Shaleigh Smith
Next Generation Sequencing Informatics
Assigned Coursework #3

The following figures were obtained by the processing and analysis of two datasets, SRR1523657 and SRR1523666. Both of the datasets were RNA sequenced by Illumina Hi-Seq 2500, resulting in paired end reads. The first row of each figure shows the ideogram of chromosome 20 from hg38 and the red indicates the locus at which the reads are shown from 43,450,000 to 43,700,000. The following three rows in each figure visualize the forward (red), reverse (blue), and both (green) coverage depth of the respective reads at the locus. The label and scale of each are defined on the left-hand side of the row: the y limit was defined as the max depth value of the paired reads (both, green) in this region. Figures 2 and 4 illustrate these values log2 transformed. The fourth row displays the genome axis track, giving the location of the reads in 500kb intervals. The fifth and final row is the UCSC genes track, showing the NCBI RefSeq gene region track of stacked transcripts for the specific locus.

See below for figures.

Figure 1. SRR1523657.

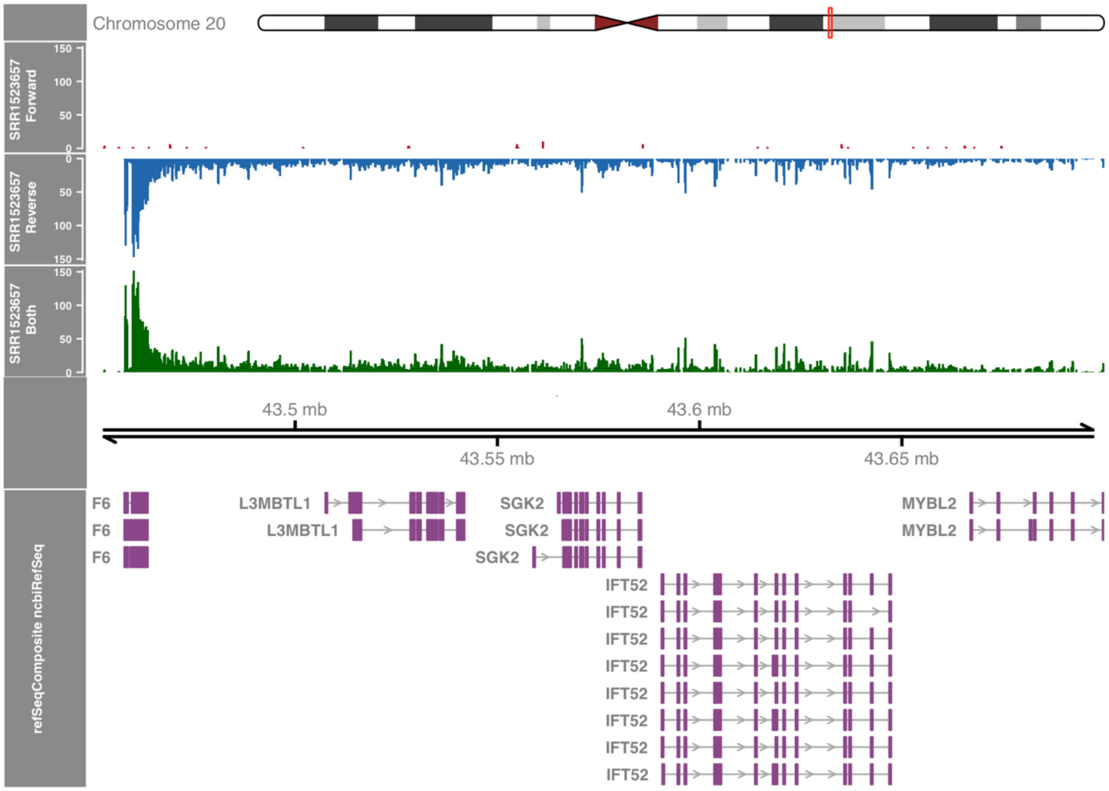


Figure 2. SRR1523657 Log Transformed.

