Code for Neelang Parghi's ASI project

Neelang Parghi

May 2021

Trim RNA-seq data

```
#!/bin/bash
#SBATCH --array=1,2
#SBATCH --mail-type=END,FAIL # Mail events (NONE, BEGIN, END, FAIL, ALL)
#SBATCH --mail-user=np788@nyu.edu # Where to send mail
#SBATCH --ntasks=4
#SBATCH --mem=32gb # Job memory request
#SBATCH --time=12:00:00 # Time limit hrs:min:sec

module load trimgalore/0.5.0
module load python/cpu/2.7.15-ES

trim_galore --paired --length 30 -o /gpfs/scratch/np788/project4/
rna/trimmed /gpfs/scratch/np788/project4/rna/
K562_Control_RNA_Rep${SLURM_ARRAY_TASK_ID}_R1.fastq.gz

/gpfs/scratch/np788/project4/rna/
K562_Control_RNA_Rep${SLURM_ARRAY_TASK_ID}_R2.fastq.gz

trim_galore --paired --length 30 -o /gpfs/scratch/np788/project4/
rna/trimmed /gpfs/scratch/np788/project4/rna/
K562_FOXM1_CRISPR_RNA_Rep${SLURM_ARRAY_TASK_ID}_R1.fastq.gz
/gpfs/scratch/np788/project4/rna/
K562_FOXM1_CRISPR_RNA_Rep${SLURM_ARRAY_TASK_ID}_R2.fastq.gz
```

Trim ChIP-seq data

```
#!/bin/bash
#SBATCH --array=1,2,3
#SBATCH --mail-type=END,FAIL # Mail events (NONE, BEGIN, END, FAIL, ALL)
#SBATCH --mail-user=np788@nyu.edu # Where to send mail
```

```
#SBATCH --ntasks=4
#SBATCH --mem=32gb # Job memory request
#SBATCH --time=12:00:00 # Time limit hrs:min:sec

module load trimgalore/0.5.0
module load python/cpu/2.7.15-ES

trim_galore --paired --length 30 -o /gpfs/scratch/np788/project4/
atac/trimmed
/gpfs/scratch/np788/project4/atac/
K562_ATACseq_${SLURM_ARRAY_TASK_ID}_R1.fastq.gz
/gpfs/scratch/np788/project4/atac/
K562_ATACseq_${SLURM_ARRAY_TASK_ID}_R2.fastq.gz
```

Trim ATAC-seq data

```
#!/bin/bash
#SBATCH --array=1,2,3
#SBATCH --mail-type=END,FAIL # Mail events (NONE, BEGIN, END, FAIL, ALL)
#SBATCH --mail-user=np788@nyu.edu # Where to send mail
#SBATCH --ntasks=4
#SBATCH --mem=32gb # Job memory request
#SBATCH --time=12:00:00 # Time limit hrs:min:sec

module load trimgalore/0.5.0
module load python/cpu/2.7.15-ES

trim_galore --paired --length 30 -o /gpfs/scratch/np788/project4/
atac/trimmed
/gpfs/scratch/np788/project4/atac/
K562_ATACseq_${SLURM_ARRAY_TASK_ID}_R1.fastq.gz
/gpfs/scratch/np788/project4/atac/
K562_ATACseq_${SLURM_ARRAY_TASK_ID}_R2.fastq.gz
```

Build HISAT2 index for RNA-seq

```
hisat2-build -p 16 /gpfs/scratch/np788/Homo_sapiens/UCSC/hg38/
Sequence/WholeGenomeFasta/genome.fa hg38
```

