**GTseq Bioinformatics**

This protocol generates genotype data from NextSeq run. Red and green fonts are terminal commands. Note that green font is unique to your project.

**Barcode file TO DO LIST**

1. Open GT-seq Extraction Template.xlsx file
2. There should be eight columns in ‘master’ BCsplit tab.
3. Check i7 names to see that they match what you used in lab.
4. *SampleType*, column two, must contain **only** one of these strings: *initial*, *fillin*, *f1*, *f2*, *f3, r1* or *qc*. Note that NTC of intitial, fillin or qc must have correct *SampleType*
5. *SampleStatus*, column three, can be *normal*, *UP*, *NS* etc. It is essentially a comment field.
6. Copy/paste header row and **ONLY** samples in the library that you are about to process into a new tab as values.
7. Rename new tab as e.g. barcodeT1-T8\_L0166.csv or bcT8-T9\_L0071.csv. Then save new tab as either .csv or .txt file in the BCsplit folder of your progject. It’s best practice to avoid spaces in file name. Your barcode file name **MUST** begin with one of these strings: *barcode*, *bcsplit*, *BCsplit* or *bc*

**Processing NGS Data**

Tipsbefore starting analysis:

* Know date of NextSeq run e.g. a run started on May 11th, 2016, NextSeq sees this date as 160511.
* Shortcut keys in MobaXterm terminal e.g. Ctrl-C or Ctrl-V are not the same as in Windows.
* Use the ‘Refresh folder’ button, located in top of left panel to see most recent changes in folder.
* **IMPORTANT**\*\* Avoid renaming/deleting files from left panel, as server will create hidden files that cannot be erase

1. Open linux terminal with MobaXterm on your Windows computer
2. Click ‘efglserv’ icon near bottom –middle of window to open terminal
3. Click ‘Follow terminal folder’ located on bottom of left panel, to see list of files in current directory
4. Now navigate to bcl directory: $ cd /media/efglserv/seqspace/NextSeq/1801026(hit tab to complete)
5. Convert bcl data into one fastq file with this lengthy command $ nohup bcl2fastq --no-lane-splitting --output-dir /media/efglserv/radspace/Library/2018
6. Navigate to archive directory $ cd /media/efglserv/radspace/Library/2018
7. Change name of extracted file: $ mv Undetermined\_S0\_R1\_0001.fastq.gz L0166.fq.gz
8. Archive library summary (can be executed in any directory): $ libSum.py
9. Navigate to radspace,drive and create library directory: $ mkdir L0166
10. **If last line of terminal looks similar to picture below, proceed. Else STOP and navigate to correct drive.**



1. Navigate to library directory and unzip fastq file:

$ gunzip –c /media/efglserv/radspace/Library/2018/L0166.fq.gz > L0166.fq

1. Create a subdirectory for each subproject: $ mkdir SY2018\_HenryLake
2. Navigate to and drop barcode file into subdirectory
3. Demultiplex fastq file: $ runDMX\_v1.0.py bcT1-T8\_L0166.csv ../L0166.fq

Check to see if new fastq files contain data by refreshing the left panel.

1. Make genotype calls for each sample: $ callinGeno\_v1.0.py
2. Compile genotype table and result files: $ compG\_v1.0.py
3. Perform duplicate search $ Dfunc\_v1.0.py
4. Click refresh button on left panel then drag/drop result files into your project folder. Note that not all files contain data. For example, if you library only has initial run samples, fGeno.csv and qcGeno.csv should be empty. Delete empty files or simply do not place them in your results folder. Below is a list of all possible result files.
   1. T\*\*\*\_L0166\_report.csv
   2. T\*\*\*\_L0166\_iGeno.csv, T\*\*\*\_L0166\_fGeno.csv, T\*\*\*\_L0166\_qcGeno.csv
   3. T\*\*\*\_L0166\_Dfunc.csv
   4. T\*\*\*\_L0166\_gCount.csv
   5. T\*\*\*\_L0166\_dGaps.csv and T\*\*\*\_L0166\_dGapsMap.csv
   6. L0166\_duplicateResults.csv
5. To exit MobaXterm session $ exit<enter><enter>
6. Perform QA/QC of results. Once you determine that your run data is correct, inform Jesse or ninh that you are done with your library, *L0166*. If you mess up, I recommend going back to step 14 and redo. But before you do make all the necessary changes to barcode file and address all known issues. Consult Jesse/ninh if you need help. If you are ready to go back to step 14, first in terminal remove all .fastq, .genos files $ rm \*.fastq \*.genos

Note that this step is nonreversible that is once you execute the rm command those files will be forever gone. Use caution with the rm command.

1. Run duplicate search for entire project if you are ready to do so.
   1. Open MobaXterm and navigate to your folder or create a folder in radspace
   2. Navigate to your directory then drag/drop all xxx\_Dfunc.csv files to left panel
   3. Run duplicate search $ Dfunc\_v1.0.py
   4. Things to consider:
      1. You may have to lower genotyping success rate and/or concordance threshold to obtain best results. I recommend starting with 90% success and 95% concordance rates.
      2. Remember that Dfunc can generate two tables if some QCs do not have matches. Scroll down to bottom of .csv results to find out.
      3. A QC sample that failed to find it’s match may not have the initial/fillin sample in directory or thresholds were set too high.
      4. Four or less mismatches per sample pair is a good indicator of correct match. While 10+ mismatches indicate an erroneous match which can occur if thresholds are set too low.

**Commands:**

View i7s: head -1000000 L####.fq | grep ‘^@NB’ | egrep –o ‘:[CAGT]{6}’ | sort | uniq –c | sort -g

Reads per plate: grep ‘^@NB’ L####.fq | egrep –o ‘:[CAGT]{6}’ | sort | uniq –c | sort –g

**Dfunc file format (no header row):**

|  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| ID | type | status | tray# | i7 | i7name | well | wellname | genos | genos |
| "initial"&ID | initial | initial | 1 | i001 | AAAAAA | A01 | AAAAAA | AA | TT |