FQStat

A parallel architecture for very high-speed assessment of sequencing quality metrics

FQStat (http://www.otulab.unl.edu/FQStat) performs quality control (QC) analysis for DNA/RNA sequencing fastq files. FQStat is written in Python and uses a parallel programming architecture. In contrast to existing tools that assess the QC of sequencing data, FQStat introduces the following improvements:

1. Automatic configuration of system parameters (e.g., core assignment and file segmentation) for optimum performance.
2. Analysis of multiple data sets for comparative assessment of QC parameters.
3. Not being coupled with other preprocessing steps (e.g., low quality base trimming) for an easy-to-use, simple, and fast calculation of QC parameters only.
4. Generating analysis results separately at the lane-, sample-, and experiment-level so the users can pick and choose high quality subsets of the sample and/or experiment data.
5. Flagging low quality lanes and/or samples that warrant further analysis.
6. Generating publication quality output figures and tables.

Input

FQStat handles four experimental categories:

1. Paired-End sequencing, each sample run on multiple lanes
2. Paired-End sequencing, each sample run on a single lane
3. Single-End sequencing, each sample run on multiple lanes
4. Single-End sequencing, each sample run on a single lane

The fastq files and sample name descriptions should be arranged as follows:

- Two experiments or datasets (e.g., "**Raw**" and "**Trimmed**") in their respective folders (e.g., under one main folder, "**data**") containing the corresponding fastq files.
- One (or two, if the experiment is paired-end) text file(s) (e.g., **R1.txt** and **R2.txt**) that represents the fastq file names.
- The sample naming should consist of a prefix for sample ID (e.g., S1) followed by lane ID (e.g., a/b/c/d), followed by R1 or R2 (for forward or reverse paired reads).
- An example sample name sequence could be "S1aR1.fastq", "S1bR1.fastq", "S2aR1.fastq", "S2bR1.fastq", etc.

For a sample data structure, please consider the "**data**" folder distributed with the FQStat package.

FQStat can also handle compressed fastq files. Each sequencing .fastq file can be compressed using the .gz or .zip format.

Output

The comparative statistics provided by FQStat are

- Read Count
- Mean Read Length
- Mean Quality Score

- %bps above PHRED 25.0

FQStat generates two HTML files (one for graphs, one for tables) along with tabular and graphical data representing lane-level, sample-level, and experiment-level statistics for reads R1 (single-end) or R1 and R2 (paired-end) based on the type of experiment. The folder in which the output files are stored is determined by the <**Project ID**> option provided by the user.

Installation

FQStat is written in Python. The installation is complete upon downloading the FQStat package from our website.

There are two separate implementations, **FQStatGUI.py** (for the GUI version) and **FQStatCL.py** (for the command line version).

The folder "**data**" in the distribution contains an example 2-sample, 4-lane, paired-end dataset with accompanying text and fastq files.

The results of the analysis of this dataset with FQStat using default parameters are in the folder "**results**" that is included in the package. Users can utilize this dataset as a test case to make sure they can use FQStat correctly.

Command line Usage

Let <pathPY> denote the local python interpreter in your computer. Command line FQStat can be executed using

<pathPY> FQStatCL.py <OPTIONS>

Options Description:

-1 <Text file with the R1 fastq file names>

-2 <Text file with the R2 fastq file names >

-Q <PHRED quality score offset value used in the fastq files> (default:33)

-E <Type of experiment, choose **1 – 4**. 1: Paired-end/Multiple Lanes, 2: Paired-end/Single Lane, 3: Single-end/Multiple Lanes, 4: Single-end/Single Lane>

-R <Directory with the Experiment1 files (with R1 and R2 subdirectories holding respective fastq files)>

-T <Directory with the Experiment2 files (with R1 and R2 subdirectories holding respective fastq files)>

-P <Project_ID>

-D <DPI value for the constructed images > (default: 300)

-H <Image height in inches> (default: 12)

-M <Parallel/Serial Processing (choose **S** or **P**)>

-Z <z-score value cut-off to flag outlier samples/lanes> (default: 1.5)

-K <Experiment1 name without spaces (e.g., Raw)>

-L < Experiment2 name without spaces (e.g., Trimmed)>

-C <*max_core* maximum number of cores per file to be used in parallel processing> (default: 55)

-U <High quality score value> (default: 25)

-A <Path to python interpreter (e.g., /usr/bin/python3 in Linux and C:/Python35/python in Windows)>

Sample Commands for Parallel Processing

(for Serial Processing please use -M S instead of -M P)

The required files to run these sample commands are included in the distribution under the "**data**" folder.

In these examples, the local python interpreter is executed by the command "**python3**" and is located at **/usr/bin/python3**.

The files that contain the names of the fastq files are **R1.txt** and **R2.txt** under the "**data**" folder in the installation folder.

The two experiment names are **RAW** and **TRIMMED**

The fastq files for the two experiments are under the folders **/data/Raw_Fastq** and **/data/Trimmed_Fastq**

Image DPIs are **300** and image heights are **12"**.

A high-quality base is defined as a base that has a PHRED score above **25**.

Samples with QC parameters that have an absolute z-value > **1.5** will be flagged.

Paired-End, Multiple Lanes:

python3 FQStatCL.py -1 ./data/R1.txt -2 ./data/R2.txt -Q 33 -E 1 -R ./data/Raw_Fastq -T ./data/Trimmed_Fastq -P 601 -D 300.0 -H 12.0 -M P -Z 1.5 -K RAW -L TRIMMED -C 55 -U 25.0 -A /usr/bin/python3

Paired-End, Single Lane:

python3 FQStatCL.py -1 ./data/R1.txt -2 ./data/R2.txt -Q 33 -E 2 -R ./data/Raw_Fastq -T ./data/Trimmed_Fastq -P 602 -D 300.0 -H 12.0 -M P -Z 1.5 -K RAW -L TRIMMED -C 55 -U 25.0 -A /usr/bin/python3

Single-End, Multiple Lanes

python3 FQStatCL.py -1 ./data/R1.txt -Q 33 -E 3 -R ./data/Raw_Fastq -T ./data/Trimmed_Fastq -P 603 -D 300.0 -H 12.0 -M P -Z 1.5 -K RAW -L TRIMMED -C 55 -U 25.0 -A /usr/bin/python3

Single-End, Single Lane:

python3 FQStatCL.py -1 ./data/R1.txt -Q 33 -E 4 -R ./data/Raw_Fastq -T ./data/Trimmed_Fastq -P 604 -D 300.0 -H 12.0 -M P -Z 1.5 -K RAW -L TRIMMED -C 55 -U 25.0 -A /usr/bin/python3

GUI Usage

Step 1: Initiate the FQStat GUI

The GUI can be initiated by running the command

<pathPY> FQStatGUI.py

where <pathPY> denotes the local python interpreter on your computer.

