Day 4 Worksheet – Part 1.2 Trimmomatic

Author: Jessica Westfall: jessica.westfall@colorado.edu

FASTQC: https://www.bioinformatics.babraham.ac.uk/projects/fastqc/
TRIMMOMATIC: http://www.usadellab.org/cms/?page=trimmomatic

Introduction: Now that we have evaluated our sequence library initially to determine if the libraries are worth moving forth with, we will do some "cleaning up" by trimming away unwanted sequences such as adapters sequences. This step is necessary for improved alignment and mapping to the reference genome downstream. Once trimming is completed we will reevaluate our sequence library again with FastQC for quality to decide if we will move forth with mapping.

! Note: The directory and username used in the screenshot will be for my working directory and username and will be different than yours.

Make working directories

In the previous worksheet, we make working directories for running fastQC. Repeat the same process, but this time we will make a directory for trimmomatic.

1. Use command **pwd** to determine what directory you are in and if necessary, **cd** to the directory that you want to place your new trimmomatic directory in.
2. Make a new directory using the **mkdir** command. Use command **ls -lsh** to confirm the folders are present.

```
[bash-4.2$ cd /scratch/Users/jewe1055/sread/
[bash-4.2$ mkdir trimmomatic
[bash-4.2$ ls -lsh
 total 1.5K
 512 drwxrwxr-x 2 jewe1055 dowelldegrp 6 Jul 15 23:21 eofiles
 512 drwxrwxr-x 2 jewe1055 dowelldegrp 2 Jul 15 23:21 fastqc
 512 drwxrwxr-x 2 jewe1055 dowelldegrp 2 Jul 16 00:00 scripts
   0 drwxrwxr-x 2 jewe1055 dowelldegrp 0 Jul 16 00:16 trimmomatic
```

Trimmomatic

3. Copy (**rsync** or **cp**) the **d4_trim_qc.sbatch** script into your script directory that you made in the previous exercise. Below I am copying from the workshop directory to my directory. I then use **ls -lsh** to confirm the file is present in the directory. You can **ls** with a absolute path as well as relative path.

To copy the script, the command syntax is **rsync <input><output>**

```
[-bash-4.2$ rsync /scratch/Shares/dowell/sread/scripts/day4/d4_trim_qc.sbatch /scratch/Users/jewe1
055/sread/scripts/
[-bash-4.2$ ls -lsh /scratch/Users/jewe1055/sread/scripts/
total 5.5K
2.5K -rwx------ 1 jewe1055 dowelldegrp 2.1K Jul 15 23:21 d4_fastqc.sbatch
3.0K -rw-r--r-- 1 jewe1055 dowelldegrp 2.9K Jul 16 07:44 d4_trim_qc.sbatch
```

4. Edit the sbatch script by using **vim <sbatch>** to open a text editor on your sbatch script. Type **i** to toggle into edit/insert mode. Similar to the previous exercise you will need to change the job name, user email, and the standard output and error log directories. Change the **–job-name=<JOB_NAME>** to a name related to the job you will be running, for example 'trim_qc'. Additionally you will want to change the **–mail-user=<YOUR_EMAIL>** to your email, as well as the path to your eofiles directory for the standard output (**--output)** and error log (**--error)**. The **%x** will be replace by your **-job-name** and the **%j** will be replace by the job id that will be assigned by the job manager when you run your sbatch script.

```
in/bash
TCH --job-name=<JOB_NAME>                    # Job name
TCH --mail-type=ALL                          # Mail events (NONE, BEGIN, END, FAIL, AL
TCH --mail-user=<YOUR_EMAIL>                 # Where to send mail
TCH --nodes=1                                # Number of nodes requested
TCH --ntasks=8                               # Number of CPUs (processor cores/tasks)
TCH --mem=8gb                                # Memory limit
TCH --time=01:30:00                          # Time limit hrs:min:sec
TCH --partition=short                        # Partition/queue requested on server
TCH --output=/scratch/Users/<USERNAME>/eofiles/%x.%j.out
TCH --error=/scratch/Users/<USERNAME>/eofiles/%x.%j.err
```

For this script, I will be change my CPU and nodes for trimmomatic which can use multiple processors per input file. I am going to request 1 node, 8 tasks, 8gb of memory and 90 minutes of wall time.

