MCBL Documentation

Release 1.0

Saranga

Jul 07, 2020

About The MCBL

1 MCIC Computational Biology Lab (MCBL) 3

2 MCBL membership 5

3 MCBL servers and computing resources 7

4 Download from BaseSpace 11

5 Share runs and projects in BaseSpace 17

6 Using BaseMount 23

7 Download from hudsonalpha.org 31

8 Filter a CASAVA-generated fastq file 33

9 Fastq adapter removal and QC 35

10 DESeq2 with phyloseq 39

11 GBS 43

12 A basic microbiome analysis 47

13 Indices and tables 51

Python Module Index 53

Index 55


MCBL

MCBL

CHAPTER 1

MCIC Computational Biology Lab (MCBL)

1.1 Our goal

Our mission is to build core support and intellectual leadership in the area of bioinformatics to support research at the OARDC, by providing an engaging work environment, space, infrastructure and training for performing research involving biological data analysis.

We aspire for the MCBL to become the place to be for learning and performing bioinformatics research at the OARDC, the place where ideas are discussed and exchanged, students and users learn from each other and get help and support from our experienced staff when needed, and we as a community move our bioinformatics knowledge forward.

We specialize in working with High-throughput Sequencing (HTS) data and providing training in reproducible science using modern tools.

We are happy to help you to carry out your own analysis. This will include helping with experimental design, discussing the most effective way to carry out your data analysis, providing the computational infrastructure (software, scripts and computers), interpreting results, and answering questions you might come across along the way. We can also process and analyze HTS data for you, using either standardized or custom pipelines.

1.2 MCBL services

Data analysis	Contacts
• Processing, analyzing, and interpreting HTS data • WGS, RADseq/GBS, RNAseq, metagenomics, microbiomics, ... • Quality control • Genotyping • Genome assembly and annotation • Differential expression analyses • Population genetic and genomic analyses • GWAS, (e)QTL analyses	Jelmer Poelstra

Workshops & training	Contacts
• UNIX command line and bash scripting • Running analyses at the Ohio Supercomputer Center (OSC) • Running analyses in the cloud • `R` - general, bioinformatics, R markdown, dashboarding • `Python` - general, bioinformatics • Reproducible compute environments with containers and `conda` • Reproducible analysis pipelines with `Snakemake` • Version control with `git` and Github • Data management and deposition in public repositories	Jelmer Poelstra

High-throughput Sequencing services	Contacts
• Illumina Miseq	• Tea Meulia (director) • Fiorella Cisneros Carter

See also:

Main MCIC page for more details on sequencing services

CHAPTER 2

MCBL membership

2.1 MCBL membership benefits

- Access to the MCBL and MCBL computers 24/7.
- Free access to all MCBL activities: workshops, user group meeting, etc.
- Access to 1 TB data storage space for the duration of the membership.
- Assistance with experimental design, bioinformatic analyses, and interpretation of your data.

2.2 How to apply for an MCBL membership

Step 1 Fill out and submit the MCBL application form: MCBL Application.

Step 2 Submit your membership fee to MCIC.

Step 3 Contact Jelmer Poelstra for login credentials.

Note: Access to MCBL resources will be granted till we receive the payment. Once the form is completed and submitted a notification e-mail will be sent to the membership applicant and the PI.

2.3 MCBL membership duration

Membership is offered for a 6 month or 1 year period at a time.

2.4 MCBL membership termination

MCBL membership will be terminated after the membership period ends, or upon a written request from user or PI to terminate the membership.

