* Receive files from sequencer
* Update where files are saved on amphiprion in sample\_data sheet
* Make project directory:

[michelles@amphiprion ~]$ mkdir 01\_seq03

* Make a data directory inside the seq dir
  + cd 01...
  + mkdir data
* Make a “bcsplit" directory to house all of the files produced by barcode splitter
* In the bcsplit directory, in nano make an index file for barcode splitter: Pool012(tab)ATCACG(return) Pool013 TGACCA Pool014 CAGATC Pool015 TAGCTT
* Run barcode splitter on all lanes from the split directory
  + [michelles@amphiprion bcsplit]$ barcode\_splitter.py --bcfile index --idxread 2 --suffix .fastq.gz /local/shared/pinsky\_lab/sequencing/hiseq\_2014\_08\_07/clownfish-ddradseq-seq03-for-222-cycles-ha1wgadxx\_1\_read\_1\_passed\_filter.fastq.gz /local/shared/pinsky\_lab/sequencing/hiseq\_2014\_08\_07/clownfish-ddradseq-seq03-for-222-cycles-ha1wgadxx\_1\_read\_2\_index\_read\_passed\_filter.fastq.gz
  + can ctrl-z, bg to run in background (or nohup)
* move all of these files into a "lane 1 directory" just in case it overwrites
  + [michelles@amphiprion split]$ barcode\_splitter.py --bcfile index --idxread 2 --suffix .fastq.gz /local/shared/pinsky\_lab/sequencing/hiseq\_2014\_08\_07/clownfish-ddradseq-seq03-for-222-cycles-ha1wgadxx\_2\_read\_1\_passed\_filter.fastq.gz /local/shared/pinsky\_lab/sequencing/hiseq\_2014\_08\_07/clownfish-ddradseq-seq03-for-222-cycles-ha1wgadxx\_2\_read\_2\_index\_read\_passed\_filter.fastq.gz
* combine all of the results:
  + [michelles@amphiprion split]$ cat ./lane1/Pool012-read-1.fastq.gz ./Pool012-read-1.fastq.gz > Pool12.fastq.gz
  + [michelles@amphiprion split]$ cat ./lane1/Pool013-read-1.fastq.gz ./Pool013-read-1.fastq.gz > Pool13.fastq.gz
  + [michelles@amphiprion split]$ cat ./lane1/Pool014-read-1.fastq.gz ./Pool014-read-1.fastq.gz > Pool14.fastq.gz
  + [michelles@amphiprion split]$ cat ./lane1/Pool015-read-1.fastq.gz ./Pool015-read-1.fastq.gz > Pool15.fastq.gz
* Move Lane 2 pools into a Lane 2 directory
  + [michelles@amphiprion split]$ mkdir lane2
  + [michelles@amphiprion split]$ mv Pool0\* ./lane2
* make a new names\_barcodes tab delimited file: copy pool contents and barcode sequence for each pool into a new sheet and save as tab delimited names\_Pool12.tsv, etc.
* Open the 4 documents in Komodo and fix the end of line characters by right clicking on the tab and looking at the properties (doing this with google sheets they were already in Unix)
* Create output directories for the different pools (mkdir Pool12, etc)
* Copy the names files into the correct pool directories: scp -r ~/Downloads/names\_Pool12.tsv michelles@amphiprion.deenr.rutgers.edu:~/01\_seq03/Pool12
* sign back into the server and run process radtags on the files
  + [michelles@amphiprion 01\_seq03]$ process\_radtags -f ./split/Pool12.fastq.gz -o Pool12 -r --renz\_1 pstI --renz\_2 mluCI -b ./data/barcodes -i gzfastq -q
  + [michelles@amphiprion 01\_seq03]$ process\_radtags -f ./split/Pool13.fastq.gz -o Pool13 -r --renz\_1 pstI --renz\_2 mluCI -b ./data/barcodes -i gzfastq -q
  + [michelles@amphiprion 01\_seq03]$ process\_radtags -f ./split/Pool14.fastq.gz -o Pool14 -r --renz\_1 pstI --renz\_2 mluCI -b ./data/barcodes -i gzfastq -q
  + [michelles@amphiprion 01\_seq03]$ process\_radtags -f ./split/Pool15.fastq.gz -o Pool15 -r --renz\_1 pstI --renz\_2 mluCI -b ./data/barcodes -i gzfastq -q
* After process\_radtags, all of the pools are still in separate directories.
* Rename the files using rename.for.dDocent\_se (for single end reads):
  + [michelles@amphiprion 01\_seq03]$ cd Pool12
  + [michelles@amphiprion Pool12]$ sh rename.for.dDocent\_se names\_Pool12.tsv
  + [michelles@amphiprion Pool12]$ cd ../Pool13
  + [michelles@amphiprion Pool13]$ sh rename.for.dDocent\_se names\_Pool13.tsv
  + [michelles@amphiprion Pool13]$ cd ../Pool14
  + [michelles@amphiprion Pool14]$ sh rename.for.dDocent\_se names\_Pool14.tsv
  + [michelles@amphiprion Pool14]$ cd ../Pool15
  + [michelles@amphiprion Pool15]$ sh rename.for.dDocent\_se names\_Pool15.tsv
* if final barcode is not getting assigned, nano the names\_ file, get rid of the final end of line character, and re-run. It will look like it might not be working but “ls” will show that the final file was renamed.
* Move all of the files into one directory for further processing:
  + [michelles@amphiprion Pool15]$ cd ..
  + [michelles@amphiprion 01\_seq03]$ mkdir samples
  + [michelles@amphiprion 01\_seq03]$ cd Pool12
  + [michelles@amphiprion Pool12]$ mv A\* ../samples/
  + [michelles@amphiprion Pool12]$ cd ../Pool13
  + [michelles@amphiprion Pool13]$ mv A\* ../samples/
  + [michelles@amphiprion Pool13]$ cd ../Pool14
  + [michelles@amphiprion Pool14]$ mv A\* ../samples/
  + [michelles@amphiprion Pool14]$ cd ../Pool15
  + [michelles@amphiprion Pool15]$ mv A\* ../samples/
* Make an output directory:
  + [michelles@amphiprion Pool15]$ cd ..
  + [michelles@amphiprion 01\_seq03]$ mkdir stacks
* List all of the samples in a format that can be copied and pasted into denovo\_map
  + [michelles@amphiprion 01\_seq03]$ ls -l ./samples/\*.fq | mawk '{print "-s", $9, "\\"}'
* Decide if you’re going to run denovo\_map.pl or run by hand
* Run denovo\_map:
  + nohup denovo\_map.pl -b 2 -B SEQ03 -m 2 -M 4 -n 8 -t -T 10 -o ./stacks -s ./samples/APCL13\_220L255.fq…etc.
  + (running this as batch 2 (-b 2) because I’ve already run a batch in SEQ03 database
* On 8/25/2014 I forgot to nohup:
  + hit ctrl-C to cancel the process and tried again with nohup at the front, terminal returned: nohup: ignoring input and appending output to `nohup.out’; no idea if process is running or not but will see when I get to the lab.

