A Tutorial: Genome-based RNA-Seq Analysis Using the TUXEDO Package

The following data and software resources are required for following the tutorial.

Data:
ftp://ftp.broad.mit.edu/pub/users/bhaas/rnaseq_workshop/rnaseq_workshop_data.tgz

Software requirements:

Bowtie
http://sourceforge.net/projects/bowtie-bio/files/bowtie/0.12.7/

TopHat (install **version 1.3.2**)
http://tophat.cbcb.umd.edu/downloads/

Cufflinks
http://cufflinks.cbcb.umd.edu/

Samtools
http://sourceforge.net/projects/samtools/files/samtools/0.1.18/samtools-0.1.18.tar.bz2/download

GenomeView
ftp://ftp.broad.mit.edu/pub/users/bhaas/rnaseq_workshop/genomeview_1951_package.tgz

R and **CummeRbund** (Bioconductor) installed:
http://www.r-project.org/

Install CummeRbund and like so:

```
source("http://bioconductor.org/biocLite.R")
biocLite("cummeRbund")
```

Align Illumina paired-end reads to the genome using TopHat (v1.3.2):

(~30 seconds each)

% tophat -I 1000 -i 20 -o condA_tophat_out genome condA.left.fa condA.right.fa

% tophat -I 1000 -i 20 -o condB_tophat_out genome condB.left.fa condB.right.fa

Run Cufflinks to assemble transcripts from the tophat alignments:

(~30 seconds each)

% cufflinks -o condA_cufflinks_out condA_tophat_out/accepted_hits.bam

% cufflinks -o condB_cufflinks_out condB_tophat_out/accepted_hits.bam

Merge separately assembled transcript structures into a cohesive set:

First, create a file that lists the names of the files containing the separately reconstructed transcripts, which can be done like so:

first writes the file
% echo condA_cufflinks_out/transcripts.gtf > assemblies.txt

writes in append mode to add the second filename
% echo condB_cufflinks_out/transcripts.gtf >> assemblies.txt

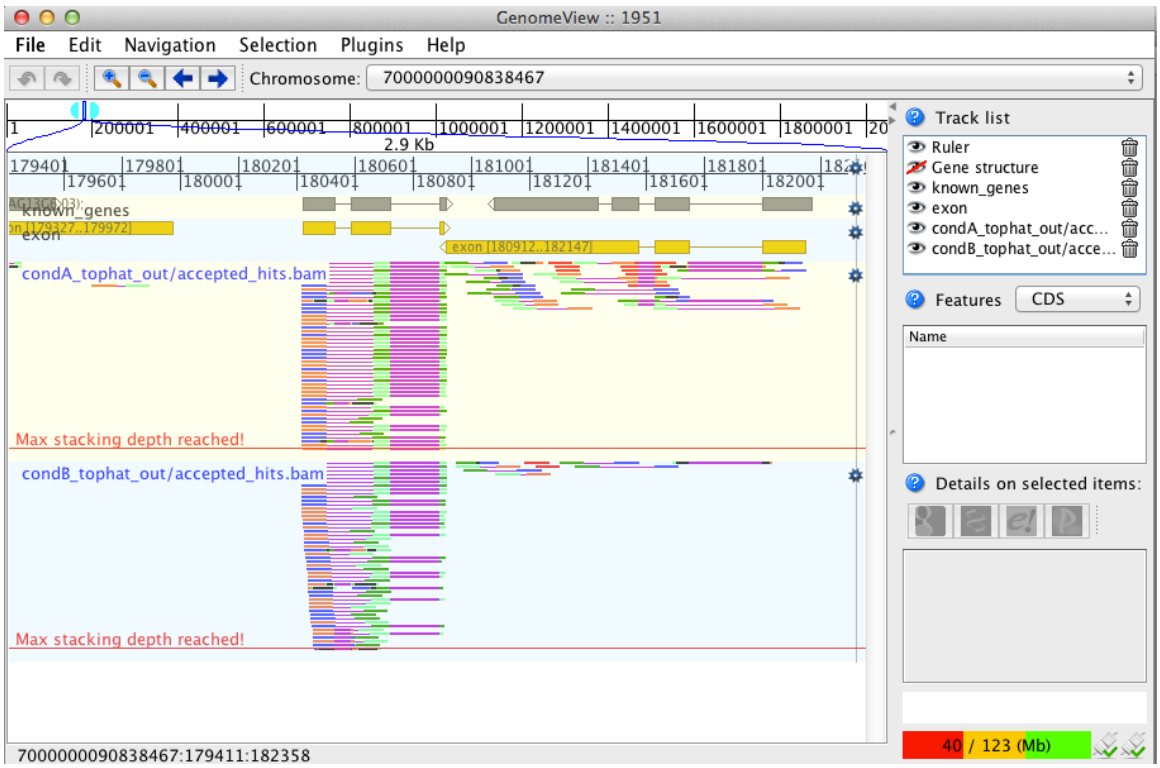
verify that this file now contains both filenames:
% cat assemblies.txt
condA_cufflinks_out/transcripts.gtf
condB_cufflinks_out/transcripts.gtf

