

RNAseq Analysis: A Practical Walkthrough (part 1/n)

C. Ryan Campbell

Duke University

c.ryan.campbell@duke.edu

12 Oct 2017

Overview

1 Workflow

2 Data Quality

- fastqc
- trimmomatic

3 Genomes

- Downloading
- Indexing

4 Data Prep

5 Alignment

- tophat

Today's Goals

- Clean Data
- Prepare genome files
- Run tophat2 (?)
 - (at least give you the tools to run it...)

Workflow

- 1 Get in your groups
- 2 Fill in the missing blanks on my diagram
 - Fill in a description and software
 - Should take about 10 minutes

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

Data Quality

- So far you've downloaded data
- Next step is to check the quality
- And if the quality is poor, “trim” the data

Data Quality

- This will simultaneously accomplish two goals:
 - 1 Check that the data is as expected (length, format)
 - Important because you otherwise can't see the data
 - 2 Confirm that the data is a good quality across samples
- We'll be using fastqc

fastqc

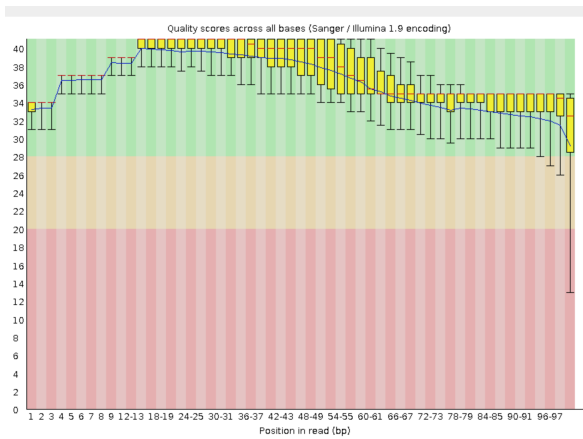
- Runs on the cluster - (small files on slogin, large as an sbatch)
- Output html file describes the fastq file (length, base composition, phred score quality, etc)
- To run fastqc:

```
cd /work/cc216/490S/<your netid>
export PATH=/work/cc216/490S/software/FastQC/:$PATH
EXAMPLE: fastqc [-f fastq|bam|sam] seqfile1 seqfile2 ..
seqfileN
fastqc -f fastq /work/cc216/490S/cc216/test_data/RNAseq_r1.fq
/work/cc216/490S/cc216/test_data/RNAseq_r2.fq
```

- The html output is generated but on the cluster, what is the last step?

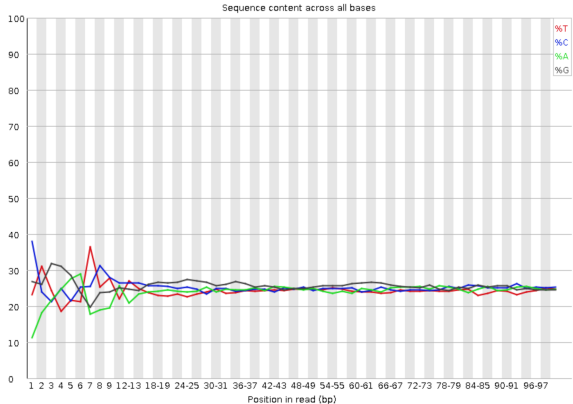
fastqc output

- Run Quality
- Base Content
- Run Length



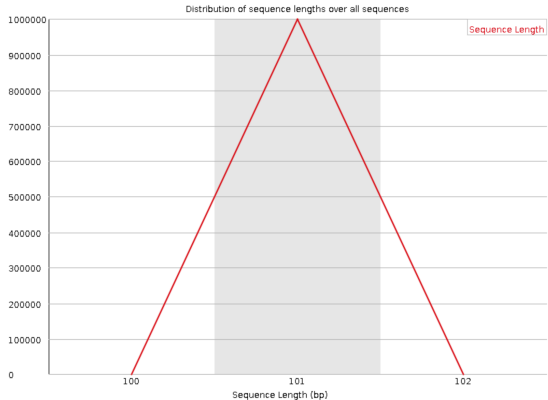
fastqc output

- Run Quality
- Base Content
- Run Length



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

- Runs on the cluster (sbatch script)
- Writes a new fastq file (or pair, R1 & R2)
- To run trimmomatic:

```
cd /work/cc216/490S/<your netid>
```

