Homework 5
BIOS-7659/CPBS-7659
Due 10/23 in class

1. Next Generation Sequencing: Sample Size Estimates

   - Install the `RNASeqPower` package from BioConductor.
   - Install the `edgeR` and `cqn` packages from BioConductor.
   - From `cqn`, use `data(montgomery.subset)` to load human RNA-seq data from lymphoblastoid cells from 10 subjects (see Montgomery et al., *Nature* 464:773-777). Use `data(uCovar)` to load the GC content and length of the genes in this data set.

   (a) Using `rnapower()`, recreate Figure 3 from the journal club paper using : Hart et al., (2013) "Calculating sample size estimates for RNA sequencing data." *J Comput Biol.* 20:970-8. What does this figure show?

   (b) For the Montgomery data, create a row in Table 1 in the Hart et al. paper. Note that genes that have zero counts in all 10 samples were already excluded (see `help(montgomery.subset)`), so "% mapped", will be 100%. How does the Montgomery data compare to the other data sets in Table 1?

   (c) Calculate the biological coefficient of variations (CV) from the Montgomery human lymphoblastoid data (use the function `estimateTagwiseDisp()` in edgeR, see problem 3). Plot the histogram and empirical cdf (use `ecdf()` function) and report the median and 90% percentile. How do the CVs compare with the examples in the Hart et al., paper in Figure 2?

   Hint: Note that Hart et al., "estimated the coefficient of variation (CV) in expression across samples in the data set using a negative binomial model (edgeR)." See the section "Biological coefficient of variation" in `http://www.ncbi.nlm.nih.gov/pmc/articles/PMC3378882/` for a description on how the CVs are obtained and the edgeR User's Guide, see section 2.8.2-2.8.3 and 2.9.1

   `http://bioconductor.org/packages/release/bioc/vignettes/edgeR/inst/doc/edgeRUsersGuide.pdf`

   (d) Using `rnapower()`, recreate Figure 3 from Hart et al. again but with two curves using the median and the 90% percentile CV across genes for the Montgomery data. What sample size do your recommend?

   (e) Using `rnapower()`, recreate the curve (not the histogram) in the top Figure 4 from the Hart et al., paper. If you cannot recreate the figure, please explain any differences.

2. Next Generation Sequencing: Pre-Processing

- Install the `yeastRNASeq` and `EDASeq` packages from BioConductor. Familiarize yourself with these packages by looking at the manuals and the reference Risso et al. (2011) "GC-Content Normalization for RNA-Seq Data." *BMC Bioinformatics* 12(1), pp. 480.
- Use `data(geneLevelData)` to load the `geneLevelData` data, which is a yeast RNA-seq data set on two mutant and two wildtype strains (see Lee A et al. (2008) *PLoS Genet* 4(12): e1000299). Use `data(yeastGC)` to also load the GC content and length (`data(yeastLength)`) of annotated yeast genes.

(a) Within `geneLevelData`, how many genes have all zeros as counts? How many have at least one sample with a zero?

- Save a new object `geneLevelDataFilter` only containing genes with ≥ 10 counts (summed over all samples).
- Based on `geneLevelDataFilter`, run the following code to create a `SeqExpressionSet` object for the `EDASeq` functions.

```
exprs = as.matrix(geneLevelDataFilter) # matrix of counts
sub = intersect(rownames(geneLevelDataFilter), names(yeastGC))
exprs = exprs[sub,] #only examine genes with annotated GC content/length
row.names(exprs) = NULL #remove row and column names
colnames(exprs) = NULL
#Create SeqExpressionSet, which contains counts, labels for the
#samples and GC content/length
counts = newSeqExpressionSet(counts=exprs,
phenoData=data.frame(conditions = colnames(geneLevelDataFilter)),
featureData=AnnotatedDataFrame(data.frame(gc=yeastGC[sub],
length = yeastLength[sub])))
```

(b) For the following plots, use the log scale (look at the manual/help for options to use the log scale).

- To plot the counts by sample, use the `boxplot()` function from the `EDASeq` package on `counts`. Do you see a need for normalization?

- To plot the mean by variance plot, use `meanVarPlot()`. What trends do you see?

- To assess any biases by GC content, use `biasPlot()`. What trends do you see?

- To assess any biases by length, use `biasPlot()`. What trends do you see?

(c) Apply `withinLaneNormalization()` to normalize by GC content. See help file and try different methods for normalization with the "which" option. Describe the different methods. Use `biasPlot()` before and after normalization. How do the methods compare? Save the normalized results in a new object to use in the next part.

(d) Using the within-lane normalized data from the previous part, apply `betweenLaneNormalization()` to normalize across samples. Try different methods for normalization with the "which" option. Describe the different methods. Use `boxplot()` before and after normalization. How do the methods compare?