And now we're ready to merge the transcripts using cuffmerge:

(~30 seconds)

% cuffmerge -s genome.fa assemblies.txt
View the reconstructed transcripts and the tophat alignments like so:

% java -jar $GENOMEVIEW/genomeview.jar genome.fa merged_asm/merged.gtf genes.bed condA_tophat_out/accepted_hits.bam condB_tophat_out/accepted_hits.bam



Pan the genome, examine the alignments, known genes and reconstructed genes.

Do the alignments agree with the known gene structures (ex. Intron placements)?

Do the cufflinks-reconstructed transcripts well represent the alignments?

Do the cufflinks-reconstructed transcripts match the structures of the known transcripts?

Differential expression analysis using cuffdiff and cummeRbund:

(~ 1 ½ minutes)

% cuffdiff -o diff_out -b genome.fa -L condA,condB -u merged_asm/merged.gtf condA_tophat_out/accepted_hits.bam condB_tophat_out/accepted_hits.bam

Examine the output files generated in the diff_out/ directory.

(the rest is interactive with little to no waiting time)

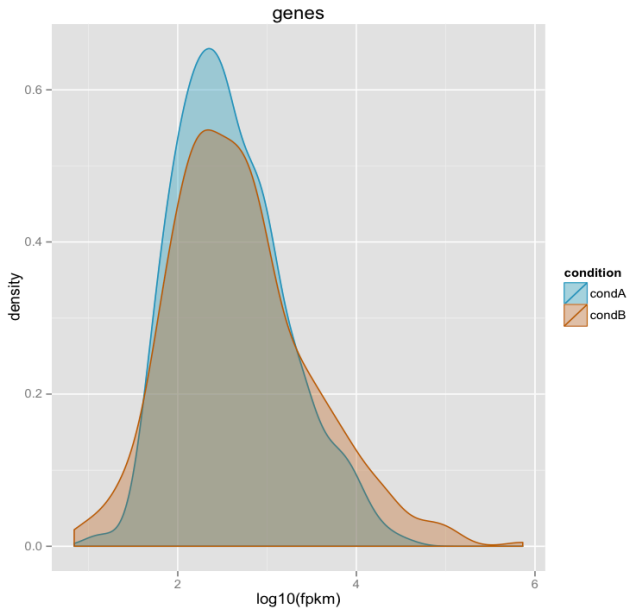
Use 'cummeRbund' to analyze the results from cuffdiff:

% R
(note, to exit R, type cntrl-D, or type "q()").

load the cummerbund library into the R session
> library(cummeRbund)

