sub-Saharan Africa and low to non-existent in developed countries.[9] *V. cholerae* infects the host through two main virulence factors, namely the toxin-coregulated pilus (TCP) and cholera toxin (CT). TCP promotes the colonization of the host’s gut mucosal layer, where CT is subsequently released to cause watery diarrhea.[9] Due to its global public health impact, high infectious dose, rapid replication rate, and ease of genetic manipulation, *V. cholerae* is used as a model organism to answer essential questions related to virulence and antibiotic resistance of Gram-negative enteric pathogens.[9,10] Its genome, first sequenced in 2000 using the El Tor N16961 strain, deviates from the single chromosome rule found in most bacteria. Instead, the *V. cholerae* genome is divided into two circular chromosomes of unequal sizes that together encode 3,885 open reading frames.[11,12]

Description of the genome files of *Escherichia coli* and *Vibrio cholerae* species

The reference genomes of both species were obtained from GenBank, a comprehensive public database that serves as a repository for nucleotide sequences and is maintained by the National Center for Biotechnology Information (NCBI).[13,14] The genome files were downloaded as gzip (.gz) compressed files containing a single GenBank flat file (.gbff). The latter is a text-based format with a record-oriented structure that allows for the comprehensive storage of genetic information, where each record corresponds to a distinct nucleotide sequence entry from a single molecule type and contains a set of fields that provide a wealth of information. The key fields in a GenBank record include the (1) LOCUS, which contains several different

data elements, including locus name, sequence length and molecule type; (2) DEFINITION, which provides a brief description of the sequence and also stores information concerning the source organism; (3) ACCESSION, which corresponds to the unique identifier assigned to a sequence record; (4) VERSION, a sequence identification number that represents a single, specific sequence in the GenBank database; (5) DBLINK, which identifies the sequencing projects associated with a GenBank sequence record; (6) KEYWORDS, words or phrases associated with the sequence record; (7) SOURCE, with information about the organism from which the sequence record was derived, including its scientific name and taxonomy; (8) REFERENCE, corresponding to publications by the authors of the sequence that discuss the data described in the record; (9) COMMENT, with additional notes provided by the submitter regarding the sequence record, (10) FEATURES, with information about genes and coding regions, as well as regions of biological significance reported in the sequence; and (11) ORIGIN, which contains the actual nucleotide sequence data.[15]

To open the compressed gzip files for reading, the “gzip” module of Python was used. The sequence record was then read using the “Bio.SeqIO.parse()” function to return a SeqRecord iterator, which was subsequently converted into a list of SeqRecord objects using the built-in Python function list(). Since this procedure needs to be carried out twice for the two chosen species, a function called “parse_gzip_files(fileName)” was developed to avoid repeating unnecessary code (lines 36-60 with documentation available in the “Script_G13.py” file). Then, the ID, name and description of each SeqRecord object, which correspond to the previously mentioned VERSION, ACESSION and DEFINITION fields of the GenBank file, respectively, were retrieved and placed within a pandas DataFrame. Again, to avoid duplicating code, a function named “create_records_dataframe(records)” was developed (lines 63-83 with documentation available in the “Script_G13.py” file).

Escherichia coli

Compressed GenBank File Name: “GCF_000005845.2_ASM584v2_genomic.gbff.gz”

Compressed GenBank File Size: 3.26 MB

The reference genome of *E. coli* is derived from the K-12 strain and the GenBank flat file includes a single record. The respective ID, name and description of the obtained SeqRecord object is illustrated below in Figure 1 and the code used to obtain this information is located between lines 90-98 of the “Script_G13.py” file.

```
Species 1: Escherichia coli

There is 1 record for the reference genome of Escherichia coli

            ID       Name                                                Description
0  NC_000913.3  NC_000913  Escherichia coli str. K-12 substr. MG1655, complete genome
```

Figure 1 – Output concerning the records contained within the reference genome file of *E. coli*. The code used to obtain these results is located between lines 90-98 of the “Script_G13.py” file.

Vibrio cholerae

Compressed GenBank File Name: “GCF_008369605.1_ASM836960v1_genomic.gbff.gz”

Compressed GenBank File Size: 2.90 MB

The GenBank flat file for the reference genome of *Vibrio cholerae* includes a total of 3 records. The respective ID, name and description of each SeqRecord object is illustrated below in Figure 2 and the code used to obtain this information is located between lines 108-115 of the “Script_G13.py” file.

```
Species 2: Vibrio cholerae

There are 3 records for the reference genome of Vibrio cholerae

              ID         Name                                                  Description
0  NZ_CP043554.1  NZ_CP043554     Vibrio cholerae strain RFB16 chromosome 1, complete sequence
1  NZ_CP043556.1  NZ_CP043556     Vibrio cholerae strain RFB16 chromosome 2, complete sequence
2  NZ_CP043555.1  NZ_CP043555  Vibrio cholerae strain RFB16 plasmid unnamed, complete sequence
```

Figure 2 – Output concerning the records contained within the reference genome file of *V. cholerae*. The code used to obtain these results is located between lines 108-115 of the “Script_G13.py” file.

