**FILE RENAMING**

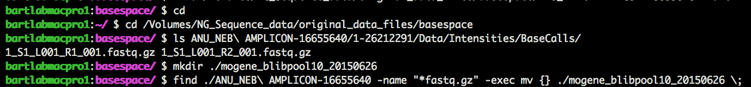
1. Download data to MacPro into /Volumes/NG\_Sequence\_data/original\_data\_files/
   1. BaseSpace (used by MOGene) has a Download application (I understand they may be trying to change the way they store data, but until then some extra steps are needed here):
      1. Files should be downloaded to the ./basespace subdirectory of the above path
      2. Files should then be moved into their own directory as:

cmd: cd /Volumes/NG\_Sequence\_data/original\_data\_files/basespace

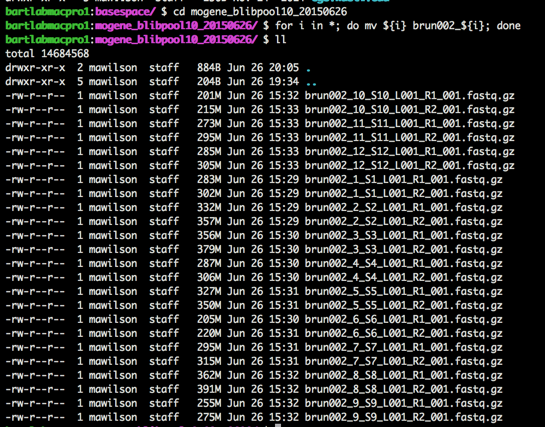
cmd: mkdir <descriptive\_directory\_name\_here>

cmd: find ./ -name "\*fastq.gz" -exec mv {} <descriptive\_directory\_name\_here> \;

for example:



NOTE: MOGene files need to be differentiated from each other for the pattern matching to work as shown in the screenshot here:



* 1. GTAC is easily downloaded using wget to the above path

1. Copy files to a temporary directory on apollo such as /home/bbart/data/blib/tmp\_raw\_data

for example, from apollo:

scp blab:/Volumes/NG\_Sequence\_data/original\_data\_files/basespace/mogene\_blibpool10\_20150626/\\* /home/bbart/data/blib/tmp\_raw\_data

1. Create regular expression patterns for the file renaming (this part is a little weird):
   1. The patterns are stored in JSON format in /home/bbart/data/blib/match.patterns which is similar to a Python dictionary. (JSON files don’t allow comments, otherwise there would be some more information in that file) The JSON is broken into two main parts:
      1. “run” - this contains the pattern to match the raw data filenames
         1. the key field of each entry is the pattern and contains two parenthesized parts that are used as keys to indicate:
            1. library
            2. paired read file indicator
         2. the data field of each entry is a list of 3 things:
            1. key indicating the data pool in the “pool” section described below
            2. string containing sequencing type and date to be inserted in the filenames
            3. list indicating order the parenthesized sections of the pattern come in since sequencing facilities do this differently
      2. “pool” - this contains the data pools to associate blib numbers with the files. Each entry in the pools contain:
         1. key field that comes from the input filenames
         2. data field is a list of a blib number and a dictionary that associates read numbers to 1 or 2 since GTAC assigns 1 and 3 to their paired end read files
   2. New entries should be created for new datasets. Care should be taken that each pattern is different in the “run” section and that each pool indicator is different in the “pool” section:
      1. “run” section for the example used in this guide:



* + 1. “pool” section for the example used in this guide:



1. Create blib directories if necessary. Example:

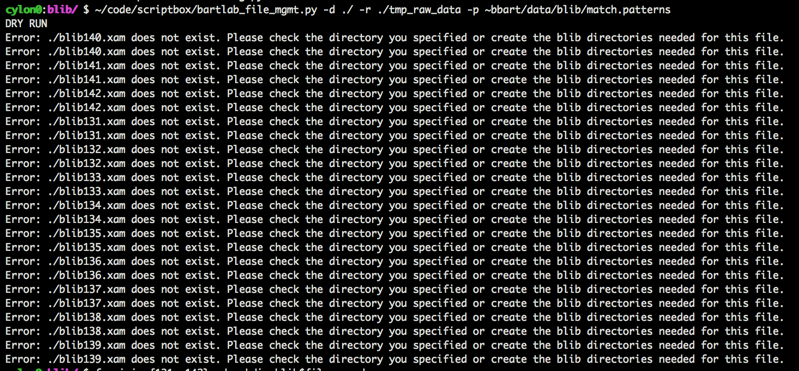


1. Move files using /home/mawilson/code/scriptbox/bartlab\_file\_mgmt.py This command is meant to be run in two steps:
   1. DRY RUN: This mode is available to check that the patterns were entered correctly and files aren’t overwritten or moved improperly:

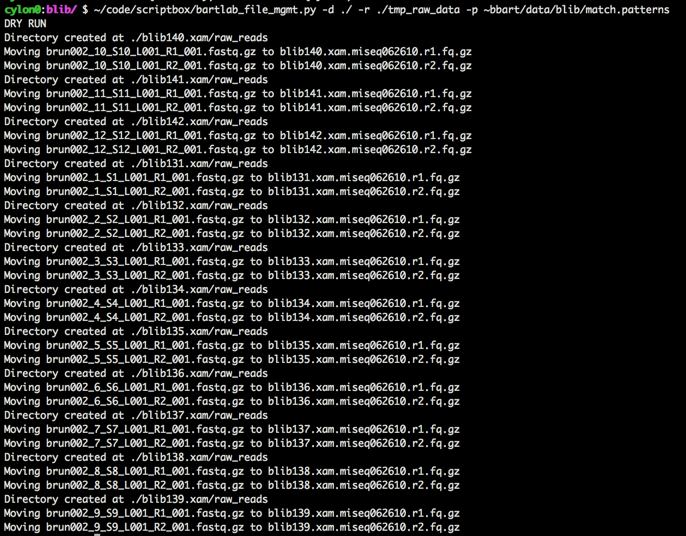
cmd: cd /home/bbart/data/blib/

cmd: ~/code/scriptbox/bartlab\_file\_mgmt.py -d ./ -r ./tmp\_raw\_data -p ~bbart/data/blib/match.patterns

* + 1. If you have forgotten to add the blib directories, this is what happens:



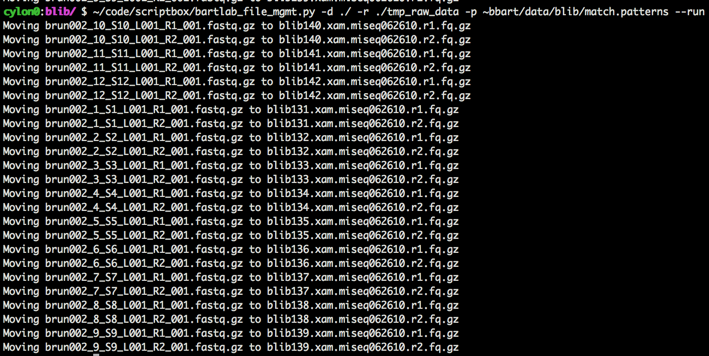
* + 1. If the Dry Run is successful:



* 1. WET RUN: Once the Dry Run is successful and you’ve checked that the filenames are changing as expected, you can do a real run by adding the --run flag to the command:

cmd: ~/code/scriptbox/bartlab\_file\_mgmt.py -d ./ -r ./tmp\_raw\_data -p ~bbart/data/blib/match.patterns --run

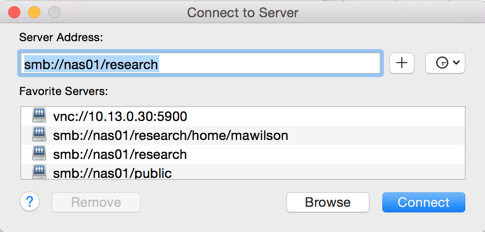
* + 1. Successful:



1. Now the files have been moved to apollo and renamed according to the Bart Lab way

**BACKUP ORIGINAL FILES**

1. Connect the MacPro to the research drive:
   1. Open Finder
   2. Press Command-K
      1. This window should open:



* 1. Under Server Address enter smb://nas01/research and hit connect
  2. You will be prompted for credentials, enter your Danforth username and password as you would to access the W3 or your email