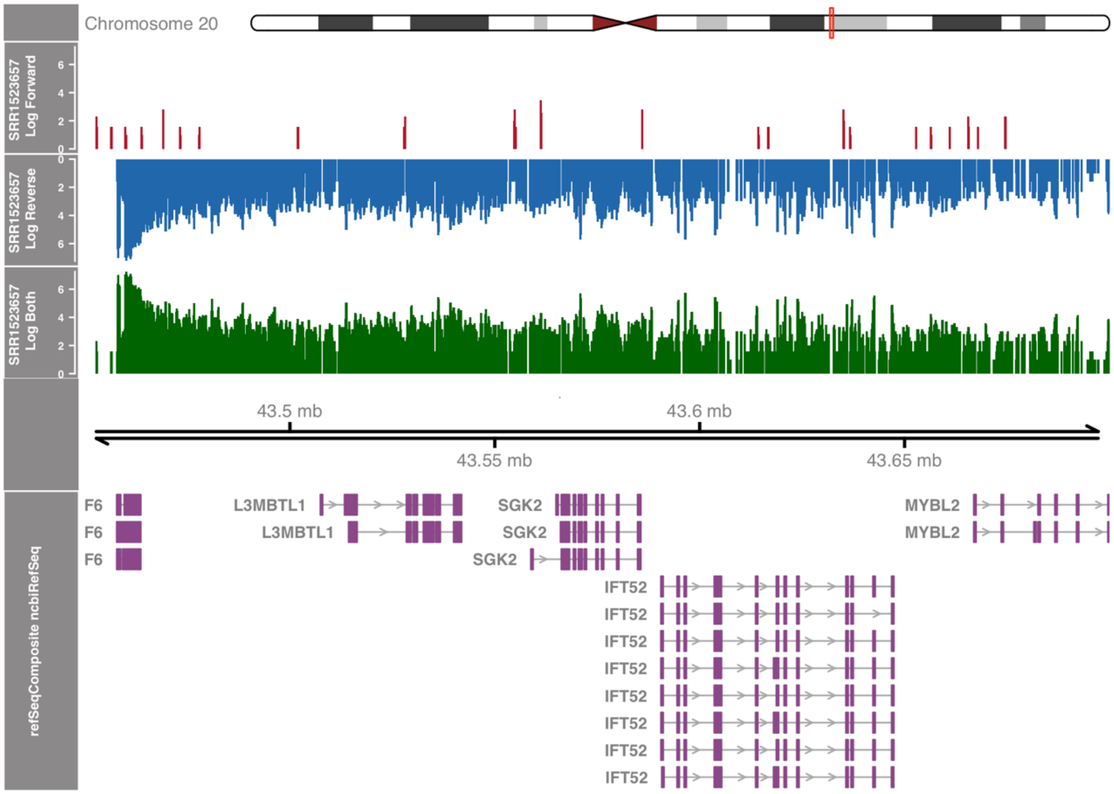


Figure 3. SRR1523666.