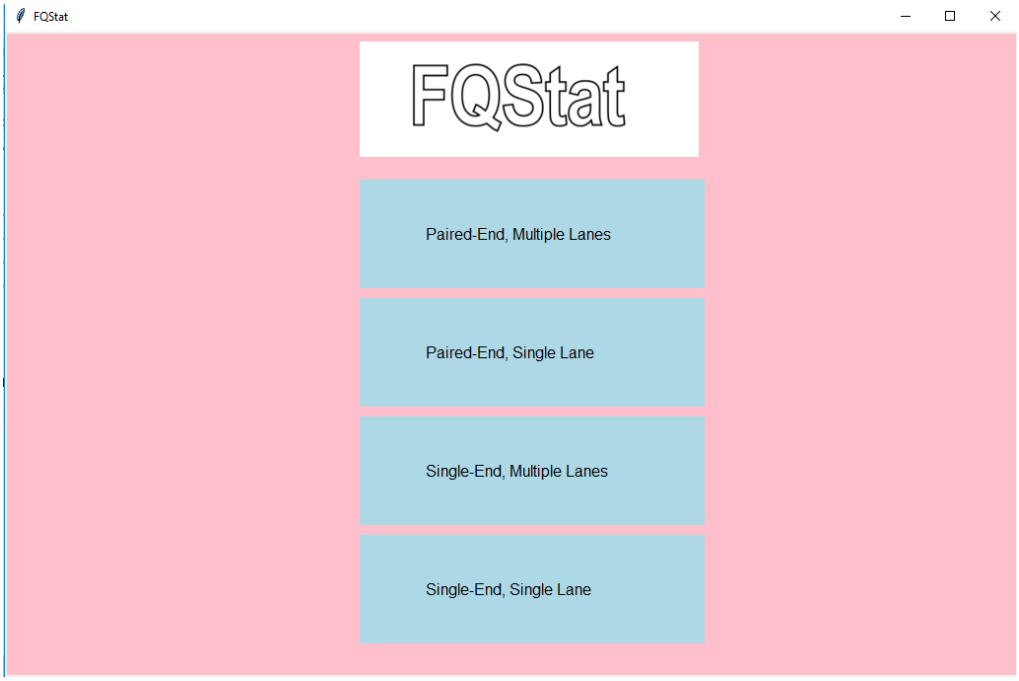

Figure 1: FQStat interface window with options to select from multiple experimental categories.

Once the experiment type is selected, the respective experiment-based main windows will open for data submission (Figures 2, 3).

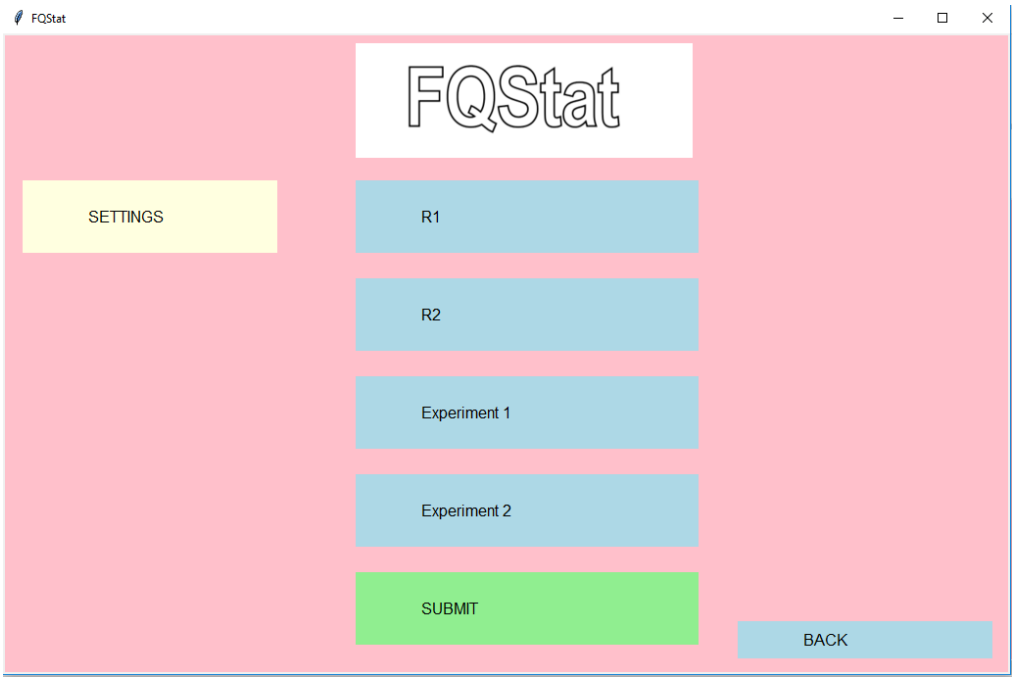

Figure 2: Main window for experiment categories “Paired-end, Multiple Lanes” and “Paired-End, Single Lane” data submission.