2.5 Contacts

Person	Information
Tea Meulia (director)	Questions regarding membership
Jelmer Poelstra	MCBL server access and remote access
Jody Whittier	MCBL payments

CHAPTER 3

MCBL servers and computing resources

3.1 Servers

Server	Processors	Cores	Memory	Local Disk
mcic-ender-svr	four 2.40GHz ten-core Intel® Xeon processors E7-4870	40	1.0 TB	16TB
mcic-ender-svr2	four 2.00GHz ten-core Intel® Xeon processors E7-4850	40	1.2 TB	10TB
mcic-ent-srvr	two 2.67GHz six-core Intel® Xeon processors X7542	12	250GB	2.0TB

3.2 Workstations

Workstation	Processors	Cores	Memory	Local Disk
mcic-galaxy-srvr	two 3.47GHz six-core Intel® Xeon processors X5690	12	94 GB	2.6 TB
mcic-mac-srvr	two 2.93GHz six-core Intel® Xeon processors X5670	12	64 GB	4.0 TB

3.3 Desktops

Desktop	Processors	Cores	Memory	Local Disk
mcic-sel019-d1	one 3.00GHz four-core Intel® Xeon processors i7-4578U	7	32 GB	1.0 TB
mcic-sel019-d2	one 3.00GHz four-core Intel® Xeon processors i7-4578U	7	32 GB	1.0 TB
mcic-sel019-d3	one 3.00GHz four-core Intel® Xeon processors i7-4578U	7	32 GB	1.0 TB
mcic-sel019-d4	one 3.00GHz four-core Intel® Xeon processors i7-4578U	7	32 GB	1.0 TB
mcic-sel019-d5	one 3.00GHz four-core Intel® Xeon processors i7-4578U	7	32 GB	1.0 TB
mcic-sel019-d6	one 3.00GHz four-core Intel® Xeon processors i7-4578U	7	32 GB	1.0 TB
mcic-sel019-d7	one 3.00GHz four-core Intel® Xeon processors i7-4578U	7	32 GB	1.0 TB

3.4 Software

The following bioinformatics software are availabe through the MCBL. Please contact Jelmer Poelstra for details on availability of the software.

3.4.1 Commercial Software

Application	Version	Description
CLCBio Workbench	8.5.1	A comprehensive and user-friendly analysis package for analyzing comparing and visualizing next generation sequencing data
Blast2GO Pro	Pro	A complete framework for functional annotation and analysis

3.4.2 Open Source Software

Application	Version	Description	Module Name
Bowtie	1.1.0/2-2.2.3	An ultrafast and memory-efficient tool for aligning sequencing reads to long reference sequences	Bowtie-<version>
Cd-hit	4.6.1	A very widely used program for clustering and comparing protein or nucleotide sequences	cd-hit-v<version>
Cutadapt	1.4.2/1.8.1	Removes adapter sequences from high-throughput sequencing data	Cutadapt/<version>
Exonerate	2.2.0	A generic tool for pairwise sequence comparison	Exonerate/<version>
Express	1.5.1	eXpress is a streaming tool for quantifying the abundances of a set of target sequences from sampled subsequences	express-<version>
Fastqc	1.5.1	Aims to provide a simple way to do some quality control checks on raw sequence data coming from high throughput sequencing pipelines	Fastqc-<version>
GenomeAnalysisTK	3.4.2	GATK tools for error modeling data compression and variant calling	GenomeAnalysisTK-<version>
Maker	2.31.8	MAKER is a portable and easily configurable genome annotation pipeline.	Maker/<version>
Mothur	1.33/1.35	Tool for analyzing 16S rRNA gene sequences.	Mothur-<version>
Mummer	3.23	A system for rapidly aligning entire genomes whether in complete or draft form.	Mummer/<version>
Pandaseq	2.8	A program to align Illumina reads optionally with PCR primers embedded in the sequence and reconstruct an overlapping sequence.	Pandaseq/<version>
Qiime	1.8	A program for comparison and analysis of microbial communities primarily based on high-throughput amplicon sequencing data.	Qiime-<version>
Rsem	1.2.16	A software package for estimating gene and isoform expression levels from RNA-Seq data.	rsem-<version>
Samtools	0.1.19	Provides various utilities for manipulating alignments in the SAM format including sorting merging indexing and generating alignments in a per-position format	Samtools-<version>
SNAP	0.1.19	A new sequence aligner that is 3-20x faster and just as accurate as existing tools like BWA-mem Bowtie2 and Novoalign	SNAP/<version>
Trim-fastq	1.2.2	A Fastq quality trimmer.	Trim-fastq-<version>
Trinity	r20140717	a novel method for the efficient and robust de novo reconstruction of transcriptomes from RNA-seq data.	Trinity

CHAPTER 4

Download from BaseSpace

Note:

Purpose This short tutorial will show you how to download MiSeq sequencing data from Illumina BaseSpace.