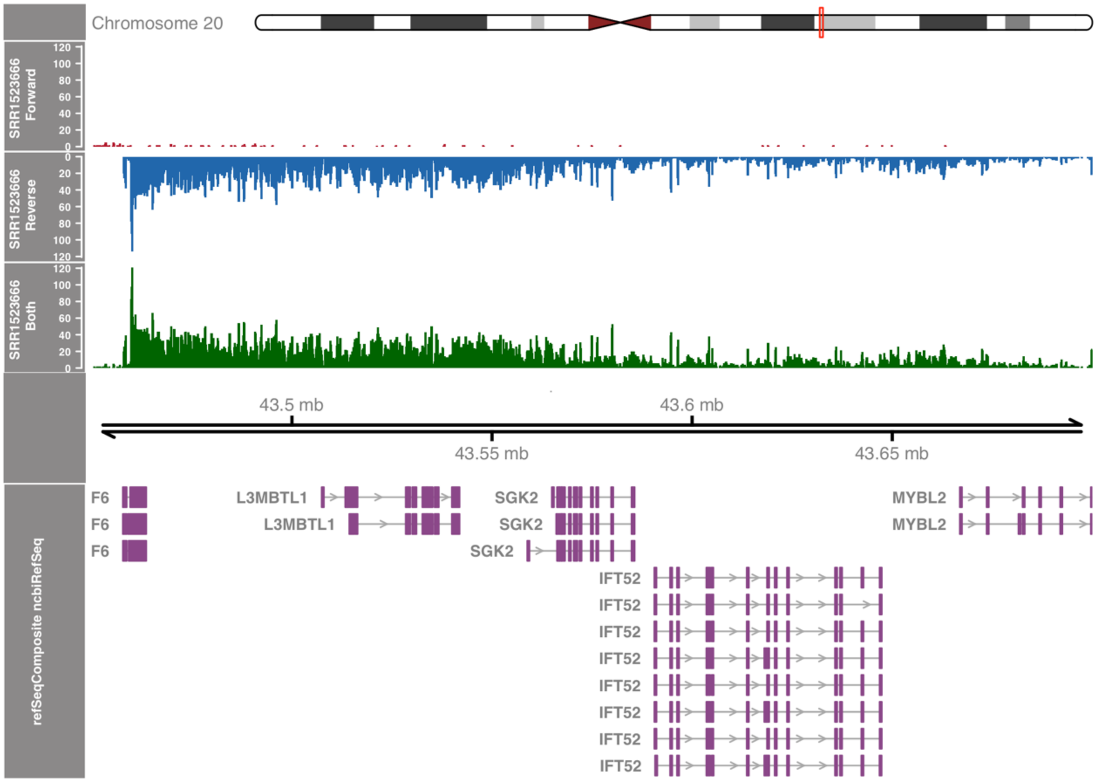
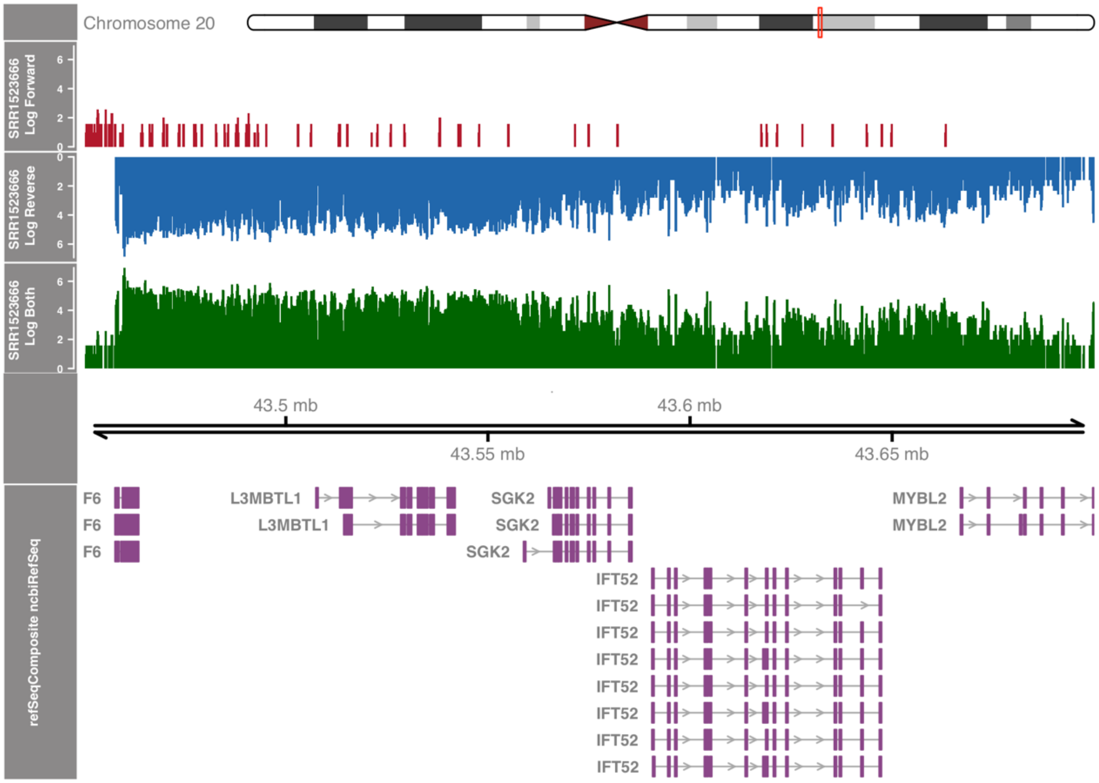


Figure 4. SRR1523666 Log Transformed.