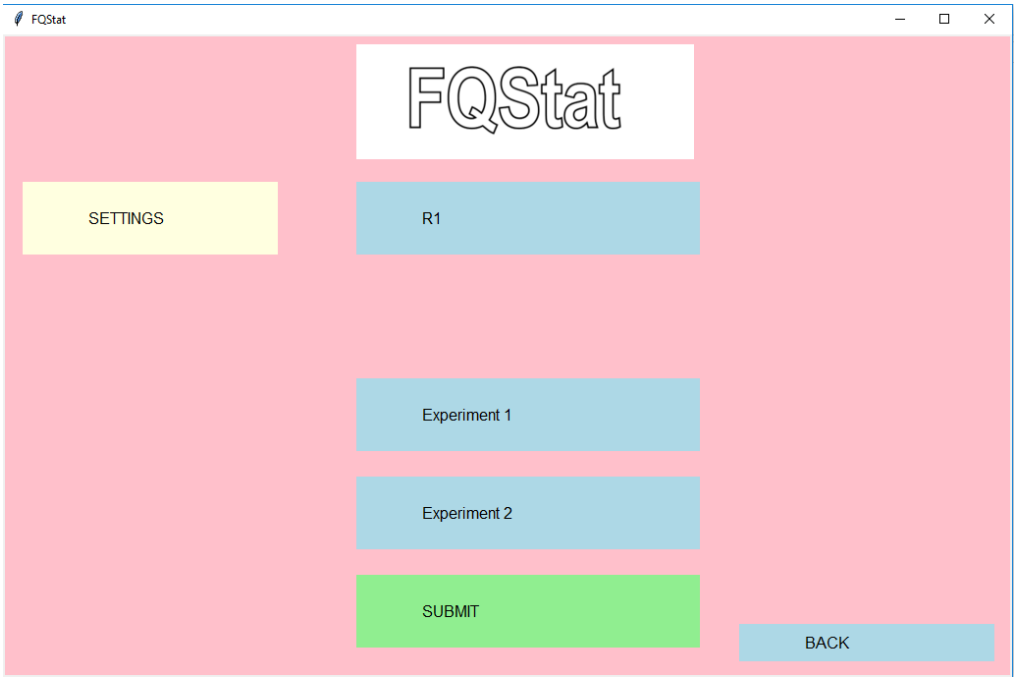

Figure 3: Main window for experiment categories “Single-end, Multiple Lanes” and “Single-end, Single Lane” data submission.

Note: Using the “Back” button in the experimental category window (Figures 2, 3), the user can go back to the initial window of the FQStat (Figure 1) to select another experimental category**.**

Step2: Press the "SETTINGS" button to set up program parameters.

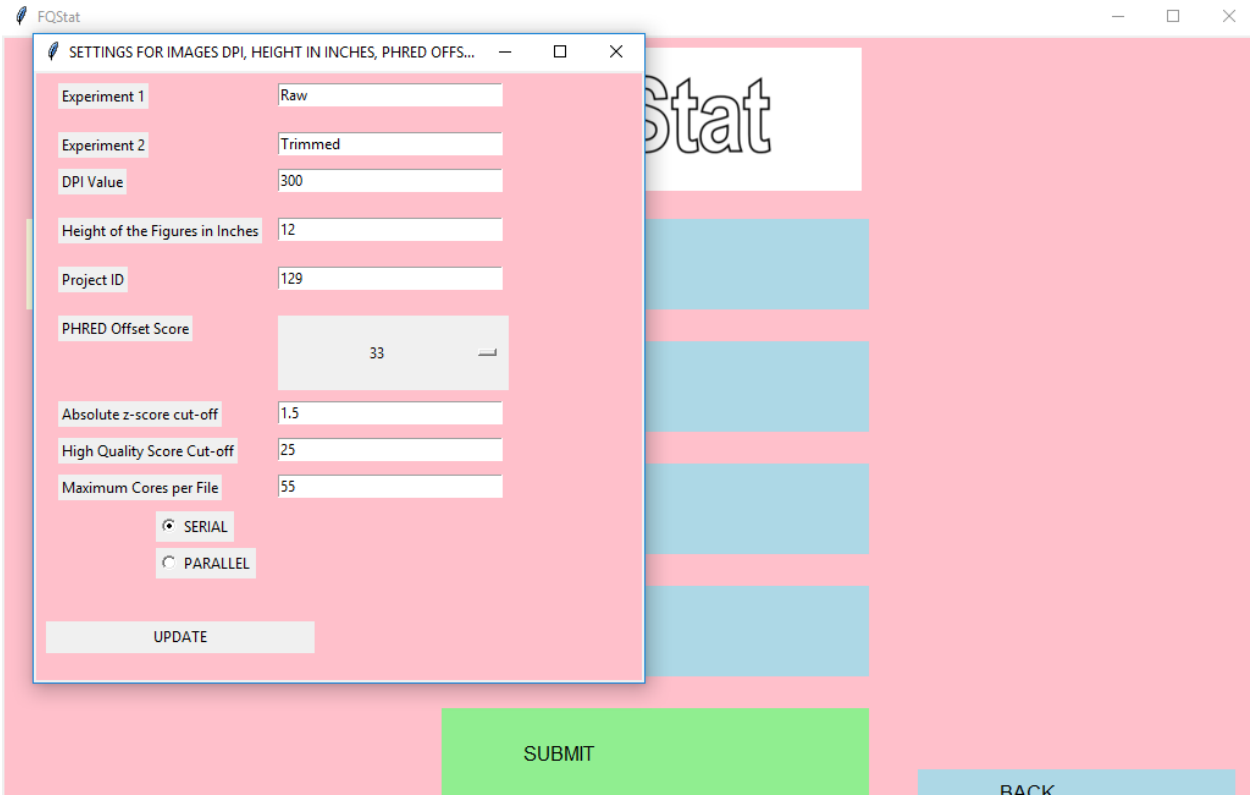

Figure 4: SETTINGS popup window and the parameters can be updated.

Table 1: List of the FQStat parameters with default and possible values.

Parameter Name	**Description**
Experiment1	Name of Experiment1
Experiment 2	Name of Experiemnt2
DPI value	Dots Per Inches value for the images. (Default: 300)
Height of the Figures in Inches	(Default: 12)
Project ID	Name of the output folder
PHRED Offset Score	(Default: 33)
Absolute z-score cut-off	Samples with a QC parameter that has a z-score above this cut-off value will be flagged. (Default: 1.5)
High Quality Score Cut-off	Bases with a quality score above this value will be considered as high-quality bases (Default: 25)
Maximum Cores per File	Maximum number of cores that can be assigned to a fastq file (Default: 55)
Radio Button: SERIAL/PARALLEL	The processing mode for FQStat

Users can click on the "UPDATE" button to set the chosen parameters. In order for Experiment1 and Experiment2 button names to be updated to the selected names for these experiments, the user needs to press the "BACK" button and reselect the experiment type.

Step 3: Submission of the data for FQSTAT Analysis

Once user sets preferred over "default parameters values" using "SETTINGS" popup window then user can submit data for the QC analysis in the Main window of the experimental category.

Here you are provided with sample submission process for the experimental categories "Paired-end, Multiple Lanes" and "Paired-end, Single Lane." For "Single-end, Multiple Lanes" and "Single-end, Single Lane" data submissions, omit step 3.2.

Step 3.1: Submit the file with the file names of the forward reads (R1) by clicking on the "R1" button (Figure 5).

