DIFFERENTIAL GENE EXPRESSION ANALYSIS

Module 1: Sequence Read Metrics and QC

COPY DATASET, OBTAIN METRICS, AND RUN QUALITY CHECK

You will be copying fastq files into your workspace and running metrics on them to obtain the read lengths and total number of reads per sample. You will then do a quality check on the reads using a program called fastqc. Once you have gone through the fastqc results, you will download the human genome and annotation files from relevant databases and run alignments using a program called HISAT2. Once alignments are generated, you will obtain alignment metrics using some Bash commands.

COPYING DATA

```
cd /home/$USER/

mkdir DGE_Virtual/

cd DGE_Virtual/

mkdir raw_reads/

cd raw_reads/

pwd
/home/$USER/DGE_Virtual/raw_reads/

cp /home/elavelle/DGE_Virtual/raw_reads/*.gz ./
```

RUNNING SEQUENCE DATA METRICS

Lets understand the AWK "NR" VARIABLE and the AWK MODULO ARITHMETIC since we will using both to generate sequence read metrics.

AWK "NR" VARIABLE: There are several AWK built-in variables. One of them is "NR" (Number of Rows/

Records). It contains line number and can be used to determine the total number of lines in a file. The following example illustrates one use of this built-in variable.

```
cd /home/<username>/DGE_Virtual_Jan2021/raw_reads/

zcat 2S1Flag-p5-2.fq.gz | awk '{print NR}' | head
1
2
3
4
5
6
7
8
9
10

# In the above example, "awk '{print NR}' | head" prints the line number of the first 10 lines

zcat 2S1Flag-p5-2.fq.gz | awk '{print NR}' | tail
47265979
47265980
47265981
47265982
47265983
47265984
47265985
47265986
47265987
47265988

# In the above example, the "awk '{print NR}' | tail" prints the line number of the last 10 lines
# The return indicates that "2S1Flag-p5-2.fq.gz" file has 47,265,988 lines
```

AWK MODULO ARITHMETIC: The modulo operator, "%", returns the remainder of division. The expression "5 % 2", for example, would evaluate to 1, while "9 % 3" would evaluate to 0 because the dividend (9) is a multiple of the divisor (3).

NR			Remainder
1	%	4	1
2	%	4	2
3	%	4	3
4	%	4	0
5	%	4	1
6	%	4	2
7	%	4	3
8	%	4	0
9	%	4	1
10	%	4	2
11	%	4	3
12	%	4	0
13	%	4	1
14	%	4	2
15	%	4	3
16	%	4	0

Now that we have looked at both the AWK "NR" VARIABLE and AWK MODULO ARITHMETIC, lets use both to find the read length and number reads present in a FASTQ file.

```
# Find read length for one sample at a time

cd /home/$USER/DGE_Virtual/raw_reads/

zcat 2S1Flag-p5-2.fq.gz | awk '{ if ( NR % 4 == 2) print length ($1)}' | head

50
50
50
50
50
50
50
50
50
50

# Using a loop to retrieve sequence lengths for multiple files

for file in *.gz; do echo ${file}; zcat ${file} | awk '{ if ( NR % 4 == 2) print length ($1)}' | head; done

# Find number of reads for one file at a time

zcat 2S1Flag-p5-2.fq.gz | awk '{ if ( NR % 4 == 2) print $0}' | wc -l
11,816,497

# Using a loop to find the number of reads for multiple files

for file in *.gz; do echo ${file}; zcat ${file} | awk '{ if ( NR % 4 == 2) print $0}' | wc -l; done
```

Sample File Name		Read Length	Total Number of Reads
2S1Flag-p5-2.fq.gz			
2S1Flag-p6-3.fq.gz			
2S1-Flag-p7-2.fq.gz			
759_7-p5-2.fq.gz			
759_7-p6-1-1.fq.gz			
759_7-p6-2-2.fq.gz			
pCDNA_p6-3.fq.gz			
pCDNA_p7-2.fq.gz			
pCDNA_p8-3.fq.gz			
Scram_1-3.fq.gz			
Scram_1_p3-1.fq.gz			
Scram_1_p3-3.fq.gz			

FASTQC

```
cd /home/$USER/DGE_Virtual/raw_reads/

mkdir fastqc

source activate fastQC

fastqc --help

#Run fastqc on one file

fastqc -o fastqc/ -f fastq 2S1Flag-p5-2.fq.gz

#Run fastqc on two files

fastqc -o fastqc/ -f fastq 2S1Flag-p5-2.fq.gz 2S1Flag-p6-3.fq.gz

#Run fastqc on all .gz files using a wildcard (*)

fastqc -o fastqc/ -f fastq *.gz

# -o: Create all output files in the specified output directory.
# Please note that this directory must exist as the program will not create it.
# If this option is not set then the output file for each sequence file is
# created in the same directory as the sequence file which was processed.

# -f -> Bypasses the normal sequence file format detection and
# forces the program to use the specified format.
# Valid formats are bam,sam,bam_mapped,sam_mapped and fastq

# *.gz -> Will run FASTQC for all files in the current folder that ends with .gz

# FastQC on all the files will take approximately 10 minutes to run to completion

ls -ltr fastqc/

# scp (from server to desktop) to view the content of the .html output files in a web browser

# Open terminal and cd into your Desktop (local computer)

scp -P 44111 -r <username>@gateway.training.ncgr.org:/home/<username>/DGE_Virtual/raw_reads/fastqc ./
```

FASTQC ONLINE RESOURCES:

https://www.bioinformatics.babraham.ac.uk/projects/fastqc/

https://www.bioinformatics.babraham.ac.uk/projects/fastqc/Help/

GOOD ILLUMINA DATA

https://www.bioinformatics.babraham.ac.uk/projects/fastqc/good_sequence_short_fastqc.html

BAD ILLUMINA DATA

https://www.bioinformatics.babraham.ac.uk/projects/fastqc/bad_sequence_fastqc.html

ADAPTER DIMER CONTAMINATED RUN

https://www.bioinformatics.babraham.ac.uk/projects/fastqc/RNA-Seq_fastqc.html

SMALL RNA WITH READ-THROUGH ADAPTER

https://www.bioinformatics.babraham.ac.uk/projects/fastqc/small_rna_fastqc.html

REDUCED REPRESENTATION BS-SEQ

https://www.bioinformatics.babraham.ac.uk/projects/fastqc/RRBS_fastqc.html

PACBIO

https://www.bioinformatics.babraham.ac.uk/projects/fastqc/pacbio_srr075104_fastqc.html

454

https://www.bioinformatics.babraham.ac.uk/projects/fastqc/454_SRR073599_fastqc.html