

RNAseq Analysis: A Practical Walkthrough (part 2 and a half/3)

C. Ryan Campbell

Duke University

c.ryan.campbell@duke.edu

31 Oct 2017

Overview

- 1 Workflow
- 2 Review Steps
 - Indexing
- 3 Alignment
 - tophat
- 4 Counting
- 5 Tutorial Files
 - Commands
 - DESeq2

Today's Goals

- Logon to interactive SLURM node
- Run tophat2
- Run htseq-count

- 1 Fill in the missing blanks on my diagram
 - Fill in either a file status or a software/process
 - Should take about 1 minute

slogin OR sbatch script

- What is the difference?
- Make sure you're making a conscious choice between the two
- Today we'll be working on slogin with SMALL files
- (why does this matter?)
- When you analyze your files, make sure to use an sbatch script

SLURM Interactive Node

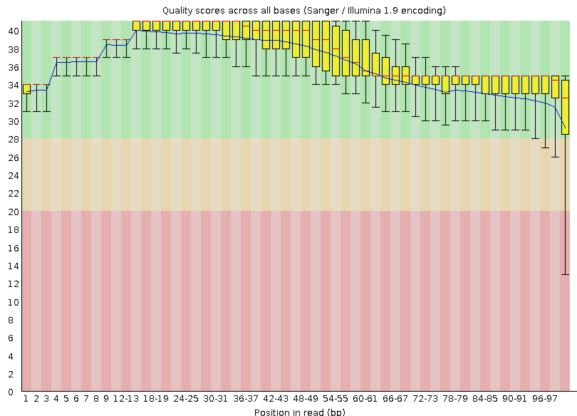
- Later we'll be troubleshooting `htseq-count`
- For that you should use an “interactive node”
- This runs like a sbatch job, but it appears as a terminal that you can interact with

```
srun -mem-per-cpu=4000MB -pty bash -i
```

- You've just requested a 16GB (powerful laptop) size node on SLURM

fastqc output

- Run Quality
- Base Content
- Run Length



Trimming Data

- Remove data that is low quality
 - You have TONS of data, taking out 5% is OK
- We will set a couple of parameters:
 - Minimum quality at the end of the read
 - Average quality along a sliding window
 - Overall read quality
- If the read doesn't meet some, or all, of these the whole read is tossed
- We'll be using trimmomatic

trimmomatic example

- To run trimmomatic:

EXAMPLE:

```
java -jar  
/work/cc216/490S/software/Trimmomatic-0.36/trimmomatic-0.36.jar  
<PE or SE> -phred33 -trimlog <output log> <Read 1.fq> <Read  
2.fq> <Read 1 output> <Read 1 output unpaired> <Read 2 output>  
<Read 2 output unpaired> LEADING:3 TRAILING:3 SLIDINGWINDOW:5:20  
MINLEN:50
```

- Remove (leading/trailing) low quality or N bases (below quality 3)
- Scan the read with a 5-base wide sliding window, cutting when the average quality per base drops below 20
- Drop reads which are less than 50 bases long after these steps