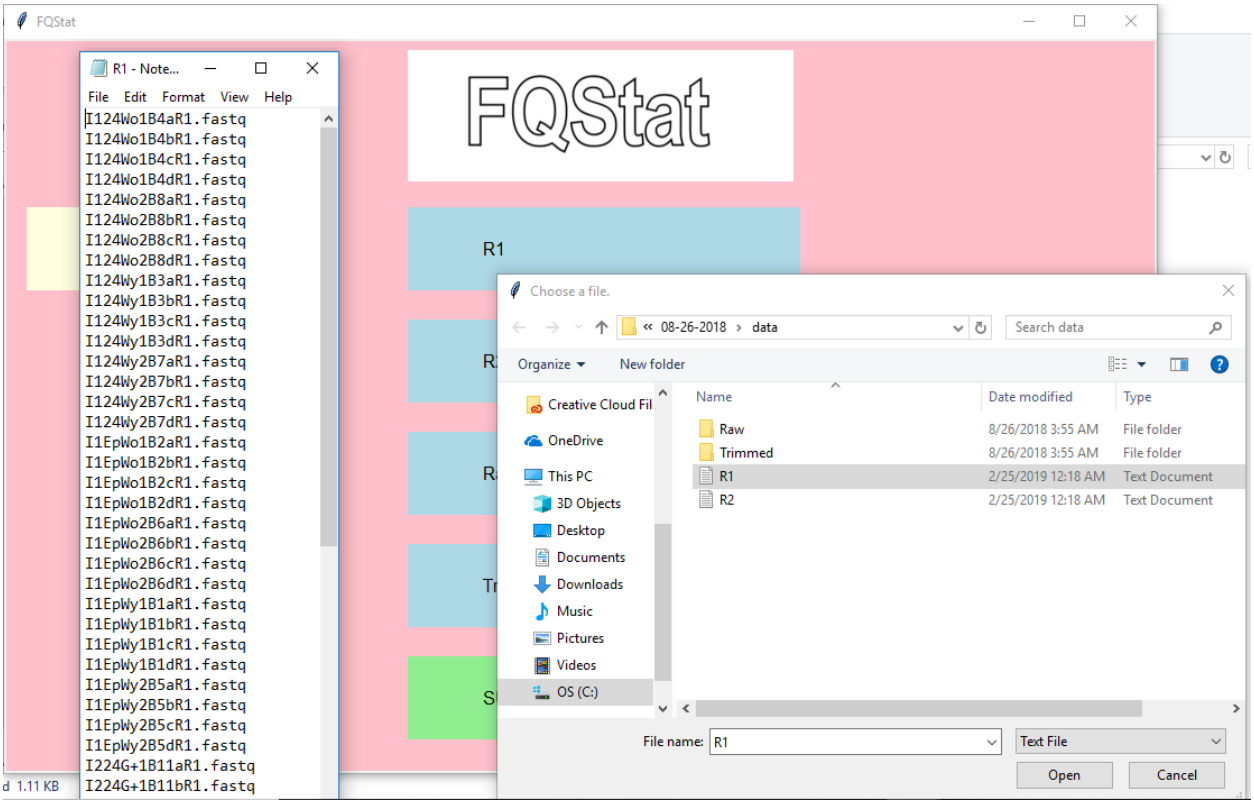

Figure 5: R1.txt file with the names of the fastq files that contain the forward reads.

After submitting the R1 file, if there are any errors in the format of the fastq file names, an error report window will pop up as shown in Figure 6. Otherwise, as shown in Figure 7, a pop-up window will appear stating that the R1 file has been successfully submitted.

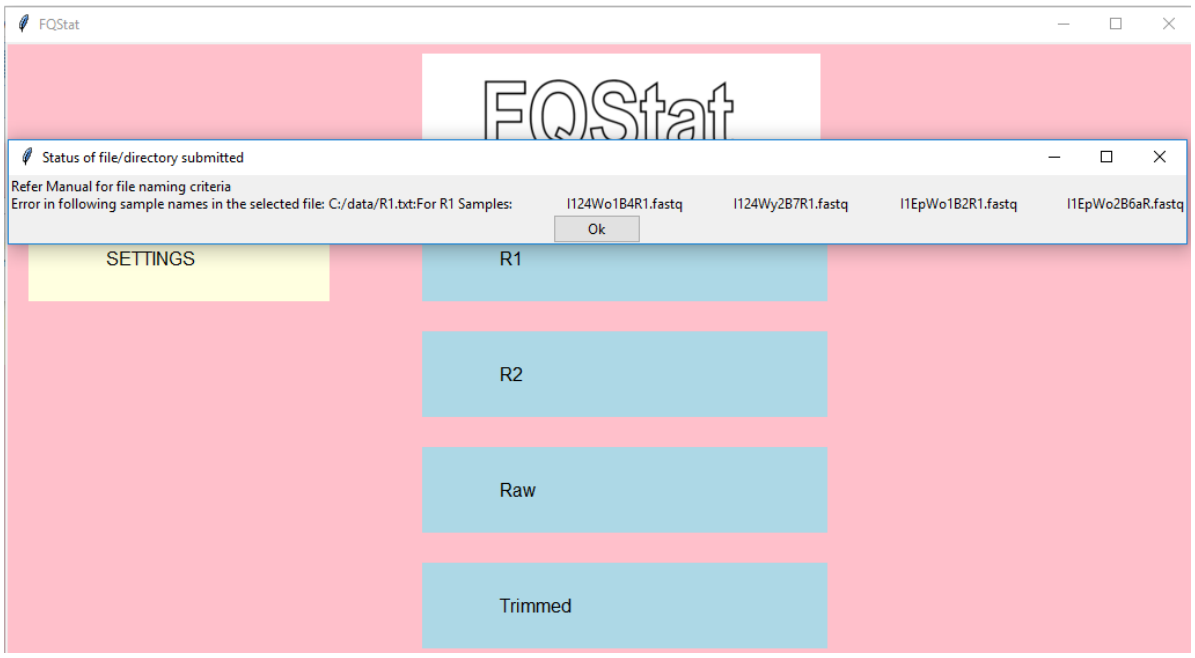

Figure 6: Error notification if the fastq file names provided in R1/R2 files are not following the file naming convention described in the "Input" section of this tutorial.