Code:

```
### Script for download, quality control, trimming, aligning, indexing, and converting to bedgraphs
### SRR1523657
#!/bin/bash
#SBATCH --job-name=ngs3_1  # 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=4 # Run on a single CPU
#SBATCH --mem=64gb # Job memory request
#SBATCH --time=24:00:00 # Time limit hrs:min:sec
#SBATCH --output=ngs3_%j.log # Standard output and error log
#SBATCH -p cpu_medium

### Script for NGS Coursework 3

### 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 bbmap/38.25
module load samtools/1.3
module load bedtools/2.26.0

### Download datasets
fastq-dump --split-files SRR1523657 --gzip -O /gpfs/scratch/sas1531/ngs3_coursework/
rm -r ~/ncbi # fastq-dump creates a temp dir that needs to be removed

### Run fastQC on datasets
fastqc -o /gpfs/scratch/sas1531/ngs3_coursework/ /gpfs/scratch/sas1531/ngs3_coursework/SRR1523657_1.fastq.gz
/gpfs/scratch/sas1531/ngs3_coursework/SRR1523657_2.fastq.gz

### Trim datasets and run fastQC again
trim_galore --q 20 --phred33 --paired -o /gpfs/scratch/sas1531/ngs3_coursework/clean --fastqc
/gpfs/scratch/sas1531/ngs3_coursework/SRR1523657_1.fastq.gz
/gpfs/scratch/sas1531/ngs3_coursework/SRR1523657_2.fastq.gz

######### SRR1523657
### Align dataset against the human genome
bbmap.sh -Xmx26G
ref=/gpfs/scratch/sas1531/hg38/Homo_sapiens/UCSC/hg38/Sequence/WholeGenomeFasta/genome.fa
in=/gpfs/scratch/sas1531/ngs3_coursework/clean/SRR1523657_1_val_1.fq.gz
in2=/gpfs/scratch/sas1531/ngs3_coursework/clean/SRR1523657_2_val_2.fq.gz
outm=/gpfs/scratch/sas1531/ngs3_coursework/clean/SRR1523657_out.sam minid=0.90 ambiguous=random

### Parse alignment to generate sorted and indexed bam files
samtools view -b -o /gpfs/scratch/sas1531/ngs3_coursework/clean/SRR1523657_out.bam
/gpfs/scratch/sas1531/ngs3_coursework/clean/SRR1523657_out.sam
samtools sort -o /gpfs/scratch/sas1531/ngs3_coursework/clean/SRR1523657_out.sorted.bam
/gpfs/scratch/sas1531/ngs3_coursework/clean/SRR1523657_out.bam
samtools index /gpfs/scratch/sas1531/ngs3_coursework/clean/SRR1523657_out.sorted.bam
```

```
### Parse for forward strand and generate sorted and indexed bam files
samtools view -b -f99 /gpfs/scratch/sas1531/ngs3_coursework/clean/SRR1523657_out.sorted.bam >
/gpfs/scratch/sas1531/ngs3_coursework/clean/SRR1523657_out.for2.bam
samtools view -b -f147 /gpfs/scratch/sas1531/ngs3_coursework/clean/SRR1523657_out.sorted.bam >
/gpfs/scratch/sas1531/ngs3_coursework/clean/SRR1523657_out.for1.bam
samtools merge -f /gpfs/scratch/sas1531/ngs3_coursework/clean/SRR1523657_out_sorted.forward.bam
/gpfs/scratch/sas1531/ngs3_coursework/clean/SRR1523657_out.for1.bam
/gpfs/scratch/sas1531/ngs3_coursework/clean/SRR1523657_out.for2.bam
samtools index /gpfs/scratch/sas1531/ngs3_coursework/clean/SRR1523657_out_sorted.forward.bam

### Parse for reverse strand and generate sorted and indexed bam files
samtools view -b -f83 /gpfs/scratch/sas1531/ngs3_coursework/clean/SRR1523657_out.sorted.bam >
/gpfs/scratch/sas1531/ngs3_coursework/clean/SRR1523657_out.rev2.bam
samtools view -b -f163 /gpfs/scratch/sas1531/ngs3_coursework/clean/SRR1523657_out.sorted.bam >
/gpfs/scratch/sas1531/ngs3_coursework/clean/SRR1523657_out.rev1.bam
samtools merge -f /gpfs/scratch/sas1531/ngs3_coursework/clean/SRR1523657_out_sorted.reverse.bam
/gpfs/scratch/sas1531/ngs3_coursework/clean/SRR1523657_out.rev1.bam
/gpfs/scratch/sas1531/ngs3_coursework/clean/SRR1523657_out.rev2.bam
samtools index /gpfs/scratch/sas1531/ngs3_coursework/clean/SRR1523657_out_sorted.reverse.bam

### Parse and generate bedgraphs for Gvis
samtools view -b /gpfs/scratch/sas1531/ngs3_coursework/clean/SRR1523657_out_sorted.forward.bam |
genomeCoverageBed -ibam stdin -bg -split -g
/gpfs/scratch/sas1531/hg38/Homo_sapiens/UCSC/hg38/Sequence/WholeGenomeFasta/genome.fa >
/gpfs/scratch/sas1531/ngs3_coursework/clean/SRR1523657_out_sorted.forward.bedgraph
samtools view -b /gpfs/scratch/sas1531/ngs3_coursework/clean/SRR1523657_out_sorted.reverse.bam |
genomeCoverageBed -ibam stdin -bg -split -g
/gpfs/scratch/sas1531/hg38/Homo_sapiens/UCSC/hg38/Sequence/WholeGenomeFasta/genome.fa >
/gpfs/scratch/sas1531/ngs3_coursework/clean/SRR1523657_out_sorted.reverse.bedgraph
samtools view -b /gpfs/scratch/sas1531/ngs3_coursework/clean/SRR1523657_out.sorted.bam |
genomeCoverageBed -ibam stdin -bg -split -g
/gpfs/scratch/sas1531/hg38/Homo_sapiens/UCSC/hg38/Sequence/WholeGenomeFasta/genome.fa >
/gpfs/scratch/sas1531/ngs3_coursework/clean/SRR1523657_out.sorted.bedgraph

### Samtools Notes
# Explanantion for sam flags: https://broadinstitute.github.io/picard/explain-flags.html
# Explanantion of sam paired flags: https://ppotato.wordpress.com/2010/08/25/samtool-bitwise-flag-paired-reads/

### Submit job using sbatch
# sbatch ngs3_script1.sh

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

### Script for download, quality control, trimming, aligning, indexing, and converting to bedgraphs

### SRR1523666