Align RNA-seq data, convert to sorted BAM format, and index

```
#!/bin/bash
#SBATCH --ntasks=4 # Run on a single CPU
#SBATCH --mem=160gb # Job memory request
#SBATCH --time=8:00:00 # Time limit hrs:min:sec
#SBATCH --array=1-2
#SBATCH -p cpu_short

module load samtools/1.9

# Align RNA data - HISAT2 OK

hisat2 -p 16 -x /gpfs/scratch/np788/project/hg38/hg38 \
-1 /gpfs/scratch/np788/project/rna/trimmed/
K562_Control_RNA_Rep$SLURM_ARRAY_TASK_ID\_R1_val_1.fq.gz \
-2 /gpfs/scratch/np788/project/rna/trimmed/
K562_Control_RNA_Rep$SLURM_ARRAY_TASK_ID\_R2_val_2.fq.gz \
-S /gpfs/scratch/np788/project/rna/
K562_Control_RNA_Rep$SLURM_ARRAY_TASK_ID.sam

hisat2 -p 16 -x /gpfs/scratch/np788/project/hg38/hg38 \
-1 /gpfs/scratch/np788/project/rna/trimmed/
K562_FOXM1_CRISPR_RNA_Rep$SLURM_ARRAY_TASK_ID\_R1_val_1.fq.gz \
-2 /gpfs/scratch/np788/project/rna/trimmed/
K562_FOXM1_CRISPR_RNA_Rep$SLURM_ARRAY_TASK_ID\_R2_val_2.fq.gz \
-S /gpfs/scratch/np788/project/rna/
K562_FOXM1_CRISPR_RNA_Rep$SLURM_ARRAY_TASK_ID.sam

#SAM to sorted BAM

samtools view -b -o /gpfs/scratch/np788/project/rna/
K562_Control_RNA_Rep$SLURM_ARRAY_TASK_ID.bam /gpfs/
scratch/np788/project/rna/K562_Control_RNA_Rep$SLURM_ARRAY_TASK_ID.sam

samtools sort -o /gpfs/scratch/np788/project/rna/
sorted_K562_Control_RNA_Rep$SLURM_ARRAY_TASK_ID.bam /gpfs/scratch/np788/
project/rna/K562_Control_RNA_Rep$SLURM_ARRAY_TASK_ID.bam

samtools index /gpfs/scratch/np788/project/rna/
sorted_K562_Control_RNA_Rep$SLURM_ARRAY_TASK_ID.bam

samtools view -b -o /gpfs/scratch/np788/project/rna/
```

```
K562_FOXM1_CRISPR_RNA_Rep$SLURM_ARRAY_TASK_ID.bam
/gpfs/scratch/np788/project/rna/
K562_FOXM1_CRISPR_RNA_Rep$SLURM_ARRAY_TASK_ID.sam

samtools sort –o /gpfs/scratch/np788/project/rna/
sorted_K562_FOXM1_CRISPR_RNA_Rep$SLURM_ARRAY_TASK_ID.bam
/gpfs/scratch/np788/project/rna/
K562_FOXM1_CRISPR_RNA_Rep$SLURM_ARRAY_TASK_ID.bam

samtools index /gpfs/scratch/np788/project/rna/
sorted_K562_FOXM1_CRISPR_RNA_Rep$SLURM_ARRAY_TASK_ID.bam
```

Convert RNA-seq BAM files to R format

```
module load r/intel/3.6.0
R
library(Rsubread)

read_counts = featureCounts(files=dir(pattern=”bam”),
annot.inbuilt=”hg38”,
isPairedEnd=TRUE)
nrow(read_counts$counts)
save(read_counts, file = ”inbuilt_mapq20.RData”)
```