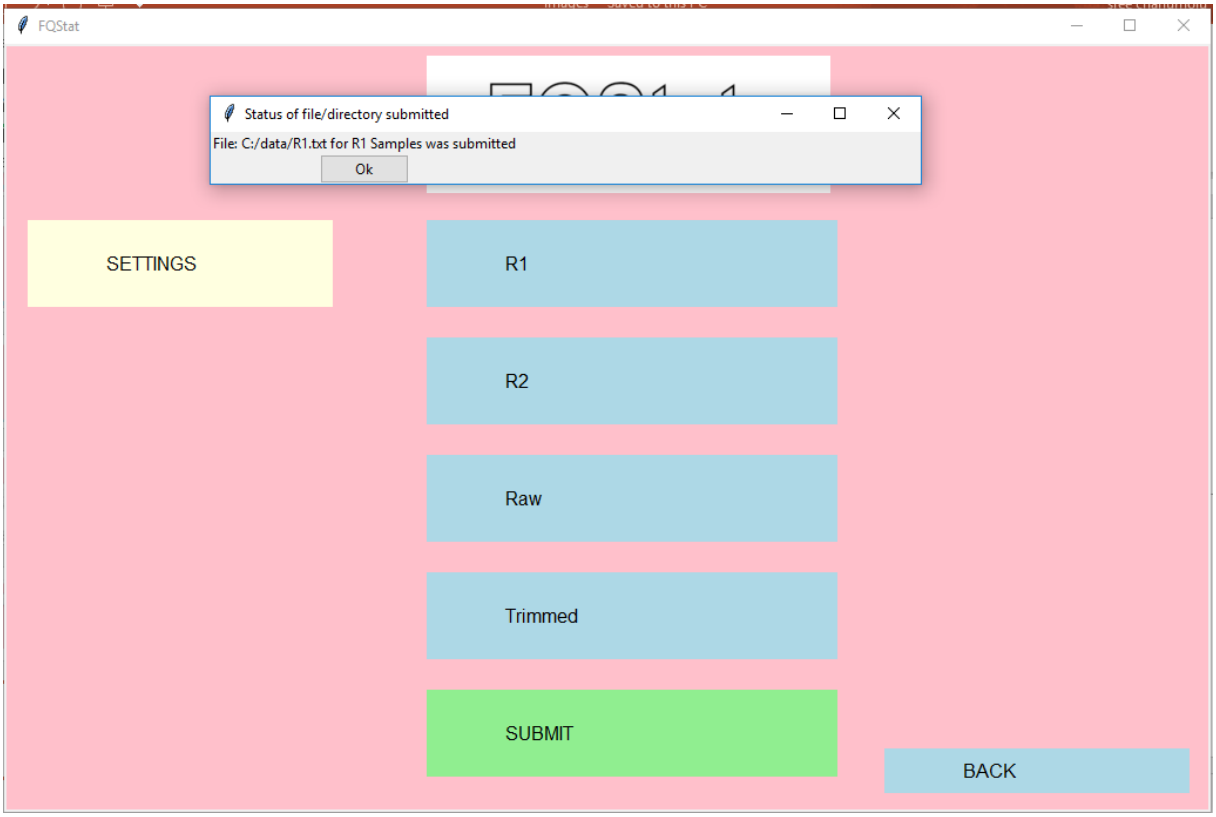

Figure 7: Pop-up window notification stating that the R1/R2 file has been successfully submitted.

Step 3.2: Submit the file with the file names of the reverse reads (R2) by clicking on the "R2" button (Figure 8).

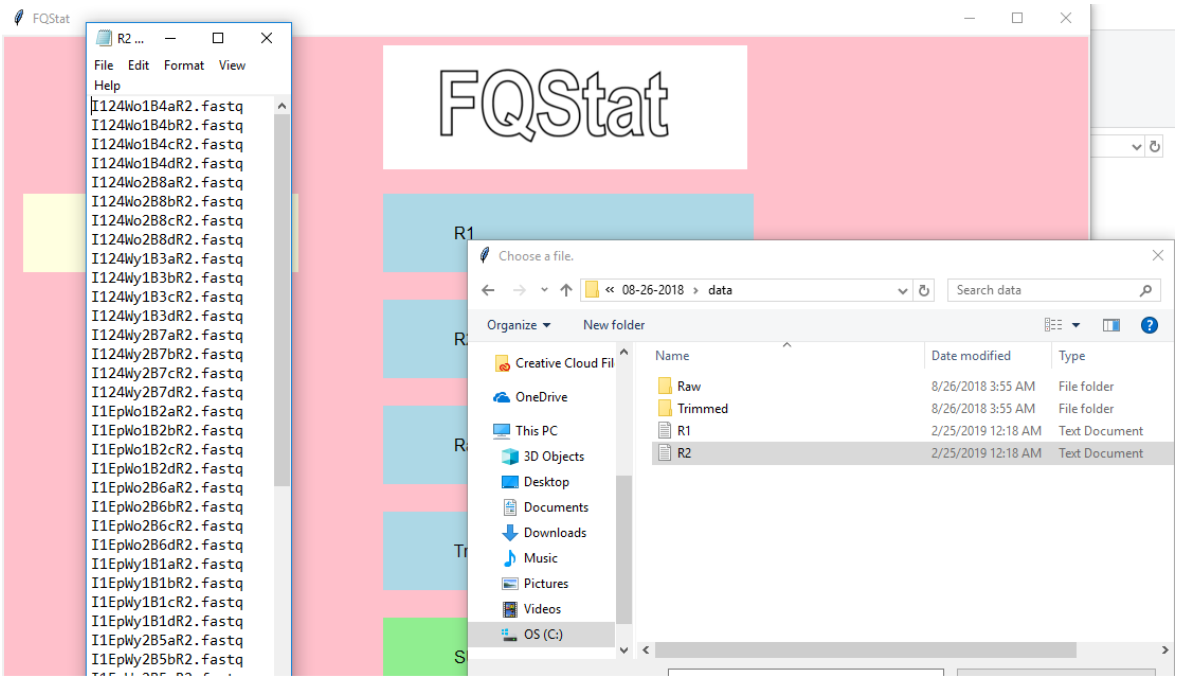

Figure 8: R2.txt file with the names of the fastq files that contain the reverse reads.

After submitting the R2 file, if there are any errors in the format of the fastq file names, an error report window will pop up as shown in Figure 6. Otherwise, as shown in Figure 7, a pop-up window will appear stating that the R2 file has been successfully submitted.

Step 3.3: Submit Experiment1 data (in this example by clicking the "Raw" button and selecting the "Raw" folder that holds the Experiment1 fastq files, Figure 9).