```
#!/bin/bash
#SBATCH --job-name=ngs3_2  # 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=4 # Run on a single CPU
#SBATCH --mem=64gb # Job memory request
#SBATCH --time=24:00:00 # Time limit hrs:min:sec
```

```
#SBATCH --output=ngs3_%j.log # Standard output and error log
#SBATCH -p cpu_medium

### Script for NGS Coursework 3

### 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 bbmap/38.25
module load samtools/1.3
module load bedtools/2.26.0

### Download datasets
fastq-dump --split-files SRR1523666 --gzip -O /gpfs/scratch/sas1531/ngs3_coursework/
rm -r ~/ncbi # fastq-dump creates a temp dir that needs to be removed

### Run fastQC on datasets
fastqc -o /gpfs/scratch/sas1531/ngs3_coursework/ /gpfs/scratch/sas1531/ngs3_coursework/SRR1523666_1.fastq.gz
/gpfs/scratch/sas1531/ngs3_coursework/SRR1523666_2.fastq.gz

### Trim datasets and run fastQC again
trim_galore --q 20 --phred33 --paired -o /gpfs/scratch/sas1531/ngs3_coursework/clean --fastqc
/gpfs/scratch/sas1531/ngs3_coursework/SRR1523666_1.fastq.gz
/gpfs/scratch/sas1531/ngs3_coursework/SRR1523666_2.fastq.gz

######### SRR1523666
### Align dataset against the human genome
bbmap.sh -Xmx26G
ref=/gpfs/scratch/sas1531/hg38/Homo_sapiens/UCSC/hg38/Sequence/WholeGenomeFasta/genome.fa
in=/gpfs/scratch/sas1531/ngs3_coursework/clean/SRR1523666_1_val_1.fq.gz
in2=/gpfs/scratch/sas1531/ngs3_coursework/clean/SRR1523666_2_val_2.fq.gz
outm=/gpfs/scratch/sas1531/ngs3_coursework/clean/SRR1523666_out.sam minid=0.90 ambiguous=random

### Parse alignment to generate sorted and indexed bam files
samtools view -b -o /gpfs/scratch/sas1531/ngs3_coursework/clean/SRR1523666_out.bam
/gpfs/scratch/sas1531/ngs3_coursework/clean/SRR1523666_out.sam
samtools sort -o /gpfs/scratch/sas1531/ngs3_coursework/clean/SRR1523666_out.sorted.bam
/gpfs/scratch/sas1531/ngs3_coursework/clean/SRR1523666_out.bam
samtools index /gpfs/scratch/sas1531/ngs3_coursework/clean/SRR1523666_out.sorted.bam

