Erica Chio, Assigned Coursework #8

Due date: April 23rd (two weeks), 11pm EST
Submission: PDF document by email to daniel.depledge@nyulangone.org

- You are going to learn how to perform RNA modification detection using NanoCompore – this uses a comparative approach to compare datasets.
- **Dataset 1: VA-24h** – this dataset contains sequencing data from RNAs with base modifications [all modifications are present]
- **Dataset 2: VA-M3KO** – this dataset contains sequencing data from RNAs without m^6A modifications [other base modifications are present]
- **Dataset 3: VA-pure** – this dataset contains sequencing data from RNAs without any base modifications [no modifications are present]
- All datasets are generated using a targeted nanopore direct RNA sequencing approach against an adenovirus RNA called VA (Ad5_VAI.fasta)
- Full instructions for using NanoCompore can be found here - https://nanocompore.rna.rocks/data_preparation/
- Provide details on where all putative modified sites are located – with a visualization
- Provide details on where m^6A putativee modified sites are located – with a visualization

Notes

1. Copy the raw data folders to your own scratch space

```
cd /gpfs/data/courses/bminga3004/Practicum11/Assignment/
Ad5_VAI.fasta  VA_24h  VA_M3KO  VA_pure

cp -R * /gpfs/scratch/ebc308/AIS/coursework8data/data
```

2. First you will need to re-basecall the data with the latest version of Guppy (3.4.5)

```
#!/bin/bash
#SBATCH --job-name=guppy # Job name
#SBATCH --mail-type=END,FAIL # Mail events (NONE, BEGIN, END, FAIL, ALL)
#SBATCH --mail-user=Erica.Chio@nyulangone.org # Where to send mail
#SBATCH --nodes=1 # Nodes
```

```
#SBATCH --gres=gpu:2 # Calling gpu nodes
#SBATCH --ntasks=8 # Run on a single CPU
#SBATCH --mem=32gb # Job memory request
#SBATCH --time=12:00:00 # Time limit hrs:min:sec
#SBATCH --output=guppy_%j.log # Standard output and error log
#SBATCH -p gpu8_short # Specifies location to submit job

module purge
module load guppy/3.4.5

# --compress_fastq Compress the fastq output
# --flowcell FLO-MIN106 (MinIONflowcell)
# --kit SQK-RNA002 (RNA-002)
# -r recursively go into directories
# --trim_strategy rna (because working with RNA dataset)
# --reverse_sequence true (reverse so its 5` to 3`)
# --u_substitution true (covert u to t for better alignment)
# -i where the raw read files are located
# -s where the basecalled files will be saved
# --device (similar to -x, assigning graphics card)
# https://github.com/tleonardi/nanocompore_pipeline/blob/master/pipeline.nf
nanopore pipeline

guppy_basecaller --compress_fastq --flowcell FLO-MIN106 --kit SQK-RNA002 -r --
trim_strategy rna --reverse_sequence true --u_substitution true -x auto -i /
gpfs/scratch/ebc308/AIS/coursework8data/data/$1/fast5_pass -s /gpfs/scratch/
ebc308/AIS/coursework8data/$2
```

```
sbatch guppy.sh VA_24h VA_24h
sbatch guppy.sh VA_M3KO/20200221_1731_MN24978_FAL81867_a9aa11a1 VA_M3KO
sbatch guppy.sh VA_pure/20200316_1810_MN24978_FAL86847_f04e883a VA_pure
```

```
cat VA_24h/*fastq.gz > VA_24h.fastq.gz
cat VA_M3KO/*fastq.gz > VA_M3KO.fastq.gz
```

```
cat VA_pure/*fastq.gz > VA_pure.fastq.gz
```

