Sequencing Data QC and Preprocessing

Nick Bartelo

2/7/2021

Q1

1. Download more FASTQ files from the Gierlinski data set so that you have all the technical replicates for the first three WT and SNF2 samples (= 6x7 FASTQ files). Place each set of 7 technical replicates into one sensibly named folder respectively.

To begin solving this problem, we first create a new directory for this exercise. We cd to /home/nib4003/ANGSD_2021_hw and create a directory called "sequencing_data_qc_and_preprocessing". We then cd into this directory. Here, we copy the files we used in the previous exercise containing the necessary information to download all 42 fastq files for this exercise. To do this, we use the commands `cp /home/nib4003/ANGSD_2021_hw/sequencing_data/info_for_accession_numbers .` and `cp /home/nib4003/ANGSD_2021_hw/sequencing_data/filereport_read_run _PRJEB5348_tsv.txt .` which copy the files to the directory we are currently in. Using these files, we can find the run accession numbers corresponding to the samples. First, we use the command `mkdir gierlinski_data_wt_1` to make a folder to contain the first biological replicates. We keep this naming scheme for all folders, but of course replacing wt with snf2 for the three snf2 samples and 1, 2, or 3 based on the replicate. We then cd into each directory and run the command below where `egrep -w '1$'` is replaced by 1, 2, or 3 depending on the biological replicate being searched for, and 'WT' is changed to 'SNF2' for these three samples. An example for SNF2 is also shown below.

```
accession_numbers_wt=$(egrep 'WT' /home/nib4003/ANGSD_2021_hw/
sequencing_data_qc_and_preprocessing/info_for_accession_numbers
| egrep -w '1$' | awk '{print $1}')
```

```
accession_numbers_snf2=$(egrep 'SNF2' /home/nib4003/ANGSD_2021_hw/
sequencing_data_qc_and_preprocessing/info_for_accession_numbers
| egrep -w '1$' | awk '{print $1}')
```

We now look through the accession numbers for all respective samples and run the following command, as we had done in the previous homework, to download all files into current folder we are in. Therefore, we need to cd into each folder where we want the 7 files per replicate to be placed before running the following command. An example for WT and SNF2 are shown below. These result in all 42 files being downloaded, 7 per directory.

```
for i in $accession_numbers_wt;
do link_wt=$(egrep "$i" /home/nib4003/ANGSD_2021_hw/sequencing_data_qc_and_preprocessing
/filereport_read_run_PRJEB5348_tsv.txt | cut -f 7); wget ftp://${link_wt}; done
```

```
for i in $accession_numbers_snf2;
do link_snf2=$(egrep "$i" /home/nib4003/ANGSD_2021_hw/sequencing_data_qc_and_preprocessing
/filereport_read_run_PRJEB5348_tsv.txt | cut -f 7); wget ftp://${link_snf2}; done
```

Q2

2. Write a for-loop that will run FastQC on all (6x7) of the FASTQ files that you have downloaded from the Gierlinski dataset. Select one sample for which you write an additional for-loop that will:
3. run TrimGalore
4. run FastQC on the trimmed datasets

To begin answering this question, we first load all tools that we need using the commands `spack load fastqc` and `spack load -r trimgalore` (needs -r to create the environment). We then use the following code to run fastqc on all 42 files at once. ‘

```
for file in g*/E*; do fastqc $file --extract; done
```

This for loop involves first specifying which directories we need to loop over and the files we want to run fastqc on using `for file in g*/E*` which means to run fastqc on all files beginning with ‘E’ in all folders beginning with ‘g’. Since we have named our folders in such a manner, this simple for loop gives us the desired result. The `--extract` results in the zipped output file being uncompressed in the same directory after it has been created.

Next, we want to run TrimGalore and FastQC on one of the samples. We first create a directory where all the output files will be stored called trimmed_files. We run the processes for WT1 using the following code which runs both TrimGalore and FastQC for all specified files in the folder.

```
for file in gierlinski_data_wt_1/*.gz;
do trim_galore $file --illumina --fastqc_args "--extract" -o /home/nib4003/ANGSD_2021_hw/
sequencing_data_qc_and_preprocessing/gierlinski_data_wt_1/trimmed_files/; done
```

The `--illumina` option is used because it is specified that Illumina sequencing was used on the Project: PRJEB5348 website. This option specifies the adapter sequence to be trimmed is the first 13bp of the Illumina universal adapter ‘AGATCGGAAGAGC’ instead of the default auto-detection of adapter sequence. The `--fastqc_args "--extract"` command runs fastqc on the trimmed file and creates a summary of the results as we had when running fastqc without the trimming. The `-o` option specifies where to store all the files that are created, which we specified as our new trimmed files folder.