Author Wirat Pipatpongpinyo

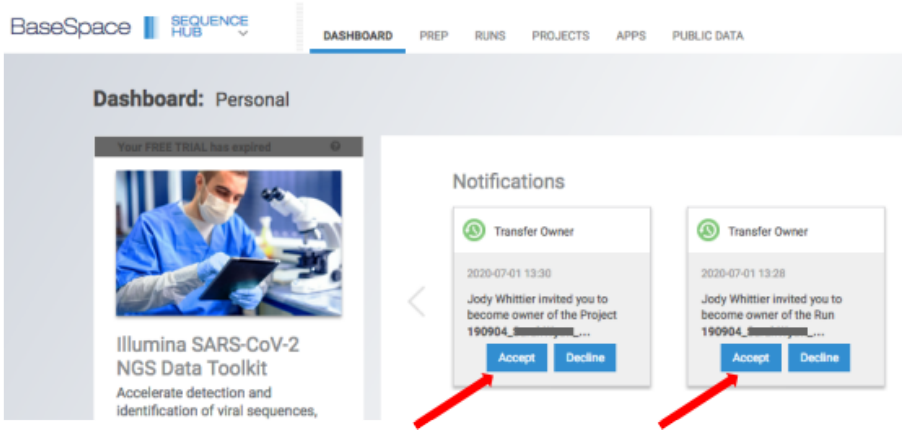
Date July 6, 2020

4.1 Step-by-step guide to download your sequence data from BaseSpace

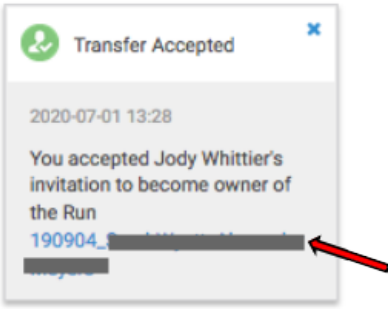
1. You will receive two invitation emails from `basespace-noreply@illumina.com` for the transfer of ownership to you, one for the **sequencing run** and another for the **project**. To be able to fully access the data, you must accept both invitations.

2. Once you click the link (`Click here to accept this transfer of ownership`) in one of the invitation emails, you will be taken to the Illumina BaseSpace website, where you can log in to your account.

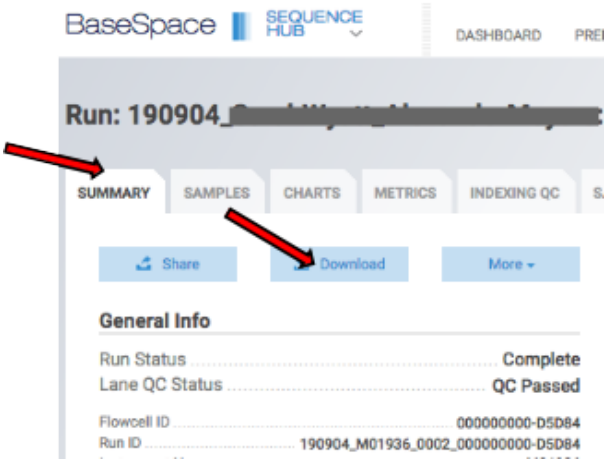
3. After you have logged in, BaseSpace will bring you to the `DASHBOARD` tab. Click `Accept` in both boxes to take the ownership of both the run and project.