3. You will then need to align the data against the VA sequence (Ad5_VAI.fasta) using miniMap2 (think carefully about parameters)

```
#!/bin/bash
#SBATCH --job-name=minimap2 # Job name
#SBATCH --mail-type=END,FAIL # Mail events (NONE, BEGIN, END, FAIL, ALL)
#SBATCH --mail-user=Erica.Chio@nyulangone.org # Where to send mail
#SBATCH --ntasks=8 # Run on a single CPU
#SBATCH --mem=16gb # Job memory request
#SBATCH --time=24:00:00 # Time limit hrs:min:sec
#SBATCH --output=minimap2_%j.log # Standard output and error log
#SBATCH -p cpu_short # Specifies location to submit job

module purge
module load minimap2/2.15
module load samtools/1.9-new

# -a Long-read alignment with CIGAR:
# -x preset
# map-ont Slightly more sensitive for Oxford Nanopore to reference mapping (-k15). For PacBio reads, HPC minimizers consistently leads to faster performance and more sensitive results in comparison to normal minimizers. For Oxford Nanopore data, normal minimizers are better, though not much. The effectiveness of HPC is determined by the sequencing error mode.
# -L Write CIGAR with >65535 operators at the CG tag. Older tools are unable to convert alignments with >65535 CIGAR ops to BAM. This option makes minimap2 SAM compatible with older tools. Newer tools recognizes this tag and reconstruct the real CIGAR in memory.

# -b option makes the output compressed
# -h causes the SAM headers to be output
# -F filter out samtool flags 2324 (read unmapped, read reverse strand, not primary alignemnt, supplementary alignment)
# -q quality
```

```
minimap2 -ax map-ont -L /gpfs/scratch/ebc308/AIS/coursework8data/data/
Ad5_VAI.fasta /gpfs/scratch/ebc308/AIS/coursework8data/$1.fastq.gz | samtools
view -bh -F 2324 -q 10 | samtools sort -O bam > /gpfs/scratch/ebc308/AIS/
coursework8data/minimap2/$1_aligned_reads.bam

samtools index /gpfs/scratch/ebc308/AIS/coursework8data/
minimap2/$1_aligned_reads.bam
```

```
sbatch minimap2.sh VA_24h
sbatch minimap2.sh VA_M3KO
sbatch minimap2.sh VA_pure
```

4. You will need to read the NanoCompore paper and manual to perform the analyses and make sense of the data output

Pre Processing Data (Nanopolish / NanopolishComp)

```
#!/bin/bash
#SBATCH --job-name=nanopolish # Job name
#SBATCH --mail-type=END,FAIL # Mail events (NONE, BEGIN, END, FAIL, ALL)
#SBATCH --mail-user=Erica.Chio@nyulangone.org # Where to send mail
#SBATCH --ntasks=4 # Run on a single CPU
#SBATCH --mem=32gb # Job memory request
#SBATCH --time=12:00:00 # Time limit hrs:min:sec
#SBATCH --output=nanopolish_%j.log # Standard output and error log
#SBATCH -p cpu_short # Specifies location to submit job

module purge
module load python/cpu/3.6.5
module load nanopolish/0.11.1

nanopolish index -s /gpfs/scratch/ebc308/AIS/coursework8data/$1/
sequencing_summary.txt -d /gpfs/scratch/ebc308/AIS/coursework8data/data/$2/
fast5_pass /gpfs/scratch/ebc308/AIS/coursework8data/$1.fastq.gz

nanopolish eventalign --reads /gpfs/scratch/ebc308/AIS/coursework8data/
```

```
$1.fastq.gz --bam /gpfs/scratch/ebc308/AIS/coursework8data/
minimap2/$1_aligned_reads.bam --genome /gpfs/scratch/ebc308/AIS/
coursework8data/data/Ad5_VAI.fasta --samples --print-read-names --scale-events
--samples > /gpfs/scratch/ebc308/AIS/coursework8data/nanopolish/
$1_eventalign_reads.tsv

NanopolishComp Eventalign_collapse -i /gpfs/scratch/ebc308/AIS/coursework8data/
nanopolish/$1_eventalign_reads.tsv -o /gpfs/scratch/ebc308/AIS/coursework8data/
nanopolish/$1_eventalign_collapsed_reads.tsv
```

```
sbatch nanopolish.sh VA_24h VA_24h; sbatch nanopolish.sh VA_M3KO VA_M3KO/
20200221_1731_MN24978_FAL81867_a9aa11a1; sbatch nanopolish.sh VA_pure VA_pure/
20200316_1810_MN24978_FAL86847_f04e883a
```

Nanocompore

```
#!/bin/bash
#SBATCH --job-name=nanocompore # Job name
#SBATCH --mail-type=END,FAIL # Mail events (NONE, BEGIN, END, FAIL, ALL)
#SBATCH --mail-user=Erica.Chio@nyulangone.org # Where to send mail
#SBATCH --ntasks=4 # Run on a single CPU
#SBATCH --mem=32gb # Job memory request
#SBATCH --time=12:00:00 # Time limit hrs:min:sec
#SBATCH --output=nanocompore_%j.log # Standard output and error log
#SBATCH -p cpu_short # Specifies location to submit job

module purge
module load python/cpu/3.6.5

nanocompore sampcomp \
    --file_list1 /gpfs/scratch/ebc308/AIS/coursework8data/nanopolish/
$1_eventalign_collapsed_reads.tsv/out_eventalign_collapse.tsv \
    --file_list2  /gpfs/scratch/ebc308/AIS/coursework8data/nanopolish/
$2_eventalign_collapsed_reads.tsv/out_eventalign_collapse.tsv \
    --label1 $1 \
    --label2 $2 \
```

```
    --fasta /gpfs/scratch/ebc308/AIS/coursework8data/data/Ad5_VAI.fasta \
    --outpath /gpfs/scratch/ebc308/AIS/coursework8data/nanocompore/$1_$2/
```

```
sbatch nanocompore.sh VA_24h VA_pure; sbatch nanocompore.sh VA_24h VA_M3KO
```

