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
```