EXAMPLE:

```
java -jar /work/cc216/490S/software/Trimmomatic-0.36/trimmomatic-0.36.jar PE -phred33 -trimlog  
RNAseq.trimlog RNAseq_r1.fastq RNAseq_r2.fastq RNAseq_r1.trimm.5.20.fastq  
RNAseq_r1.trimm.unpaired.5.20.fastq RNAseq_r2.trimm.5.20.fastq RNAseq_r2.trimm.unpaired.5.20.fastq  
LEADING:3 TRAILING:3 SLIDINGWINDOW:5:20 MINLEN:50
```

- trimmomatic is a java file, you're essentially running the command "java" then pointing it to the .jar file for instructions
- Do you need to move this file down to your laptop?

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:DOUFGACXX:3:1101:11445:2184 length=101  
101 0 101 0  
  
SRR848963.64 ILLUMINA:322:DOUFGACXX:3:1101:11909:2032 length=101  
97 1 98 3  
  
SRR848963.64 ILLUMINA:322:DOUFGACXX:3:1101:11909:2032 length=101  
98 0 98 3
```

- At this point, you should run fastqc again
- Why?

So you say you need a genome...

- Why do we need a genome?
- Where do we get it?
- And if you don't know...

Downloading genomes

- Hopefully your googling led you to NCBI
- Model species have a page like this
- Download from the “FASTA format for genome” and “annotation in GFF” links

The screenshot shows the NCBI website interface for the *Drosophila melanogaster* genome. The browser address bar shows the URL: https://www.ncbi.nlm.nih.gov/genome/47?genome_assembly_id=204923. The page has a navigation bar with "NCBI", "Resources", and "How To". Below this is a search bar with "Genome" selected. The main content area is titled "Drosophila melanogaster" and includes links for downloading sequences in FASTA format, genome annotation in GFF, GenBank, or tabular format, and BLAST against the *Drosophila melanogaster* genome. It also lists "All 8 genomes for species" with links to browse the list or download sequences from RefSeq or GenBank. The page has a "Display Settings" dropdown and an "Overview" link. The "Organism Overview" section is titled "Drosophila melanogaster (fruit fly)" and provides reference sequences (RefSeq) of the *Drosophila melanogaster* genome. The "Lineage" is listed as: 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]. The "Summary" section provides details about the genome assembly, including the submitter (The FlyBase Consortium/Berkeley Drosophila Genome Project/Celera Genomics), assembly level (Chromosome), assembly (GCA_000001215.4 Release 6 plus ISO1 MT scaffolds: 2,442 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

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: 2,442 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)

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)

bowtie index

- This is going to be a computationally intensive process
- So write a submission script to do it
- You can start with my template:

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

```
cat /work/cc216/490S/cc216/genome_index.submit
```

```
cp /work/cc216/490S/cc216/genome_index.submit  
/work/cc216/490S/duke-bio490s/projects/<your_project>/
```

- Now, edit the file to index the genome you downloaded, in your folder

Preparing your data files

- We're now ready to align the RNAseq reads to the genome
- At this point, you're going to start generating a lot of different files
- So this is a good point to plan out file names with your group
- So that everyone knows which files are which, it is best to rename files:
 - e.g. change SRR1201401_1.fastq to adultSample1_r1.fastq
 - Keep a file with a record of these changes
 - (The same file can be a for loop to make the changes)

- 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
```

- No template this time!
- Run tophat2 command and read over the manual
- The next slide has an example command – be careful, the defaults are suitable for mammals, if you're working with non-mammalian data check your paper to for suggestions

- tophat2 example (all one line):

```
tophat -p 4 -o RNAseq1 -G  
/work/cc216/490S/cc216/genomes/dmel_20171011.gff dmel  
/work/cc216/490S/cc216/test_data/RNAseq_r1.trimm.5.20.fastq  
/work/cc216/490S/cc216/test_data/RNAseq_r2.trimm.5.20.fastq
```

translated:

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

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

The End