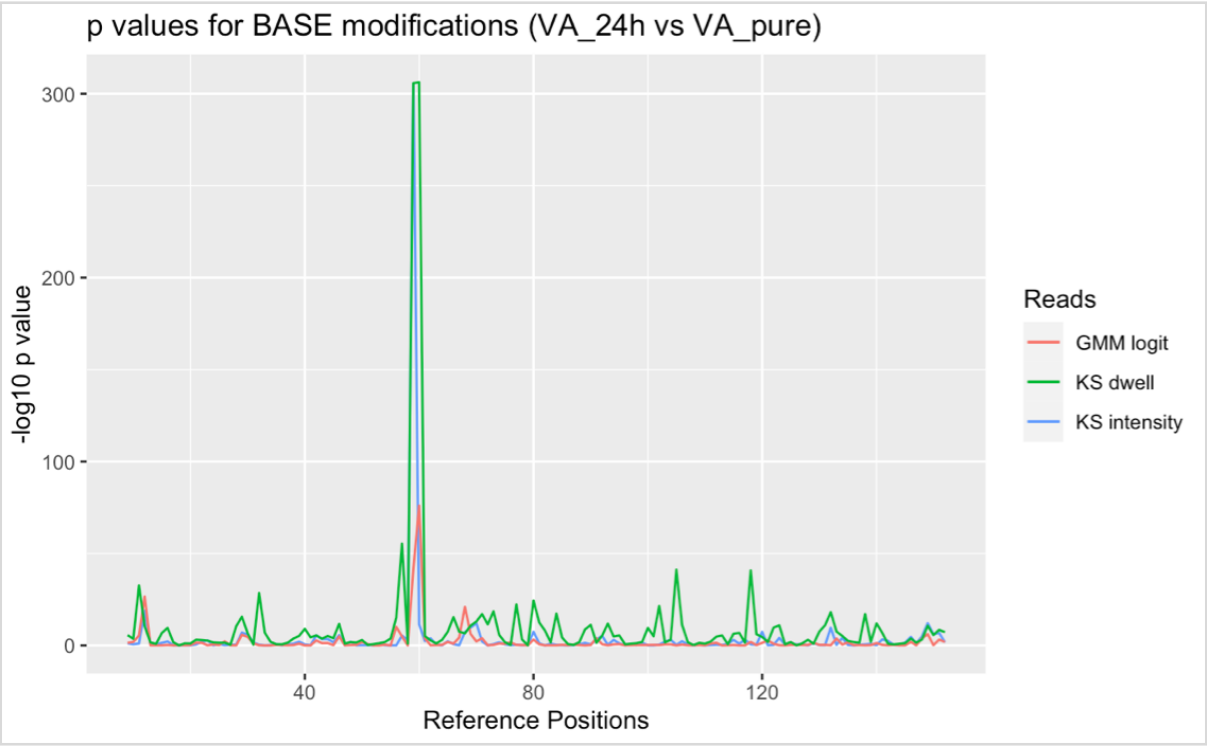
Visualization

```
library(ggplot2)
library(ggpubr)
library(cowplot)
```

VA_24h vs VA_Pure to get where any modified bases are

```
VA_24_results <- read.table("/Users/ericachio/Documents/sackler/applied
informatics sequencing/coursework8/VA_24h_VA_pure/out_nanocompore_results.tsv",
header = TRUE)
VA_24_shift_stats <- read.table("/Users/ericachio/Documents/sackler/applied
informatics sequencing/coursework8/VA_24h_VA_pure/
out_nanocompore_shift_stats.tsv", header = TRUE)

plot1 <- ggplot(data=VA_24_shift_stats, aes(x=pos)) + geom_line(aes(y =
c1_median_intensity, color = "red")) + geom_line(aes(y = c2_median_intensity,
color="blue")) + ylab("Median Signal Intensity") + xlab("Reference Positions")
+ scale_color_discrete(name = "Reads", labels = c("C1", "C2")) +
ggtitle("Modified Signals for BASE modifications (VA_24h vs VA_pure)")

plot2 <- ggplot(data=VA_24_shift_stats, aes(x=pos)) + geom_line(aes(y =
log10(c1_median_dwell), color = "red")) + geom_line(aes(y =
log10(c2_median_dwell), color="blue")) + ylab("Log10 (Dwell Time)") +
xlab("Reference Positions") + scale_color_discrete(name = "Reads", labels =
c("C1", "C2"))

plot_grid(plot1, plot2,
          labels = c("A", "B"),
          ncol = 1, nrow = 2)
```