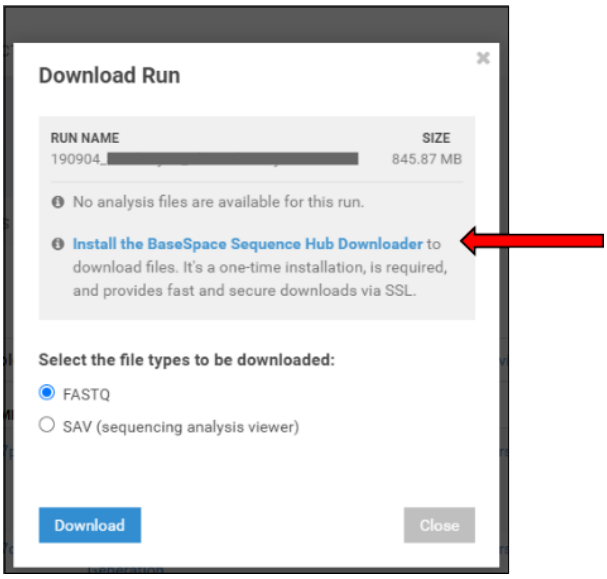
4. Once transfer is accepted, click on the name of the run.



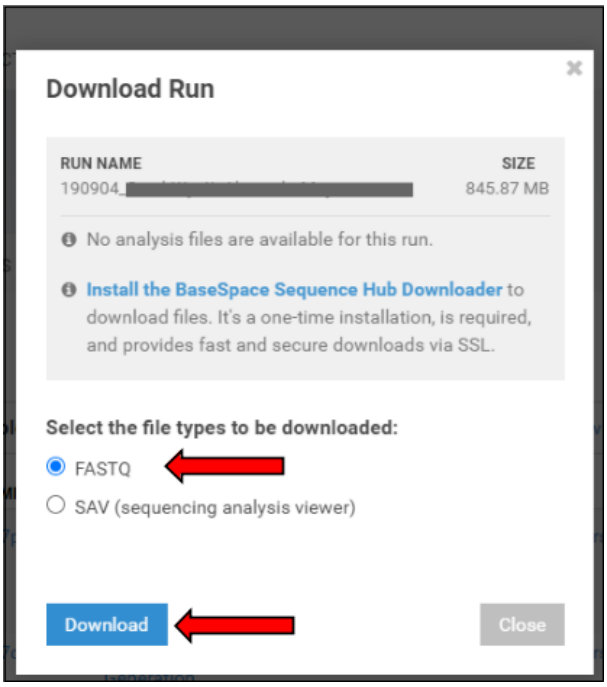
5. Under the `SUMMARY` tab, click `Download`.



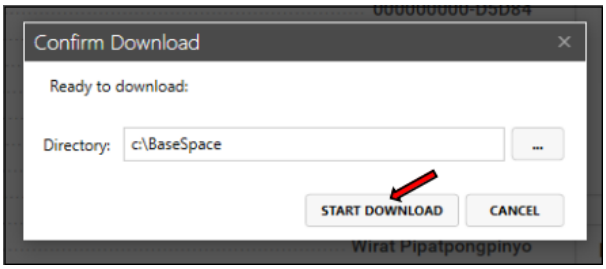
6. The `Download Run` screen will pop up.
If this
is the first time you download from BaseSpace, you will need to install the downloader software: click `Install the BaseSpace Sequence Hub Downloader`.



7. After the BaseSpace downloader has been installed, select `FASTQ` as the file type, and click `Download`.



8. The `Confirm Download` screen will pop up, where you can select a directory to download the files into. In this case, the files will be stored at `C:\BaseSpace`. Click `START DOWNLOAD`.



CHAPTER 5

Share runs and projects in BaseSpace

Note:

Purpose These two short tutorials will show you how to share sequencing runs and projects, respectively, on Illumina BaseSpace.