1. Sync the original\_data\_files with the research drive:
   1. Open a terminal
   2. Use rsync to sync the data:

cmd: rsync -arvh /Volumes/NG\_Sequence\_data/original\_data\_files/ /Volumes/research/bart\_lab/original\_data\_files\_bkup

**TRIM READS**

1. ssh to apollo
2. ssh to one of the cylons
3. cd /home/bbart/data/blib
4. For new blibs make sure the blib directories have a trimmed\_reads directory, for example:

cmd: for i in {131..142}; do mkdir blib${i}/trimmed\_reads; done

1. This example of the command is built to loop through multiple blib datasets. In order to use this with just one blib, instead of a range entered with {092..101}, you can replace that with just the number. Notes about the command:
   1. On apollo a directory different from /tmp is often needed for JAVA commands, I use /home/mawilson/tmp. This isn’t accessible to anyone else so make sure you change this to your own home directory once you create your own
   2. Make sure the number of threads you’ve selected is appropriate for the machine you’re using, however this command never seems to use more than 3 CPUs
   3. This command is set up to use the example parameters provided on the Trimmomatic home page. This may not be appropriate for your dataset, but it seems to be fine most of the time
   4. This command is set up as 3 nested for loops to make it easier to change things like blib number, species suffix, and run info. This allows you to change this information in one place instead of several

cmd: for i in {131..142}; do for k in xam; do echo blib${i}.${k}; for j in miseq062615; do echo ${j}; java -Djava.io.tmpdir=/home/mawilson/tmp -Xmx4g -jar /nfs4shares/bioinfosw/installs\_current/Trimmomatic-0.32/trimmomatic-0.32.jar PE -threads 13 -phred33 blib${i}.${k}/raw\_reads/blib${i}.${k}.${j}.r1.fq.gz blib${i}.${k}/raw\_reads/blib${i}.${k}.${j}.r2.fq.gz blib${i}.${k}/trimmed\_reads/blib${i}.${k}.${j}.p1.fq.gz blib${i}.${k}/trimmed\_reads/blib${i}.${k}.${j}.u1.fq.gz blib${i}.${k}/trimmed\_reads/blib${i}.${k}.${j}.p2.fq.gz blib${i}.${k}/trimmed\_reads/blib${i}.${k}.${j}.u2.fq.gz ILLUMINACLIP:/nfs4shares/bioinfosw/installs\_current/Trimmomatic-0.32/adapters/TruSeq3-PE.fa:2:30:10 SLIDINGWINDOW:4:15 MINLEN:36 &> blib${i}.${k}/trimmed\_reads/blib${i}.${k}.${j}.trim; done; done; done

**ASSEMBLE READS**

1. ssh to apollo
2. ssh to one of the cylons
3. cd /home/bbart/data/blib
4. For new blibs make sure the blib directories have an assembly directory, for example:

cmd: for i in {131..142}; do mkdir blib${i}/assembly; done

1. Notes on loops and threads in the TRIM READS section apply here as well. Assemble reads:

cmd: for i in {131..142}; do for j in miseq062615; do for k in xam; do echo ${i} ${k} ${j} spades run; /nfs4shares/bioinfosw/installs\_current/SPAdes-3.0.0-Linux/bin/spades.py -t 10 -o blib${i}.${k}/assembly/spades\_v1 --pe1-1 blib${i}.${k}/trimmed\_reads/blib${i}.${k}.${j}.p1.fq.gz --pe1-2 blib${i}.${k}/trimmed\_reads/blib${i}.${k}.${j}.p2.fq.gz --pe1-s blib${i}.${k}/trimmed\_reads/blib${i}.${k}.${j}.u1.fq.gz --pe1-s blib${i}.${k}/trimmed\_reads/blib${i}.${k}.${j}.u2.fq.gz -k 33,53,73,93 &> blib${i}.${k}/assembly/blib${i}.${k}.${j}\_spades.out; done; done; done