5. Assigning path variables will make your scripts easier to read. In addition, this makes it easier to reference to a given path and utilize it in your scripts. For the **INDIR=**change the path to where the data files directories are located and specifically the fastq data. For the **OUTDIR=**, point to the appropriate output file directories for our fastQC and trimmed fastq files. I also use the command **mkdir -p** just in case for my output directories.

```
################ ASSIGNS PATH VARIABLES ################################################
## the fastq files will be used as input to fastqc and trimmomatic
## trimmed reads will then be passed on to the mapping step

INDIR=/scratch/Shares/dowell/sread/data_files/day4
FASTQ=${INDIR}/fastq

OUTDIR=/scratch/Users/<USERNAME>
FASTQC=${OUTDIR}/fastqc
TRIM=${OUTDIR}/trimmomatic

FILENAME=chr21Eric_repA

mkdir -p ${OUTDIR}
mkdir -p ${FASTQC}
mkdir -p ${TRIM}
```

6. Load the require modules for running this pipeline. We will be using fastQC and the trimming program trimmomatic. Similar to fastqc, if you are not sure which version of the program is available on the cluster you can use the command **module spider <string>** to find the available versions.

```
-bash-4.2$ module spider trimmomatic

----------------------------------------------------------------------------
  trimmomatic: trimmomatic/0.36
----------------------------------------------------------------------------
    Description:
      No Description Given

    This module can be loaded directly: module load trimmomatic/0.36
```

Now I can add the appropriate versions for the modules I want to load in the pipeline.

```
###############################################################################################
################ LOAD REQUIRED MODULES ########################################################
module load fastqc/0.11.8
module load trimmomatic/0.36
```

7. For the meat of the script, we will be running 3 steps in the pipeline. (1) To run fastQC on the sample, (2) trim the fastQC and (3) reevaluate the quality of the trimmed fastq with fastQC.

```
###############################################################################################
################## RUN PIPELINE ###############################################################
##1: Run fastqc on the samples (here run on example file ${FILENAME}.RNA.end1.fastq)
fastqc ${FASTQ}/${FILENAME}.RNA.end1.fastq -o ${FASTQC}
fastqc ${FASTQ}/${FILENAME}.RNA.end2.fastq -o ${FASTQC}

##2: Trim FASTQ Files


java -jar /opt/trimmomatic/0.36/trimmomatic-0.36.jar PE \
 -threads 8 \
 -phred33 \
 -trimlog ${TRIM}/trimlog \
 ${FASTQ}/${FILENAME}.RNA.end1.fastq ${FASTQ}/${FILENAME}.RNA.end2.fastq \
 ${TRIM}/${FILENAME}.RNA.end1.trimmed.fastq ${TRIM}/${FILENAME}.RNA.end1.unpaired.fastq \
 ${TRIM}/${FILENAME}.RNA.end2.trimmed.fastq ${TRIM}/${FILENAME}.RNA.end2.unpaired.fastq \
 ILLUMINACLIP:/opt/trimmomatic/0.36/adapters/TruSeq3-PE.fa:2:30:10 \
 CROP:20

##3: Check Post-Trimming QC stats
fastqc ${TRIM}/*.trimmed.fastq -o ${FASTQC}

echo Job finished at `date +"%T %a %d %b %Y"`
```

In this script we are running paired end reads. Trimmomatic can be used on both single-end or paired-end reads. When setting your parameters use the appropriate adapters. Below are the syntaxes needed to run trimmomatic

Illuminaclip parameter (see below for quick reference to trimming)
ILLUMINACLIP:<path_adapters_fasta>:<seed_mismatches>:
<palindrome_clip_threshold>:<simple_clip_threshold> LEADING:<quality>

