Shaleigh Smith
Next Generation Sequencing Informatics
Assigned Coursework #6

This study aligned m6A data against the human genome and determined which genes produced m6A-modified mRNAs in test vs. control conditions. The data included three input control RNA datasets (iCTRL1-3), three input test RNA datasets (iDS1-3), three m6A enriched RNA control datasets (mCTRL1-3) and three m6A-enriched test RNA datasets (mDS1-3). The data consisted of single-end reads obtained from RIP-seq.

The twelve datasets were downloaded from the SRA and trimmed for adapters along with a quality score above 30. Each dataset was then aligned using Bowtie2. The alignment used FASTQ files (-q), removed all unmapped reads (--no-unal) and had maximum sensitivity (--very-sensitive). The sensitivity parameters allow no mismatches (-N 0), sets the length of the seed substrings (-L 20), sets the intervals in respect to seed length (-I s, 1, 0, 0.50), and limits seed extension and maximum numbers of repetitive seed events (-D 20 -R 3). Interestingly, using the --no-unal results in truncated sam and bam files; however, these still successfully ran through exomePeak and gave significantly differential peaks.

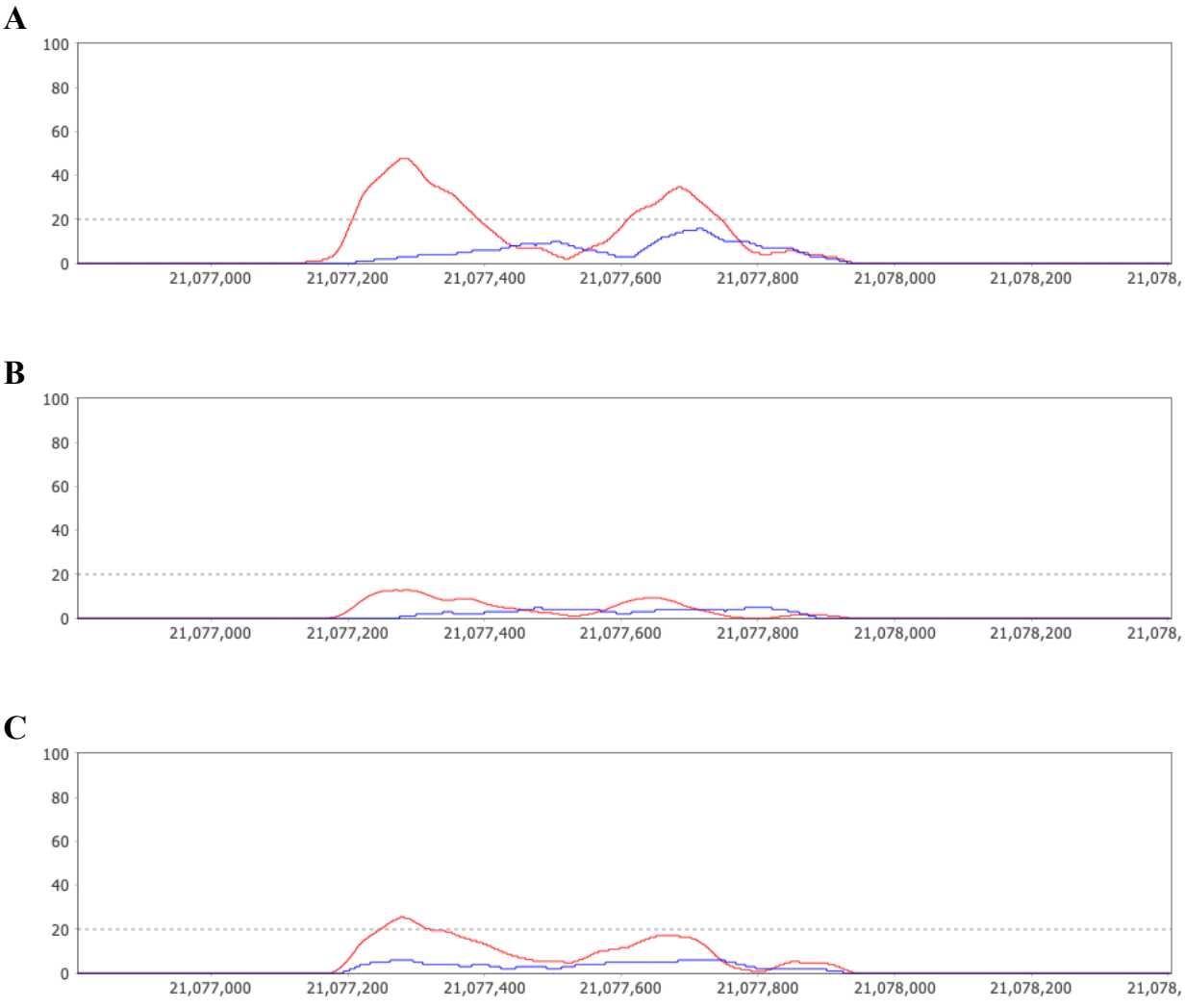
ExomePeak was run four times. The first identified genes producing m6A modified transcripts that differ between all test and control datasets. The other three identified these genes that were modified in each individual dataset. Each individual dataset had a different number of genes modified: 338 genes were identified in mDS1, 346 genes were identified in mDS2 had, and 4116 genes were identified in mDS3. In all datasets, 395 genes were modified consistently, and 478 genes were modified in at least two datasets (Table 1). The peak structure of Interferon, Beta-1 (IFNB1) was different across the three datasets but generally illustrated two peaks against the background in mDS1 and mDS3 (Figure 1).

