RNAseq analysis methods

Prerequisites:

Python 2.7

R

EdgeR (installation described below within R)

HISAT2 v2.1.0 https://ccb.jhu.edu/software/hisat2/index.shtml

StringTie https://ccb.jhu.edu/software/stringtie/index.shtml#install

prepDE.py https://ccb.jhu.edu/software/stringtie/dl/prepDE.py

Samtools http://www.htslib.org/download/

Goal:

Determine which genes are differentially expressed in a human dataset. For computational ease. I have limited the dataset to RNAseq data from chr22 of the human genome. The reads have already been de-multiplexed, adapter trimmed, and quality trimmed. The necessary reference files (gtf and fasta) have already been downloaded and are available in the data folder.

The files were downloaded from:

Fasta file for chr22 is found on this page:
http://hgdownload.cse.ucsc.edu/goldenPath/hg38/chromosomes/

Gtf file for whole genome is found here:
ftp://ftp.ensembl.org/pub/release-81/gtf/homo_sapiens

I limited the analysis to genes found on chromosome 22, so I used this command to pull out those genes: grep ^22 humangenome.gtf > chr22.gtf

A bit more about gtf format:
http://www.ensembl.org/info/website/upload/gff.html

Worksheet:

1. Download all data from the UM BOX.

2. Unzip data.tar.gz, if it wasn't automatically unzipped by your computer

   tar -xzf data.tar.gz

3. Unzip reference files, if it wasn't automatically unzipped by your computer

   mkdir ref

```
gunzip chr22.fa.gz
mv chr22.fa ref/
gunzip chr22_genes.gtf.gz
mv chr22_genes.gtf ref/
```

4. Index the reference using hisat2 build (Note: hisat2 executables must be in your PATH or specify the entire path to the executable)

```
cd ref
hisat2-build chr22.fa chr22
cd ..
```

5. Align the reads to the reference fasta using hisat2. Note: hisat2 must be in your PATH. First it is necessary to make the directory that all data will be written to, note: this directory only needs to be created once.

```
mkdir alignments
hisat2 -f -x ref/chr22 -1 data/sample_01_1.fasta -2 data/sample_01_2.fasta -S
alignments/sample01.sam
```

6. Now that you have a bam file, you will want to check the mapping quality. Note: The output of samtools idxstats is tab-delimited with each line consisting of reference sequence name, sequence length, # mapped reads and # unmapped reads.

   Use samtools view to convert the SAM file into a BAM file

```
cd alignments
samtools view -bSh -o sample01.bam sample01.sam
```

   Use samtools sort to convert the BAM file to a sorted BAM file.

```
samtools sort -o sample01.sort.bam sample01.bam
samtools index sample01.sort.bam
samtools idxstats sample01.sort.bam
```

7. Next use stringtie to generate a gtf for each sample for each gene in the reference annotation set on a per individual basis, this was an unstranded library preparation (it's important to know how your data was generated), -e limits matches to the specified reference annotation file

```
cd ..
mkdir ballgown
```

```
stringtie alignments/sample01.sort.bam -G ref/chr22_genes.gtf -e > ballgown/sample01.gtf
```

8. Repeat for all samples. Let's write a shell script that will do this for us! The shell script that works on my mac is below, modifications may be required, depending on your operating system. I used vi to paste this script into a file called script.sh.

```
for i in 0{1..9} {10..20}
do
echo ${i}
hisat2 -f -x ref/chr22 -1 data/sample_${i}_1.fasta -2 data/sample_${i}_2.fasta -S
alignments/sample${i}.sam
samtools view -bSh -o alignments/sample${i}.bam alignments/sample${i}.sam
samtools sort -o alignments/sample${i}.sort.bam alignments/sample${i}.bam
samtools index alignments/sample${i}.sort.bam
samtools idxstats alignments/sample${i}.sort.bam > alignments/sample${i}.stats
stringtie alignments/sample${i}.sort.bam -G ref/chr22_genes.gtf -e -B -o
ballgown/${i}/sample${i}.gtf
done
```

   At this point, you will have 19 bam files, one for each individual

9. We want to generate counts data for each individual. There are many ways to do this, we will use prepDE.py, which is a python script provided with stringtie.

```
prepDE.py
```

10. We'll now move to R for the remainder of our analyses, I have posted the code on github

    https://github.com/jokelley/congen-2019/blob/master/RNAseqAnalysisMethods.md