III. Goal 2 - Sequence alignment of the genomes and similarity calculation between the species

Basic Local Alignment Search Tool, colloquially known as BLAST, is a widely used bioinformatics technique for detecting similarity between sequences of interest.[16] Thus, to align and assess the similarity of the genomic sequences of the two chosen species, *E. coli* and *V. cholerae*, a BLAST run was performed. There are several ways to conduct a BLAST run, namely remotely by invoking the NCBI online BLAST service from within a Python script or via the NCBI-web service, or locally using our own computational resources and stored sequence databases. The NCBI-web service (which can be accessed using this link: https://blast.ncbi.nlm.nih.gov/BlastAlign.cgi) was employed at this stage due to its simplicity of use and user-friendly graphical interface to conduct BLAST runs.[17]

After choosing the "Nucleotide BLAST" option and being re-directed to another webpage, the "Align two or more sequences" field was selected. Then, the ID value associated with the single record in the reference genome file of *E. coli* (NC_000913.3) was copied into the "Enter Query Sequence" field. Meanwhile, the IDs corresponding to the three records in the reference genome of *V. cholerae* (NZ_CP043554.1, NZ_CP043556.1 and NZ_CP043555.1) were placed into the "Enter Subject Sequence" field. The "blastn" program for somewhat similar sequences was chosen due to its suitability for cross-species comparisons and the BLAST run was finally performed. Following a waiting period of a few seconds, a webpage with the alignment results was produced and these were downloaded as an XML file. The latter was subsequently parsed in our "Script_G13.py" file by importing the "NCBIXML" module from the "Bio.Blast" package of Biopython and using the "Bio.Blast.NCBIXML.read()" function. The latter function assumes that just one query sequence in a BLAST run was employed, which was the case here.[18] Subsequently, the similarity score between the genomes of the two chosen species and the total number of alignments below an e-value threshold of 1.00×10^{-30} were determined. The e-value represents the number of different alignments with scores equivalent to or better than those that would be expected to occur in a database search by chance. The lower the e-value, the more significant the score and the alignment. These local alignments that achieve one of the highest alignment scores in a given search are known as high-scoring segment pairs (HSPs).[19] In this context, a similarity score of 75.35 % between the *E. coli* and *V. cholerae* genomes was obtained, while a total number of 921 alignments below the specified e-value threshold were registered (Figure 3). The code developed to obtain these results is located between lines 132-169 of the "Script_G13.py" file.

```
The similarity score between Escherichia coli and Vibrio cholerae is 75.35 %

Sequence title, length and e-value for the alignments below a threshold of 1.00e-30
****Alignment****
sequence: gi|1741166712|ref|NZ_CP043554.1| Vibrio cholerae strain RFB16 chromosome 1, complete sequence
length: 2948589
e value: 0.0
```

Figure 3 – The similarity score obtained between the genomes of *E. coli* and *V. cholerae* was 75.35 %. The sequence title, length and e-value for the first HSP below an e-value threshold of 1.00×10^{-30} is shown here.

IV. Goal 3 - Retrieval of research papers related with each species

The final goal of this project consisted in accessing Entrez to retrieve from PubMed the 10 most recent research papers for each species, as well as those for two of their genes selected at random. Entrez is a retrieval system that provides users with access to NCBI's databases, including PubMed, GenBank and others. It can be accessed from a web browser (https://www.ncbi.nlm.nih.gov/Web/Search/entrezfs.html) or through the "Bio.Entrez" module of Biopython to enter queries from within a Python script. Due to the repetitive nature of the task at hand, the latter option was chosen to streamline and automate the retrieval process. In this context, the "Bio.Entrez" module from Biopython was used.

To perform an Entrez search of the PubMed database for the 10 most recent research papers related to each of the two chosen species, the "Bio.Entrez.esearch()" function was invoked with the following parameters: db = "pubmed", term = "species name[title]", retmax = 10 and sort = "pub_date". The first parameter specifies that the search is to be carried out in the PubMed database, the second defines the search for articles where the species name (set to either Escherichia coli or Vibrio cholerae) appears in the title, the third determines the maximum number of unique identifiers (known as UIDs, which in the case of a PubMed search are known as PMIDs) retrieved by the esearch() function, and the fourth sorts the PMIDs by their publication date. The [title] field tag, which narrows down the search to the article title, was used to validate the obtained results, since it is easier and faster to check whether the latter are correct when the specified term appears in the title, rather than somewhere else in the article. To obtain the research papers containing the search term anywhere in the article, the same code can be re-used by just omitting the [title] field tag within the term parameter. As previously mentioned, the "Bio.Entrez.esearch()" function returns a list containing the PMIDs of research papers satisfying a particular search query. Since this procedure needs to be performed multiple times to retrieve the research papers according to distinct search queries, a function called "search_pubmed_records(query_term, retmax=10, sort="pub_date")" was developed to avoid repeating duplicated code (lines 184-209 with documentation available in the "Script_G13.py" file).