RNA-seq analysis/visualization in R

```
‘‘‘{r setup, include=FALSE}
knitr::opts_chunk$set(echo = TRUE)
‘‘‘

‘‘‘{r, echo = TRUE}
### LOAD REQUIRED LIBRARIES
library(”DESeq2”)
library(”pheatmap”)
library(”RColorBrewer”)
library(”vsn”)
library(”AnnotationDbi”)
library(”org.Hs.eg.db”)
library(”genefilter”)
library(”biomaRt”)
library(”IHW”)
library("annotate")
library("ggplot2")
library(DOSE)

setwd("~/Downloads/asi/project/rna")

load(file="inbuilt_mapq20.RData")
CountTable <- as.data.frame(read_counts$counts)
samples <- read.table("sample.txt", header=TRUE)
rownames(CountTable) <- as.character(row.names(CountTable))

dds <- DESeqDataSetFromMatrix(countData = CountTable, colData=samples,
design=~condition)

dds = DESeq(dds)
counts(dds)->raw_counts
raw_counts<-as.data.frame(counts(dds))


#Create a normalized matrix
norm_counts = counts(dds, normalized = TRUE)

(VST) that roughly mirrors how DeSeq2 models the data.
vsd1 <- varianceStabilizingTransformation(dds, blind=FALSE)

plotPCA(vsd1, "condition")

plotPCA(vsd1, c("condition","batch"))

using the limma::removeBatchEffect function
vsd2 <- varianceStabilizingTransformation(dds, blind=FALSE)
assay(vsd2) <- limma::removeBatchEffect(assay(vsd2), vsd2$batch)
plotPCA(vsd2, "condition")
data <- plotPCA(vsd2, "condition",returnData=TRUE)

sampleDists <- dist( t( assay(vsd1) ) )
sampleDists
sampleDistMatrix <- as.matrix( sampleDists )
colnames(sampleDistMatrix) <- NULL
colors <- colorRampPalette( rev(brewer.pal(9, "Reds")) )(255)
pheatmap(sampleDistMatrix, clustering_distance_rows=sampleDists,
clustering_distance_cols=sampleDists, col=colors)

# corrected
sampleDistsCorr <- dist( t( assay(vsd2) ) )
sampleDistsCorr
```

~~~~
sampleDistCorrMatrix <- as.matrix( sampleDistsCorr )
colnames(sampleDistCorrMatrix) <- NULL
colors <- colorRampPalette( rev(brewer.pal(9, "Blues")) )(255)
pheatmap(sampleDistCorrMatrix, clustering_distance_rows=sampleDists,
clustering_distance_cols=sampleDists, col=colors)




data <- plotPCA(vsd2, returnData=TRUE)
percentVar <- round(100 * attr(data, "percentVar"))
ggplot(data, aes(PC1, PC2, color=condition)) +
  theme_bw() +
  theme(panel.background = element_blank())+
  theme(panel.border = element_rect(colour = "black", fill=NA, size=1))+
  geom_point(size=5)+
  scale_color_manual(values=c("#74c476","#ec7014","purple"))+
  xlab(paste0("PC1:␣",percentVar[1],"%␣variance")) +
  ylab(paste0("PC2:␣",percentVar[2],"%␣variance")) +
  theme(plot.title = element_blank())+
  theme(text = element_text(size=20))+
  theme(legend.position="bottom") +
  theme(axis.text = element_text( color = "black", size = 25))+
  theme(legend.title=element_blank())

```

```{r BASIC DGE ANALYSIS USING DESEQ2}

# Run DESEQ and generate a simple plot showing the distribution
of regulated and unregulated genes

DatasetProcessed <- DESeq(dds) # runs DESEQ
par(mfrow=c(1,1))
#plotMA(DatasetProcessed, main="DESeq2", ylim=c(-5,5))

DatasetProcessed$condition <- relevel(DatasetProcessed$condition,
"Ctrl")

res1 <- lfcShrink(DatasetProcessed, coef=2)


baseMeanCtrl = rowMeans(counts(DatasetProcessed,normalized=TRUE)
[,DatasetProcessed$condition == "Ctrl"])
baseMeanCspr = rowMeans(counts(DatasetProcessed,normalized=TRUE)
[,DatasetProcessed$condition == "Cspr"])
~~~~

~~~~
res1 = cbind(as.data.frame(res1), baseMeanCtrl, baseMeanCspr)


res1$entrez <- getSYMBOL(row.names(res1), data='org.Hs.eg')

write.csv(res1, "DGEanalysis.csv", row.names=TRUE)
```