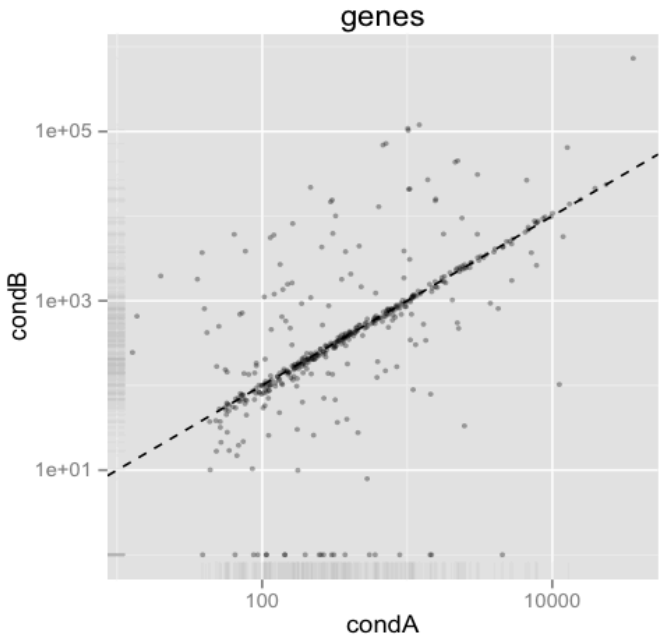
import the cuffdiff results
>cuff = readCufflinks('diff_out')

examine the distribution of expression values for the reconstructed transcripts
>csDensity(genes(cuff))



Examine transcript expression values in a scatter plot
Expression values are typically log-normally distributed. This is just a sanity check.

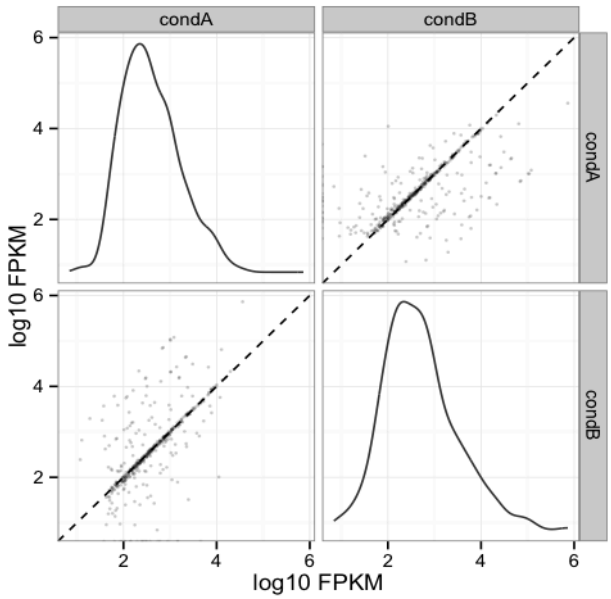
>csScatter(genes(cuff), 'condA', 'condB')



Strongly differentially expressed transcripts should fall far from the linear regression line.

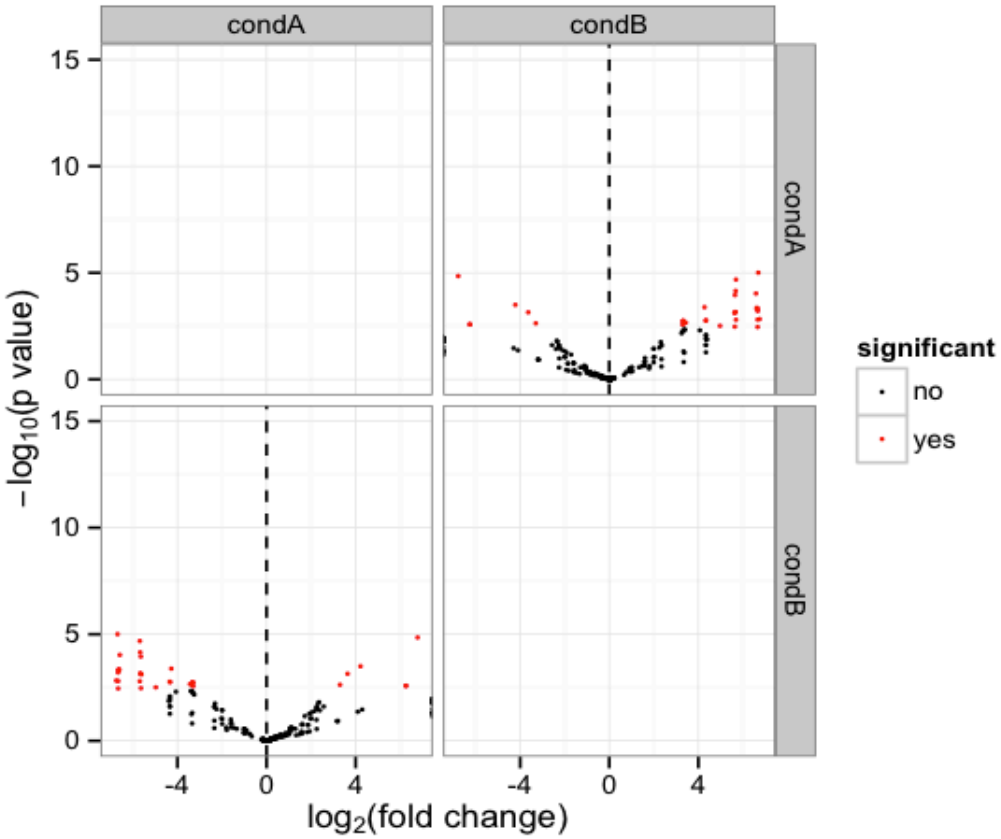
Examine individual densities and pairwise scatterplots together.