### Parse for forward strand and generate sorted and indexed bam files
samtools view -b -f99 /gpfs/scratch/sas1531/ngs3_coursework/clean/SRR1523666_out.sorted.bam >
/gpfs/scratch/sas1531/ngs3_coursework/clean/SRR1523666_out.for2.bam
samtools view -b -f147 /gpfs/scratch/sas1531/ngs3_coursework/clean/SRR1523666_out.sorted.bam >
/gpfs/scratch/sas1531/ngs3_coursework/clean/SRR1523666_out.for1.bam
samtools merge -f /gpfs/scratch/sas1531/ngs3_coursework/clean/SRR1523666_out_sorted.forward.bam
/gpfs/scratch/sas1531/ngs3_coursework/clean/SRR1523666_out.for1.bam
/gpfs/scratch/sas1531/ngs3_coursework/clean/SRR1523666_out.for2.bam
samtools index /gpfs/scratch/sas1531/ngs3_coursework/clean/SRR1523666_out_sorted.forward.bam

### Parse for reverse strand and generate sorted and indexed bam files
samtools view -b -f83 /gpfs/scratch/sas1531/ngs3_coursework/clean/SRR1523666_out.sorted.bam >
/gpfs/scratch/sas1531/ngs3_coursework/clean/SRR1523666_out.rev2.bam
samtools view -b -f163 /gpfs/scratch/sas1531/ngs3_coursework/clean/SRR1523666_out.sorted.bam >
/gpfs/scratch/sas1531/ngs3_coursework/clean/SRR1523666_out.rev1.bam
```

```
samtools merge -f /gpfs/scratch/sas1531/ngs3_coursework/clean/SRR1523666_out_sorted.reverse.bam
/gpfs/scratch/sas1531/ngs3_coursework/clean/SRR1523666_out.rev1.bam
/gpfs/scratch/sas1531/ngs3_coursework/clean/SRR1523666_out.rev2.bam
samtools index /gpfs/scratch/sas1531/ngs3_coursework/clean/SRR1523666_out_sorted.reverse.bam

### Parse and generate bedgraphs for Gvis
samtools view -b /gpfs/scratch/sas1531/ngs3_coursework/clean/SRR1523666_out_sorted.forward.bam |
genomeCoverageBed -ibam stdin -bg -split -g
/gpfs/scratch/sas1531/hg38/Homo_sapiens/UCSC/hg38/Sequence/WholeGenomeFasta/genome.fa >
/gpfs/scratch/sas1531/ngs3_coursework/clean/SRR1523666_out_sorted.forward.bedgraph
samtools view -b /gpfs/scratch/sas1531/ngs3_coursework/clean/SRR1523666_out_sorted.reverse.bam |
genomeCoverageBed -ibam stdin -bg -split -g
/gpfs/scratch/sas1531/hg38/Homo_sapiens/UCSC/hg38/Sequence/WholeGenomeFasta/genome.fa >
/gpfs/scratch/sas1531/ngs3_coursework/clean/SRR1523666_out_sorted.reverse.bedgraph
samtools view -b /gpfs/scratch/sas1531/ngs3_coursework/clean/SRR1523666_out.sorted.bam |
genomeCoverageBed -ibam stdin -bg -split -g
/gpfs/scratch/sas1531/hg38/Homo_sapiens/UCSC/hg38/Sequence/WholeGenomeFasta/genome.fa >
/gpfs/scratch/sas1531/ngs3_coursework/clean/SRR1523666_out.sorted.bedgraph

### Samtools Notes
# Explanantion for sam flags: https://broadinstitute.github.io/picard/explain-flags.html
# Explanantion of sam paired flags: https://ppotato.wordpress.com/2010/08/25/samtool-bitwise-flag-paired-reads/

