A Tutorial: De novo RNA-Seq Assembly and Analysis Using Trinity and edgeR

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:

Trinity (includes **RSEM** in the distribution)
http://sourceforge.net/projects/trinityrnaseq/files/trinityrnaseq_r2012-10-05.tgz/download

GMAP
http://research-pub.gene.com/gmap/src/gmap-gsnap-2012-07-20.tar.gz

GenomeView
http://sourceforge.net/projects/genomeview/files/GenomeView/1.9991/genomeview-1.9991.zip/download

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

Install bioconductor and related tools like so:

```
source("http://bioconductor.org/biocLite.R")
biocLite("edgeR")
biocLite("ctc")
install.packages('gplots')
install.packages('ape')
```

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

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

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

Data contents

After downloading the data file ‘rnaseq_workshop_data.tgz’, expand the contents like so:

```
%   tar –zxvf rnaseq_workshop_data.tgz
```

In the folder nraseq_workshop_data/, in part you’ll find the following files containing rna-seq reads. (note, these were simulated from Schizosacharomyces pombe known transcripts):

condA.left.fa
condA.right.fa

condB.left.fa
condB.right.fa

The reads are paired-end and correspond to two conditions (A, B).

Also, in this folder, you’ll find data corresponding to a small (~2 Mb) annotated region of the S. pombe genome. Although these data are not going to be used for assembly (the reads alone are used for that), we’ll use these data later for examining the results of our de novo assembly in the context of the known transcripts and genomic regions from which they were derived.

De novo assembly of reads using Trinity

To generate a reference assembly that we can later use for analyzing differential expression, first combine the read data sets for the different conditions together into a single target for Trinity assembly. Combine the left reads and the right reads of the paired ends separately like so:

```
% cat condA.left.fa condB.left.fa > both.left.fa
% cat condA.right.fa condB.right.fa > both.right.fa
```

Now run Trinity:
(note, we refer to $TRINTY_HOME/ as the installation directory for the Trinity software and included utilities).

(~9 minutes)

```
% $TRINITY_HOME/ Trinity.pl --seqType fa --left both.left.fa --right both.right.fa --
CPU 4 --JM 4G
```

The assembled transcripts will be found at 'trinity_out_dir/Trinity.fasta'.

How many contigs were assembled?

```
% grep '>' trinity_out_dir/Trinity.fasta | wc -l
    622
```

One way to measure the contiguity of the assembly is by the N50 length (such that at least 50% of the assembled bases reside in contigs at least that length). Compute the N50 value like so:

```
% $TRINITY_HOME/util/N50.pl trinity_out_dir/Trinity.fasta
N50: 1230
```

Since we happen to have a reference genome and a set of reference transcript annotations that correspond to this data set, we can align the Trinity contigs to the genome and examine them in the genomic context.

Align the transcripts to the genome using the GMAP software like so.

First, prepare the genomic region for alignment by GMAP like so:

```
% gmap_build -d genome -D . -k 13 genome.fa
```

Now, align the Trinity transcript contigs to the genome, outputting in SAM format, which will simplify viewing of the data in our genome browser.

```
% gmap -D . -d genome trinity_out_dir/Trinity.fasta -f samse > trinity_gmap.sam
```

Convert to a coordinate-sorted BAM (binary sam) format like so:

```
% samtools view –Sb trinity_gmap.sam > trinity_gmap.bam
% samtools sort trinity_gmap.bam trinity_gmap
```

Now index the bam file to enable rapid navigation in the genome browser:

```
% samtools index trinity_gmap.bam
```

View the Trinity genomic alignments using GenomeView:

Before doing so, let's align the combined read set against the genome so that we'll be able to see how the input data matches up with the Trinity-assembled contigs. Do this by running TopHat like so:

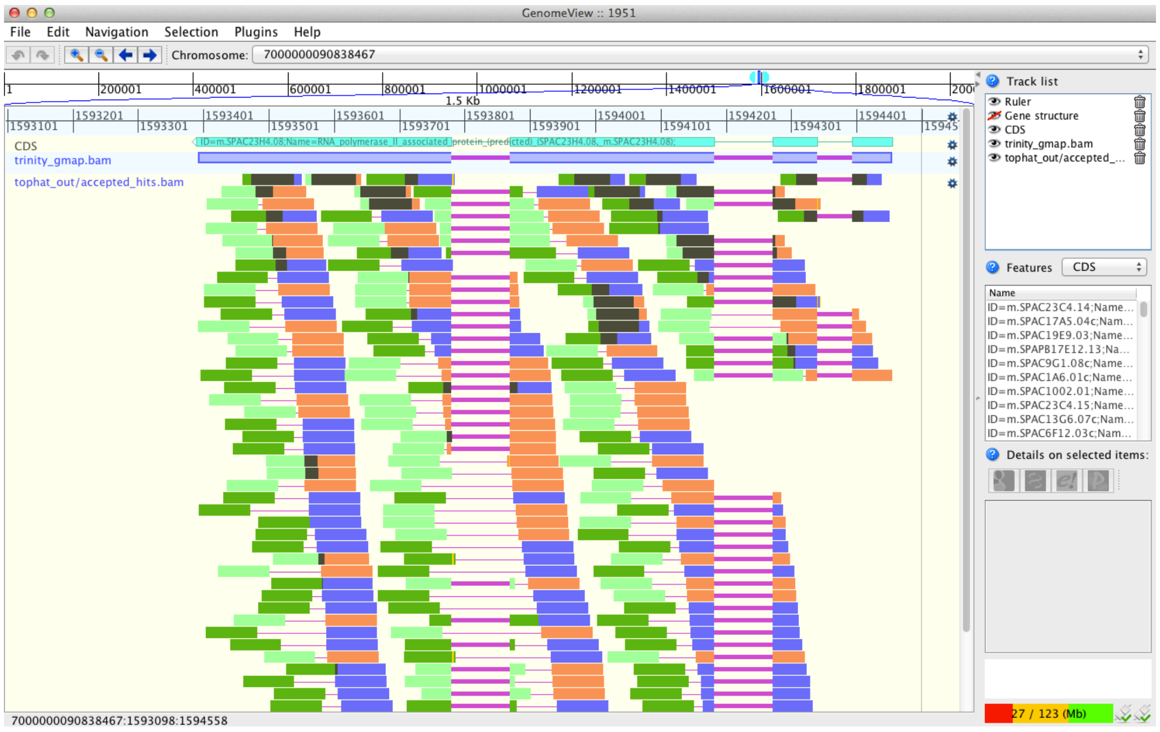
```
# prep the genome for running tophat
```

```
% bowtie-build genome.fa genome

# now run tophat:
% tophat -I 1000 -i 20 genome both.left.fa both.right.fa

# index the tophat bam file needed by the viewer:
% samtools index tophat_out/accepted_hits.bam

# run genomeview to now visualize all the data together:
% java -jar $GENOMEVIEW/genomeview.jar genome.fa genes.bed
tophat_out/accepted_hits.bam trinity_gmap.bam
```



Does Trinity fully or partially reconstruct transcripts corresponding to the reference transcripts and yielding correct structures as aligned to the genome?

Are there examples where the de novo assembly resolves introns that were not similarly resolved by the alignments of the short reads, and vice-versa?

Abundance estimation using RSEM

To estimate the expression levels of the Trinity-reconstructed transcripts, we use the strategy supported by the RSEM software. We first align the original rna-seq reads back against the Trinity transcripts, then run RSEM to estimate the number of rna-seq fragments that map to each contig. Because the abundance of individual transcripts may significantly differ between samples, the reads from each sample must be examined separately, obtaining sample-specific abundance values.

For the alignments, we use 'bowtie' instead of 'tophat'. There are two reasons for this. First, because we're mapping reads to reconstructed cDNAs instead of genomic sequences, properly aligned reads do not need to be gapped across introns. Second, the RSEM software is currently only compatible with gap-free alignments.

The RSEM software is wrapped by scripts included in Trinity to facilitate usage in the Trinity framework.

```
# Align the condition A reads.
% $TRINITY_HOME/util/alignReads.pl --aligner bowtie --seqType fa --left
condA.left.fa --right condA.right.fa --target trinity_out_dir/Trinity.fasta --output
condA_bowtie
```

```
#Before running RSEM, move into that output directory 'condA_bowtie/'
% cd condA_bowtie/
```

The above alignment step will have generated a bam file called 'condA_bowtie.nameSorted.PropMapPairsForRSEM.bam', which will next be used as input to RSEM for estimating abundance values, like so:

```
% $TRINITY_HOME/util/RSEM_util/run_RSEM.pl --transcripts
../trinity_out_dir/Trinity.fasta --name_sorted_bam
condA_bowtie.nameSorted.PropMapPairsForRSEM.bam --paired
```

Once finished, RSEM will have generated two files: 'RSEM.isoforms.results' and 'RSEM.genes.results'. These files contain the Trinity transcript and component (the Trinity analogs to Isoform and gene) rna-seq fragment counts and normalized expression values.