To retrieve the records for research papers from a list of PMIDs, the “Bio.Entrez.efetch()” function was invoked. Here, the following parameters were used: db = “pubmed”, id = idlist, rettype = “medline” and retmode = “text”. The first parameter specifies PubMed as the database from which the records are to be fetched, the second parameter passes the list of PMIDs (assigned to a variable named “idlist”) to the function, while the third and fourth parameters define the desired records to be returned as Medline in plain-text format. The “Bio.Medline” module was then used to parse these records, which were iterated to obtain their respective titles, author names, journals and publication dates.[20] To avoid repeating unnecessary code, a function named “retrieve_medline_records(id_list)” was developed (lines 212-244 with documentation available in the “Script_G13.py” file). By invoking the “search_pubmed_records(query_term, retmax=10, sort="pub_date")” function within the “retrieve_medline_records(id_list)” function, it was thus possible to obtain the 10 most recent research papers for any specified value of the “query_term” parameter. The code used to obtain the 10 most recent research papers of *E. coli* and *V. cholerae* is located between lines 254-262 in the “Script_G13.py” file.

To select the 10 most recent research papers for two random genes belonging to each chosen species, a list of gene names from the original GenBank records was first created. This was performed by sequentially iterating over each element in the list of SeqRecord objects created during the first goal of the project, then iterating over each feature to check whether it is a gene and appending its name (if it exists) to a newly created list. A function named “retrieve_gene_names(records)” was developed to avoid duplicated code (lines 276-305 with documentation available the “Script_G13.py” file). After checking the resulting gene names, two were chosen for each species. For *E. coli*, the selected genes were *recA* and *lacZ*, while for *V. cholerae* the choice was made for the *hlyA* and *rtxA* genes. To confirm that the selected gene names existed in the GenBank records and to automate the process of gene selection without requiring prior inspection of the existing gene names within the original files, another function called “select_genes(possible_genes, gene_names)” was developed. The latter compares a list of possible gene names with the existing ones in the original GenBank records (lines 308-331 with documentation available in the “Script_G13.py” file). The code used to select the two genes for *E. coli* and *V. cholerae* is located between lines 336-347 and 370-380, respectively, in the “Script_G13.py” file.

The same procedure described previously to retrieve the 10 most recent research papers from PubMed for each species was followed here, but now for the selected genes instead. The function “search_pubmed_records(query_term, retmax=10, sort=“pub_date”)” was invoked within the “retrieve_medline_records(records)” function. The former function was called with query_term = “gene name[title]” to define the search for research papers where the gene name appears in the title. As mentioned previously, this approach was followed to validate the obtained results, since it is easier and faster to check whether the latter are correct when the specified term appears in the title, rather than somewhere else in the article. To obtain the research papers containing the search term anywhere else in the article, the [title] field tag can be omitted. The “search_pubmed_records(query_term, retmax=10, sort=“pub_date”)” function returns a list of PMIDs which are then passed to the “retrieve_medline_records(records)” function to retrieve the 10 most recent research papers for the chosen genes. The code used to obtain this information for genes *recA* and *lacZ* of E. coli is located between lines 357-365, while for genes *hlyA* and *rtxA* of *V. cholerae* the code can be found in lines 390-397 of the “Script_G13.py” file. It should be noted that some of the obtained research papers display a publication date that is in the future. While at first strange, following further inspection of these articles we concluded that these have already been published online. Thus, the publication date in these cases reflects the future date in which they will be physically printed in their respective journals.

Unlike what was done previously for project goals 1 and 2, in which some of the obtained results were placed in this report, we opted not to specifically mention some of the 10 most recent research papers for either the species or the genes, because these might vary in a future run of the “Script_G13.py” file.

V. Conclusions

This project focused on the use of Biopython to perform a comparative analysis of the genomic sequences of *E. coli* and *V. cholerae*, as well as to retrieve relevant research papers associated with each of these species. For the first goal, the *E. coli* and *V. cholerae* species were briefly described and detailed information concerning their genome files, including the number of records, IDs, names and description of each record, was provided. The second goal consisted in aligning the genomes of *E. coli* and *V. cholerae* to calculate the similarity between them. Here, a similarity score of 75.35% was obtained between the genomic sequences of the two species and a total of 921 alignments below a specified e-value threshold of 1.00×10^{-30} were registered. For the third goal, the 10 most recent research papers related to each species, as well as those for two of their genes selected at random were retrieved from PubMed by accessing the NCBI’s Entrez database.