See below for figures and code.

Table 1. ExomePeak identification of genes producing m6a modified transcripts that differ between test and control datasets. Genes that had multiple modified transcripts were only counted once.

	Number of Genes Identified
mDS1	338
mDS2	346
mDS3	4116
All Datasets	395
At Least Two Datasets	478

Figure 1. Plots derived from m6a viewer showing the m6a peak structure of IFNB1 in each test dataset (red) against its paired input control (blue). X axis is the coordinates on chromosome 9 in base pairs, centered on IFNB1 at chr9:21,077,104-21,077,962 **(a)** Test dataset mDS1. **(b)** Test dataset mDS2. **(c)** Test dataset mDS3.



Code

The code below downloads, trims, aligns, and converts file format:

```
#!/bin/bash
#SBATCH --job-name=ngs6  # Job name
#SBATCH --mail-type=END,FAIL # Mail events (NONE, BEGIN, END, FAIL, ALL)
#SBATCH --mail-user=shaleigh.smith@nyulangone.org # Where to send mail
#SBATCH --ntasks=8 # Run on a multiple CPU
#SBATCH --mem=64gb # Job memory request
#SBATCH --time=12:00:00 # Time limit hrs:min:sec
#SBATCH --output=/gpfs/scratch/sas1531/ngs6_coursework/ngs6_%j.log # Standard output
and error log
#SBATCH -p cpu_short

# Load modules
module load sratoolkit/2.9.1
module load fastqc/0.11.7
module load trimgalore/0.5.0
module load python/cpu/2.7.15-ES ### CutAdapt is hidden in here
module load bowtie2/2.3.4.1
module load samtools/1.9

# Download datasets
fastq-dump ${1} --gzip -O /gpfs/scratch/sas1531/ngs6_coursework/

# Remove fastq-dump directory
rm -r ~/ncbi

# Rename files
mv/gpfs/scratch/sas1531/ngs6_coursework/${1}*fastq.gz
/gpfs/scratch/sas1531/ngs6_coursework/${2}.fastq.gz

# Trim
# These are single end reads (not paired) sequenced using RIP-Seq
trim_galore --q 30 \
--phred33 \
-o /gpfs/scratch/sas1531/ngs6_coursework/ \
--fastqc ./${2}.fastq.gz

# Build bowtie2 index (done once)
#bowtie2-build                                                                  -f
/gpfs/scratch/sas1531/hg38/Homo_sapiens/UCSC/hg38/Sequence/WholeGenomeFasta/genome.fa
hg38_index

# Align to hg38
bowtie2 -q \
--end-to-end \
--very-sensitive \
--no-unal \
-x /gpfs/scratch/sas1531/hg38/Homo_sapiens/UCSC/hg38/Sequence/Bowtie2Index/genome \
-U ./${2}_trimmed.fq.gz \
```

```
-S ./${2}.sam

# Convert to bam and sort
samtools view -S -b ${2}.sam > ${2}.bam
samtools sort ${2}.bam -o ${2}_sorted.bam
samtools index ${2}_sorted.bam
```

