Short Read Sequencing Workshop Outline

DnA Labs

July 8-19th, 2019

Before Day 1

There are a few items that will help expedite Day 1 of the workshop if they are done beforehand. For additional instructions, workshop resources, and videos, see the following link: Short Read Sequencing Workshop

1. Set-up/install terminal application (if you are a Windows user)
2. Set-up/install X2Go Client (& Quartz for Mac users)
3. Create a GitHub account and email BIT your user ID (bit-help@colorado.edu) **include subject line [sr2019]**
4. Add a public key to GitHub

Day 1: Intro to Sequencing and Overview of Pipelines

Overview

Cover the basic principles of High-throughput Sequencing (HTS) and the subsequent steps involved in processing this data (analysis pipelines).

Key Takeaways

1. Different sequencing experiment types will require a different read depth & sequencing protocol
2. It is important to communicate with your sequencing facility to get sequencing and library prep recommendations
3. The library prep and sequencing type will affect your data quality & subsequent analysis which will impact the conclusions you can draw from your experiment

Day 2: Intro to Unix & Vim

Overview

Introduction to basic Unix commands and text editors (Vim). You can Google search Unix commands/vim commands and go to images for handy cheat-sheets for all of these tools.

Key Takeaways

1. Become comfortable navigating directory tree, moving directories, and creating new directories/files
2. Understand basic file types (e.g. .gz, .tar, .bed) as well as how to view, search, and manipulate these different file types

Additional Resources

For a quick reference of common Unix commands, see this cheat sheet.

Day 3: Intro to servers & downloading public data

Understand the basics of high-performance computing, servers, and job/workload managers. It is also important to know how to access public data and transfer this data to and from the server.

1. Understand what is meant by a compute cluster and what is meant by an EC2 instance
2. Have a basic knowledge of Slurm workload management system and the items necessary to generate an sbatch script
3. Become mindful of compute resources (e.g. how much time/memory/CPUs are needed for a given job) and how to successfully regulate these requests both in your script and in the compute cluster
4. Understand what it means to secure shell and how to transfer data between a local and remote system

Day 4: Quality Control, Mapping & IGV (Genome Browser Visualization)

Quality control, mapping, and visualization are the first steps that take place after obtaining your sequencing data. It is important to assess that your sequencing experiment is successful before moving on to downstream analyses.

1. Understand the difference between fa/fastq files – NEVER EDIT THIS RAW DATA
2. Be able to read and understand each of the 4 lines in a fastq file (esp. grasping read quality)
3. Run FastQC and be able to assess html output of sequence quality measures – remember this assessment and pass/fail measures will be highly dependent on your experiment type (ChIP, ATAC, Nascent, RNA-seq, etc.)
4. Understand the basics of read trimming (e.g. Trimmomatic, TrimGalore, BBDuk –*preferred*–)
5. Understand the basics of mapping and how to adjust these setting based on experiment type (e.g. HISAT2)

Additional Resources

For an excellent paper that compares sequence alignment, see here: Barruzo2016

QC Fail: a useful blog page for teasing apart common sequencing experiment failures and how they will present in FastQC: QC Fail

MultiQC: a program that will assess (in batch) all quality control output from a given experiment for user-friendly visualization

Day 5: Assessment

Quick assessment to make sure everyone is comfortable with the first four days of the workshop. We will give you a short task which should ideally take less than an hour to complete. If it takes longer, it is a good idea to go home over the weekend and review/practice skills from the first week. The second week will be a much quicker pace.

Key Takeaways

1. Be able to run FastQC on raw and trimmed reads and interpret quality reports
2. Understand the basic principles of trimming and be able to run the trimming software of your choice on fastq files for both single- and paired-end sequencing runs
3. Identify whether the trimming was efficient/sufficient
4. Be able to run mapping software (e.g. HISAT2) on your trimmed reads and determine what constitutes a good alignment

Day 6: RNA-seq)

For RNA-seq, we will cover read counting using featureCounts, isoform analysis using Stringtie/Ballgown & creating a custom GTF file from these annotations, and differential expression analysis (DEA) using DESeq2.

Key Takeaways

1. Learn the basics of R including setting variables and command line usage
2. Understand the nuances that are involved in read counting as well as determining which annoation file to use
3. Know that isoform analysis requires a higher read-depth and understand the principles of running Stringtie/Ballgown
4. Be able to understand the big data statistics involved in differential expression analysis and the importance of replicates in your analysis

5. Successfully run DESeq2 and be able to interpret both the QC (e.g. PCA & dispersion plots) as well as the differential gene expression analysis

6. Incorporate DESeq2 into an R script and run it as a job using an sbatch script

Day 7: Post-Mapping QC & Nascent Sequencing

We will learn how to assess the quality of our data post-mapping (e.g. mostly analyzing BAM files). We will also learn how to use MultiQC to combine outputs from multiple samples into one concatenated QC report. In the second half of the day, we will learn how to annotate nascent sequencing data as most of the annotations are based off of ChIP (for enhancers) and RNA-seq/Steady-State (for genes). In nascent analyses, we can capture elements such as intergenic & intragenic transcription regulatory elements, altnerative 5'-end RNA polymerase initiation, and 3'-end run-on. As such, we need to be able to quickly capture all of these elements to analyze using methods such as motif displacement analysis, differential transcription analysis, and comparitive analyses with RNA/ChIP/ATAC-seq. We will use FStitch to capture these unnanotated regions and learn the principles of Tfit and DAStk beforehand in the video as well as in principle in the hands-on portion of the workshop.

