FASTQC Bash Script

1.0 Key features of the script

1.1 Fully scalable

- Renames fq.gz files based on information retrieved from sample detail file, for easier identification. Due to this, it will work for other lifecycle stages, and if additional samples were added, or removed. With the user defining the raw data path, and all files being named and searched for with variables, and loops, it is fully scalable and built to accommodate different directories, reference genomes and data sets. The single tab delimited plain txt file has also been scripted to accommodate more than 2 lifecycles.

1.1 Following FASTQC analysis, user has the ability to define which samples he wishes to view more in-depth details on, with a function that opens its html file.

- After seeing both tables of fastqc summary and warned/failed modules. The user may want to see in-depth FASTQC analysis details of specific samples.
- By typing "*variable*" , the script will search for all html files containing the word "*variable*", thus opening all the html files with the word "*variable*", allowing for in-depth analysis of each of the FASTQC modules.
    - E.g. Slender → opens all slender samples' html file
    - E.g. 222_L8 → Opens both forward and reverse 222_L8 html files

1.2 Array for user to select which samples he/she deemed to have failed FASTQC analysis. These selected samples will not be aligned.

- Following step 1.1, the user may decide that a sample has failed the FASTQC analysis. These samples deemed to have failed can be selected by an array, and will subsequently be ignored during alignment and counting of genes.

2.0 Requirements to ensure script runs successfully:

2.1 Raw data and all Required files have to be within a single parent directory

- All required files such as raw sequence data, sample details, reference genome, have to be stored within one main directory (storing them in subdirectories within the master directory is fine).

2.2 User input of Raw Data path AND output directory

- The second step of the script requires the user to enter the path of the directory where all Raw Data and Required files are. **The path entered by the user must not have a "/" on the end of it.**

    - Correct input - /localdisk/data/BPSM/Assignment1
    - Wrong input - /localdisk/data/BPSM/Assignment1/

2.3 Array for selection of samples that failed FASTQC

- As this uses paired-end sequencing data, if either the forward or reverse sequence was deemed by the user to have failed the FASTQC analysis, both the forward and reverse sequences must be ignored. The array currently requires the user to manually select both the reverse and forward sequences, if the user only selects one sequence from the pair, it would fail.

3.0 Justification of parameters/flags:

3.1 FASTQC

- **-t 50**
    - 50 threads will enable all FASTQC samples to be ran simultaneously at a significantly faster rate.
- **--quiet**
    - Prevents the terminal from being flooded with useless information
- **--extract**
    - Automatically extracts the output of FASTQC, preventing the need for an additional command to unzip

3.2 BOWTIE2

3.2.1 BOWTIE2 INDEX

- **--quiet**
    - Prevents the terminal from being flooded with useless information
- **--threads 50**
    - Enables the creation of the index at a significantly faster rate.

3.2.2 BOWTIE2 ALIGNMENT

- **--no-mixed**
    - We assume incorrectly that **the gene has NO INTRONS**. As the gene has no introns, we **do not expect any variants.** Since there was equal number of R1 and R2 reads, if only one of the mates successfully maps/aligns, we can ignore it as orphaned reads which are due to variations or from trimming. We did not trim, and we do not expect any variations.
- **--no-unal**
    - unaligned reads are not included in the bam file output. As we're assuming incorrectly that there are no introns the gene, all reads should align.
- **--no-discordant**

    - Discordant read-pairs are classified by both reads of the pair, being mapped to the reference genome, however they coordinates do not agree with the insert size, or may map to different chromosomes. These discordant pairs are important for structural variation analysis, however we are assuming incorrectly that the gene has no exons, therefore no alternative splicing etc should be present. These discordant reads are now deemed to be unaligned reads, and thus should be ignored.
- **-p 50**
    - 50 threads will enable faster alignment

3.3 SAMTOOLS

- **-@ 50**
    - Multithreading will enable faster sorting and indexing
- **-bS**
    - Converts the output from the alignments directly into BAM files

3.4 BEDTOOLS

- **Multicov -D**
    - Multicov counts multiple BAM files
    - The parameter -D includes duplicate reads in the gene count. From a recently published paper on the impact of removing duplicates on differential expression analysis by RNA-seq, it was concluded that computational removal of duplicates during differential expression analysis is not recommend. We were not provided with information on the number of PCR-cycles used to create the cDNA library, however the paper mentions that duplicates caused by artefacts from PCR-cycles are negligible if there was sufficient starting material, thus less PCR-cycles. (Parekh et al., 2016). Duplicates can also come from highly expressed genes or short RNA, therefore I've decided to keep the duplicate reads.