### Submit job using sbatch
# sbatch ngs3_script2.sh

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

### RScript for Locus Image

```
### Load Packages
library(tidyverse)
library(data.table)
library(Gviz)
library(GenomicFeatures)
library(org.Hs.eg.db)

### Set working directory
setwd("/Users/sha/Desktop/NGS_Informatics/NGS_courswork/ngs_coursework3_shaleigh_smith")

### Specify Genome and Locus
my_genome <- "hg38"
my_chr <- "chr10"
my_start <- 43450000
my_end <- 43700000

### Read in bedgraph files as simple text files
# Label column names
### SRR1523657 (1)
bedfile_1_for <- fread('./SRR1523657_out_sorted.forward.bedgraph',
                    col.names = c('chromosome', 'start', 'end', 'value'))
bedfile_1_rev <- fread('./SRR1523657_out_sorted.reverse.bedgraph',
                    col.names = c('chromosome', 'start', 'end', 'value'))
bedfile_1_both <- fread('./SRR1523657_out.sorted.bedgraph',
                     col.names = c('chromosome', 'start', 'end', 'value'))
```

```
### SRR1523666 (2)
bedfile_2_for <- fread('./SRR1523666_out_sorted.forward.bedgraph',
                      col.names = c('chromosome', 'start', 'end', 'value'))
bedfile_2_rev <- fread('./SRR1523666_out_sorted.reverse.bedgraph',
                      col.names = c('chromosome', 'start', 'end', 'value'))
bedfile_2_both <- fread('./SRR1523666_out.sorted.bedgraph',
                       col.names = c('chromosome', 'start', 'end', 'value'))

### Determine the maximium depth value within the specific locus
### SRR1523657 (1)
chr_data_1 <- bedfile_1_both[bedfile_1_both$chromosome == "chr10",]
chr_data_1_start <- chr_data_1[chr_data_1$start > my_start]
chr_data_1_end <- chr_data_1_start[chr_data_1_start$end < my_end,]
max_value_1 <- max(chr_data_1_end$value)
### SRR1523666 (2)
chr_data_2 <- bedfile_2_both[bedfile_2_both$chromosome == "chr10",]
chr_data_2_start <- chr_data_2[chr_data_2$start > my_start]
chr_data_2_end <- chr_data_2_start[chr_data_2_start$end < my_end,]
max_value_2 <- max(chr_data_2_end$value)

### Generate Data Tracks
# Type 'a' is a line plot of the column-wise average values
### SRR1523657 (1)
data_track_1_for <- DataTrack(range = bedfile_1_for, type = "a", chromosome = my_chr,
                             genome = my_genome, name = "SRR1523657 \n Forward",
                             fill = "#B2182B", col = "black", ylim = c(0, max_value_1))
data_track_1_rev <- DataTrack(range = bedfile_1_rev, type = "a", chromosom = my_chr,
                             genome = my_genome, name = "SRR1523657 \n Reverse",
                             fill = "#2166AC", col = "black", ylim = c(max_value_1, 0))
data_track_1_both <- DataTrack(range = bedfile_1_both, type = "a", chromosome = my_chr,
                              genome = my_genome, name = "SRR1523657 \n Both",
                              fill = "#006400", col = "black")
### SRR1523666 (2)
data_track_2_for <- DataTrack(range = bedfile_2_for, type = "a", chromosome = my_chr,
                             genome = my_genome, name = "SRR1523666 \n Forward",
                             fill = "#B2182B",col = "black", ylim = c(0, max_value_2))
data_track_2_rev <- DataTrack(range = bedfile_2_rev, type = "a", chromosom = my_chr,
                             genome = my_genome, name = "SRR1523666 \n Reverse",
                             fill = "#2166AC", col = "black", ylim = c(max_value_2, 0))
data_track_2_both <- DataTrack(range = bedfile_2_both, type = "a", chromosome = my_chr,
                              genome = my_genome, name = "SRR1523666 \n Both",
                              fill = "#006400", col = "black")

### Generate genome and ideogram tracks
g_track<-GenomeAxisTrack(col="black")
i_track <- IdeogramTrack(genome = my_genome, chromosome = my_chr)

### Read in UCSC genes and track
ucsc_genes_1 <- UcscTrack(genome = my_genome, table = "ncbiRefSeq",
                         track = 'NCBI RefSeq', trackType="GeneRegionTrack",
                         chromosome = my_chr, rstarts = "exonStarts", rends = "exonEnds",
                         gene = "name", symbol = 'name', transcript = "name",
                         strand = "strand", stacking = 'pack', showID = T, geneSymbol = T,
                         fill = "#8B4789", col = "#8B4789")

z <- ranges(ucsc_genes_1)
