Genome Visualisation tool (Report+Manual)

1. INTRODUCTION

This piece of software was made using Java programming language to read the FASTA file and GTF file and display it visually to the user. It also calculates the Basic statistics for the two files along with highlighting of exons in FASTA .

2. DESIGN AND REQUIREMENTS

The design of the software is quite user friendly with self-explanatory buttons which was made using NetBeans IDE 16. It includes importing of many classes for calling various components like the label, button, text fields, frames, panes, file choosing etc. included in the "*Java.awt package"* and the "*javax.swing"* packages. Packages like "*java.util"* are also used to make arrays and lists to store values . Some "*java.io"* classes are used like "*BufferedReader"* and "*IOexception"* to read the files based on character based input stream (FASTA) and to handle some exception like input file type.

3. STRUCTURE

a. Uploading FASTA:

A *jButton1* is created and it is kept in a private method *jButton1ActionPerformed* and the following tasks are performed when it is pressed, in the following manner:-

1. A file chooser box is opened with an instance "open FASTA file", and the "getDirectory()" method takes the directory of the file while "getFile()" takes the name of the file which are concatenated with "concat()".

2. Here the "readline()" method is used to read every line and the "startsWith()" method is used to ignore the line starting with ">" as it is the header of the file. The final output is displayed in the *jtextArea1*.

3. A variable "seq" of string type is created to get components present in *jtextArea1* using "getText()". Then "replaceAll()" is used to remove the whitespaces from the text to find the right number of characters.

4. The GC content is calculated and is displayed in the basic stats section along with length of the sequence and the number of G's and C's. This is achieved by iterating through the "seq" and subtracting the total length of file by characters and "replace" is used to remove all the occurrences of the letter

"G" from the string. The difference between the two values gives the occurrences of "G" and is stored in "gCount". The same is being done for "C" except that it is stored in "cCount"[The "count " method was tried but as these are Buffered and not strings the method was not working].

5. A clear button is also created to clear (replacing characters with " whitespace") the FASTA area and the basic stats area.

b. <u>Uploading GTF:</u>

A *jButton3* is created which is kept in the private method *jButton3ActionPerformed* and the following tasks are performed serially when this button is pressed:-

1. File dialog- A new instance of *FileDialog* class is created with the title "Open GTF File". As above the "concat()" method is used to concatenate the directory and filename to make a full file path.

2. Array- 9 Arrays are created with capacity of 150 elements each as there are 145 columns in the GTF files. And these arrays will be used to store the values of the 9 columns of GTF file.

3. File reading- The further part of code is responsible for reading the file "filename " which the user selected in the dialog box section, and then each line of file is read while ignoring the line starting with "##" as it is the header of the GTF file. If the line doesn't start from that, it is split by tabs and stored in the previously formed arrays. And the output is appended in the *jTextArea3* with appropriate labels above to identify the headers.

4. Exon iteration- The next chunk of code iterates a variable "i" through the feature column to look for "exon" and "gene" and store it in variables. It also keeps a track of the longest and shortest gene names by using the "split()" and calculating the geneLength by subtracting geneStart from geneEnd and adding it to totalGeneLength.
   If the value of feature column is not "exon" it increments the geneCount by 1. It then assigns the value of column 4 (geneStart) and next column (geneEnd). The code then checks of geneLength is greater than longestGeneLength, if it is it then assigns it to the variable along with that respective geneName.
   It then checks if geneLength is less than shortestGeneLength,and if it is, it assigns the gene length to shortestGeneLength and assigns geneName to shortestGeneName.

5. New exon Lists- In this part of code new lists are created, ”newExonstarts” and “newExonEnds” which will basically save the indexes of the start position and end position of the exons by subtracting with a constant (133746040), which is the start position of the gene present in the given FASTA file.

6. Indexes- The starting and ending indexes of every exon is printed on the terminal of our region of interest (i.e. in the given file of FASTA, GRCh38).

7. Display- The output is displayed in the different tabbed section in the main tab “Stats”. And the “clear” button is also designed to clear all the text areas and text panes.

c. Highlighting Exons:

- A *jButton* was created in order to highlight the exons present in the given FASTA file.
- The start index and final indexes of the exons were stored in the ArrayList named “newExonStarts” and “newExonEnds” which were calculated previously, but as the indexing starts from [0] in Java the number “1” had to be subtracted from every element of the two Arrays In order to give the exact start and end positions of the exons in the given FASTA.
- The highlighting is done to a JtextArea object called *jtextArea1* with the default highlight painter. The first number is the starting position for the indexing while the second number is the final position of the indexing to highlight.
- Now the values mentioned in second bullet point were added to this function (17 values) and the highlighting was performed.