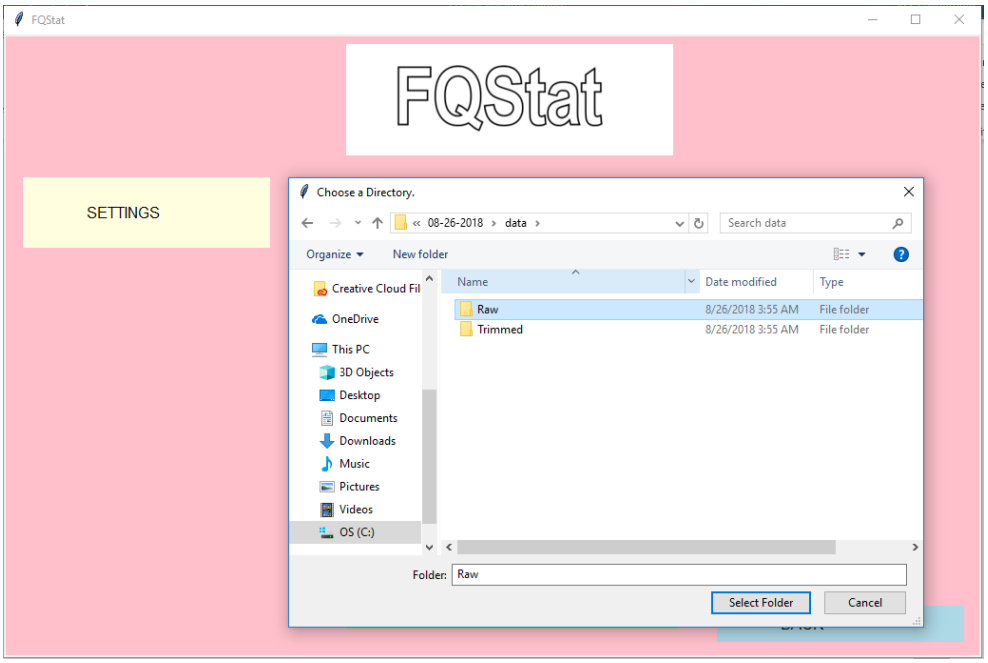

Figure 9: Submitting Experiment1 directory location

Step 3.4: Submit Experiment2 data (in this example by clicking the "Trimmed" button and selecting the "Trimmed" folder that holds the Experiment2 fastq files, Figure 10).

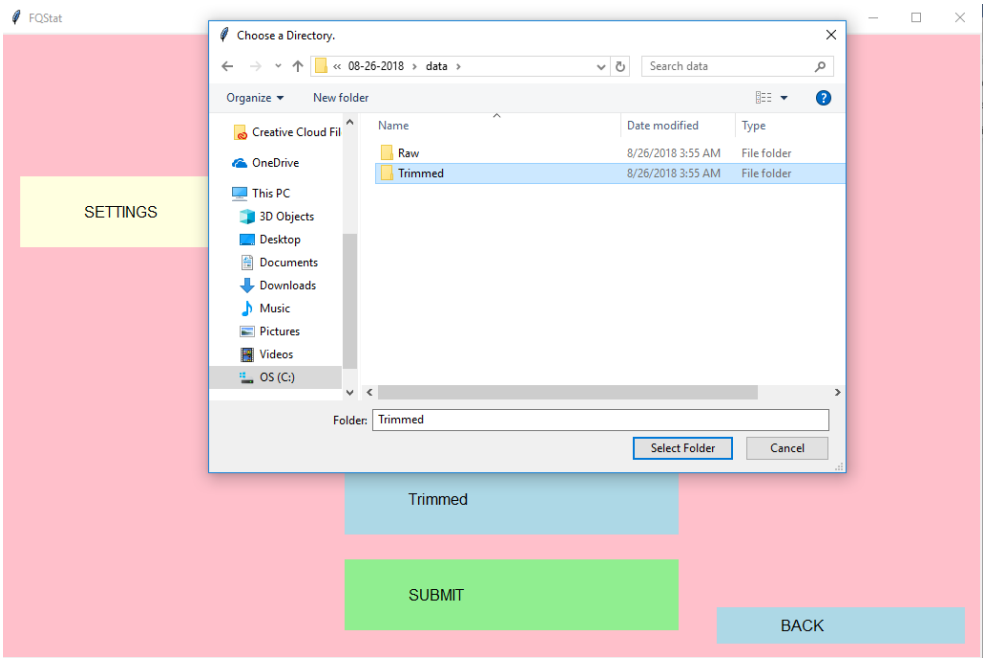

Figure 10: Submitting Experiment2 directory location.

Step 3.5: Click on the "SUBMIT" button to initiate the QC analysis.

Results

The results of the FQStat analysis are saved in "ProjectID" subdirectory under the folder "**results\projects**". A new subdirectory is created for each QC run performed by FQStat with the name provided in "Project ID" parameter in the "SETTINGS" pop-up window (Figure 11).