## Differential analysis
```{r DE analysis with DESEQ2}


Ctrl_Cspr<-res1
head(`Ctrl_Cspr`)
dim(`Ctrl_Cspr`)

adj_p_val = 0.05
abs_log2fc = 1

Ctrl_Cspr_sig = res1[res1$padj < adj_p_val & !is.na(res1$padj) &
abs(res1$log2FoldChange) >abs_log2fc ,]
head(`Ctrl_Cspr_sig`)
dim(`Ctrl_Cspr_sig`)



rna_df<-as.data.frame(`Ctrl_Cspr`)
rna_df$log10.pvalue<-(-1*log10(rna_df$padj))
rna_df$Significant.Gene<-"No"

row_number<-which(rna_df$padj<adj_p_val & abs(rna_df$log2FoldChange)>
abs_log2fc)
rna_df$Significant.Gene[row_number]<-"Yes"
rna_df$Differential_Gene<-rna_df$Significant.Gene
rna_df$Differential_Gene[which(rna_df$padj< adj_p_val &
rna_df$log2FoldChange> abs_log2fc)]<-"Up-Regulated"
rna_df$Differential_Gene[which(rna_df$padj< adj_p_val &
rna_df$log2FoldChange< -abs_log2fc)]<-"Down-Regulated"

```


## Heatmap with DEGs
~~~~

~~~~
```{r Heatmap}
sig.genes<-rownames(`Ctrl_Cspr_sig`)

#extract raw counts for these genes:
m<-match(sig.genes,rownames(norm_counts))
de_norm_counts<-norm_counts[m,]


library(pheatmap)
library(RColorBrewer)

sample<-as.data.frame(data$name)
colnames(sample)<-c("samples")
rownames(sample)<-colnames(de_norm_counts)
sample$samples<-as.character(data$condition)
sample2<-sample
sample2$samples[which(sample2$samples=="Cspr")]<-"#74c476"
sample2$samples[which(sample2$samples=="Ctrl")]<-"#ec7014"
sample3 = list(samples = c("Cspr"= "#74c476", "Ctrl"="#ec7014"))

pheatmap(de_norm_counts,scale = "row",show_rownames=FALSE,
         colorRampPalette(rev(brewer.pal(n = 10, name ="RdBu")))(25),
         clustering_method = "ward.D2",
         clustering_distance_cols ="euclidean",
         annotation_col  = sample)#, annotation_colors = sample3[1])
#dev.off()

```

```{r}
library(enrichplot)
options(ggrepel.max.overlaps = 9999)
#x = enrichDO(row_number, ont = "DO", pvalueCutoff = 0.05,
pAdjustMethod = "BH", universe, minGSSize = 10, maxGSSize = 500,
qvalueCutoff = 0.2, readable = FALSE)

x = enrichDO(row_number)

cnetplot(x, showCategory = 4)
```
~~~~

ATAC-seq alignment, SAM → BAM conversion, peak calling (broad and narrow), and peak annotation

```
#!/bin/bash
#SBATCH --ntasks=4 # Run on a single CPU
#SBATCH --mem=160gb # Job memory request
#SBATCH --time=8:00:00 # Time limit hrs:min:sec
#SBATCH --array=1-3
#SBATCH -p cpu_short

module load macs/1.4.2
module load bedtools
module load samtools/1.3
module load deeptools
module load homer

# Align ATAC data (array 1-3)
bowtie2 -x /gpfs/scratch/np788/Homo_sapiens/UCSC/hg38/Sequence/
Bowtie2Index/genome \
-1 /gpfs/scratch/np788/project/atac/trimmed/
K562_ATACseq_$SLURM_ARRAY_TASK_ID\_R1_val_1.fq.gz \
-2 /gpfs/scratch/np788/project/atac/trimmed/
K562_ATACseq_$SLURM_ARRAY_TASK_ID\_R2_val_2.fq.gz \
-S /gpfs/scratch/np788/project/atac/
K562_ATACseq_$SLURM_ARRAY_TASK_ID.sam