NOTE- a previous method to highlight the exons in FASTA dynamically was tried but it was always overwriting the previous exons and thus only giving the final exons from the file, thus it is in the script but is commented under the name “DYNAMIC HIGHLIGHTING”.

4. Limitation

1) It is unable to calculate multiple FASTA files.(Any single FASTA file is good to go).
2) It wont work for other GTF files ,as it is not coded dynamically but for the given specific file (Although the Starting index and end index for the exon can be calculated dynamically by this program and it prints to the terminal. The values need to be changed manually in *jButton4* if they are to be highlighted in the respective FASTA file).

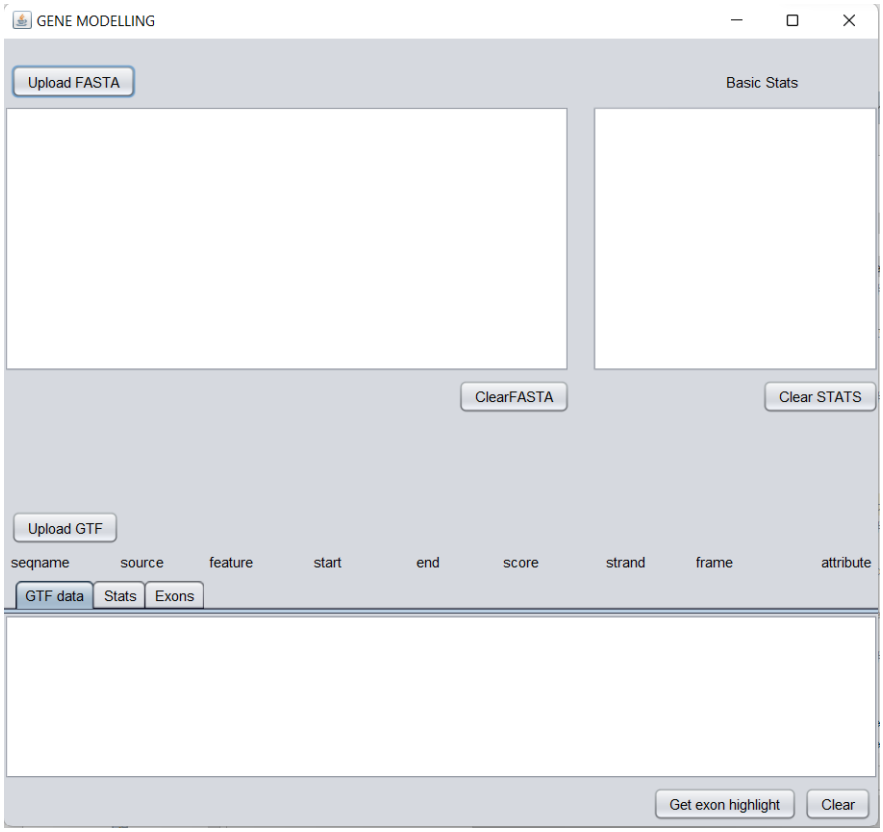
USER MANUAL

User pre-requisite: -

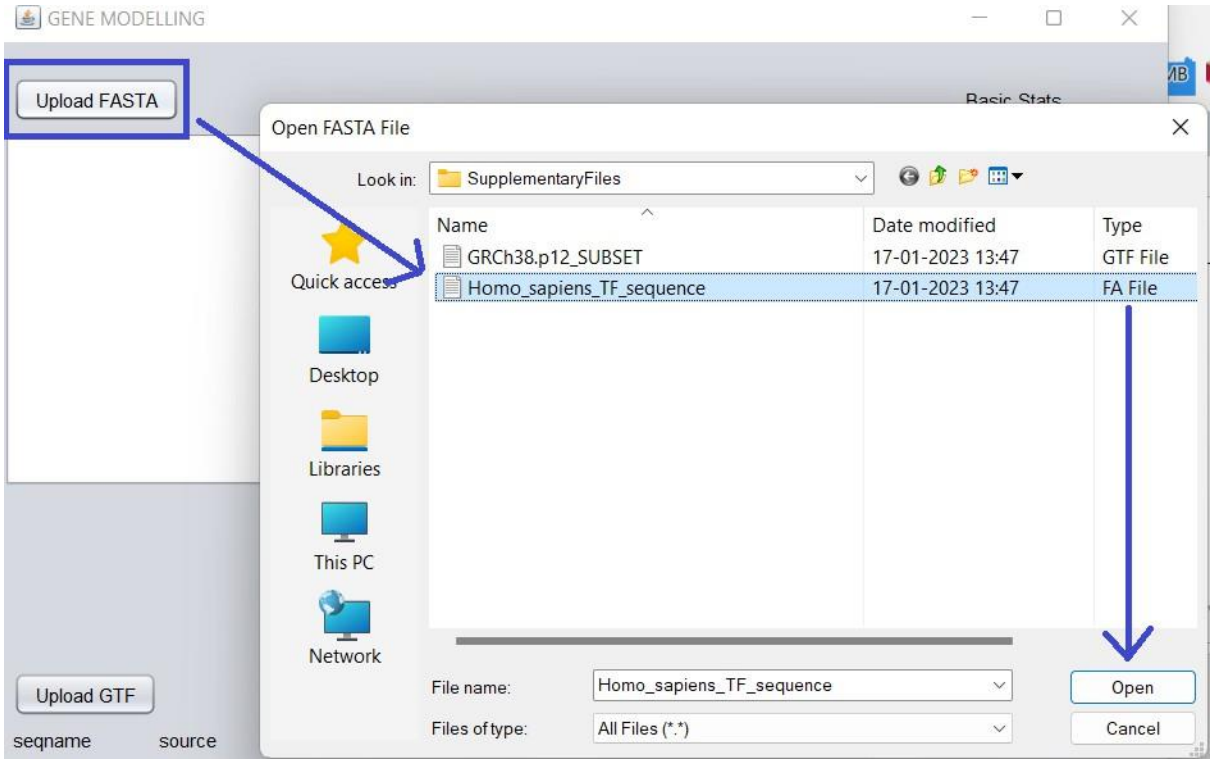
1) Latest version of Java installed.
2) FASTA and GTF files as given in the supplementary files.

Procedure:-

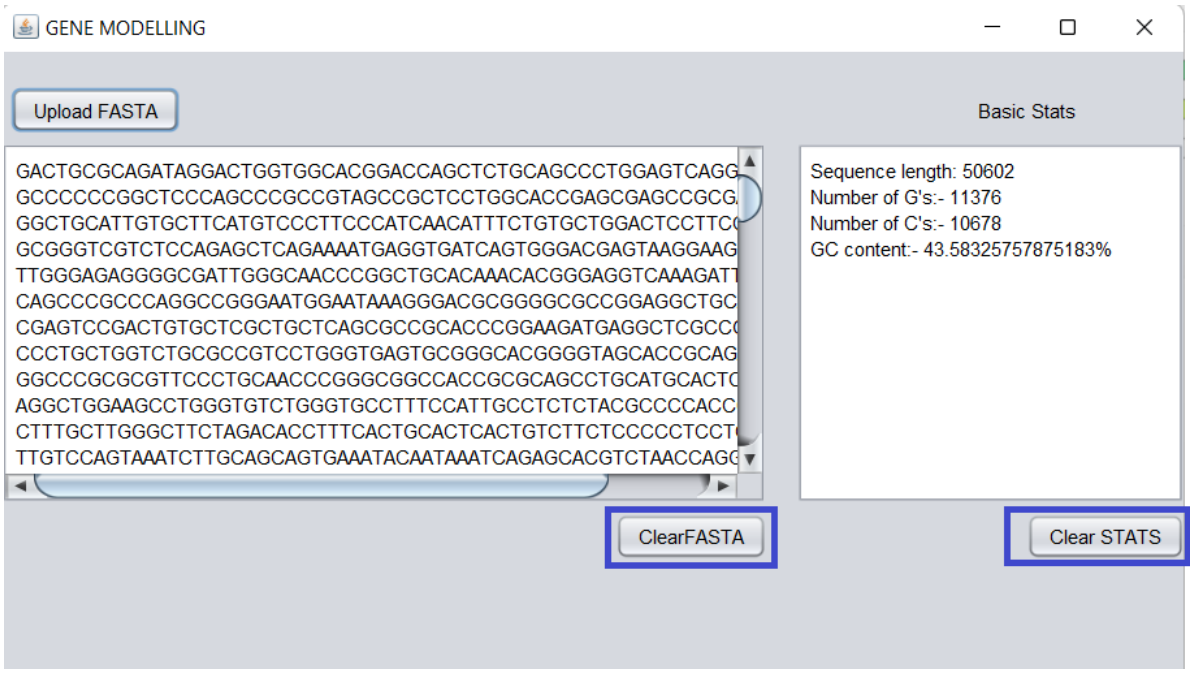
1) The user interface starts with the title “GENE MODELLING” and the user is shown this kind of screen. The user can maximise the tool for a better experience.