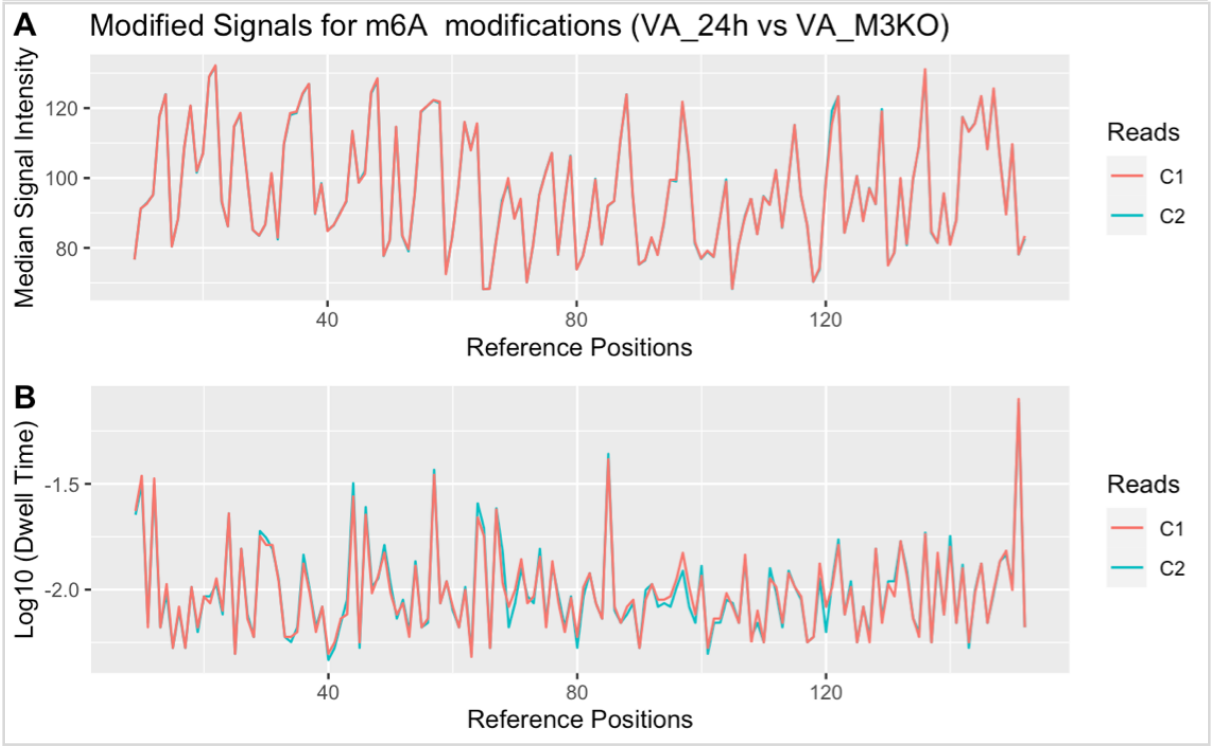

```
plot3 <- ggplot(data=VA_24_results, aes(x=pos)) + geom_line(aes(y = -
log10(GMM_logit_pvalue), color = "red")) + geom_line(aes(y = -
log10(KS_dwell_pvalue), color="blue")) + geom_line(aes(y = -
log10(KS_intensity_pvalue), color="green")) + ylab("-log10 p value") +
xlab("Reference Positions") + scale_color_discrete(name = "Reads", labels =
c("GMM logit", "KS dwell", "KS intensity")) + ggtitle("Modified Signals for
BASE modifications (VA_24h vs VA_pure)")
plot3
```