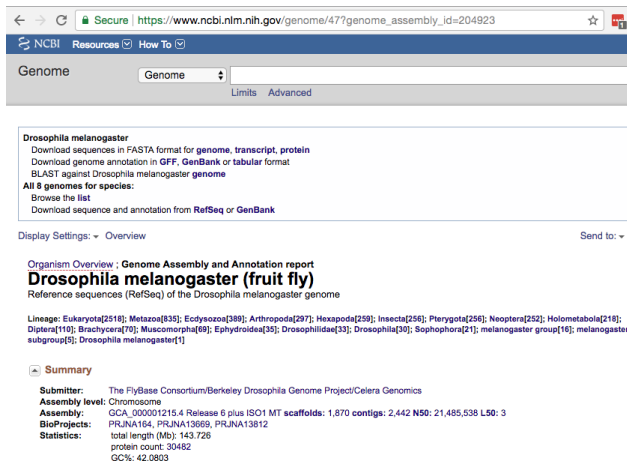
trimmomatic example

- Output:

```
Input Read Pairs: 1000000 Both Surviving: 955447 (95.54%)  
Forward Only Surviving: 29029 (2.90%) Reverse Only Surviving:  
9577 (0.96%) Dropped: 5947 (0.59%)  
  
>head RNAseq.log  
SRR848963.63 ILLUMINA:322:DOUFKACXX:3:1101:11445:2184 length=101  
101 0 101 0  
SRR848963.64 ILLUMINA:322:DOUFKACXX:3:1101:11909:2032 length=101  
97 1 98 3  
SRR848963.64 ILLUMINA:322:DOUFKACXX:3:1101:11909:2032 length=101  
98 0 98 3
```

Downloading genomes

- Model species have a page like this
- Download from the “FASTA format for genome” and “annotation in GFF” links



The screenshot shows the NCBI Genome browser interface for *Drosophila melanogaster*. The browser address bar shows the URL https://www.ncbi.nlm.nih.gov/genome/47?genome_assembly_id=204923. The NCBI logo and navigation links are at the top. The main content area has a search bar with "Genome" selected. Below the search bar, there are links for "Limits" and "Advanced". The main content area displays information about *Drosophila melanogaster*, including download links for FASTA, GFF, GenBank, and tabular format, and a link to BLAST against the *Drosophila melanogaster* genome. It also lists all 8 genomes for species and provides links to browse the list and download sequences and annotations from RefSeq or GenBank. The page includes a "Display Settings" dropdown menu, an "Overview" link, and a "Send to" button. The "Organism Overview" section provides a summary of the genome assembly and annotation report, including reference sequences (RefSeq) of the *Drosophila melanogaster* genome. The "Summary" section provides details about the submitter (The FlyBase Consortium/Berkeley Drosophila Genome Project/Celera Genomics), assembly level (Chromosome), assembly (GCA_000001215.4 Release 6 plus ISO1 MT scaffolds: 1,870 contigs; 2,442 N50: 21,485,538 L50: 3), bioProjects (PRJNA164, PRJNA13669, PRJNA13812), statistics (total length (Mb): 143.726, protein count: 30482, GC%: 42.0803).

Secure https://www.ncbi.nlm.nih.gov/genome/47?genome_assembly_id=204923

NCBI Resources How To

Genome Limits Advanced

Drosophila melanogaster
Download sequences in FASTA format for [genome](#), [transcript](#), [protein](#)
Download genome annotation in [GFF](#), [GenBank](#) or [tabular](#) format
BLAST against *Drosophila melanogaster* [genome](#)
All 8 genomes for species:
[Browse the list](#)
[Download sequence and annotation from RefSeq or GenBank](#)

Display Settings: Overview Send to:

Organism Overview ; **Genome Assembly and Annotation report**
Drosophila melanogaster (fruit fly)
Reference sequences (RefSeq) of the *Drosophila melanogaster* genome

Lineage: [Eukaryota\[2518\]](#); [Metazoa\[835\]](#); [Ecdysozoa\[389\]](#); [Arthropoda\[297\]](#); [Hexapoda\[259\]](#); [Insecta\[256\]](#); [Pterygota\[256\]](#); [Neoptera\[252\]](#); [Holometabola\[218\]](#); [Diptera\[110\]](#); [Brachycera\[70\]](#); [Muscomorpha\[69\]](#); [Ephydroidea\[35\]](#); [Drosophilidae\[33\]](#); [Drosophila\[30\]](#); [Sophophora\[21\]](#); [melanogaster group\[16\]](#); [melanogaster subgroup\[5\]](#); [Drosophila melanogaster\[1\]](#)

Summary

Submitter: The FlyBase Consortium/Berkeley Drosophila Genome Project/Celera Genomics
Assembly level: Chromosome
Assembly: GCA_000001215.4 Release 6 plus ISO1 MT scaffolds: 1,870 contigs; 2,442 N50: 21,485,538 L50: 3
BioProjects: PRJNA164, PRJNA13669, PRJNA13812
Statistics: total length (Mb): 143.726
protein count: 30482
GC%: 42.0803