# SAM to sorted BAM

samtools view -b -o /gpfs/scratch/np788/project/atac/
K562_ATACseq_$SLURM_ARRAY_TASK_ID.bam
/gpfs/scratch/np788/project/atac/
K562_ATACseq_$SLURM_ARRAY_TASK_ID.sam

samtools sort -o /gpfs/scratch/np788/project/atac/
sorted_K562_ATACseq_$SLURM_ARRAY_TASK_ID.bam
/gpfs/scratch/np788/project/atac/
K562_ATACseq_$SLURM_ARRAY_TASK_ID.bam

samtools index /gpfs/scratch/np788/project/atac/
sorted_K562_ATACseq_$SLURM_ARRAY_TASK_ID.bam
```

```
multiBamSummary bins --bamfiles sorted_K562_ATACseq_1.bam
sorted_K562_ATACseq_2.bam sorted_K562_ATACseq_3.bam -o results.npz

# Merge replicate peak files. Sort per
#https://github.com/hbctraining/Intro-to-ChIPseq/blob/master/
#lessons/data_visualization_with_bedtools.md

# BROAD
macs2 callpeak -t /gpfs/scratch/np788/project/atac/
sorted_K562_ATACseq_$SLURM_ARRAY_TASK_ID.bam --broad -f BAMPE
-n=atac$SLURM_ARRAY_TASK_ID --outdir /gpfs/scratch/np788/
project/atac/callpeak_broad

cat atac1_peaks.broadPeak atac2_peaks.broadPeak atac3_peaks.broadPeak >
atac_broad_combined.broadPeak

sort -k1,1 -k2,2n atac_broad_combined.broadPeak | bedtools merge -i - >
atac_broad_combined.bed

annotatePeaks.pl /gpfs/scratch/np788/project/atac/callpeak_broad/
atac_broad_combined.broadPeak hg38 >
/gpfs/scratch/np788/project/atac/callpeak_broad/
atac_broad_combined_broadPeak.txt

annotatePeaks.pl /gpfs/scratch/np788/project/atac/callpeak_broad/
atac_broad_combined.bed hg38 > /gpfs/scratch/np788/project/atac/
callpeak_broad/atac_broad_combined_bed.txt



# NARROW
macs2 callpeak -t /gpfs/scratch/np788/project/atac/
sorted_K562_ATACseq_$SLURM_ARRAY_TASK_ID.bam -f BAMPE
-n=atac$SLURM_ARRAY_TASK_ID --outdir /gpfs/scratch/np788/project/
atac/callpeak_narrow

cat atac1_peaks.narrowPeak atac2_peaks.narrowPeak
atac3_peaks.narrowPeak > atac_narrow_combined.narrowPeak

sort -k1,1 -k2,2n atac_narrow_combined.narrowPeak |
bedtools merge -i - > atac_narrow_combined.bed

annotatePeaks.pl /gpfs/scratch/np788/project/atac/callpeak_narrow/
atac_narrow_combined.narrowPeak hg38 >
/gpfs/scratch/np788/project/atac/callpeak_narrow/
atac_narrow_combined_narrowPeak.txt
```

```
annotatePeaks.pl /gpfs/scratch/np788/project/atac/callpeak_narrow/
atac_narrow_combined.bed hg38 >
/gpfs/scratch/np788/project/atac/callpeak_narrow/
atac_narrow_combined_bed.txt
```

ChIP-seq alignment, SAM → BAM conversion, peak calling, and peak annotation