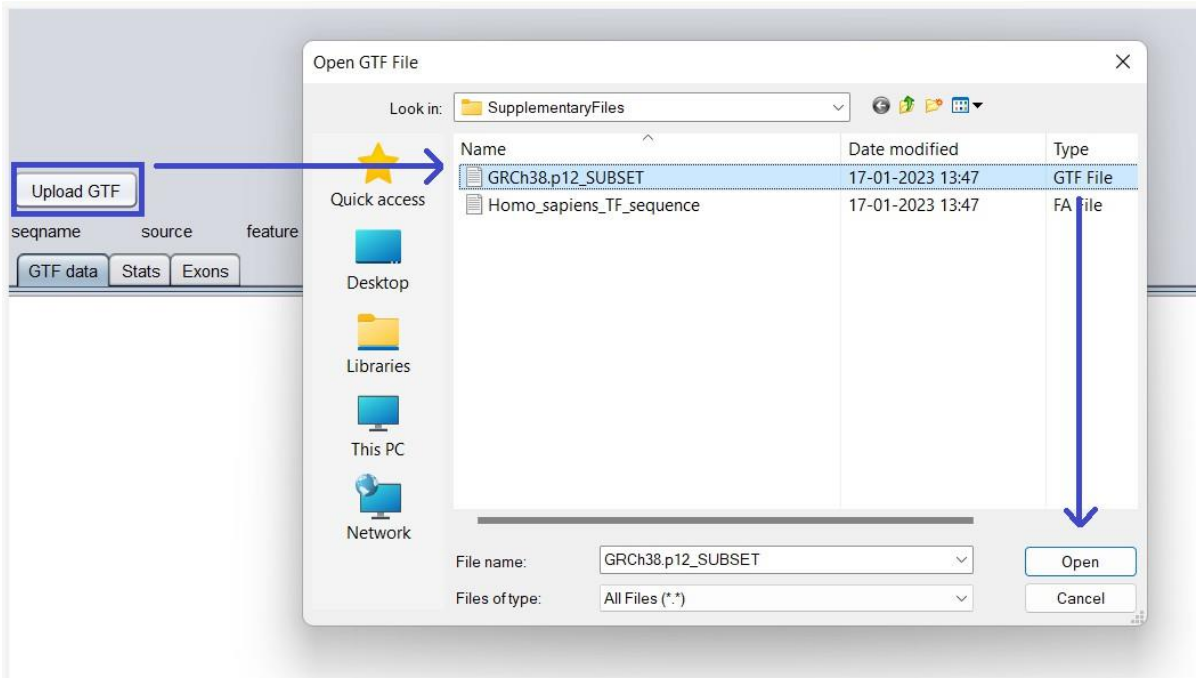
2) The user has to select the FASTA file from the button “Upload FASTA” and has to select it from the File chooser option which will pop up after the “upload FASTA” button. Then choose “open” after selecting the FASTA file.



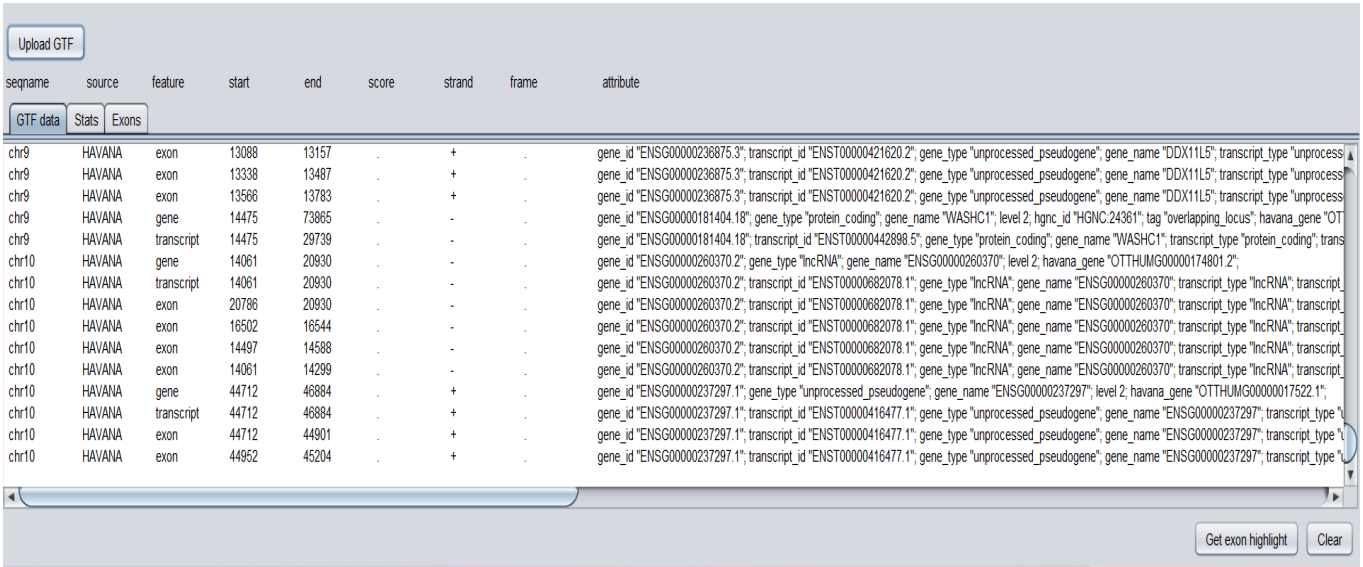
3) As soon as the file is opened the sequence is shown in the text area without the header and the basic stats of the FASTA (Number of G's ,Number of C's, GC content and sequence length) is calculated. The Basic Stats and FASTA sequence can be cleared from the clear button.