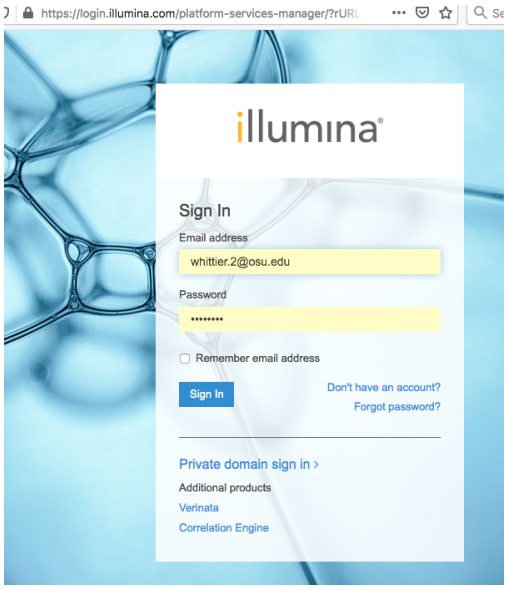
Author Wirat Pipatpongpinyo

Date July 6, 2020

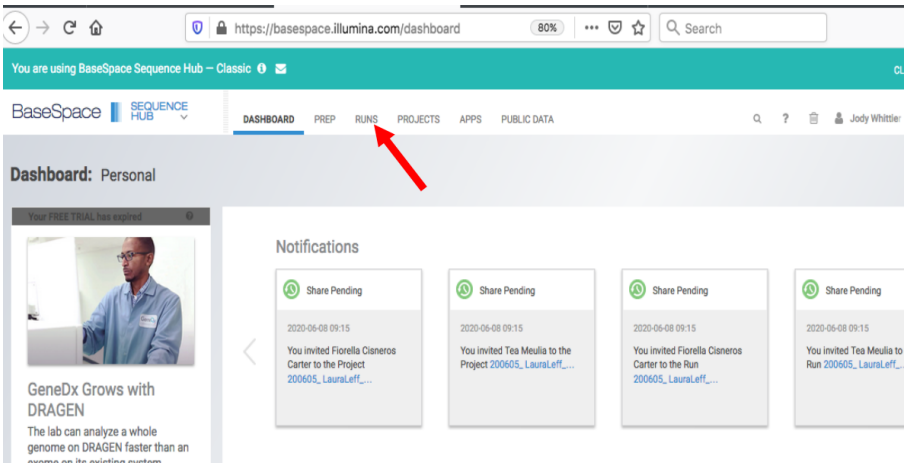
- *Sharing runs*
- *Sharing projects*

5.1 Sharing runs

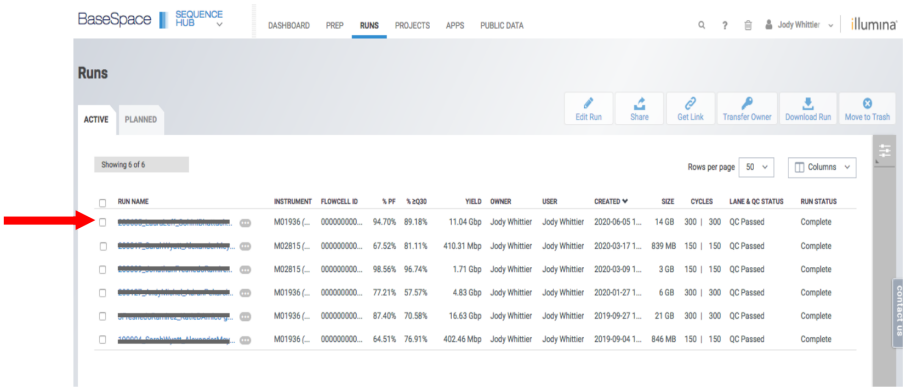
1. Go to the Illumina BaseSpace website and sign in.