4.0 Difficulties encountered:

4.1 Scalability

- The main difficulty encountered was making the **script fully scalable**, and with **minimal user input**. I've managed to minimise the user input to the bare minimum. However it will still require more user input compared to a script that is hard-coded to only analyse the Assignment's files.

4.2 Selection of samples that failed FASTQC analysis

- Another difficulty was **creating the array that would enable the user to select which samples had failed the FASTQC analysis** after a more in-depth analysis of the FASTQC results in Firefox. As mentioned earlier, the array requires the user to select both the forward and reverse sequence. If time had allowed, I would have coded it such that a single input would select both the forward and reverse sequence.

4.3 Deciding whether or not to keep/count duplicate reads while using bed tools

- This is a highly debated question with no concrete answer, as computational methods do notwhich does not distinguish PCR-artefacts from natural duplicates.

5.0 Additional features that would be beneficial to include:

5.1 Trimming of samples

- Adding this option will allow for users to trim their sequences if the FASTQC analysis brings errors on adaptor content.

5.2 Enable skipping of files that have already been analysed/aligned. So that when new data is added, previous data will not have to be re-analysed.

- This can be achieved by creating an if loop, testing if the fastqc output file for the sample exists, then skip the fastqc.
  - E.g. if [[-f "$file"]]; then

5.3 Enabling the user to input maximum and minimum fragment lengths for the bedtool paired-end parameters.

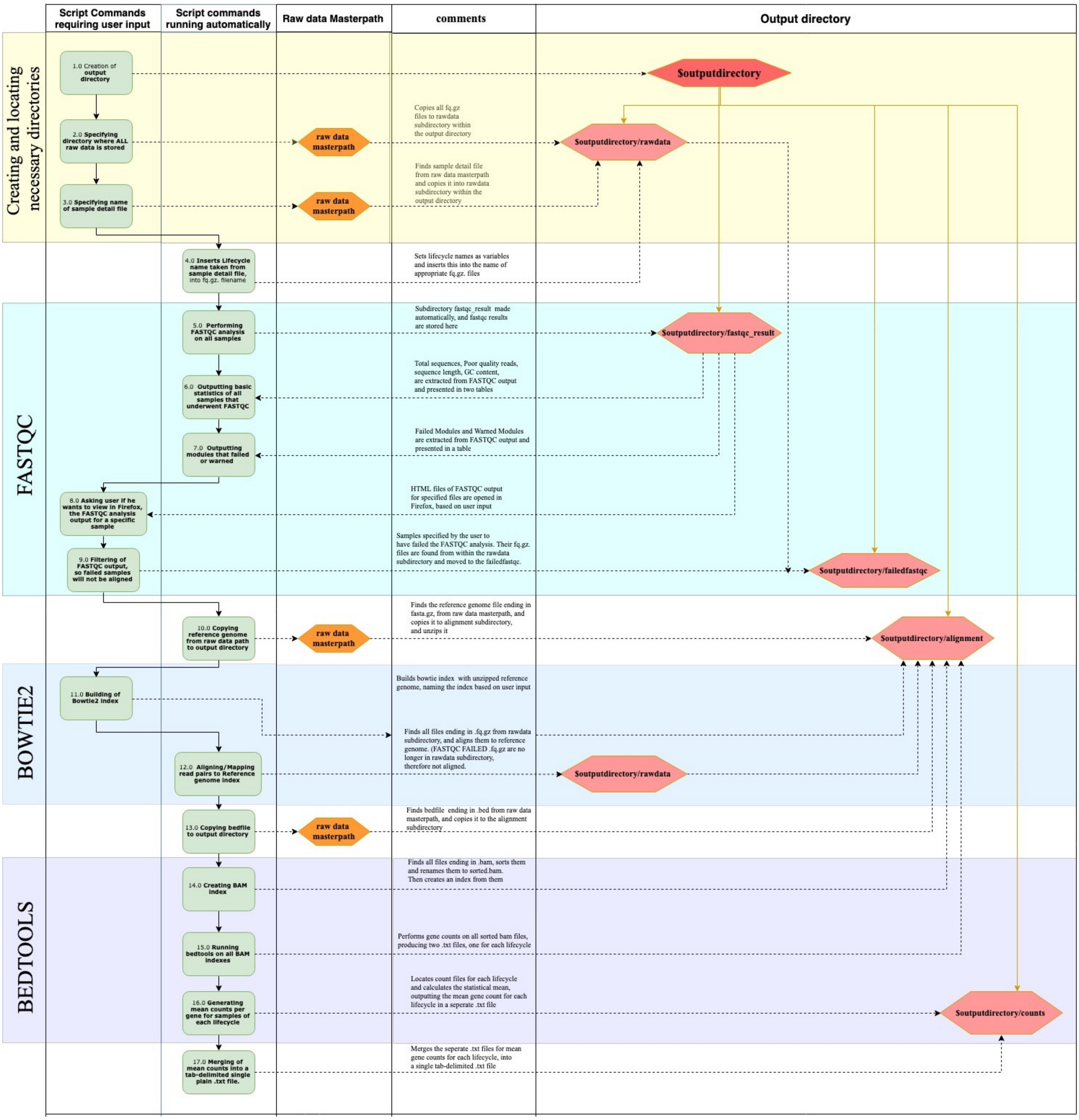
- -I/--minins <int> and -X/--maxins <int>
  - The default setting of bowtie2 is a minimum length of 0 and maximum length of 500. If the user has information about the insert size that he selected for library preparation, and was able to set these parameters manually, it would narrow the search for reads in the pair and improve the sensitivity of the alignment.

