**RSC RNAseq Analysis Pipeline**

v0.1 5/23/14

The RSC uses linux workstations at the CBSU: http://cbsu.tc.cornell.edu/lab/lab.aspx

These workstations have all the necessary software and many genome reference files (except where noted below).

The BioHPC User Guide is very helpful, especially for new users.

The genome reference files and locations are described in the ‘Databases’ section of the BioHPC User Guide.

1. Raw data QC with ‘fastqc’

cmd line: fastqc -q --noextract -o <output\_dir> <input.fastq.gz>

*-q = quiet mode*

*--noextract = output file .zip format*

output: fastqc\_report.html

Category Expected Result (RNAseq)

Basic Statistics PASS

Per base sequence quality PASS

Per sequence quality scores PASS

Per base sequence content FAIL (first 10-12nt of RNAseq reads have characteristic pattern)

Per base GC content FAIL (first 10-12nt of RNAseq reads have characteristic pattern)

Per sequence GC content PASS

Per base N content PASS

Sequence Length Distribution PASS

Sequence Duplication Levels WARNING (highly expressed genes give highly duplicated reads)

Overrepresented sequences PASS (unless adaptor sequences are found)

Kmer Content WARNING (kmer patterns in first 10-12nt)

1. Adaptor and quality trimming with ‘cutadapt’

cmd line: cutadapt -m 20 -q 20 -a AGATCGGAAGAGCAC --match-read-wildcards \

-o <output\_trimmed.fastq.gz> <input.fastq.gz> > <output\_log.out>

*-m 20 = minimum read length after trimming (else discard read)*

*-q 20 = quality cutoff to first trim low-quality ends (BWA algorithm)*

*-a ADAPTER-SEQUENCE = 3’ adaptor sequence to trim (trim to end of read)*

*--match-read-wildcards = Allow 'N's in the read as matches to the adapter*

Default params: *-e ERROR\_RATE, default: 0.1 (# errors/length of match)*

*-n COUNT, default 1 (trim the adaptor sequence at most once per read)*

*-O LENGTH, default 3 (minimum overlap length, eg at 3’ end of read)*

output: output\_trimmed.fastq.gz

1. Optional: repeat QC assessment with ‘fastqc’

cmd line: fastqc -q --noextract -o <output\_dir> <input\_trimmed.fastq.gz>

output: fastqc\_report.html

The only sections that should change are

Per base sequence quality if there is significant quality trimming

Sequence Length Distribution if there is significant trimming

Overrepresented sequences if adapter sequences were present

**The standard RSC pipeline follows the Tuxedo protocol without gene/transcript discovery**

http://www.nature.com/nprot/journal/v7/n3/full/nprot.2012.016.html *- see Box 1, alternate protocol B*

Note that the Tuxedo package likes to use directories to organize output files, not unique output filenames.

1. Mapping to transcriptome and genome with ‘tophat’ (uses Bowtie2)

Pre-built tophat GFF file (run once per reference release)

cmd line: tophat -G <latest\_genes.gtf> --transcriptome-index=<gff\_output\_dir>

output: gff\_output\_dir/latest\_genes.\*

Mapping trimmed fastq file

cmd line: tophat -p # -o <tophat\_output\_dir> --transcriptome-index= gff\_output\_dir/latest\_genes \

--no-novel-juncs </path-to-Bowtie2Index/genome> <input\_trimmed.fastq.gz>

*-p # = number of threads (processors)*

*--no-novel-juncs = don’t try to find novel splice junctions (alternate protocol B)*

output: tophat\_output\_dir/accepted\_hits.bam = mapped reads

and other files (summary, logs, indel.bed files, junctions.bed, unmapped.bam)

1. Quantify gene expression per library with ‘cuffquant’ (part of ‘cufflinks’)

Note: use cufflinks v2.2 which allows generation of CXB files using cuffquant

http://cufflinks.cbcb.umd.edu/index.html

cmd line: cuffquant -p # -o <cuffquant\_output\_dir> <latest\_genes.gff> \

<accepted\_hits.bam>

*-p # = number of threads (processors)*

output: cuffquant\_output\_dir/abundances.cxb (binary)

1. Analyze differential gene expression with ‘cuffdiff’ (part of ‘cufflinks’)

Make cuffdiff-compatible GTF file containing tss\_id and p\_id (run once per reference release)

cmd line: cuffcompare -s </path-to- genome/genome.fa> -CG -r latest\_genes.gff latest\_genes.gff

output: cuffcmp.combined.gtf

Cuffdiff

cmd line: cuffdiff -p # -o <tophat\_output\_dir> cuffcmp.combined.gtf \

<case1r1\_abundances.cxb, case1r2\_abundances.cxb,… case1rM\_abundances.cxb> \

<case2r1\_abundances.cxb, case2r2\_abundances.cxb,… case2rM\_abundances.cxb> \

…

<caseNr1\_abundances.cxb, caseNr2\_abundances.cxb,… caseNrM\_abundances.cxb>

output: many files, including count, FPKM, and read group tracking and differential expression tests

for isoforms, genes, CDS (p\_id), and tss\_groups (tss\_id); differential splicing, coding, and

promoter analysis, and run info files.

Note that Cuffdiff will give significant p- (and q-) values even when gene expression levels are low and/or when fold-change is slight. Users may want to filter on a minimum FPKM (at least one case) and/or minimum fold-change for selection of genes for validation and follow-up studies in order to be able to detect gene expression and change in gene expression using standard methods, e.g. qPCR.