Code

The code below is the submitter script for the first script:

```
#!/bin/bash
#SBATCH --job-name=job_submitter
#SBATCH --nodes=1
#SBATCH --mem=200MB
#SBATCH --time=1:00:00
#SBATCH --error=job_sub_error.txt
#SBATCH --output=job_sub_stdout.txt

sbatch --array=1 ngs6_script.sh  SRR7992458  iCTRL1
sbatch --array=1 ngs6_script.sh  SRR7992461  iCTRL2
sbatch --array=1 ngs6_script.sh  SRR7992460  iCTRL3
sbatch --array=1 ngs6_script.sh  SRR7992450  iDS1
sbatch --array=1 ngs6_script.sh  SRR7992457  iDS2
sbatch --array=1 ngs6_script.sh  SRR7992456  iDS3
sbatch --array=1 ngs6_script.sh  SRR7992455  mCTRL1
sbatch --array=1 ngs6_script.sh  SRR7992454  mCTRL2
sbatch --array=1 ngs6_script.sh  SRR7992459  mCTRL3
sbatch --array=1 ngs6_script.sh  SRR7992453  mDS1
sbatch --array=1 ngs6_script.sh  SRR7992452  mDS2
sbatch --array=1 ngs6_script.sh  SRR7992451  mDS3
```

