

Demultiplexing Illumina Sequencing Data of 10X Single Cell Genomics on UGA Clusters (Sapelo2)

Mengrui Zhang, Madgy Alabady

October 2020

1 Introduction

This tutorial is for downloading sequencing runs from Illumina BaseSpace. The tutorial also shows the conversion and demultiplexing of the raw BCL data into 10X genomics single cell data on the UGA GACRC Clusters.

2 Data Download from Illumina BaseSpace

The basespace-cli environment is installed on the GACRC transfer (xfer) and computational (sapelo2) nodes. We will use the transfer nodes to download the run because it is faster.

Log in to the xfer node using your GACRC credentials, as following:

```
$ ssh myID@xfer.gacrc.uga.edu
```

Open up an Internet browser on your local computer and log into your BaseSpace account and navigate to the run that needs download. Figure 1 is an example run dashboard. Copy the Url of this website, you will use the link address later in the command line. As an example, here, the Url link for this example run is "https://basespace.illumina.com/run/196794705/details".

Go back to your terminal where we logged in the xfer server, modify the following code with your own working directory to download the BCL dataset.

```
$ cd #directory where the data should be downloaded
$ bc cp https://basespace.illumina.com/run/196794705 ./3299
```

```
# this is the url that you copied from BaseSpace without the "/details" part.
```

After running the above code, you have to authenticate the download process by copying the link that the software will provide and paste to a browser and confirm it. After confirming the link, The run data will begin downloading immediately.

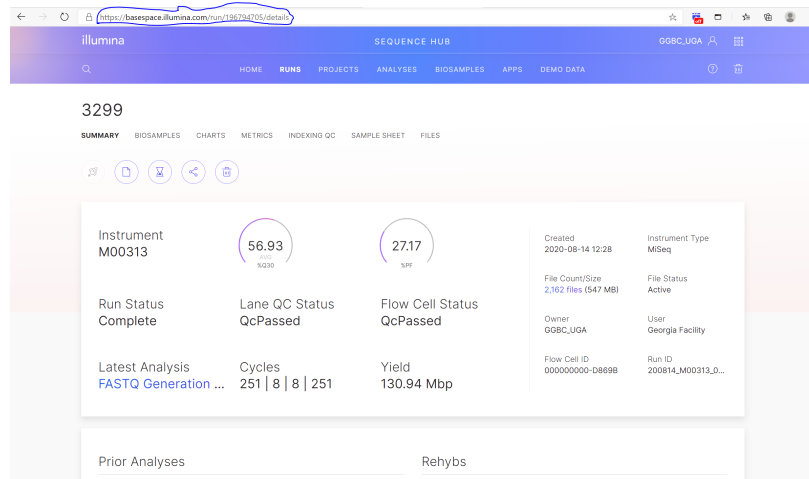


Figure 1: Example run sample sheet from BaseSpace

3 Create Sample Sheet

Now, let's create the sample sheet that's required for the demultiplexing process. Figure 2 is an example sample sheet. When creating the sample sheet in excel, please save it as the ".csv" file format. The lane column indicates Which lane(s) of the flowcell to process. Can be either a single lane, a range (e.g., 2-4) or '*' for all lanes in the flowcell. The sample column indicates the name of the sample. The index column indicates the 10x sample index that was used in library construction.

Lane, Sample, Index

*,3104,SI-GA-G9

Lane	Sample	Index
*	3104	SI-GA-G9

Figure 2: Example run dashboard from Basespace

After making the sample sheet, please upload the sample sheet file to the directory that has the downloaded BCL data. You can use either scp command or FileZilla software. Please check the following URL for transferring the file to the server. An example scp command on your local computer can be:

```
$ scp <SampleSheet.csv> myID@xfer.gacrc.uga.edu:<#directory of downloaded data>
```

Please check and make sure the sample sheet file is in the folder of downloaded BCL data before going to the next step.

4 Submit Job to UGA GACRC Cluster

Now, we need to create a bash script file that has the conversion to fastq and demultiplexing commands, and submit it to the queue of the computer cluster (sapelo2). The following code is an example of this bash script. Please make sure to modify the following. 1. Change the email address to yours and change the where the BCL files were downloaded. We can name this script as demultiplex10x.sh

```
#PBS -S /bin/bash
#PBS -q batch
#PBS -N mkfastq
#PBS -l nodes=1:ppn=6
#PBS -l walltime=24:00:00
#PBS -l mem=100gb
#PBS -M mz79277@uga.edu
#PBS -m abe

cd $PBS_O_WORKDIR

ml CellRanger/3.1.0
module load bcl2fastq2/2.20.0-foss-2016b-Python-2.7.14

cellranger mkfastq --id=<Name of output folder> \
--qc --run=<Directory of downloaded run. #For example ./3299> \
--csv=<Directory of the sample sheet file>
```

The following code can be used on your local computer to upload this script to the server:

```
$ scp <mkfastq.sh> myID@xfer.gacrc.uga.edu:<directory of downloaded data>
```

Make sure the data folder contains both the sample sheet .csv file and this script (example name: demultiplex10x.sh) before submit jobs. To submit the job, run the following code:

```
$ qsub demultiplex10x.sh
```

To check the status of the job. Run:

```
$ qstat_me
```

There is usually a waiting line for jobs. Your job might not be run immediately after you submitted. Once the demultiplexing process has finished, the newly created folder has the same name as the id name in your scripts. For example,

```
$ ./<id_name>/outs/fastq_path/
```

The above directory contains the demultiplexed fastq files for all the samples.