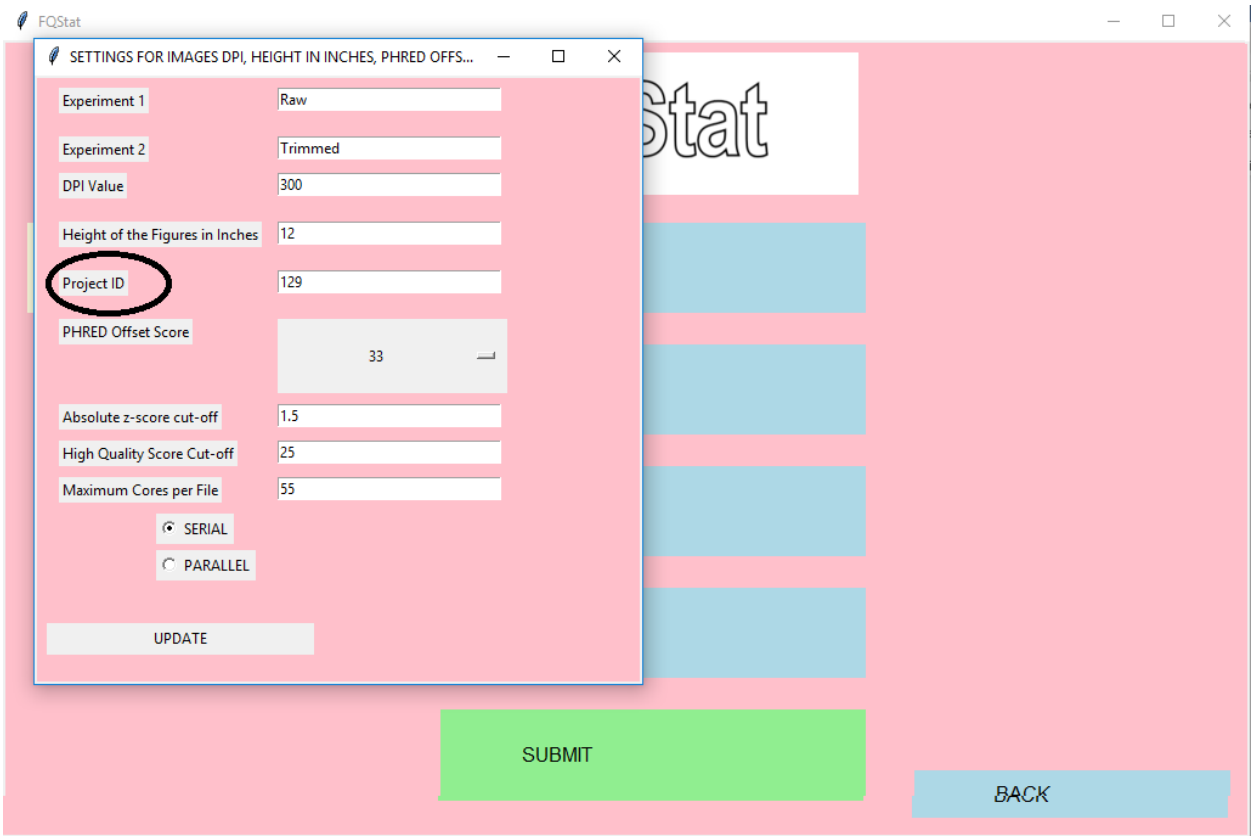

Figure 11: The Project ID information for the current QC test in the SETTINGS pop-up window.

As shown in Figure11, the user can find the current QC test results in subdirectory 129 under the "**results\projects**" directory. The contents of this directory are shown in Figures 12 and 13. Table 2 the files within this directory

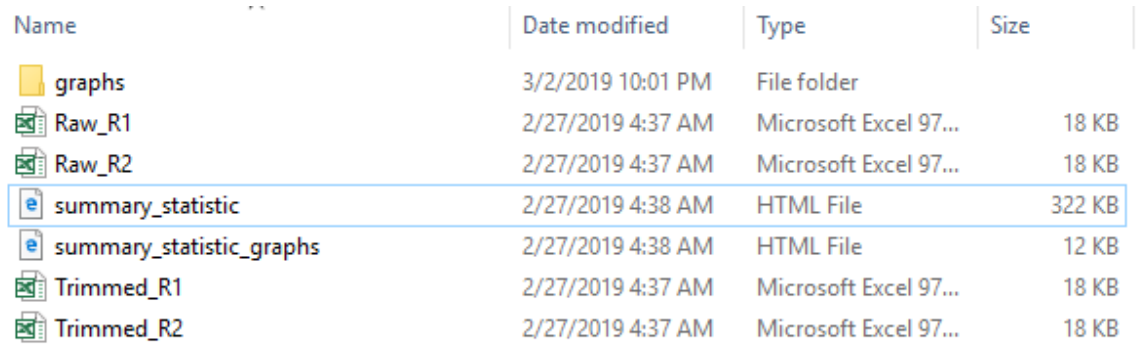

Figure 1: List of the files and directories in the <ProjectID> subdirectory.

Figure 2: The images for the QC comparison results in the subdirectory **"graphs."**

Table2: Description of the files/subdirectories in <ProjectID> directory.

File/Folder name	**Description**
Raw_R1.xls	QC statistics for the R1 reads of the Raw data
Raw_R2.xls	QC statistics for the R2 reads of the Raw data
Trimmed_R1.xls	QC statistics for the R1 reads of the Trimmed data
Trimmed _R2.xls	QC statistics for the R2 reads of the Trimmed data
summary_statistic.html	Tabulated QC comparative statistics of Raw and Trimmed data, along with flagged samples based on z-score
summary_statistic_graphs.html	All of the QC comparative statistics of Raw and Trimmed data in bar chart format
graphs	Folder containing the individual bar chart images of the QC comparative statistics

Explanation of the Results Files

Table 4: Description of the columns in the Raw_R1 (Raw_R2, Trimmed_R1, Trimmed_R2) file.

Column Name	Description
File Name	Sample file names
Read Count	Number of reads
Mean Read Length	Average of the length (in bp) of the reads
St. Dev. Read Length	Standard deviation of the length of the reads
Min. Read Length	Length of the shortest read
Max. Read Length	Length of the longest read
Median Read Length	Median of the length of the reads
25th Perc. Read Length	25th percentile of the length of the reads
75th Perc. Read Length	75th percentile of the length of the reads
Mean Quality Score (bp)	Average of the PHRED quality scores of all of the sequenced nucleotides
St. Dev. Quality Score (bp)	Standard deviation of the PHRED quality scores of all of the sequenced nucleotides
Median Quality Score (bp)	Median of the PHRED quality scores of all of the sequenced nucleotides
25th Perc. Mean Quality Score (bp)	25th percentile of the PHRED quality scores of all of the sequenced nucleotides
75th Perc. Quality Score(bp)	75th percentile of the PHRED quality scores of all of the sequenced nucleotides
%bps Above PHRED 20	Percentage of nucleotides with a PHRED score above 20
%bps Above PHRED 25	Percentage of nucleotides with a PHRED score above 25
%bps Above PHRED 30	Percentage of nucleotides with a PHRED score above 30
%bps Above PHRED User Value (Default: 25)	Percentage of nucleotides with a PHRED score above the user defined value
Mean of Mean Read Quality Score	Average of the average PHRED quality scores of the reads
St. Dev. Mean Read Quality Score	Standard deviation of the average PHRED quality scores of the reads
Min. Mean Read Quality Score	Minimum of the average PHRED quality scores of the reads
Max. Mean Read Quality Score	Maximum of the average PHRED quality scores of the reads
Median Mean Read Quality Score	Median of the average PHRED quality scores of the reads
25th Perc. Mean Read Quality Score	25th percentile of the average PHRED quality scores of the reads
75th Perc. Mean Read Quality Score	75th percentile of the average PHRED quality scores of the reads

Table 5: Description of the tables in the "**summary_statistic.html**" file.

