Shaleigh Smith

Next Generation Sequencing Informatics

Assigned Coursework #2

The SRR1523657 dataset is a transcriptomic dataset and was sequenced on Illumina HiSeq 2500 using an RNA-Seq, paired-end strategy. The data came from primary human foreskin fibroblasts (HFF) that had been infected with Strain 17 HSV-1. The sequencing included newly transcribed RNA that was labelled at one-hour intervals after infection up to eight hours. The SRX747060 dataset is also a transcriptomic dataset but was sequenced on Illumina HiSeq 2000 using a miRNA-Seq strategy that produced paired-end reads. The data originated from Kaposi's Sarcoma-derived human cells that were either infected with Kaposi's Sarcoma-Associate Herpesvirus (KSHV) or negative for infection. The resulting microRNA sequence data consisted of both human and viral samples. The ERR218285 dataset was sequenced by a 454 GS FLX Titanium instrument using the amplicon method and PCR selection, producing single-end reads. The dataset is metagenomic, consisting of calcified dental plaque sequence data from ancient human teeth to better understand how the oral microbial community changed when society shifted from hunter-gatherer to farming. While the initial fastQC report illustrated a relatively high average quality for each dataset, the per base quality figure showed that the two reads from SRR1523657 and SRX747060 as well as the third read from ERR218285 need to be trimmed for base quality. The first and second reads from ERR218285 do not need to be trimmed because they only consist of short group tags. Additionally, the fastQC report indicated that there are adapters included in all of the reads for SRR1523657 and SRX747060 which need to be removed.

Generally speaking, both trimming programs had the same performance, but that performance was highly affected by their varying parameters. This was assessed by comparing the fastQC Report for each dataset read before and after trimming for both programs (Table 1). Trimmomatic and trim galore were both able to remove low quality bases for all three datasets; however, trimmomatic was more stringent when trimming the low-quality bases. This was illustrated in the per base quality fastQC figure for the ERR218285 and SRX747060 datasets after trimming. Both programs were set to a phred score of 20 and after both datasets were processed by trim galore there were still error bars reaching below this threshold; in contrast, when it was processed by trimmomatic the error bars were all above the threshold (20). Furthermore, for each dataset trimmomatic consistency had a higher minimum per base quality than trim galore, thus illustrating the stricter trimming nature of trimmomatic (Table 1). It should be noted that this could easily be an example of over-trimming and could result in the loss of valuable read information.

In respect to trimming adapters, each method performed similarly. The only con was that trimmomatic needs to be given a list of adapter sequences to run against the reads where trim galore automatically detects these sequences. As a result, trimmomatic did a poorer job of detecting the small RNA adapters in the SRX747060 dataset and was only able to remove a portion of the adapters. This was only an issue with the SRX747060 reads: trimmomatic performs comparably to trim galore for adapter removal if given the correct adapters (Table 1).

The programs differed minimally when comparing the output report statistics. Trimmomatic and trim galore filtered out almost the same number of reads for ERR218285, SRR1523657, and SRX747060 (Table 1). The sequence length after trimming was also similar with ERR218285 and SRR1523657 but differed with SRX747060 because trim galore was able to remove the adapter more effectively than trimmomatic, making the sequence length shorter. An added plus of trim galore is that it gives the user a trimming report which includes a comprehensive summary of statistics for the run. Trimmomatic gives some simple statistics in the output log, but nothing as extensive.

Table 1. Comparison of Trimmomatic and Trim Galore using fastQC Report

	SRR1523657		SRX747060		ERR218285	
	Trim Galore	Trimmomatic	Trim Galore	Trimmomatic	Trim Galore	Trimmomatic
Read Type & Number	Paired 1	Paired 1	Paired 1	Paired 1	Single 3	Single 3
Average Read Quality	37	37	38	39	38	38
Per Base Quality Min (Error Bar)	25	28	2	23	19	25
Sequences After Filtering	33554856	33644736	10255988	11689081	1771	1765
Trimming Summary	Yes	No	Yes	No	Yes	No
Removed all Adapters	Yes	Yes	Yes	No	Yes	Yes

Base trimming had the biggest impact on SRR1523657 and ERR218285. This was shown in the per base sequence fastQC quality figure: for both datasets the tail end of the read dramatically increased in quality due to the base trimming. In contrast, SRX747060 was most affected by adapter trimming because its tail-end adapter greatly decreased base quality as shown in the per base sequence quality figure. Thus, removing the 3’ and 5’ small RNA adapters increased overall quality per base more noticeably than trimming by base quality.