If running by hand:

Run ustacks:

[michelles@amphiprion 01\_seq03]$ for file in ./samples/\*.fq; do ustacks -t fastq -o stacks -d -r -p 10 -f "$file"; done

Ustacks finished during the night.

Check the stacks directory using ls -lh…move any samples that have a file size of 0 to the problem children directory, move them out of the samples directory too

Running cstacks

Need to make list of input sample files

[michelles@amphiprion 01\_seq03]$ ls -l ./samples/\*.fq | mawk '{print "-s ./stacks/"substr($9,11,14), "\\”}'

Literal translation of this shell script is

ls - list contents of directory

-l - (lower case ell) - long version

./samples/\*.fq - any files that end in .fq

| - pipe

mawk - extract text and then…

‘{print - output to screen

“-s ./stacks” - literal output - don’t put a space between stacks/“ and substr or a space will appear in output

substr($9,11,14), - take a subset of the string item 9 in the list - starting at character 11 and including 14 characters

“\\” - literal output

}’ - end print command

Ran cstacks

[michelles@amphiprion 01\_seq03]$ cstacks -b 1 -o stacks -n 4 -p 10 -s ./stacks/APCL13\_031L410 \

> -s ./stacks/APCL13\_037L411 etc.

At sample APCL13\_164L441 - there was a segmentation fault (core dumped) - retrying

[michelles@amphiprion 01\_seq03]$ ls -l ./samples/\*.fq | mawk '{print "-s ./stacks/"substr($9,11,14), "\\”}'

[michelles@amphiprion 01\_seq03]$ cstacks -b 1 -o stacks -n 4 -p 10 -s ./stacks/APCL13\_031L410 \

> -s ./stacks/APCL13\_037L411 etc.

This time L442 caused the dump…is it the location in the analysis or the file?

L442, L446, L269, L304 all have empty .alleles.tsv files, moving those samples out of the directory and trying again…

[michelles@amphiprion 01\_seq03]$ mkdir problem\_children

[michelles@amphiprion 01\_seq03]$ mv ./samples/A\*L304.\* ./problem\_children/

[michelles@amphiprion 01\_seq03]$ mv ./samples/A\*L269.\* ./problem\_children/

[michelles@amphiprion 01\_seq03]$ mv ./samples/A\*L446.\* ./problem\_children/

[michelles@amphiprion 01\_seq03]$ mv ./samples/A\*L442.\* ./problem\_children/

trying again at 2:12pm with problem children removed from samples and stacks folders - also upped n = 8

[michelles@amphiprion 01\_seq03]$ ls -l ./samples/\*.fq | mawk '{print "-s ./stacks/"substr($9,11,14), "\\”}'

[michelles@amphiprion 01\_seq03]$ cstacks -b 1 -o stacks -n 8 -p 10 -s ./stacks/APCL13\_031L410 \

> -s ./stacks/APCL13\_037L411 etc.