Name of the Section	Description
R1/R2 Lane-Level and Sample-Level Summary statistics	Comparative statistics of Read Count, Mean Read Length, Mean Quality Score, and %bps Above PHRED 25.0 (or user given value)
R1/R2: Experiment-Level Read Count Statistics (Lane and Sample Resolution)	The mean, median, standard deviation, 25th, and 75th percentile values of statistic at the Experiment Level is either calculated using each lane file as a data point (lane resolution) or each sample file (combined lanes, if any) as a data point (sample resolution).
R1/R2: Experiment-Level Read Length Statistics (Lane and Sample Resolution)	
R1/R2: Experiment-Level Mean (bp) Quality Score Statistics (Lane and Sample Resolution)	
R1/R2: Experiment-Level %bps Above PHRED 25.0 Statistics (Lane and Sample Resolution)	

Table 6: List of the Experiment1/Experiment2 data QC comparative statistics presented in the "**summary_statistic_graphs.html**" file in bar-chart form. These bar charts can be found in the "**graphs**" folder as individual images.

Graph Type
R1/R2: Read Count Distribution (Lane and Sample Resolution)
R1/R2: Mean Read Length Distribution (Lane and Sample Resolution)
R1/R2: Mean (bp) Quality Score Distribution (Lane and Sample Resolution)
R1/R2: %bps Above PHRED 25 Distribution (Lane and Sample Resolution)

Description of the QC statistics generated by FQStat

N: Number of reads in a sample
l: Length of a read
Q: Base pair quality score
W: Number of samples
X: Number of lanes

1. l_i: Length of the i^{th} read where i € [1…N]
2. Q_{ij}: Quality score of the i^{th} read's j^{th} base pair where i € [1…N] and j € [1… l_i]
3. S_m: Name of the m^{th} sample where m € [1… W]
4. L_n: Lane name where n € [1… X]
5. R1: reads belonging to forward orientation
6. R2: reads belonging to reverse orientation
7. S_m L_nR1 represents the file name for the R1 reads that belong to $S_m{}^{th}$ sample's, $L_n{}^{th}$ lane where m € [1… W] and n € [1… X].
8. S_m L_nR2 where m € [1… W] and n € [1… X] represents names of the R2 reads file belonging to $S_m{}^{th}$ sample, $L_n{}^{th}$ lane

9. S_m R1 =$S_m\sum_{n=1}^{x} L_n$R1
10. S_m R2 =$S_m\sum_{n=1}^{x} L_n$R2
11. Mean Read Length (MRL)= $\sum_{n=1}^{x} l_i$ / N
12. Mean Quality Score (MQS)=$\sum_{i=1}^{N}\sum_{j=1}^{li} Q_{ij}/\sum_{n=1}^{x} l_i$
13. %bp above PHRED score 25 (QPHRED25)= $((\sum_{i=1}^{N}\sum_{j=1}^{li}(Q_{ij}>25)/\sum_{n=1}^{x} l_i)/MQS)*10$

- For each of the following, first a C vector is obtained as described (Separately for R1 and R2).
- Then, comparative graphs are plotted for both experiments using the data points in C.
- Experiment-level statistics at lane-resolution are calculated based on lane-level C.
- Experiment-level statistics at sample-resolution are calculated based on sample-level C.
- The calculation of the "mean" of C is given as an example at the experiment-level (both at lane-level and sample-level resolutions). Standard deviation, median, 25th and 75th percentiles of C are accordingly calculated.

A) Read Count Statistics:

Lane-Level:
Let C denote the array containing the number of reads in each lane for each sample.
C=[N_{SmLnR1}], where m € [1… W] and n € [1… X]. Size of C is WX.
$Mean$: $(\sum_{m=1}^{W}\sum_{n=1}^{X} N_{\text{Sm LnR1}})/ (W * X)$

Sample-Level:
Let C denote the array containing the number of reads in each sample.
C=[R_m] where m € [1… W] and R_m= $N_{\text{Sm R1}}$ and n € [1… X]
$Mean$: $(\sum_{m=1}^{W}((\sum_{n=1}^{X} N_{\text{Sm LnR1}})/X)/ (W)$

B) Read Length Statistics:

Lane-level:
Let C denote the array containing the average read length in each lane for each sample.
C=[MRL_{SmLnR1}] where m € [1… W] and n € [1… X]
$Mean$: $(\sum_{m=1}^{W}\sum_{n=1}^{X} MRL_{\text{Sm LnR1}})/ (W * X)$

Sample-Level:
Let C denote the array containing the average read length each sample.
C=[R_m] where m € [1… W] and R_m= $\sum_{n=1}^{X} MRL_{\text{Sm LnR1}}$ and n € [1… X]
$Mean$: $(\sum_{m=1}^{W}((\sum_{n=1}^{X} MRL_{\text{Sm LnR1}})/X)/ (W)$

C) Mean Quality Score Statistics:

Lane-level:
Let C denote the array containing the average of mean-read-quality in each lane for each sample.
C=[MQS_{SmLnR1}] where m € [1… W] and n € [1… X]
$Mean$: ($\sum_{m=1}^{W}\sum_{n=1}^{X} MQS_{\text{Sm LnR1}}$)/ (W * X)

Sample-Level:
Let C denote the array containing the average of mean-read-quality in each sample.
C=[R_m] where m € [1… W] and R_m= $\sum_{n=1}^{X} MQS_{\text{Sm LnR1}}$ and n € [1… X]
$Mean$: ($\sum_{m=1}^{W}((\sum_{n=1}^{X} MQS_{\text{Sm LnR1}})/X)$/ (W)

D) %bp with PHREAD above 25 score:

Lane-level:
Let C denote the array containing the percentage of the high-quality base pairs in each lane for each sample.
C=[$QPHRED_{SmLnR1}$] where m € [1… W] and n € [1… X]
$Mean$: ($\sum_{m=1}^{W}\sum_{n=1}^{X} QPHRED_{\text{Sm LnR1}}$)/ (W * X)

Sample-Level:
Let C denote the array containing the percentage of the high-quality base pairs in each sample.
C=[R_m] where m € [1… W] and R_m= $\sum_{n=1}^{X} QPHRED_{\text{Sm LnR1}}$ and n € [1… X]
$Mean$: ($\sum_{m=1}^{W}((\sum_{n=1}^{X} QPHRED_{\text{Sm LnR1}})/X)$/ (W)