```
#!/bin/bash
#SBATCH --ntasks=4 # Run on a single CPU
#SBATCH --mem=160gb # Job memory request
#SBATCH --time=8:00:00 # Time limit hrs:min:sec
#SBATCH --array=1-2
#SBATCH -p cpu_short

module load bowtie2/2.3.4.1
module load samtools/1.3
module load bedtools/2.26.0
module load ucscutils/374 # contains bedGraphToBigWig
module load macs/1.4.2
module load homer


bowtie2 -x /gpfs/scratch/np788/Homo_sapiens/UCSC/hg38/Sequence/
Bowtie2Index/genome
-1 /gpfs/scratch/np788/project/chip/trimmed/
K562_FOXM1_ChIP_$SLURM_ARRAY_TASK_ID\_R1_val_1.fq.gz \
-2 /gpfs/scratch/np788/project/chip/trimmed/
K562_FOXM1_ChIP_$SLURM_ARRAY_TASK_ID\_R2_val_2.fq.gz \
-S /gpfs/scratch/np788/project/chip/prac9/
K562_FOXM1_ChIP_$SLURM_ARRAY_TASK_ID.sam


samtools view -bSo /gpfs/scratch/np788/project/chip/prac9/
K562_FOXM1_ChIP_$SLURM_ARRAY_TASK_ID.bam
/gpfs/scratch/np788/project/chip/prac9/
K562_FOXM1_ChIP_$SLURM_ARRAY_TASK_ID.sam

samtools sort /gpfs/scratch/np788/project/chip/prac9/
K562_FOXM1_ChIP_$SLURM_ARRAY_TASK_ID.bam >
/gpfs/scratch/np788/project/chip/prac9/
sorted_K562_FOXM1_ChIP_$SLURM_ARRAY_TASK_ID.bam
```

```
samtools index /gpfs/scratch/np788/project/chip/prac9/
sorted_K562_FOXM1_ChIP_$SLURM_ARRAY_TASK_ID.bam

multiBamSummary bins --bamfiles sorted_K562_FOXM1_ChIP_1.bam
sorted_K562_FOXM1_ChIP_2.bam -o results.npz
plotCorrelation --corData results.npz --corMethod spearman
--skipZeros --plotTitle "Spearman␣Correlation␣of␣Read␣Counts"
--whatToPlot heatmap --colorMap RdYlBu --plotNumbers -o
heatmap_SpearmanCorr_readCounts.png --outFileCorMatrix
SpearmanCorr_readCounts.tab

macs2 callpeak -t /gpfs/scratch/np788/project/atac/
sorted_K562_ATACseq_$SLURM_ARRAY_TASK_ID.bam -f BAMPE
-n=atac$SLURM_ARRAY_TASK_ID --outdir /gpfs/scratch/np788/
project/atac/callpeak_narrow

cat /gpfs/scratch/np788/project/chip/callpeak/chip1_peaks.narrowPeak
/gpfs/scratch/np788/project/chip/callpeak/chip2_peaks.narrowPeak >
/gpfs/scratch/np788/project/chip/callpeak/chip_combined.narrowPeak

sort -k1,1 -k2,2n /gpfs/scratch/np788/project/chip/callpeak/
chip_combined.narrowPeak | bedtools merge -i - >
/gpfs/scratch/np788/project/chip/callpeak/chip_combined.bed


annotatePeaks.pl /gpfs/scratch/np788/project/chip/callpeak/
chip_combined.bed hg38 >
/gpfs/scratch/np788/project/chip/callpeak/chip_combined_bed.txt

annotatePeaks.pl /gpfs/scratch/np788/project/chip/callpeak/
chip_combined.narrowPeak hg38 >
/gpfs/scratch/np788/project/chip/callpeak/chip_combined_narrowpeak.txt
```

Find intersecting peaks from ATAC-seq and ChIP-seq peak calling data

```
#!/bin/bash
#SBATCH --ntasks=4 # Run on a single CPU
#SBATCH --mem=160gb # Job memory request
#SBATCH --time=8:00:00 # Time limit hrs:min:sec
#SBATCH --array=1-2
#SBATCH -p cpu_short
```