Indexing

- Most aligners require their genome to be indexed
- What do you think this means?
- You'll need to index using bowtie2 (the aligner tophat2 uses)

- tophat2 - alignment software
- In: Sequence data
- Out: .bam file - where that data aligns to/fits in the genome

- tophat2 is the command/software that aligns the reads to the genome
- This (also) is going to be a computationally intensive process
- So write a submission script to do it:

```
export PATH=/opt/apps/tophat-bowtie/:$PATH
```

```
tophat2
```

- tophat2 example (all one line):

```
tophat2 -p 4 -o s01 -G  
/work/cc216/490S/cc216/RNAseq_pt2/hsap_annotations.gff  
/work/cc216/490S/cc216/RNAseqpt2/hsap  
/work/cc216/490S/cc216/RNAseq_pt2/s01_hsap_norm_R1.fastq  
/work/cc216/490S/cc216/RNAseq_pt2/s01_hsap_norm_R2.fastq
```

translated:

```
tophat2 -p <number of threads> -o <output dir> -G <gff file,  
annotations> <bowtie2 index> <R1 fastq> <R2 fastq>
```

- Help can be found by running “tophat2”
- Or in the tophat2 manual online
- <http://ccb.jhu.edu/software/tophat/manual.shtml>

Counting & Analysis

- Now that we have bam files the next step is to count the reads
- And using those counts compare gene expression levels
- We'll be using HTseq

- python-based program to count reads
- Input:
- .bam file and .gtf/.gff
- Output:
- A table of counts by gene

htseq-count

- We'll be using htseq-count
- This will count the number of reads mapped to each gene
- That data will be taken into DESeq2

```
/opt/apps/rhel7/Python-2.7.11/bin/htseq-count
```

(go ahead and put it in your path)

htseq-count

- What does HTSeq do?
- What are its flags and options?

```
htseq-count <options> <alignment bam> <gff file> > <count  
output>
```

- Probably important: -f, -s, -t

Files to Use

- I've set up some example files to use for this tutorial
- They're human RNAseq files from a hypoxia experiment:

```
ls -lthr /work/cc216/490S/cc216/RNAseq_pt2
```

- What do you see? Which will you use?