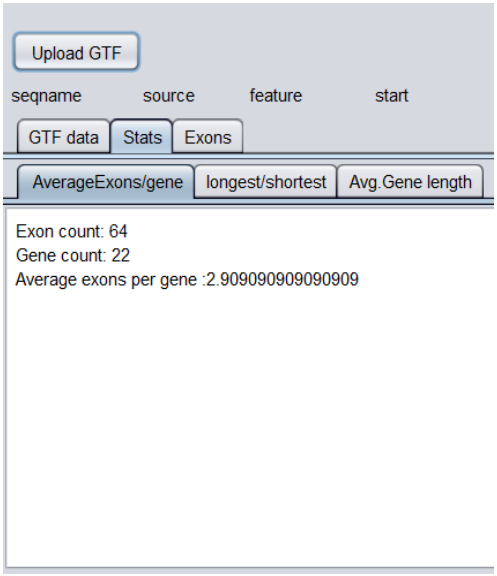
4) The user can then click on “Upload GTF” button and will be prompted to select the GTF file from the file chooser.

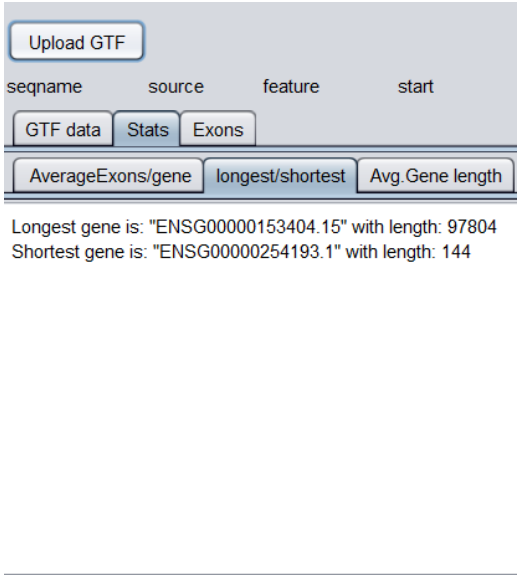


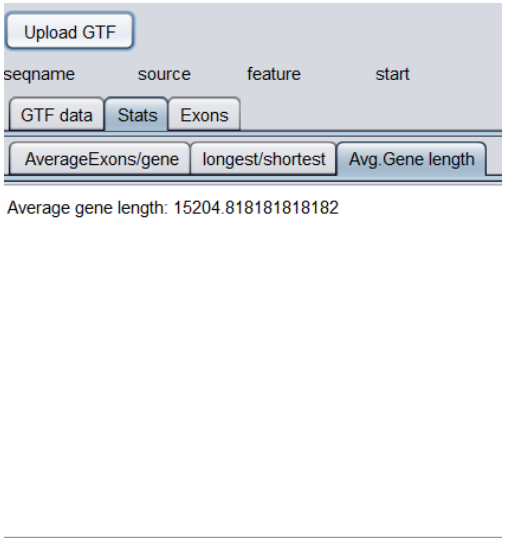
5) The user will be shown the GTF data which have been aligned with the labels above them if the screen is maximised. The user can clear the data from the text are using the “clear” button. (this clear button will remove the output from“GTF data”, “stats” and also the “Exons” tab).



6) The user can go to the Stats tab to see the basic stats for the GTF file, namely; Average exons per gene, longest/shortest gene and the average gene length.







7) The Text highlight of the exons in sequence of the given FASTA file can be seen when the “get Exon highlight ” button is pressed.

(Highlighted exons of the given FASTA file)