Code

The code below runs ExomePeak and performs downstream analysis:

```
### Shaleigh Smith
### NGS Coursework 6

### Import libraries ###
library(tidyverse)
library(exomePeak)
library(dplyr)

### Set paths to files ###
# GTF
gtf <- "../genes.gtf"
# Input RNA
```

```
i_control_1 <- "../iCTRL1_sorted.bam"
i_control_2 <- "../iCTRL2_sorted.bam"
i_control_3 <- "../iCTRL3_sorted.bam"
i_sample_1 <- "../iDS1_sorted.bam"
i_sample_2 <- "../iDS2_sorted.bam"
i_sample_3 <- "../iDS3_sorted.bam"
# m6a Enriched RNA (treated)
m_control_1 <- "../mCTRL1_sorted.bam"
m_control_2 <- "../mCTRL2_sorted.bam"
m_control_3 <- "../mCTRL3_sorted.bam"
m_sample_1 <- "../mDS1_sorted.bam"
m_sample_2 <- "../mDS2_sorted.bam"
m_sample_3 <- "../mDS3_sorted.bam"

### ExomePeak ###
# Set working directory
setwd("/Users/sha/Desktop/NGS_Informatics/NGS_courswork/ngs6_coursework_shaleigh_smit
h/exome_1")
# Run exomepeak on all datasets (with all controls)
result <- exomepeak(GENE_ANNO_GTF = gtf,
                    IP_BAM = c(m_control_1,
                               m_control_2,
                               m_control_3),
                    INPUT_BAM = c(i_control_1,
                                  i_control_2,
                                  i_control_3),
                    TREATED_IP_BAM = c(m_sample_1,
                                       m_sample_2,
                                       m_sample_3),
                    TREATED_INPUT_BAM = c(i_sample_1,
                                          i_sample_2,
                                          i_sample_3))

# Set working directory for first dataset
setwd("/Users/sha/Desktop/NGS_Informatics/NGS_courswork/ngs6_coursework_shaleigh_smit
h/exome_m1")
# Run exomepeak on first dataset against all controls
result1 <- exomepeak(GENE_ANNO_GTF = gtf,
                     IP_BAM = c(m_control_1,
                                m_control_2,
                                m_control_3),
                     INPUT_BAM = c(i_control_1,
                                   i_control_2,
                                   i_control_3),
                     TREATED_IP_BAM = c(m_sample_1),
                     TREATED_INPUT_BAM = c(i_sample_1))

# Set working directory for second dataset
setwd("/Users/sha/Desktop/NGS_Informatics/NGS_courswork/ngs6_coursework_shaleigh_smit
h/exome_m2")
# Run exomepeak on second dataset against all controls
result2 <- exomepeak(GENE_ANNO_GTF = gtf,
                     IP_BAM = c(m_control_1,
                                m_control_2,
                                m_control_3),
```

```
                    INPUT_BAM = c(i_control_1,
                                  i_control_2,
                                  i_control_3),
                    TREATED_IP_BAM = c(m_sample_2),
                    TREATED_INPUT_BAM = c(i_sample_2))

# Set working directory for third dataset
setwd("/Users/sha/Desktop/NGS_Informatics/NGS_courswork/ngs6_coursework_shaleigh_smit
h/exome_m3")
# Run exomepeak on third dataset against all controls
result3 <- exomepeak(GENE_ANNO_GTF = gtf,
                    IP_BAM = c(m_control_1,
                               m_control_2,
                               m_control_3),
                    INPUT_BAM = c(i_control_1,
                                  i_control_2,
                                  i_control_3),
                    TREATED_IP_BAM = c(m_sample_3),
                    TREATED_INPUT_BAM = c(i_sample_3))

### Downstream Analysis ###
# Results
result
result1
result2
result3

# View results (all)
con_sig_diff_1 <- read.table("./exome_1/exomePeak_output/con_sig_diff_peak.xls",
                             head = TRUE, sep = "\t")
sig_diff_1 <- read.table("./exome_1/exomePeak_output/sig_diff_peak.xls",
                         head = TRUE, sep = "\t")
diff_1 <- read.table("./exome_1/exomePeak_output/diff_peak.xls",
                     head = TRUE, sep = "\t")

# View results (m1)
con_sig_diff_m1 <- read.table("./exome_m1/exomePeak_output/con_sig_diff_peak.xls",
                              head = TRUE, sep = "\t")
sig_diff_m1 <- read.table("./exome_m1/exomePeak_output/sig_diff_peak.xls",
                          head = TRUE, sep = "\t")
diff_m1 <- read.table("./exome_m1/exomePeak_output/diff_peak.xls",
                      head = TRUE, sep = "\t")

# View results (m2)
con_sig_diff_m2 <- read.table("./exome_m2/exomePeak_output/con_sig_diff_peak.xls",
                              head = TRUE, sep = "\t")
sig_diff_m2 <- read.table("./exome_m2/exomePeak_output/sig_diff_peak.xls",
                          head = TRUE, sep = "\t")
diff_m2 <- read.table("./exome_m2/exomePeak_output/diff_peak.xls",
                      head = TRUE, sep = "\t")

# View results (m3)
con_sig_diff_m3 <- read.table("./exome_m3/exomePeak_output/con_sig_diff_peak.xls",
                              head = TRUE, sep = "\t")
sig_diff_m3 <- read.table("./exome_m3/exomePeak_output/sig_diff_peak.xls",
```

```
                          head = TRUE, sep = "\t")
diff_m3 <- read.table("./exome_m3/exomePeak_output/diff_peak.xls",
                      head = TRUE, sep = "\t")

# Modified Transcripts that differ between test and control datasets (unique)
length(unique(con_sig_diff_1$name)) # 395

# Number of genes modified in each individual dataset (unique)
length(unique(con_sig_diff_m1$name)) # 338
length(unique(con_sig_diff_m2$name)) # 346
length(unique(con_sig_diff_m3$name)) # 4116

# Number of genes present in at least two of the test datasets and none of the
control
# Select gene name interest and add data column with 1
con_m1 <- dplyr::select(con_sig_diff_m1, name)
con_m1 <- distinct(con_m1)
con_m1$data_m1 <- 1
con_m2 <- dplyr::select(con_sig_diff_m2, name)
con_m2 <- distinct(con_m2)
con_m2$data_m2 <- 1
con_m3 <- dplyr::select(con_sig_diff_m3, name)
con_m3 <- distinct(con_m3)
con_m3$data_m3 <- 1

# Full join datasets, fill NA with 0
con <- merge(con_m1, con_m2, by = "name", all = TRUE)
con <- merge(con, con_m3, by = "name", all = TRUE)
con[is.na(con)] <- 0

# Sum data columns
con$count <- rowSums(con[ ,2:4])

# Numer of genes present in at least two of the test datasets
nrow(con[con$count > 1, ]) # 478
```