Q3

3. Describe one detail of the QC results that changes after TrimGalore and one result that stays the same and explain why.

To compare the results, we need to look at the html files produced. To do this, we need to use WinSCP and transfer the html files from the SCU to our computer. We therefore create a folder using the command `mkdir transfer_files` in the ANGSD_2021_hw directory. To copy all html files from trimgalore for transfer, we cd into the new directory and execute the command

```
cp /home/nib4003/ANGSD_2021_hw/sequencing_data_qc_and_preprocessing/
gierlinski_data_wt_1/trimmed_files/*.html .
```

We also do the same for all html files from the fastqc files without the trimming for WT1 using the command above without the trimmed files directory added resulting in all html files ready for transfer using WinSCP. We then login to WinSCP as discussed in the previous homework and drag and drop our tranfer files folder to the computer. We can then double click on each html file to open their output on the computer.

As we compare the output files, we notice that one detail of the QC results that changes after TrimGalore is that Sequence Length Distribution part of the FastQC Report has a warning for every post-trimmed file, whereas none of the files before trimming have a warning. This is due to the removal of reads due to the trimming process and should not be of great concern. Also, some of the Sequence Duplication Levels that have warnings before trimming no longer have warnings after trimming. One result that stays the same before and after trimming is Per Tile Sequence Quality. This is due to the fact that this measures the quality of the flowcell itself, and this does not change for the trimmed or untrimmed reads.

Q4

4. Combine the initial FastQC results for all 6x7 FASTQ files into one document using MultiQC. You can load the tool using spack load -r py-multiqc. Export one image of either of the results where the SNF2 samples are highlighted in a different color than the WT samples and add it to this report.

We now need to import MultiQC using the command `spack load -r py-multiqc`. After navigating to `/home/nib4003/ANGSD_2021_hw/sequencing_data_qc_and_preprocessing/` we can now run the code

```
multiqc g*/
```

which searches all the directories for analysis logs and compiles a HTML report. The output of multiqc is a folder entitled "multiqc_data" and an HTML entitled "multiqc_report.html". However, we have found that since our trimmed files are in a child directory of the directory containing the WT1 files, the trimmed folders are being used by multiqc as well as the original fastqc files. Therefore, we have to move this folder by navigating to /home/nib4003/ANGSD_2021_hw/sequencing_data_qc_and_preprocessing and executing the command `mv /home/nib4003/ANGSD_2021_hw/sequencing_data_qc_and_preprocessing/gierlinski_data_wt_1/trimmed_files .` which moves the folder to the current directory. Once moving the folder, we can execute the command again and get the correct files in the multiqc analysis. We then cd into the transfer files folder and execute the command `cp /home/nib4003/ANGSD_2021_hw/sequencing_data_qc_and_preprocessing/multiqc_report.html .` so that we can transfer the HTML file to our computer using WinSCP. Once the HTML is opened on the computer, we need to make some changes to the file names so that we can distinguish the WT from the SNF2 since these labels are not included on the files after multiqc analyzes them. To do this, we click on the multiQC Toolbox on the righthand side and click on the A which stands for Rename. We then use the fact that we know from which samples the accession numbers came from, which is how multiqc labeled all files, and therefore we label all the WT files as 'accession_number'_wt to distinguish them from the SNR2 files. Now we can highlight the different samples based on their naming schemes. To do this, we highlight all WT files with red by clicking the tack on the toolbox that stands for highlight. We then click the box that says "Regex mode" and type in t$ since this represents a "t" at the end of the sentence, corresponding to our renamed samples for WT. We then press enter and click apply for these changes to occur. Next, we label the SNR2 files blue by typing [0-9]$ which means that the last character of the name of the file is a number, which represents all these files. We then go to the export tab on the toolbox and only select the fastqc_sequence_counts_plot and fastqc_per_sequence_quality_scores_plot for export. In this tab there is the option to download many of the different plots shown in the HTML. We attach the fastqc_per_sequence_quality_scores_plot to this homework.

Q5

5. Based on the QC, would you be justified in combining any of the FASTQ files given that they are technical replicates?

We would be justified in combining any of the FASTQ files given that they are technical replicates because all files have very similar GC content, adapter content, and per sequence quality score. Therefore, since the technical replicates score about the same on all tests, we would not be introducing an extra contamination to the sample by merging the replicates.