> csScatterMatrix(genes(cuff))



Volcano plots are useful for identifying genes most significantly differentially expressed.

> csVolcanoMatrix(genes(cuff), 'condA', 'condB')



Extract the 'genes' that are significantly differentially expressed (red points above)

retrieve the gene-level differential expression data
> gene_diff_data = diffData(genes(cuff))

how many 'genes'?
> nrow(gene_diff_data)

from the gene-level differential expression data, extract those that
are labeled as significantly different.
> sig_gene_data = subset(gene_diff_data, (significant == 'yes'))

how many?
> nrow(sig_gene_data)

Examine the entries at the top of the unsorted data table:

```
> head(sig_gene_data)
       gene_id sample_1 sample_2 status  value_1     value_2 log2_fold_change
11 XLOC_000011    condA    condB     OK  320.122  10051.2000          4.97261
24 XLOC_000024    condA    condB     OK  680.167  68932.0000          6.66314
29 XLOC_000029    condA    condB     OK 1211.090 119654.0000          6.62642
33 XLOC_000033    condA    condB     OK  112.935   5556.4400          5.62059
44 XLOC_000044    condA    condB     OK  102.436   1109.5000          3.43711
51 XLOC_000051    condA    condB     OK 1097.570     88.2133         -3.63717
   test_stat     p_value   q_value significant
11  -2.95865 0.003089880 0.0398694         yes
24  -2.91993 0.003501110 0.0424377         yes
29  -3.48117 0.000499220 0.0181535         yes
33  -3.36348 0.000769672 0.0192418         yes
44  -3.06523 0.002175050 0.0348008         yes
51   3.38382 0.000714839 0.0190624         yes
```

You can write the list of significantly differentially expressed genes to a file like so:

```
> write.table(sig_gene_data, 'sig_diff_genes.txt', sep = '\t', quote = F)
```

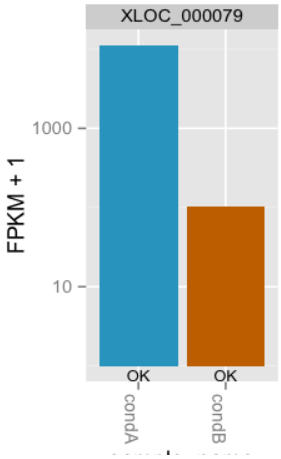
examine the expression values for one of your genes that's diff. expressed:

select expression info for the one gene by its gene identifier:
(note we're naming the variable the same as the
transcript name, so don't be confused by this)

```
>XLOC_000079  = getGene(cuff, 'XLOC_000079')  # use your gene from above, since these may be numbered differently from here.
```

now plot the expression values for the gene under each condition
(error bars are only turned off here because this data set is both simulated
and hugely underpowered to have reasonable confidence levels)