2. Click on the `RUNS` tab.

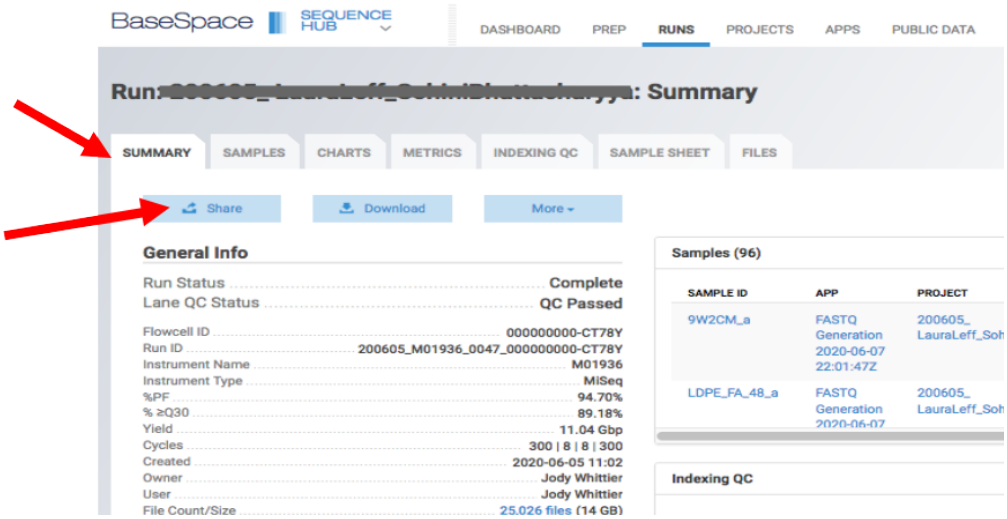


3. Select the name of the run you want to share.



4. Go to the `SUMMARY` tab.

5. Click the `SHARE` button.



6. Enter the email address (linked to a BaseSpace account) of your collaborator.

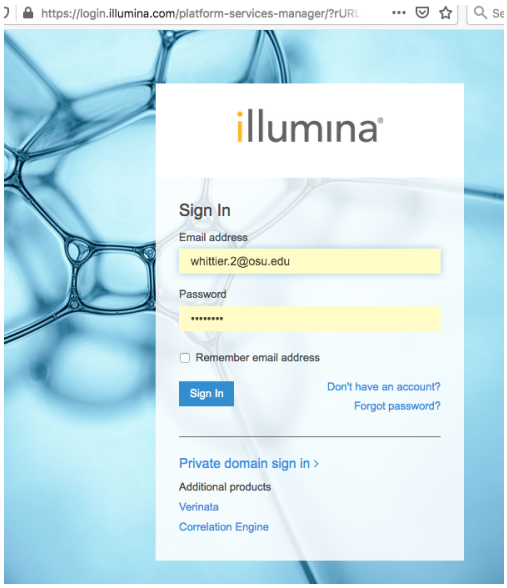
6. Click `Add Collaborator`.

(Repeat Steps 6 and 7 if you want to share with multiple collaborators.)