To conclude, we successfully achieved all the proposed goals for this project. The combination of tasks associated with each project goal, namely the retrieval of genomic sequences and literature, as well as alignment of genomes and calculation of the similarity between them allowed us to gain valuable insights regarding the usefulness and variety of tools provided in Biopython.

VI. Bibliography

1. Cock PJ, Antao T, Chang JT, et al. Biopython: freely available Python tools for computational molecular biology and bioinformatics. *Bioinformatics*. Jun 1 2009;25(11):1422-3. doi:10.1093/bioinformatics/btp163
2. Blount ZD. The Natural History of Model Organisms: The unexhausted potential of E. coli. *eLife*. Mar 25 2015;4doi:10.7554/eLife.05826
3. Motlagh AM, Yang Z. Detection and occurrence of indicator organisms and pathogens. *Water Environ Res*. Oct 2019;91(10):1402-1408. doi:10.1002/wer.1238
4. Braz VS, Melchior K, Moreira CG. Escherichia coli as a Multifaceted Pathogenic and Versatile Bacterium. *Front Cell Infect Microbiol*. 2020;10:548492. doi:10.3389/fcimb.2020.548492
5. Blattner FR, Plunkett G, 3rd, Bloch CA, et al. The complete genome sequence of Escherichia coli K-12. *Science*. Sep 5 1997;277(5331):1453-62. doi:10.1126/science.277.5331.1453
6. Serres MH, Gopal S, Nahum LA, Liang P, Gaasterland T, Riley M. A functional update of the Escherichia coli K-12 genome. *Genome Biol*. 2001;2(9):RESEARCH0035. doi:10.1186/gb-2001-2-9-research0035
7. Ali M, Nelson AR, Lopez AL, Sack DA. Updated global burden of cholera in endemic countries. *PLoS Negl Trop Dis*. 2015;9(6):e0003832. doi:10.1371/journal.pntd.0003832
8. Baker-Austin C, Oliver JD, Alam M, et al. Vibrio spp. infections. *Nat Rev Dis Primers*. Jul 12 2018;4(1):8. doi:10.1038/s41572-018-0005-8
9. Yoon SH, Waters CM. Vibrio cholerae. *Trends Microbiol*. Sep 2019;27(9):806-807. doi:10.1016/j.tim.2019.03.005
10. Sikora AE, Tehan R, McPhail K. Utilization of Vibrio cholerae as a Model Organism to Screen Natural Product Libraries for Identification of New Antibiotics. *Methods Mol Biol*. 2018;1839:135-146. doi:10.1007/978-1-4939-8685-9_12
11. Heidelberg JF, Eisen JA, Nelson WC, et al. DNA sequence of both chromosomes of the cholera pathogen Vibrio cholerae. *Nature*. Aug 3 2000;406(6795):477-83. doi:10.1038/35020000
12. Trucksis M, Michalski J, Deng YK, Kaper JB. The Vibrio cholerae genome contains two unique circular chromosomes. *Proceedings of the National Academy of Sciences of the United States of America*. Nov 24 1998;95(24):14464-9. doi:10.1073/pnas.95.24.14464
13. NCBI. Escherichia coli str. K-12 substr. MG1655 (ID 167) - Genome - NCBI. Accessed 14th of May, 2023, 2023. https://www.ncbi.nlm.nih.gov/genome/167?genome_assembly_id=161521
14. NCBI. Vibrio cholerae (ID 505) - Genome - NCBI. Accessed 14th of May, 2023, 2023. https://www.ncbi.nlm.nih.gov/genome/505?genome_assembly_id=678998
15. NCBI. Sample GenBank Record. Accessed 18th of May, 2023, https://www.ncbi.nlm.nih.gov/genbank/samplerecord/
16. Altschul SF, Gish W, Miller W, Myers EW, Lipman DJ. Basic local alignment search tool. *J Mol Biol*. Oct 5 1990;215(3):403-10. doi:10.1016/S0022-2836(05)80360-2
17. NCBI. BLAST: Basic Local Alignment Search Tool. Accessed 19th of May, 2023, https://blast.ncbi.nlm.nih.gov/BlastAlign.cgi
18. Biopython. Bio.Blast.NCBIXML module - Biopython 1.75 documentation. Biopython. Accessed 17th of May, 2023, 2023. https://biopython.org/docs/1.75/api/Bio.Blast.NCBIXML.html
19. NCBI. BLAST Glossary - BLAST Help - NCBI Bookshelf. Accessed 20th of May, 2023, https://www.ncbi.nlm.nih.gov/books/NBK62051/
20. Biopython. Bio.Medline package - Biopython 1.76 documentation. Accessed 18th of May, 2023, https://biopython.org/docs/1.76/api/Bio.Medline.html