TRAILING:<quality> SLIDINGWINDOW:<window_size>:<required_quality> MINLEN:<length>

For single-end reads

java jar /opt/trimmomatic/0.36/trimmomatic-0.36.jar SE [-threads <n>] [-phred33 | -phred64] [-trimlog <output_trimlog>] <input_file> <output_file> ILLUMINACLIP

For pair-end reads

java jar /opt/trimmomatic/0.36/trimmomatic-0.36.jar PE [-threads <n>] [-phred33 | -phred64] [-trimlog <output_trimlog>] <input_file1> <input_file2> <output_fileP1> <output_fileU1> <output_fileP2> <output_fileU2> ILLUMINACLIP

Recall that the '\' at the end is used to break the code up for clarity purpose. We can write this syntax as a single line but it is harder to read. '\' does not change color as you see above, you may have an extra space after the '\'. Remove that space or your code will not run properly.

8. Save your sbatch script. Press **esc** to exit out of edit mode, then type **:wq**. This will write/save (w) and quit (q) the script.

9. Let's run script. Submit the job to the job manager SLURM using the command **sbatch <sbatch_file>**. The job manager will assign a job id to your run.

12. This pipeline has more tasks than the previous worksheet, so you will want to check the status of your job using the command **squeue -u <username>** to see if the job is running (R) or completed (C). If there are any errors, often time these are just typos in your scripts, you will want to access your error log to make necessary corrections. I will **ls -lahtr /path/to/eofiles** to get the name of the error log for the job id so that I can view it using **more**, **less**, or **cat**. I use **-tr** with the **ls** command to get order my files based on time so I can quickly find the latest error log.

13. Check the error log to find information about the fastqc and trimming job.

```
Approx 95% complete for chr21Eric_repA.RNA.end2.fastq
TrimmomaticPE: Started with arguments:
 -threads 8 -phred33 -trimlog /scratch/Users/jewe1055/sread//trimmomatic/trimlog /scratch/Shares/
dowell/sread/data_files/day4/fastq/chr21Eric_repA.RNA.end1.fastq /scratch/Shares/dowell/sread/dat
a_files/day4/fastq/chr21Eric_repA.RNA.end2.fastq /scratch/Users/jewe1055/sread//trimmomatic/chr21
Eric_repA.RNA.end1.trimmed.fastq /scratch/Users/jewe1055/sread//trimmomatic/chr21Eric_repA.RNA.en
d1.unpaired.fastq /scratch/Users/jewe1055/sread//trimmomatic/chr21Eric_repA.RNA.end2.trimmed.fast
q /scratch/Users/jewe1055/sread//trimmomatic/chr21Eric_repA.RNA.end2.unpaired.fastq ILLUMINACLIP:
/opt/trimmomatic/0.36/adapters/TruSeq3-PE.fa:2:30:10 CROP:20
Using PrefixPair: 'TACACTCTTTCCCTACACGACGCTCTTCCGATCT' and 'GTGACTGGAGTTCAGACGTGTGCTCTTCCGATCT'
ILLUMINACLIP: Using 1 prefix pairs, 0 forward/reverse sequences, 0 forward only sequences, 0 reve
rse only sequences
Input Read Pairs: 296415 Both Surviving: 296399 (99.99%) Forward Only Surviving: 16 (0.01%) Rever
se Only Surviving: 0 (0.00%) Dropped: 0 (0.00%)
TrimmomaticPE: Completed successfully
Started analysis of chr21Eric_repA.RNA.end1.trimmed.fastq
Approx 5% complete for chr21Eric_repA.RNA.end1.trimmed.fastq
```