8. Click `Save Settings`. Your collaborator will receive an email from *basespace-noreply@illumina.com*.

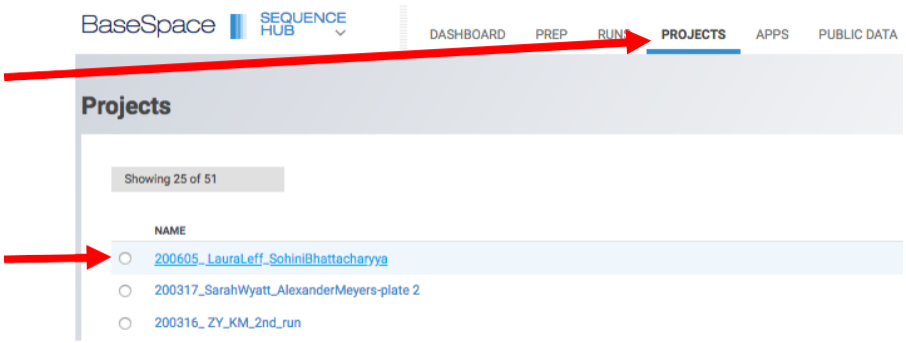
5.2 Sharing projects

1. Got to the Illumina BaseSpace website and sign in.

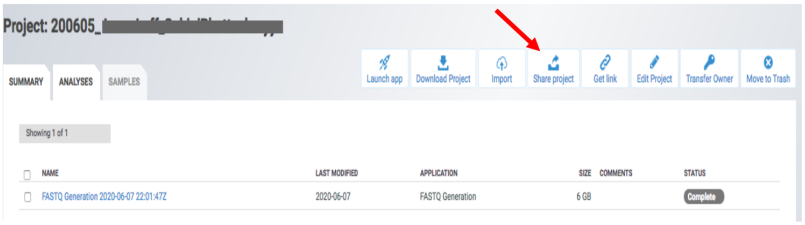


2. Click on the `PROJECTS` tab.

3. Select the Project that you would like to share.



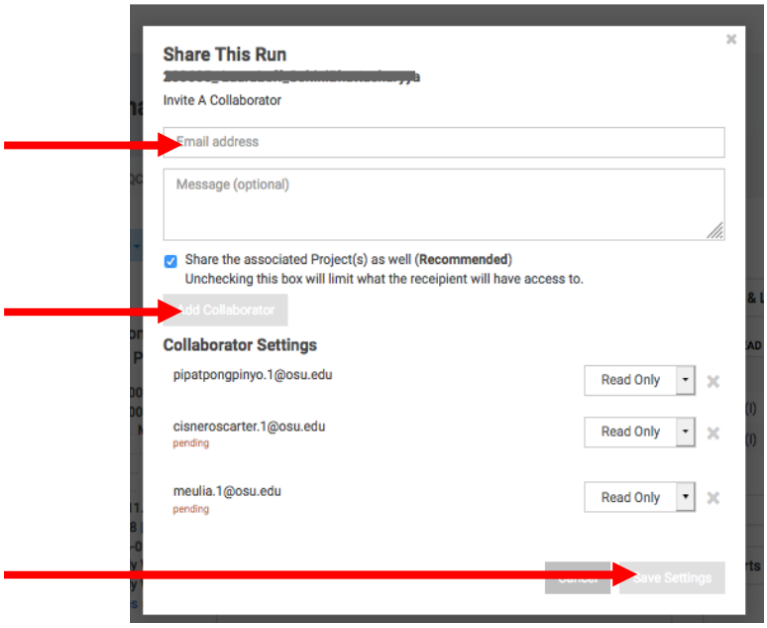
4. Click the `Share Project` button.



5. Enter the email address (linked to a BaseSpace account) of your collaborator.

6. Click `Add Collaborator`. (Repeat Steps 6 and 7 if you want to share with multiple collaborators.)

7. Click `Save Settings`. Your collaborator will receive an email from *basespace-noreply@illumina.com*.


Share This Run
Invite A Collaborator
Email address
Message (optional)
Share the associated Project(s) as well (Recommended)
Unchecking this box will limit what the receipient will have access to.
Collaborator Settings
pipatpongpinyo.1@osu.edu
Read Only
cisneroscarter.1@osu.edu
pending
Read Only
meulia.1@osu.edu
pending
Read Only
Save Settings


MCBL

CHAPTER 6

Using BaseMount

- *Introduction to BaseMount*
- *Getting Started with BaseMount*
  - *Example*
- *Basic analysis of fastq files*
- *Basic analysis on Alignment files (BAM)*
- *Metadata*
- *Limitations of BaseMount*

6.1 Introduction to BaseMount

What is BaseMount?