Key Takeaways

1. Understand how to run pileup.sh to determine genomic coverage

2. Understand how to run read_distribution.py (part of the RSeQC suite) to determine read distribution relative to gene annotations

3. Be able to run MultiQC to batch assess the quality of our data

4. Understand how to generating a training file for FStitch in IGV

5. Use the training file to run FStitch segment and generate genome-wide annotations

6. Using the output from FStitch segment, run the bidir python module to capture regions bidirectional transcription indicative of RNAP loading and initiation. These elements are also sometimes referred to as eRNAs, but should be used with caution as eRNA implies these have transcriptional regulatory function – not all bidirectionals appear to regulate transcription or have a clear purpose

Additional Resources

MultiQC: a program that will assess (in batch) all quality control output from a given experiment for user-friendly visualization

FStitch GitHub Repository and Documentation: FStitch

Tfit GitHub Repository and Documentation: Tfit

DAStk GitHub Repository and Documentation: DAStk

Day 8: Variant Calling, DNA-seq, & Single-Cell Sequencing

Variant calling using GATK and single-cell sequencing analysis.

Day 9: ChIP-seq & ATAC-seq

This section will cover the basic analysis of ChIP-seq and ATAC-seq. We will cover peak calling using MACS2, the different setting required in peak calling depending on the type type of data, and motif displacement (MD) analysis using DAStk (also covered in Day 7 homework).

1. Know the importance of generating an input (control) sample in ChIP-seq data analysis and how this principle can also be applied to ATAC-seq
2. Understand the difference between narrowPeak and broadPeak settings in MACS2 and when to use each
3. Be able to identify an appropriate p/q-value cutoff for peak calling depending on the type of experiment and how this affects downstream analyses
4. Understand that because motifs are typically generated using data from ChIP-seq analyses, if your ChIP-seq data is for a transcription factor (TF), running the motif displacement analysis is circular
5. Successfully run the MD score analysis on both ChIP-seq and ATAC-seq datasets

Day 10: Downstream Analysis

Overview

Day 10 is meant to get you started learning additional tools that might be helpful in downstream analysis of short read data. This year (2019) we will cover downloading data from ENCODE, some basic BedTools commands, and open biomedical ontologies. Additionally, an advanced homework assignment will go over creating command-line runnable bash scripts with for-loops, if-statements, and user-input.

Key Takeaways

1. Be able to search and download data from ENCODE
2. Be familiar with BedTools, how to run it and what tools are available
3. Understand the basics of ontologies
4. Be familiar with some types of ontology enrichment tools, and what can be obtained from them
5. Understand the basics of for-loops and if-statements in Bash

Additional Resources

1. ENCODE
2. BedTools
3. Official Bash Documentation
4. The OBO Foundry
5. The Gene Ontology
6. Reactome Pathway Analyzer

Day 7 Homework | QC & Nascent Sequencing

Overview

Go through QC on ChIP/ATAC-seq datasets and compare the outputs to nascent. Work through some of the applications of FStitch/Tfit output and nascent sequencing analysis.

Homework Items

1. Run pileup, read_dist, and fastqc on ChIP/ATAC-seq data. Generate a report using MultiQC
2. Generate an sbatch script and run MultiQC on /scratch/Workshop/SR2019/GRO-seq/qc – save the output to your /scratch/Users/USERNAME directory (**HINT**: remember to set
   export PATH=~/.local/bin:$PATH in your script)
3. Using the skills we learned in week 1 (*hint*: think awk), generate a new BED file for samples
   SRR4090{098,099,100,101}.tfit_bidirs.bed
   containing only chr6 annotations found in the directory
   /scratch/Workshop/SR2019/7_nascent/tfit
4. Using the script **7_merge.sbatch**, edit it such that you merge the biological replicates for chr6:
   DMSO: SRR4090{098,099}_chr6.tfit_bidirs.bed
   Nutlin: SRR4090{100,101}_chr6.tfit_bidirs.bed
5. Edit the **7_process_atac.sbatch** script to run process_atac and calculate MD scores for each motif
6. Edit the **7_differential_md_score.sbatch** script to calculate the differential MD scores between the DMSO and Nutlin treatments

If you struggle with the latter 2 items, don't worry! We will get more practice with similar concepts on days 9/10. You can always come back to them.

Day 7 Worksheet | QC & Nascent Sequencing

Overview

We will first go through some more in-depth QC on RNA/GRO-seq datasets and compare the outputs using MultiQC. Then, we will learn how to annotate our nascent data using FStitch. After, we will cover the basics of Tfit/MD score analysis which you will try to apply/run in the homework.

