Workshop hands on session –Basic RNA-seq analysis with Tuxedo suite

Note: This is intended as a step by step guide of doing basic RNA-seq analysis with Illumina reads using Tohat/Cufflinks. The data set is a subset of real data due to the time and computation resource constraints at the workshop here. Therefore the results here are only for demonstration of workflow purpose. We are only using the most basic parameter setting.

For details of parameter setting with Tophat/cufflinks, see:

Tophat manual: <http://tophat.cbcb.umd.edu/manual.html>

Cufflinks manual: <http://cufflinks.cbcb.umd.edu/manual.html>

Bowtie manual: <http://bowtie-bio.sourceforge.net/manual.shtml>

Samtools manual: <http://samtools.sourceforge.net/samtools.shtml>

1. Download Yeast reference genome from UCSC genome browser download site. Use wget command on HPC:

$ wget <http://hgdownload.soe.ucsc.edu/goldenPath/sacCer3/bigZips/chromFa.tar.gz>

This should download a zipped version of yeast genome sequences by chromosome in your current directory. You can use ls to see it.

$ gunzip chromFa.tar.gz

$ tar -xvf chromFa.tar

Gunzip and tar should uncompress your downloaded file into 17 FASTA files for 17 chromosomes. Concatenate the17 FASTA files into one file as your reference genome using cat command. \*.fa lists all files that end with .fa and > directs the output to a file called ref.fa.

$ cat \*.fa > ref.fa

1. Use bowtie to index your reference genome ref.fa. –f indicates that your file is in FASTA format. Ref is the prefix of the output index files.

$ module load bowtie

$ bowtie-build -f ref.fa ref

You should have six files of indexes now: ref.1-4.ebwt and ref.rev.1/2.ebwt. Bowtie indexes the genome with a Burrows-Wheeler index to keep its memory footprint small. If you are interested in why you need to index your reference and what the .ebwt files mean, refer to

<http://bowtie-bio.sourceforge.net/index.shtml>

1. Download a sample annotation file in GTF format I prepared for the tutorial. This is a small subset of annotation so your Tophat run will not take too long.

$ wget http://hpc.oit.uci.edu/biolinux/ref/ref.gtf

1. Download the sample read files for condition 1 and 2. It will take a few minutes depending on your internet speed. You should be able to use ls to see them in your current directory.

$ wget <http://hpc.oit.uci.edu/biolinux/reads/s1.fastq>

$ wget <http://hpc.oit.uci.edu/biolinux/reads/s2.fastq>

1. Run Tophat with the download reads: s1.fastq and s2.fastq respectively. This should run much faster than a normal Tophat run, which would take hours. –T forces only align the reads to the transcriptome to save running time here. –o specify the output folder, c1 for condition1 and c2 for condition 2. –G specifies your annotation file.

$ module load samtools

$ module load tophat

$ tophat --bowtie1 –T –G ref.gtf –o c1 ref s1.fastq &

$ tophat --bowtie1 –T –G ref.gtf –o c2 ref s2.fastq &

You do not have to wait for tophat runs to finish to move to the next step!

1. Index and sort the 2 tophat mapping using samtools command so the mapping can be imported to IGV. First get the prepared bam files instead of waiting for your tophat run to finish.

$ wget <http://hpc.oit.uci.edu/biolinux/tophat/s1.bam>

$ wget <http://hpc.oit.uci.edu/biolinux/tophat/s2.bam>

$ samtools sort s1.bam s1.sorted

$ samtools sort s2.bam s2.sorted

$ samtools index s1.sorted.bam

$ samtools index s2.sorted.bam

1. Visualize your indexed and sorted mapping in IGV.

$ module load igv

$ igv &

Click File->load from file and choose s1.sorted.bam. Use yeast genome and go to e.g. chrIII:167,540-167,956 in the search box and you should see your mapped reads! Load s2.sorted.bam in the same way. Change your coordinates to wherever you like.

1. Running cufflinks to assemble transcripts and estimate abundance for each sample. –o specifies cufflinks output folder as cf1 for condition1 and cf2 for condition 2:

$ module load cufflinks

 $ cufflinks –o cf1 s1.sorted.bam &

 $ cufflinks –o cf2 s2.sorted.bam &

You should have two folders cf1 and cf2 for cufflinks output when you are done. Use

$ ls –ltr cf1 to see what is inside.

The file transcripts.gtf is your assembled transcripts. Genes.fpkm\_tracking and isoforms.fpkm\_tracking lists the FPKM values with its corresponding transcripts/genes ID and genomic location.

1. Use cuffmerge to combine your assembled transcripts from all samples. The manifest.txt contains the path to the two assembled transcripts.gtf files.

$ wget http://hpc.oit.uci.edu/biolinux/manifest.txt

$ cuffmerge -s ref.fa manifest.txt

You should have a folder generated by cuffmerge called merge\_asm when you are done. There is a merged.gtf file in this folder which is your merged GTF format transcripts.

1. Use cuffdiff to do differential expression analysis between condition 1 and 2. –o specifies the output directory of cuffdiff. –L gives labels to the 2 conditions as c1 and c2. –-FDR sets the false discovery rate threshold to 0.05. merged\_asm/merged.gtf is the merged transcriptome annotation you generated in step 9.

$ cuffdiff -o cd\_1\_2 -L c1,c2 --FDR=0.05 merged\_asm/merged.gtf s1.sorted.bam s2.sorted.bam

You should have a folder called cd\_1\_2 with your cuffdiff results.

$ ls –ltr cd\_1\_2

will show you the results list. There are around 22 files in the folder and you should be mostly interested in those .diff file such as gene\_exp.diff.

$ less gene\_exp.diff

For details of cuffdiff output, see http://cufflinks.cbcb.umd.edu/manual.html#cuffdiff\_output