## Now, go back a directory to our work area and rerun the above alignment and abundance estimation steps on condition B. This is briefly done like so:

```
% cd ../    # go back to our work directory
% $TRINITY_HOMEutil/alignReads.pl --aligner bowtie --seqType fa --left
condB.left.fa --right condB.right.fa --target trinity_out_dir/Trinity.fasta --output
condB_bowtie
```

```
% cd condB_bowtie/
% $TRINITY_HOME/util/RSEM_util/run_RSEM.pl --transcripts
../trinity_out_dir/Trinity.fasta --name_sorted_bam
condB_bowtie.nameSorted.PropMapPairsForRSEM.bam --paired
% cd ../  # back to our working directory again
```

Differential Expression Using EdgeR

To run edgeR and identify differentially expressed transcripts, we need a data table containing the raw rna-seq fragment counts for each transcript and sample analyzed. We can combine the RSEM-computed isoform fragment counts into a matrix file like so:

```
# first, for convenience sake, create symbolic links to the RSEM.isoforms.results
files, naming them according to the sample:

% ln -s condA_bowtie/RSEM.isoforms.results condA.counts
% ln -s condB_bowtie/RSEM.isoforms.results condB.counts

# merge them into a matrix like so:
% $TRINITY_HOME/util/RSEM_util/merge_RSEM_frag_counts_single_table.pl
condA.counts condB.counts > rsem.counts.matrix

# before running edgeR, we need the transcript length information, which we can
extract from one of the RSEM.isoforms.results files like so:
% cat condA_bowtie/RSEM.isoforms.results | cut -f 1,3,4 > trans_lengths.txt

# now, run edgeR via the helper script provided in the Trinity distribution:
% $TRINITY_HOME/Analysis/DifferentialExpression/run_EdgeR.pl --matrix
rsem.counts.matrix --transcript_lengths trans_lengths.txt --no_eff_length --output
edgeR
```

Examine the contents of the edgeR/ directory.

```
% cd edgeR
```

The file 'all_diff_expression_results.txt' lists all the transcripts identified as differentially expressed. How many transcripts are identified?

```
% cat all_diff_expression_results.txt | cut -f 3 | wc -l
    75
```

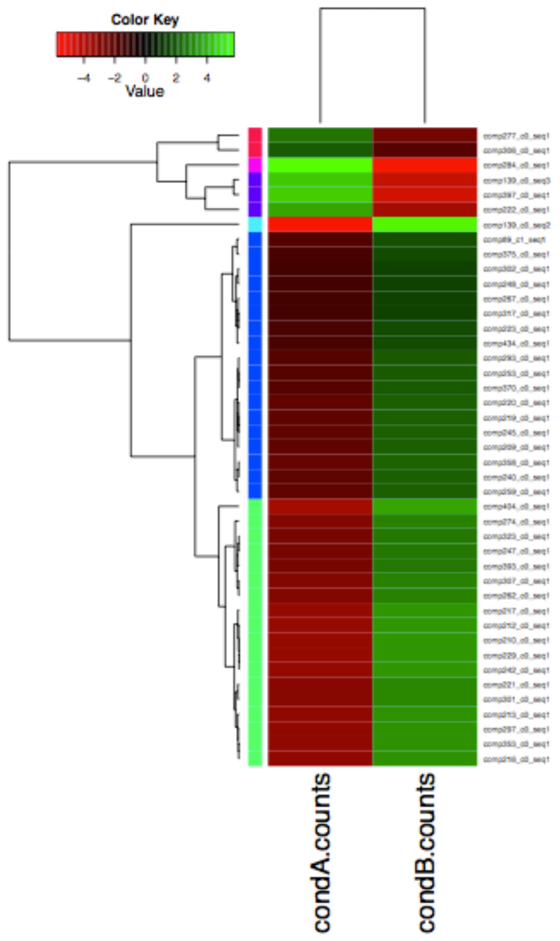
Note, that since the above counts the column header, there are actually 74 transcripts identified.

The file ‘TMM_info.txt’ includes the results from running the TMM normalization step, and the new ‘effective’ library sizes (depth of read sequencing) are indicated. These adjusted library sizes are used to recomputed the FPKM expression values, as provided in the file ‘matrix.TMM_normalized.FPKM’.

Extract those differentially expressed transcripts that are at least 4-fold differentially expressed at a significance of <= 0.001

```
% $TRINITY_HOME/Analysis/DifferentialExpression/analyze_diff_expr.pl --matrix
matrix.TMM_normalized.FPKM
```

This will generate a clustered heatmap file called ‘diffExpr.P0.001_C2.matrix.heatmap.eps’ as shown below.



Note: *The R-script that generates this file is called 'diffExpr.P0.001_C2.matrix.R'. If the image needs to be made smaller to fit inside the page, edit the line of the script below:*

postscript(file="diffExpr.P0.001_C2.matrix.heatmap.eps", horizontal=FALSE, width=7, height=10, paper="special");

resetting the width and height values as set above, and you can regenerate the image like so:

```
%   R --vanilla -q < diffExpr.P0.001_C2.matrix.R
```

More information on Trinity and supported downstream applications can be found from the Trinity software website: http://trinityrnaseq.sf.net