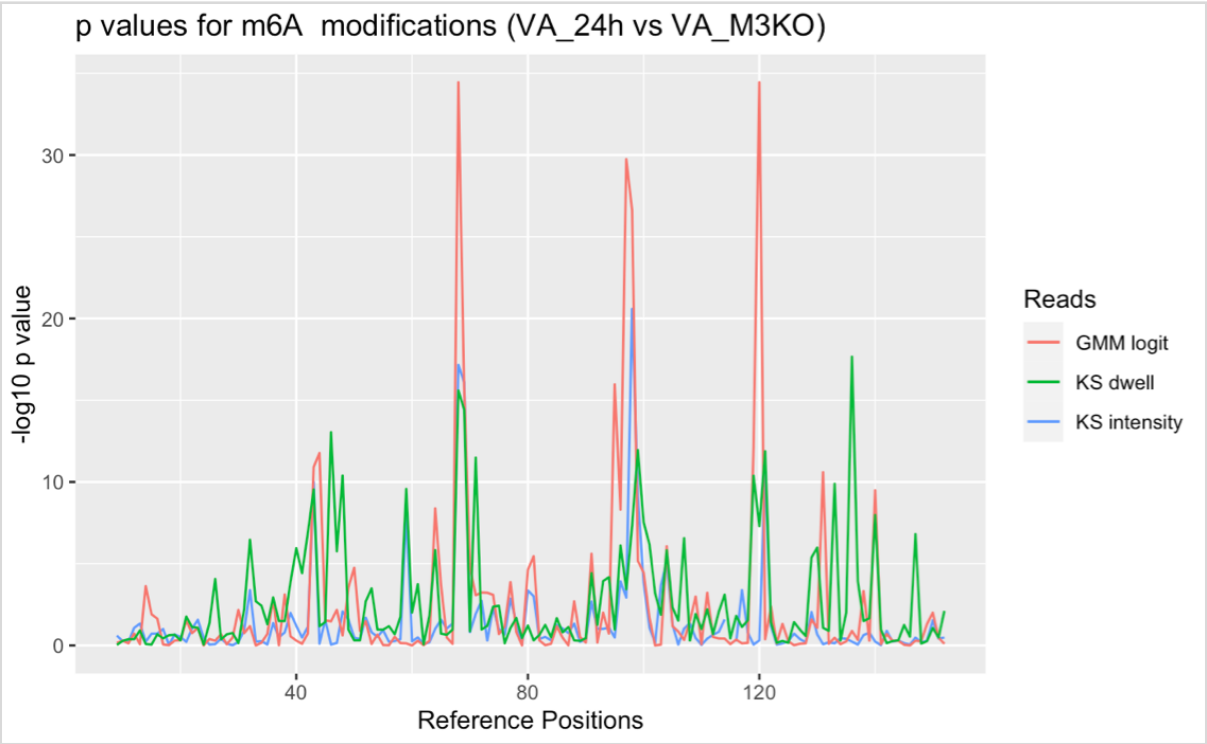
VA_24h vs VA_M3KO to get where any m^{6}A modifications are

```
VA_M3KO_results <- read.table("/Users/ericachio/Documents/sackler/applied
informatics sequencing/coursework8/VA_24h_VA_M3KO/out_nanocompore_results.tsv",
header = TRUE)
VA_M3KO_shift_stats <- read.table("/Users/ericachio/Documents/sackler/applied
informatics sequencing/coursework8/VA_24h_VA_M3KO/
out_nanocompore_shift_stats.tsv", header = TRUE)

plot4 <- ggplot(data=VA_M3KO_shift_stats, aes(x=pos)) + geom_line(aes(y =
c1_median_intensity, color = "red")) + geom_line(aes(y = c2_median_intensity,
color="blue")) + ylab("Median Signal Intensity") + xlab("Reference Positions")
+ scale_color_discrete(name = "Reads", labels = c("C1", "C2")) +
ggtitle('Modified Signals for m6A  modifications (VA_24h vs VA_M3KO)')

plot5 <- ggplot(data=VA_M3KO_shift_stats, aes(x=pos)) + geom_line(aes(y =
log10(c1_median_dwell), color = "red")) + geom_line(aes(y =
log10(c2_median_dwell), color="blue")) + ylab("Log10 (Dwell Time)") +
xlab("Reference Positions") + scale_color_discrete(name = "Reads", labels =
c("C1", "C2"))

plot_grid(plot4, plot5,
          labels = c("A", "B"),
```

```
          ncol = 1, nrow = 2)
```




```
plot6 <- ggplot(data=VA_M3KO_results, aes(x=pos)) + geom_line(aes(y = -
log10(GMM_logit_pvalue), color = "red")) + geom_line(aes(y = -
log10(KS_dwell_pvalue), color="blue")) + geom_line(aes(y = -
log10(KS_intensity_pvalue), color="green")) + ylab("-log10 p value") +
xlab("Reference Positions") + scale_color_discrete(name = "Reads", labels =
c("GMM logit", "KS dwell", "KS intensity")) + ggtitle('Modified Signals for m6A
modifications (VA_24h vs VA_M3KO)')
plot6
```



Provide full coding and final figures/tables (with legends) - all in a single pdf file

#appliedsequencinginformatics