See below for scripts: SRA Download, FASTQC, Trim Galore, and Trimmomatic

### SRA Download Script

```
#!/bin/bash
#SBATCH --job-name=sra_job # Job name
#SBATCH --mail-type=END,FAIL # Mail events (NONE, BEGIN, END, FAIL, ALL)
#SBATCH --mail-user=first.last@nyumc.org # Where to send mail
#SBATCH --ntasks=1 # Run on a single CPU
#SBATCH --mem=4gb # Job memory request
#SBATCH --time=06:00:00 # Time limit hrs:min:sec
#SBATCH --output=sra_%j.log # Standard output and error log
#SBATCH --partition=cpu_short # Which partition you want to run it on

# This script is used to download fastq data directly from SRA ncbi, split the data into two fastq files and compress the results files to a gzip (fastq.gz)

module load sratoolkit/2.9.1

fastq-dump --split-files SRR7992453 --gzip -O /gpfs/scratch/sas1531/
fastq-dump --split-files SRX747060 --gzip -O /gpfs/scratch/sas1531/
fastq-dump --split-files ERR218285 --gzip -O /gpfs/scratch/sas1531/

rm -r ~/ncbi # fastq-dump creates a temp dir but doesn't remove it... plays merry havoc with home storage space

### Submit job using sbatch
# sbatch sra_download_script.sh

### View queue
# squeue -u sas1531
```

### FASTQC Script

```
#!/bin/bash
#SBATCH --job-name=fastqc_job # Job name
#SBATCH --mail-type=END,FAIL # Mail events (NONE, BEGIN, END, FAIL, ALL)
#SBATCH --mail-user=first.last@nyumc.org # Where to send mail
#SBATCH --ntasks=1 # Run on a single CPU
#SBATCH --mem=4gb # Job memory request
#SBATCH --time=02:00:00 # Time limit hrs:min:sec
#SBATCH --output=fastqc_%j.log # Standard output and error log
#SBATCH --p=cpu_short # Which partition you want to run it on

# This script is used to generate an FASTQC report from a fastqc.gz files

module load fastqc

fastqc -o /gpfs/scratch/sas1531/ /gpfs/scratch/sas1531/SRR1523657_1.fastq.gz
/gpfs/scratch/sas1531/SRR1523657_2.fastq.gz /gpfs/scratch/sas1531/SRX747060_1.fastq.gz
/gpfs/scratch/sas1531/SRX747060_2.fastq.gz /gpfs/scratch/sas1531/ERR218285_1.fastq.gz
/gpfs/scratch/sas1531/ERR218285_2.fastq.gz /gpfs/scratch/sas1531/ERR218285_3.fastq.gz

### Submit job using sbatch
# sbatch fastqc_report_script.sh

### View queue
# squeue -u sas1531

### Export html files using filezilla
```

### Trim Galore Script

```
#!/bin/bash
#SBATCH --job-name=trimgalore_test # Job name
#SBATCH --mail-type=END,FAIL # Mail events (NONE, BEGIN, END, FAIL, ALL)
#SBATCH --mail-user=Shaleigh.Smith@nyumc.org # Where to send mail
#SBATCH --ntasks=1 # Run on a single CPU
#SBATCH --mem=4gb # Job memory request
#SBATCH --time=6:00:00 # Time limit hrs:min:sec
#SBATCH --output=trim_galore_%j.log # Standard output and error log
#SBATCH --partition=cpu_short # Which partition you want to run it on

# This script is used to trim sequence reads using trimgalore

module load trimgalore/0.5.0
module load python/cpu/2.7.15-ES ### CutAdapt is hidden in here
module load fastqc

# Trim sequences with PHRED quality score above 30 and remove adapters (trim galore will automatically detect
these), run FASTQC on output:

trim_galore --q 20 --phred33 -o /gpfs/scratch/sas1531/ngs2_coursework/ --fastqc
/gpfs/scratch/sas1531/ngs2_coursework/ERR218285_3.fastq.gz

trim_galore --q 20 --phred33 --paired -o /gpfs/scratch/sas1531/ngs2_coursework/ --fastqc
/gpfs/scratch/sas1531/ngs2_coursework/SRR1523657_1.fastq.gz
/gpfs/scratch/sas1531/ngs2_coursework/SRR1523657_2.fastq.gz

trim_galore --q 20 --phred33 --small_rna --paired -o /gpfs/scratch/sas1531/ngs2_coursework/ --fastqc
/gpfs/scratch/sas1531/ngs2_coursework/SRX747060_1.fastq.gz
/gpfs/scratch/sas1531/ngs2_coursework/SRX747060_2.fastq.gz

# Don't need to trim these:
# ERR218285_1.fastq.gz
# ERR218285_2.fastq.gz

### For command help
# trim_galore --help

### Submit job using sbatch
# sbatch trimgalore_script.sh

### View queue
# squeue -u sas1531

### Export html files using filezilla
```