Part 1: Quality Control

Install Required Python Packages

```
$ sh /scratch/Workshop/SR2019/7_nascent/scripts/install_python.sh
```

Copy Scripts to Users Directory

```
$ scp /scratch/Workshop/7_nascent/scripts/*
    ↪ /scratch/Users/USERNAME/scripts
```

Running Fastqc

1. Edit your 7_fastqc.sbatch script. Make sure to change your email, username, and std err/out if they differ from the template
2. Run the 7_fastqc.sbatch script (sbatch 7_fastqc.sbatch)

Remember the arguments required to run fastqc are...

```
$ fastqc SRR.fastq -o OUTDIR/
```

For a full list of arguments, see here.

Running Pileup

1. Edit your 7_pileup.sbatch script. Make sure to change your email, username, and std err/out if they differ from the template
2. Run the 7_pileup.sbatch script (sbatch 7_pileup.sbatch)

The basic arguments to run pileup.sh (part of the BBMap suite of tools) are...

```
$ pileup.sh in=SRR.sorted.bam out=SRR.coverage_stats.txt
```

For a full list of arguments, see here.

Running Read Distribution

1. Edit your 7_read_dist.sbatch script. Make sure to change your email, username, and std err/out if they differ from the template
2. Run the 7_read_dist.sbatch script (sbatch 7_read_dist.sbatch)

The basic arguments to run read_distribution.py (part of the RSeQC suite of tools) are...

```
$ read_distribution.py -i SRR.sorted.bam -r GENOME.bed >
   ↪ SRR.read_dist.txt
```

For a full list of arguments, see here.

Running MultiQC

First, set a PATH to your locally install executables:

```
$ export PATH=~/.local/bin:$PATH
```

Then, run MultiQC on your qc/ directory:

```
$ multiqc qc/ -o multiqc/
```

The minimum required arguments for MultiQC are actually:

```
$ multiqc .
```

which will search your current directory and all sub-directories for any files which match the patterns supported by the program and the default output for the report will be your current directory.

For a full list of supported modules, output options, and example reports see here.

Once we have out .html output, we can open up the report in X2Go. I will demonstrate this in-class.

Part 2: Nascent Analysis

For full help instructions in running FStitch, see here.

Running FStitch Train

We need to begin by generating our training file. Begin by logging into X2Go and load IGV:

```
$ sh /opt/igv/2.3.75/igv.sh
```

Once IGV is loaded, we will import by going to the menu and selecting:

Regions → Import Regions ...

and loading the training file I started located here:
/scratch/Workshop/SR2019/7_nascent/chr1_hct116.bed

We will add another 3 "ON" (transcriptionally active) regions and 3 "OFF" (transcriptionally inactive) regions. To do so, we will need to make sure our region navigator is opened by selecting the following from the top menu:

Regions → Region Navigator ...

Once we have 40 total regions, we can export our new BED file (chr, start, stop, description) to our /scratch/Users/USERNAME directory with the filename of our choice:

Regions → Export Regions ...

Now that our training file is prepared, we can edit our FStitch train script as follows:

1. Edit your 7_fstitch_train.sbatch script. Make sure to change your email, username, and std err/out if they differ from the template
2. Add the full path to your completed training file
3. Run the 7_fstitch_train.sbatch script (sbatch 7_fstitch_train.sbatch)

The basic arguments to run FStitch train are...

```
$ ./FStitch train --bedgraph SRR.cat.bedGraph --strand + --train
    ↪ hg38_train.pos.bed --output PROJECTNAME.hmminfo
```

Running FStitch Segment

We will check to make sure our output parameters (.hmminfo file) are non-zero values. If our training file looks good, we can then edit our fstitch segment script as follows which will annotate our genome into transcriptionally active and inactive regions of interest:

1. Edit your 7_fstitch_segment.sbatch script. Make sure to change your email, username, and std err/out if they differ from the template
2. Add the full path to your paramter file (.hmminfo)
3. Run the 7_fstitch_segment.sbatch script (sbatch 7_fstitch_segment.sbatch)

The basic arguments to run FStitch segment are...

```
$ FStitch segment --bedgraph SRR.cat.bedGraph --strand (+/-) --params
    ↪ PROJECTNAME.hmminfo --output SRR.fstitch.{pos,neg}.bed
```

Notice in the script we are also concatenating our positive and negative strands and sorting using BEDTools (we will see more of this type of file manipulation on day 10). You will use another BEDTools module, merge, in your homework. Don't worry about the details, yet – the script is all set for you.

Running FStitch Bidir

Once our segment module is complete, we will run a python add-on "bidir" that will parse these active regions into regions of bidirectional transcription (putitive eRNAs/regulatory elements/sites of RNAP loading). There are a number of options that will be specific to your data (e.g. due to quality) that you may have to adjustas needed. Once your segment is complete, edit your bidir script as follows:

1. Edit your 7_fstitch_bidir.sbatch script. Make sure to change your email, username, and std err/out if they differ from the template
2. Run the 7_fstitch_bidir.sbatch script (sbatch 7_fstitch_bidir.sbatch)

The basic arguments to run FStitch bidir are...

```
$ bidir --bed SRR.cat.fstitch.bed --genes gene_ref.bed --output
    ↪ SRR.fstitch_bidirs.bed
```