6.0 How it helps biologically

This differential gene expression analysis compared the expression levels of multiple genes from multiple samples of T. Brucei, from different stages of their lifecycle. This will help researchers to identify the molecular basis of phenotypic differences, enabling them to select gene-expression targets for further in-depth study. It will also advance our understanding of the stage-specific processes of the parasite, which could possibly reveal additional mechanisms behind the parasite's pathology.

Furthermore, this could ultimately lead to the innovation of a drug that causes the overexpression of genes that expressed during the Stumpy stage of the lifecycle, where it is non-proliferative. This knowledge will help us tackle the catastrophic "sleeping sickness" disease.

7.0 Assignment1 Shell script flowchart:


Script Commands requiring user input
Script commands running automatically
Raw data Masterpath
comments
Output directory
Creating and locating necessary directories
1.0 Creation of output directory
2.0 Specifying directory where ALL raw data is stored
3.0 Specifying name of sample detail file
raw data masterpath
raw data masterpath
Copies all fq.gz files to rawdata subdirectory within the output directory
Finds sample detail file from raw data masterpath and copies it into rawdata subdirectory within the output directory
$outputdirectory
$outputdirectory/rawdata
4.0 Inserts Lifecycle name taken from sample detail file, into fq.gz. filename
Sets lifecycle names as variables and inserts this into the name of appropriate fq.gz. files
FASTQC
5.0 Performing FASTQC analysis on all samples
Subdirectory fastqc_result made automatically, and fastqc results are stored here
$outputdirectory/fastqc_result
6.0 Outputting basic statistics of all samples that underwent FASTQC
Total sequences, Poor quality reads, sequence length, GC content, are extracted from FASTQC output and presented in two tables
7.0 Outputting modules that failed or warned
Failed Modules and Warned Modules are extracted from FASTQC output and presented in a table
8.0 Asking user if he wants to view in Firefox, the FASTQC analysis output for a specific sample
HTML files of FASTQC output for specified files are opened in Firefox, based on user input
9.0 Filtering of FASTQC output, so failed samples will not be aligned
Samples specified by the user to have failed the FASTQC analysis. Their fq.gz. files are found from within the rawdata subdirectory and moved to the failedfastqc.
$outputdirectory/failedfastqc
10.0 Copying reference genome from raw data path to output directory
raw data masterpath
Finds the reference genome file ending in fasta.gz, from raw data masterpath, and copies it to alignment subdirectory, and unzips it
$outputdirectory/alignment
BOWTIE2
11.0 Building of Bowtie2 index
Builds bowtie index with unzipped reference genome, naming the index based on user input
Finds all files ending in .fq.gz from rawdata subdirectory, and aligns them to reference genome. (FASTQC FAILED .fq.gz are no longer in rawdata subdirectory, therefore not aligned.
12.0 Aligning/Mapping read pairs to Reference genome index
$outputdirectory/rawdata
13.0 Copying bedfile to output directory
raw data masterpath
Finds bedfile ending in .bed from raw data masterpath, and copies it to the alignment subdirectory
BEDTOOLS
14.0 Creating BAM index
Finds all files ending in .bam, sorts them and renames them to sorted.bam. Then creates an index from them
15.0 Running bedtools on all BAM indexes
Performs gene counts on all sorted bam files, producing two .txt files, one for each lifecycle
16.0 Generating mean counts per gene for samples of each lifecycle
Locates count files for each lifecycle and calculates the statistical mean, outputting the mean gene count for each lifecycle in a seperate .txt file
$outputdirectory/counts
17.0 Merging of mean counts into a tab-delimited single plain .txt file.
Merges the seperate .txt files for mean gene counts for each lifecycle, into a single tab-delimited .txt file

Apologies for the small size, zoom shall fix that. This gives a full run down of what the shell script is doing, both the processes that are visible to the naked eye, and processes happening behind the scenes.