mcols(z)$symbol <- mapIds(org.Hs.eg.db, gsub("\\.[1-9]$", "", mcols(z)$symbol), "SYMBOL","REFSEQ")
ucsc_genes_2 <- ucsc_genes_1
```

```
ranges(ucsc_genes_2) <- z

### Plot and export figure
### SRR1523657 (1)
tiff("SRR1523657_ngs3.tiff", units="in", width=7, height=5, res=300)
plotTracks(list(i_track, data_track_1_for, data_track_1_rev,
            data_track_1_both, g_track, ucsc_genes_2),
        collapseTranscripts = "meta", transcriptAnnotation = "symbol",
        from = my_start, to = my_end, sizes = c(0.05,0.15,0.15,0.15,0.15,0.4),
        type = "hist", col.histogram = NA, cex.title = 0.5, cex.axis = 0.4,
        axis = NA,title.width = 1, background.title="#8B8989")
dev.off()
### SRR1523666 (2)
tiff("SRR1523666_ngs3.tiff", units="in", width=7, height=5, res=300)
plotTracks(list(i_track, data_track_2_for, data_track_2_rev,
            data_track_2_both, g_track, ucsc_genes_2),
        collapseTranscripts = "meta", transcriptAnnotation = "symbol",
        from = my_start, to = my_end, sizes = c(0.05,0.15,0.15,0.15,0.15,0.4),
        type = "hist", col.histogram = NA, cex.title = 0.5, cex.axis = 0.4,
        axis = NA,title.width = 1, background.title="#8B8989")
dev.off()

### Calculate Log2 of max
max_value_1_log <- log2(max_value_1 + 1)
max_value_2_log <- log2(max_value_2 + 1)

# Create Tracks using new log parameters
### SRR1523657 (1)
data_track_1_for_log <- DataTrack(range = bedfile_1_for, type = "a", chromosome = my_chr,
                                  genome = my_genome, name = "SRR1523657 \n Log Forward",
                                  fill = "#B2182B", col = "black",
                                  ylim = c(0, max_value_1_log))
data_track_1_rev_log <- DataTrack(range = bedfile_1_rev, type = "a", chromosom = my_chr,
                                  genome = my_genome, name = "SRR1523657 \n Log Reverse",
                                  fill = "#2166AC", col = "black",
                                  ylim = c(max_value_1_log , 0))
data_track_1_both_log <- DataTrack(range = bedfile_1_both, type = "a", chromosome = my_chr,
                                   genome = my_genome, name = "SRR1523657 \n Log Both",
                                   fill = "#006400", col = "black")
### SRR1523666 (2)
data_track_2_for_log <- DataTrack(range = bedfile_2_for, type = "a", chromosome = my_chr,
                                  genome = my_genome, name = "SRR1523666 \n Log Forward",
                                  fill = "#B2182B", col = "black",
                                  ylim = c(0, max_value_2_log))
data_track_2_rev_log <- DataTrack(range = bedfile_2_rev, type = "a", chromosom = my_chr,
                                  genome = my_genome, name = "SRR1523666 \n Log Reverse",
                                  fill =  "#2166AC", col = "black",
                                  ylim = c(max_value_2_log , 0))
data_track_2_both_log <- DataTrack(range = bedfile_2_both, type = "a", chromosome = my_chr,
                                   genome = my_genome, name = "SRR1523666 \n Log Both",
                                   fill = "#006400", col = "black")

### Log Plots
### SRR1523657 (1)
tiff("SRR1523657_log_ngs3.tiff", units="in", width=7, height=5, res=300)
plotTracks(list(i_track, data_track_1_for_log, data_track_1_rev_log,
            data_track_1_both_log, g_track, ucsc_genes_2),
        collapseTranscripts = "meta", transcriptAnnotation = "symbol",
        from = my_start, to = my_end, sizes = c(0.05,0.15,0.15,0.15,0.15,0.4),
        type = "hist", col.histogram = NA, cex.title = 0.5, cex.axis = 0.4,
        axis = NA,title.width = 1, background.title="#8B8989",
        transformation=function(x){log2(x+1)})
dev.off()
### SRR1523666 (2)
tiff("SRR1523666_log_ngs3.tiff", units="in", width=7, height=5, res=300)
plotTracks(list(i_track, data_track_2_for_log, data_track_2_rev_log,
            data_track_2_both_log, g_track, ucsc_genes_2),
        collapseTranscripts = "meta", transcriptAnnotation = "symbol",
        from = my_start, to = my_end, sizes = c(0.05,0.15,0.15,0.15,0.15,0.4),
        type = "hist", col.histogram = NA, cex.title = 0.5, cex.axis = 0.4,
        axis = NA,title.width = 1, background.title="#8B8989",
        transformation=function(x){log2(x+1)})
dev.off()
```