### Trimmomatic Script

```
#!/bin/bash
#SBATCH --job-name=trimmomatic_job_test # Job name
#SBATCH --mail-type=END,FAIL # Mail events (NONE, BEGIN, END, FAIL, ALL)
#SBATCH --mail-user=Shaleigh.Smith@nyumc.org # Where to send mail
#SBATCH --ntasks=1 # Run on a single CPU
#SBATCH --mem=4gb # Job memory request
#SBATCH --time=06:00:00 # Time limit hrs:min:sec
#SBATCH --output=trimmomatic_%j.log # Standard output and error log
#SBATCH --partition=cpu_short # Which partition you want to run it on

# This script is used to trim sequence reads using trimmomatic

module load trimmomatic/0.36
module load fastqc

#Single End:
java -jar /gpfs/share/apps/trimmomatic/0.36/trimmomatic-0.36.jar SE -phred33
/gpfs/scratch/sas1531/ngs2_coursework/ERR218285_3.fastq.gz
/gpfs/scratch/sas1531/ngs2_coursework/ERR218285_3_trimmomatic_out.fastq.gz
ILLUMINACLIP:/gpfs/scratch/sas1531/ngs2_coursework/adapter.fasta:2:30:10 SLIDINGWINDOW:4:20

#Paired End:
java -jar /gpfs/share/apps/trimmomatic/0.36/trimmomatic-0.36.jar PE -phred33
/gpfs/scratch/sas1531/ngs2_coursework/SRR1523657_1.fastq.gz
/gpfs/scratch/sas1531/ngs2_coursework/SRR1523657_2.fastq.gz
/gpfs/scratch/sas1531/ngs2_coursework/SRR1523657_1_trimmomatic_paired_out.fastq.gz
/gpfs/scratch/sas1531/ngs2_coursework/SRR1523657_1_trimmomatic_unpaired_out.fastq.gz
/gpfs/scratch/sas1531/ngs2_coursework/SRR1523657_2_trimmomatic_paired_out.fastq.gz
/gpfs/scratch/sas1531/ngs2_coursework/SRR1523657_2_trimmomatic_unpaired_out.fastq.gz
ILLUMINACLIP:/gpfs/scratch/sas1531/ngs2_coursework/adapter.fasta:2:30:10 SLIDINGWINDOW:4:20

java -jar /gpfs/share/apps/trimmomatic/0.36/trimmomatic-0.36.jar PE -phred33
/gpfs/scratch/sas1531/ngs2_coursework/SRX747060_1.fastq.gz
/gpfs/scratch/sas1531/ngs2_coursework/SRX747060_2.fastq.gz
/gpfs/scratch/sas1531/ngs2_coursework/SRX747060_1_trimmomatic_paired_out.fastq.gz
/gpfs/scratch/sas1531/ngs2_coursework/SRX747060_1_trimmomatic_unpaired_out.fastq.gz
/gpfs/scratch/sas1531/ngs2_coursework/SRX747060_2_trimmomatic_paired_out.fastq.gz
/gpfs/scratch/sas1531/ngs2_coursework/SRX747060_2_trimmomatic_unpaired_out.fastq.gz
ILLUMINACLIP:/gpfs/scratch/sas1531/ngs2_coursework/adapter.fasta:2:30:10 SLIDINGWINDOW:4:20

# Don't need to trim these:
# ERR218285_1.fastq.gz ERR218285_2.fastq.gz

### Put each output file through fastqc to generate report
fastqc -o /gpfs/scratch/sas1531/ngs2_coursework /gpfs/scratch/sas1531/ngs2_coursework/*trimmomatic*.gz

### Manual: http://www.usadellab.org/cms/uploads/supplementary/Trimmomatic/TrimmomaticManual_V0.32.pdf

### Submit job using sbatch
# sbatch fastqc_report_script.sh

### View queue
# squeue -u sas1531

### Export html files using filezilla
```