**Bioinformatics Python Project**

This project was conducted after the completion of my Introduction to Bioinformatics using Python module during my MSc in Applied Bioinformatics course at Cranfield University, UK.

**Technical Report**

The *summarise\_picard\_metrics.py* script was designed to work on plain Python, not Jupyter notebook or anything else, hence the “.y”. This program takes as input multiple subfolders containing text files, provided as a single input folder, analyses the text files in each subfolder by counting the number of files in each subfolder, makes summary text files and plots and then hands back the results to a user-defined output folder. For information on how to run this program, please see the user manual of this program attached to this report.

**Design/Structure of the Program**

The program has the following structure and design:

**Required libraries** – the required libraries/modules for this program were imported here. Libraries such as the *os module* (for interacting with files), *matplotlib* (for making plots), *pandas* (for creating data frames), *argparse* (for parsing command line arguments) and *numpy* (for generating numpy arrays) were imported and utilized in different functions of the script.

**Functions**

The following functions were used in the program:

1. *count\_source\_text\_files()* – this function counts the number of source files in the input folder. It takes as an argument, the folder path and returns the number of source text files in each subfolder in the path. This function was defined to count the number of samples in hs\_metrics, insert\_sizes and dedup\_metrics folders provided for this task.
2. *remove\_headers()* – this function removes headers starting with ‘#’ and ‘##’ that are not required for the analysis of the text files. It takes as an argument line and reads through the lines of each sample text file of the subfolders.
3. *combine\_and\_save\_summary\_text()* – this is the function that reads the content of the subfolders provided in the input folder, combines the text files and saves the summary text files to the output folder. It takes two arguments: the subfolder path and the output folder. The function also prints to the user in the command line the path to the folder where the summary files were saved. The function was also designed to exclude headers not starting with the “#” or “##” and maintain a single header in the summary text file. These summary text files were later used for plotting the sample summaries.
4. *combine\_and\_save\_hist\_text()* – this function extracts histogram data from the source files, combines them and saves to the output folder. It takes a subfolder path and an output folder as arguments. It also prints to the user the path to the output folder where the combined text files were saved. These files are generated only for duplicate and insert size metrices and were used to plot the histograms.
5. *hs\_summary\_plot()* – this function plots the hybridization selection metrics summary file. It takes as arguments the summary file, the output folder (where the result is saved) and the variables to be plotted. It also informed the user where this plot is saved.

A graph of a graph with blue and orange lines

Description automatically generated

**Figure 1**: Simple line summary plot of some selected hybridization metrics variables generated by *hs\_summary\_plot()*

**Interpretation** – the plot shows a correlation between the mean bait coverage and mean target coverage across all 12 samples of the hybridization selection metrics.

1. *dedup\_summary\_plot()* – this function plot the mark duplicates metrics summary file. It takes as arguments the summary text file, the output folder and the selected variables to be plotted. The generated example plot by this function is shown below.

A graph of data with blue and orange bars

Description automatically generated

**Figure2**: A summary barplot of unpaired reads examined and unpaired reads duplicates of duplicate metrics generated by the *dedup\_summary\_plot()*.

1. *insert\_summary\_plot()* – this function plot the insert size metrics as the two functions above. An example plot generated by this function is shown below.

A graph of blue and orange bars

Description automatically generated

**Figure3**: A summary barplot for median insert sizes and median absolute deviation variables of insert sizes metrics generated by the *insert\_summary\_plot()* function.

**Interpretation** – the median insert sizes and median absolute deviation of the insert size metrics show a similar pattern.

1. *dedup\_summary\_hist()* – the function plots a histogram of the extracted histogram values. It takes the summary histogram text file and the output folder as arguments. An example of a plot generated by this function is shown below.

A graph of a number of blue bars

Description automatically generated

**Figure 4**: A summary histogram for duplicate metrics generated by the *dedup\_summary\_hist()* function.

**Interpretation** – the histogram of the duplicate metrics does not show a normal distribution across the samples.

1. *insert\_summary\_hist()* – this plot a histogram of the summary histogram file and prints to the output folder. Below is an example of a histogram generated by this function.

A graph of a number of people

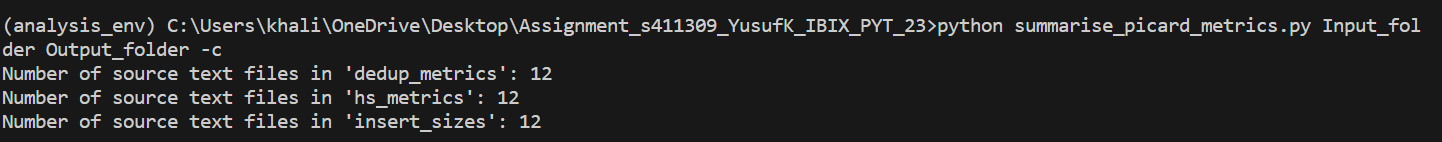
Description automatically generated with medium confidence

**Figure 5**: A summary histogram for insert size metrics generated by the *insert\_summary\_hist()* function.

**Interpretation** – the insert size histogram indicates that the smaller insert sizes are abundantly observed compared to the larger inserts.

1. *main()* – this serves as the entry point to the program. It contains all the *argparse* arguments for interacting with the user via the command prompt. The parser arguments provided by this are as follows:
2. **Required arguments** – the required arguments here are input and output folders. These arguments are compulsory must be provided by the user when running the program. The input folder is the folder containing the subfolders of the sample text files, while the output folder is where the result of the analyses will be printed.
3. **Optional arguments** – five optional arguments can be provided in addition to the required arguments and include the following:

* **args.counts**: provided using “-c” or “—counts”, when provided, the program calls the *count\_source\_text\_files()* and counts the number of source text files for each of the *hs\_metrics, insert\_sizes()* and *dedup\_metrics()* folders, prints to the user in the command prompt and exit the program. The screenshot below was generated using the “-c” argument.



* **args.hybridization**: provided using “-hyb” or “—hybridization”, when provided, the argument calls *count\_source\_text\_files()*, *combine\_and\_save\_summary\_text()* and *hs\_summary\_plot()* functions which provide the number of source text files, summary text file and plot for the hybridization selection metrics. See picture below.

A black screen with white text

Description automatically generated

* **args.inserts**: can be provided using “-ins” or “—inserts”, and will call the *count\_source\_text\_files()*, *combine\_and\_save\_summary\_text()*, *combine\_and\_save\_hist\_text()*, *insert\_summary\_plot()* and *insert\_summary\_hist()* and these functions will print the number of source text files, print to the output folder the summary text/hist files and plot and histogram respectively for the insert sizes metrics as shown below.

A screen shot of a computer screen

Description automatically generated

* **args.duplicates**: using “-dup” or “duplicates”, calls the *count\_source\_text\_files()*, *combine\_and\_save\_summary\_text()*, *combine\_and\_save\_hist\_text()*, *dedup\_summary\_plot()* and *dedup\_summary\_hist()* functions. They print the number of source text files, print to the output folder the summary text/hist files and plot and histogram respectively for the duplicate metrics as shown below.

A screen shot of a computer screen

Description automatically generated

* **args.all**: provided using “-a” or “—all” and calls all the functions and executes all the above tasks as shown below.

A screen shot of a computer

Description automatically generated

All the example plots, summary text files, and histograms obtained while testing the program were provided in a folder called ‘Output folder’, as well as the input folder containing the test data.