Pre- and post-trim fastQC

Pre-trimming

FastQC Report

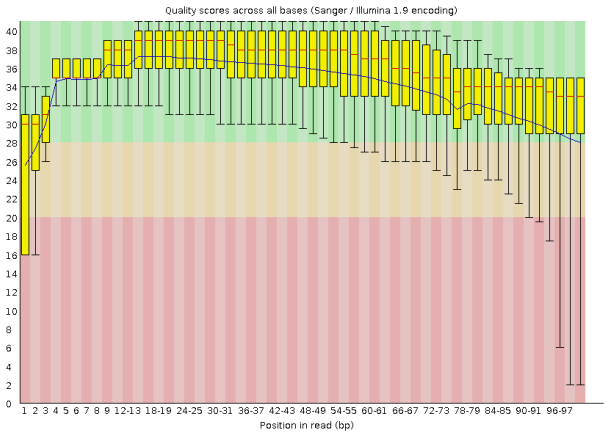
Summary

- Basic Statistics
- Per base sequence quality
- Per tile sequence quality
- Per sequence quality scores
- Per base sequence content
- Per sequence GC content
- Per base N content
- Sequence Length Distribution
- Sequence Duplication Levels
- Overrepresented sequences
- Adapter Content

Basic Statistics

Measure	Value
Filename	chr21Eric_repA.RNA.end1.fastq
File type	Conventional base calls
Encoding	Sanger / Illumina 1.9
Total Sequences	296415
Sequences flagged as poor quality	0
Sequence length	100
%GC	42

Per base sequence quality



Post-trimming

FastQC Report

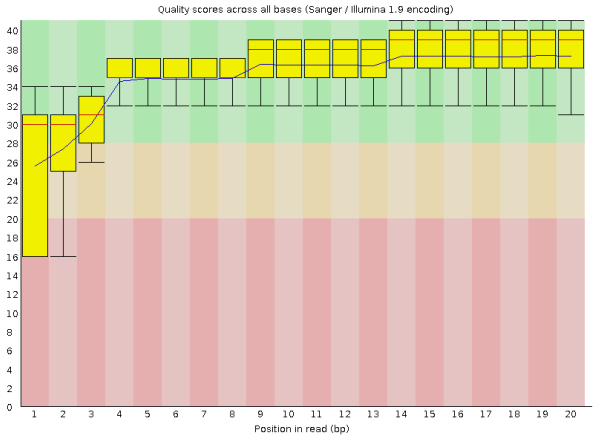
Summary

- Basic Statistics
- Per base sequence quality
- Per tile sequence quality
- Per sequence quality scores
- Per base sequence content
- Per sequence GC content
- Per base N content
- Sequence Length Distribution
- Sequence Duplication Levels
- Overrepresented sequences
- Adapter Content

Basic Statistics

Measure	Value
Filename	chr21Eric_repA.RNA.end1.trimmed.fastq
File type	Conventional base calls
Encoding	Sanger / Illumina 1.9
Total Sequences	296399
Sequences flagged as poor quality	0
Sequence length	20
%GC	43

Per base sequence quality



Reference to Illumina parameters:
http://www.usadellab.org/cms/uploads/supplementary/Trimmomatic/TrimmomaticManual_V0.32.pdf

Implemented trimming steps (Quick reference)

Trimmomatic performs a variety of useful trimming tasks for illumina paired-end and single ended data. The selection of trimming steps and their associated parameters are supplied on the command line.

The current trimming steps are:

- ILLUMINACLIP: Cut adapter and other illumina-specific sequences from the read.
- SLIDINGWINDOW: Performs a sliding window trimming approach. It starts scanning at the 5' end and clips the read once the average quality within the window falls below a threshold.
- MAXINFO: An adaptive quality trimmer which balances read length and error rate to maximise the value of each read
- LEADING: Cut bases off the start of a read, if below a threshold quality
- TRAILING: Cut bases off the end of a read, if below a threshold quality
- CROP: Cut the read to a specified length by removing bases from the end
- HEADCROP: Cut the specified number of bases from the start of the read
- MINLEN: Drop the read if it is below a specified length
- AVGQUAL: Drop the read if the average quality is below the specified level
- TOPHRED33: Convert quality scores to Phred-33
- TOPHRED64: Convert quality scores to Phred-64