Q6

6. Even if the answer to the previous question is "no", what command(s) would you use to combine the several FASTQ files into one?

The command used to combine several FASTQ files into one are

```
zcat fastq1.gz fastq2.gz fastq3.gz fastqn.gz > fastq_combined.gz
```

Here we are just concatenating the fastq files together into one final file.

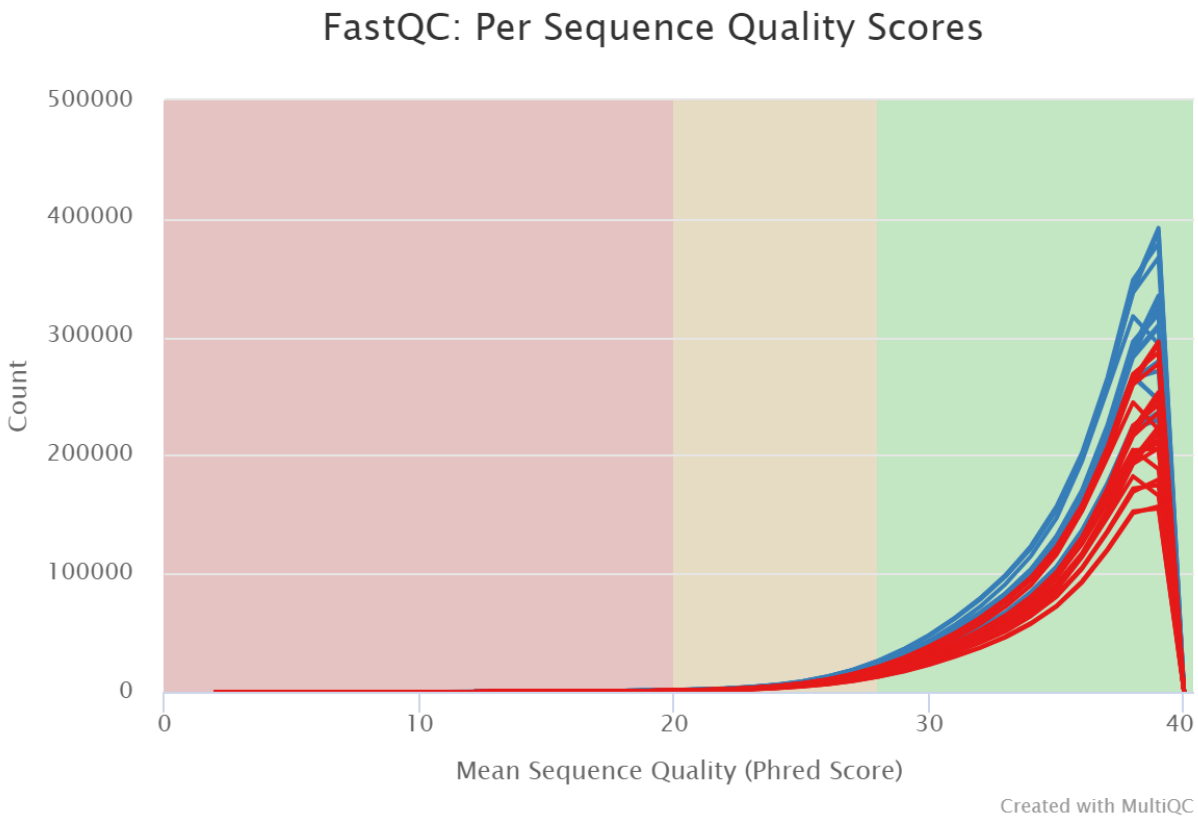

Figure 1: Plot of per sequence quality scores with WT files in red and SNF2 files in blue.

Q7

7. Bonus point: If you had to determine the version of the Sanger quality score encoding used in a given FASTQ file without the help of FastQC, what would you do?

To determine the version of the Sanger quality score encoding used in a given FASTQ file without the help of FastQC, I would look at the range of the ASCII characters used and the range of the quality scores. For example, for identifying the quality score used between Sanger standard, Solexa/early Illumina, and Illumina 3+, we could first look at the range of the quality scores. Since Solexa/early Illumina has scores from -5 to 62 and the other quality scores do not contain negative numbers, we would know that the version is Solexa/early Illumina by searching for any negative numbers in the quality scores. To differentiate between Sanger standard and Illumina 1.3+, the range of ASCII letters for the Sanger standard version is 33-126 and the Illumina 1.3+ version is 64-126. Therefore, we can search the quality scores for ASCII characters between 33 and 63, which will tell us which version is being used.