```
module load hisat2/2.1.0

#Build HISAT2 index
#hisat2-build -p 16 /gpfs/scratch/np788/Homo_sapiens/UCSC/hg38/
Sequence/WholeGenomeFasta/genome.fa hg38

bedtools intersect -a /gpfs/scratch/np788/project/atac/callpeak_broad/
atac_broad_combined.bed -b /gpfs/scratch/np788/project/chip/
callpeak/chip_combined.bed > /gpfs/scratch/np788/project/
broad_int.bed

annotatePeaks.pl /gpfs/scratch/np788/project/broad_int.bed hg38 >
/gpfs/scratch/np788/project/broad_int.txt

bedtools intersect -a /gpfs/scratch/np788/project/atac/callpeak_narrow/
atac_narrow_combined.bed -b /gpfs/scratch/np788/project/chip/
callpeak/chip_combined.bed > /gpfs/scratch/np788/project/narrow_int.bed

annotatePeaks.pl /gpfs/scratch/np788/project/narrow_int.bed hg38 >
/gpfs/scratch/np788/project/narrow_int.txt
```

Find intersections for all three datasets and find significant GO terms and transcription factors

```
library(gprofiler2)

# Q2

atac_narrow = read.csv("~/Downloads/asi/project/atac/callpeak_narrow/
atac_narrow_combined_bed.csv")

atac_broad = read.csv("~/Downloads/asi/project/atac/callpeak_broad/
atac_broad_combined_bed.csv")

rna = read.csv("~/Downloads/asi/project/rna/rna_df.csv")
chip = read.csv("~/Downloads/asi/project/chip/callpeak/
chip_combined_bed.csv")

#broad1 = intersect(as.character(rna$X), rownames(atac_broad))
broad1 = intersect(rna$X, atac_broad$Entrez.ID)
broad2 = intersect(broad1, chip$Entrez.ID)
length(broad2)
```

```
narrow1 = intersect(rna$X, atac_narrow$Entrez.ID)
narrow2 = intersect(narrow1, chip$Entrez.ID)
length(narrow2)


write.table(broad2, ”˜/Downloads/asi/project/broad.csv”, sep = ”,”,
row.names = FALSE, col.names = FALSE)

write.table(narrow2, ”˜/Downloads/asi/project/narrow.csv”, sep = ”,”,
row.names = FALSE, col.names = FALSE)

gostres <- gost(query = broad2,
                organism = ”hsapiens”, ordered_query = FALSE,
                multi_query = FALSE, significant = TRUE,
                exclude_iea = FALSE,
                measure_underrepresentation = FALSE, evcodes = FALSE,
                user_threshold = 0.05, correction_method = ”fdr”,
                domain_scope = ”annotated”, custom_bg = NULL,
                numeric_ns = ””, sources = NULL, as_short_link = FALSE)

gostplot(gostres, capped = FALSE, interactive = TRUE)


# Q3

broad_int = read.csv(”˜/Downloads/asi/project/broad_int.csv”)
broad_rna = intersect(rna$X, broad_int$Entrez.ID)
length(broad_rna)

narrow_int = read.csv(”˜/Downloads/asi/project/narrow_int.csv”)
narrow_rna = intersect(rna$X, narrow_int$Entrez.ID)
length(narrow_rna)

write.table(broad_rna, ”˜/Downloads/asi/project/broad_rna.csv”,
sep = ”,”, row.names = FALSE, col.names = FALSE)
write.table(narrow_rna, ”˜/Downloads/asi/project/narrow_rna.csv”,
sep = ”,”, row.names = FALSE, col.names = FALSE)

gostres <- gost(query = narrow_rna,
                organism = ”hsapiens”, ordered_query = FALSE,
                multi_query = FALSE, significant = TRUE,
                exclude_iea = FALSE,
                measure_underrepresentation = FALSE, evcodes = FALSE,
                user_threshold = 0.05, correction_method = ”fdr”,
                domain_scope = ”annotated”, custom_bg = NULL,
                numeric_ns = ””, sources = NULL, as_short_link = FALSE)
```

```r
gostplot(gostres, capped = FALSE, interactive = TRUE)
```