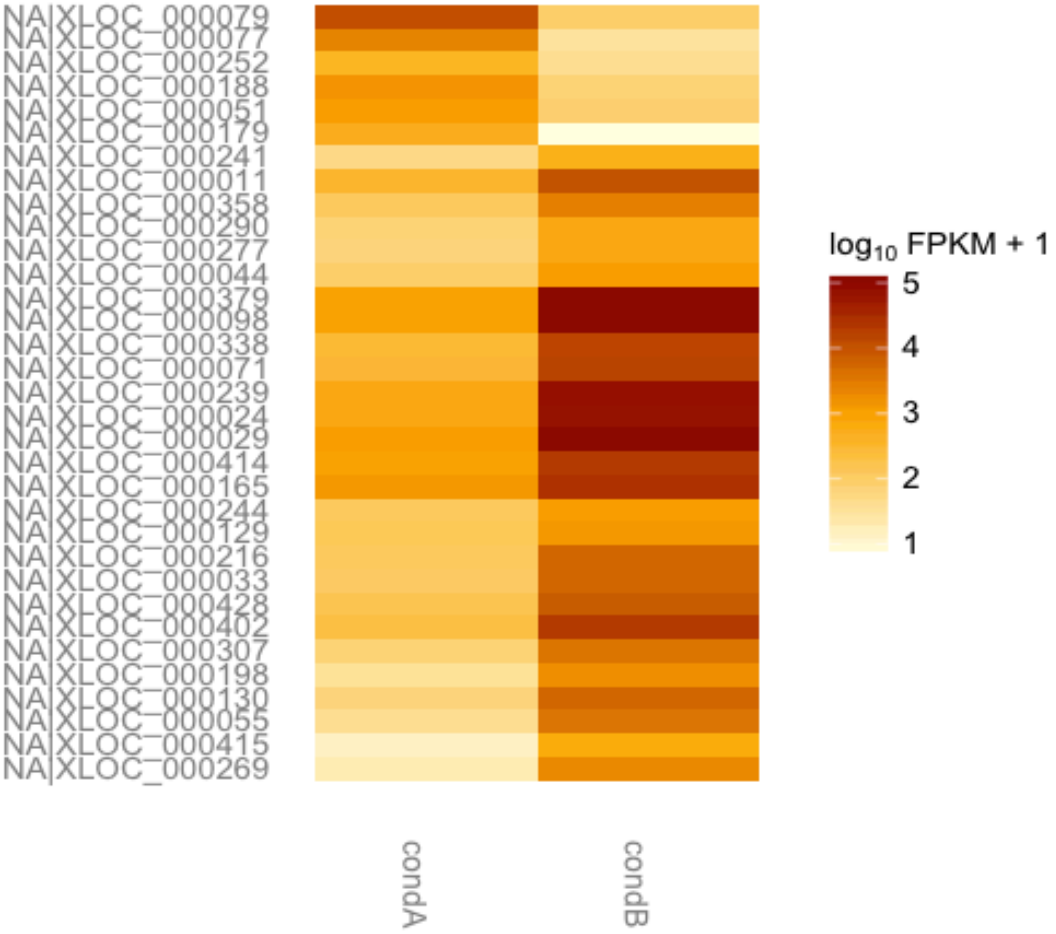
```
>expressionBarplot( XLOC_000079, logMode=T, showErrorbars=F)
```



Draw a heatmap showing the differentially expressed genes

first retrieve the 'genes' from the 'cuff' data set by providing a
a list of gene identifiers like so:
>sig_genes = getGenes(cuff, sig_gene_data$gene_id)

# now draw the heatmap
csHeatmap(sig_genes, cluster='both')



More information on using the Tuxedo package can be found at:

Trapnell C, Roberts A, Goff L, Pertea G, Kim D, Kelley DR, Pimentel H, Salzberg SL, Rinn JL, Pachter L.
Differential gene and transcript expression analysis of RNA-seq experiments with TopHat and Cufflinks. Nat Protoc. 2012 Mar 1;7(3):562-78. doi: 10.1038/nprot.2012.016.
http://www.nature.com/nprot/journal/v7/n3/full/nprot.2012.016.html

The CummeRbund manual:
http://compbio.mit.edu/cummeRbund/manual_2_0.html

(note, most of the tutorial provided here is based on the above two resources)

and the Tuxedo tool websites:

TopHat: http://tophat.cbcb.umd.edu/
Cufflinks: http://cufflinks.cbcb.umd.edu/
CummeRbund: http://compbio.mit.edu/cummeRbund/