Bulk RNA-seq Workflow

```
#View reference and FASTQ files.
#FASTQ files are from the Griffith lab.
#https://github.com/griffithlab/rnaseq_tutorial/wiki/RNAseq-Data
seqkit stats hg38.fasta
cat hg38.fasta | head -5
seqkit stats *.fastq.gz

#Make file of root names.
cat > rootNames.txt
brain1
brain2
brain3
cancer1
cancer2
cancer3

#Align sequencing reads to genome using hisat2.
mkdir -p bam

cat rootNames.txt | parallel "hisat2 -x ../Genomics/OptimizedPipeline/hg38 -1 {}.read1.fastq.gz -2
{}.read2.fastq.gz | samtools sort > bam/{}.bam"

#Index BAM files.

cat rootNames.txt | parallel "samtools index bam/{}.bam"

#Load BAM files into IGV.

#Count features, remembering that data is stranded.
cat rootNames.txt | parallel -j 1 echo 'bam/{}.bam' | xargs featureCounts -p -s 1 -a
../Genomics/OptimizedPipeline/hg38.ensGene.gtf -o counts.csv


#Perform transcript classification analysis.
#Activate salmon environment.
conda activate salmon_env

#Build salmon index.
salmon index -t GRCh38_latest_rna.fna -i salmon.idx

#Run salmon quantification.
cat rootNames.txt | parallel -j 4 "salmon quant -i salmon.idx -l A --validateMappings -1
{}.read1.fastq.gz -2 {}.read2.fastq.gz -o salmon/{}"

#Save quant.sf files as TSV files. Then open as spreadsheets and copy and paste each NumReads
#column into a new spreadsheet to make one file of counts (salmonCounts.csv).
```

#Run edgeR on counts files to perform differential expression analysis. I will use code from the
#“Biostars Handbook” to run edgeR.
#First for hisat2 featureCounts data.
Rscript code/edger.r

#This outputs a file entitled results.csv that includes FDR of genes.

#Next, for salmon data.
Rscript code/edger_salmon.r

#This outputs a file entitled salmon_results.csv that includes FDR of genes.

#Make heatmap of hisat2 featureCounts data with FDR < .05.
Rscript code/create_heatmap.r

#Brain and cancer samples segregate very well.

#Make heatmap of salmon data with FDR < .05.
Rscript code/create_salmon_heatmap.r

#The salmon heatmap has more genes with FDR < .05. It segregates well by group, although there are
#two or three genes that do not segregate that well.

#Calculate transcript integrity number (TIN) as a QC step. Use tin.py to calculate TIN for each
#transcript using hisat2 BAM files.
python tin.py -i bam -r hg38_RefSeq.bed

#This outputs a .tin.xls file for each BAM file, as well as a .summary.txt file giving the average TIN for
#that BAM file. I will only use genes that have TIN > 60.

#The next step is to perform gene set enrichment analysis (GSEA) on genes with FDR < .05 and
#TIN > 60 using Panther and g:Profiler (g:GOSt). These tools will return gene ontology (GO)
#annotations that are statistically over-represented compared to background.