Files to Use

```
ls -l /work/cc216/490S/cc216/RNAseq_pt2
```

```
cc216@dcc-yoderlab-01 /work/cc216/490S/cc216/RNAseq_pt2 $ ls -l
total 6716846
-rw-r--r-- 1 cc216 root 1031141111 Oct 27 17:36 hsap.1.bt2
-rw-r--r-- 1 cc216 root 770146216 Oct 27 17:36 hsap.2.bt2
-rw-r--r-- 1 cc216 root 16208 Oct 27 16:55 hsap.3.bt2
-rw-r--r-- 1 cc216 root 770146209 Oct 27 16:55 hsap.4.bt2
lrwxrwxrwx 1 cc216 root 59 Oct 25 23:02 hsap_annotations.gff -> /work/keh65/genomes/GCF_000001405.36_GRCh38.p10_genomic.gff
lrwxrwxrwx 1 cc216 root 59 Oct 30 10:12 hsap.fa -> /work/keh65/genomes/GCF_000001405.36_GRCh38.p10_genomic.fna
lrwxrwxrwx 1 cc216 root 59 Oct 27 16:51 hsap_genome.fna -> /work/keh65/genomes/GCF_000001405.36_GRCh38.p10_genomic.fna
-rw-r--r-- 1 cc216 root 519799 Oct 26 07:48 hsap_hypox_0.10.counts
-rw-r--r-- 1 cc216 root 21183034 Oct 26 01:01 hsap_hypox.counts
-rw-r--r-- 1 cc216 root 0 Oct 26 08:02 hsap_norm_0.10.counts
-rw-r--r-- 1 cc216 root 1031141111 Oct 27 18:19 hsap.rev.1.bt2
-rw-r--r-- 1 cc216 root 770146216 Oct 27 18:19 hsap.rev.2.bt2
drwxr-xr-x 3 cc216 root 249 Oct 30 13:10 s01
lrwxrwxrwx 1 cc216 root 21 Oct 31 09:10 s01_accepted_hits.bam -> s01/accepted_hits.bam
-rw-r--r-- 1 cc216 root 85287123 Oct 27 17:08 s01_hsap_norm_R1.fastq
-rw-r--r-- 1 cc216 root 84966017 Oct 27 17:08 s01_hsap_norm_R2.fastq
drwxr-xr-x 3 cc216 root 249 Oct 30 15:28 s02
lrwxrwxrwx 1 cc216 root 21 Oct 31 09:10 s02_accepted_hits.bam -> s02/accepted_hits.bam
-rw-r--r-- 1 cc216 root 85232659 Oct 27 17:10 s02_hsap_norm_R1.fastq
-rw-r--r-- 1 cc216 root 84768847 Oct 27 17:10 s02_hsap_norm_R2.fastq
drwxr-xr-x 3 cc216 root 249 Oct 30 17:44 s03
lrwxrwxrwx 1 cc216 root 21 Oct 31 09:11 s03_accepted_hits.bam -> s03/accepted_hits.bam
-rw-r--r-- 1 cc216 root 85052823 Oct 27 17:11 s03_hsap_hypo_R1.fastq
-rw-r--r-- 1 cc216 root 84843767 Oct 27 17:11 s03_hsap_hypo_R2.fastq
drwxr-xr-x 3 cc216 root 249 Oct 30 19:57 s04
lrwxrwxrwx 1 cc216 root 21 Oct 31 09:11 s04_accepted_hits.bam -> s04/accepted_hits.bam
-rw-r--r-- 1 cc216 root 85325174 Oct 27 17:12 s04_hsap_hypo_R1.fastq
-rw-r--r-- 1 cc216 root 85165288 Oct 27 17:12 s04_hsap_hypo_R2.fastq
drwxr-xr-x 2 cc216 root 0 Oct 27 17:13 unused
```

symbolic links

- Some of the files in this directory are symbolic links or “sym-links”
- It looks like the file is there, but it actually just points to the true location
- Essentially a shortcut, you can call the link as if it were the file

```
ln -s <file to link to> <link to create>
```

```
ln -s s01/accepted_hits.bam ./s01_accepted_hits.bam
```

(the -s stands for “soft” which allows links across directories and file systems)

SLURM Interactive Node

- Now that we're about to troubleshoot htseq-count hopefully your SLURM node is open

```
srun -mem-per-cpu=4000MB -pty bash -i
```

htseq-count

- What does HTSeq do?
- What are its flags and options?

```
htseq-count <options> <alignment bam> <gff file> > <count  
output>
```

- Probably important: -f, -s, -t

- tophat2 example (all one line):

```
tophat2 -p 4 -o s01 -G  
/work/cc216/490S/cc216/RNAseq_pt2/hsap_annotations.gff  
/work/cc216/490S/cc216/RNAseqpt2/hsap  
/work/cc216/490S/cc216/RNAseq_pt2/s01_hsap_norm_R1.fastq  
/work/cc216/490S/cc216/RNAseq_pt2/s01_hsap_norm_R2.fastq
```

translated:

```
tophat2 -p <number of threads> -o <output dir> -G <gff file,  
annotations> <bowtie2 index> <R1 fastq> <R2 fastq>
```

- Help can be found by running “tophat2”
- Or in the tophat2 manual online
- <http://ccb.jhu.edu/software/tophat/manual.shtml>

Today's Goals

- 1 Logon to interactive SLURM node
- 2 Run tophat2
- 3 Run htseq-count

DESeq2

- What does DESeq2 do?
- Compares the count matrices from many samples
- Where do you run it?
- In R on your laptop

DESeq2

- R-based program to analyze expression
- Input:
 - A table of counts by gene
- Output:
 - Graphs and (hopefully) Answers!!!

DESeq2 guides

- We'll get into DESeq2 next week
- If you want to get started here are some guides:
- Walkthrough
- Bioconductor Manual
- Bioconductor Walkthrough

The End