BaseMount is Illumina software that enables access to your **BaseSpace storage** as a **Linux file system** on the command line

Advantages of BaseMount:

- Have access to your Projects, Runs, and App results as your local files.
- Can run **local apps** on basespace data **without downloading data to your local computer.**
- Can save your local storage space:

> *"Although BaseMount does facilitate file download, we would recommend that since BaseMount allows convenient, fast, cached access to your BaseSpace metadata and files, you may find that many*

operations can be carried out without the need to download locally. During our testing, we have used BaseMount to grep through fastq files, extract blocks of reads from bam files and even use IGV on the bam files directly all without downloading files locally. This can be more convenient than including a download step and saves on the overheads of local storage." -Illumina

Official page

6.2 Getting Started with BaseMount

Quick Install

```
sudo bash -c "$(curl -L https://basemount.basespace.illumina.com/install/)"
```

Manual install

Supported Operating Systems: Ubuntu, CentOS

Ubuntu 14 & 15:

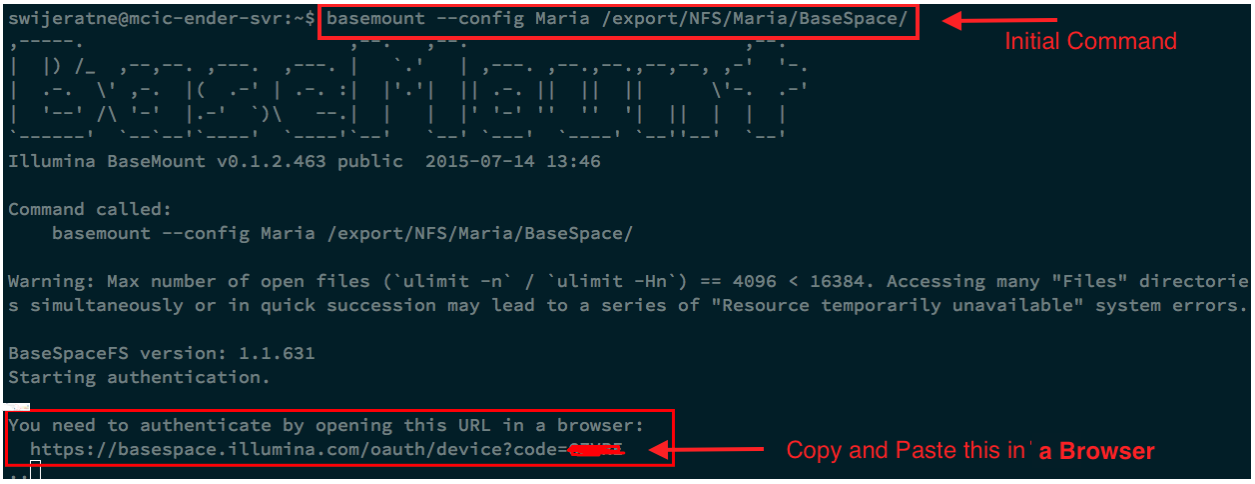
```
wget https://bintray.com/artifact/download/basespace/BaseSpaceFS-DEB/bsfs_1.1.631-1_amd64.deb
wget https://bintray.com/artifact/download/basespace/BaseMount-DEB/basemount_0.1.2.463-20150714_amd64.deb
sudo dpkg -i --force-confmiss bsfs_1.1.631-1_amd64.deb
sudo dpkg -i basemount_0.1.2.463-20150714_amd64.deb
```

Mounting Your BaseSpace Account

```
basemount --config {config_file_prefix} {mount-point folder}
basemount --config user1 ~/BaseSpace_Mount
```

6.2.1 Example

```
mkdir /export/NFS/Saranga/BaseSpace
mkdir /export/NFS/Maria/BaseSpace
basemount --config Maria /export/NFS/Maria/BaseSpace/
```



Then open your internet browser:

After you click "Accept",

```
It worked!
Enter new passphrase for access token encryption:
```
Enter a Password

```
Enter same passphrase again:
```
Re-Enter the password

```
Your identification has been saved with the new passphrase.
Api Server: https://api.basespace.illumina.com//

Mounting BaseSpace account.
To unmount, run: basemount --unmount /export/NFS/Maria/BaseSpace/

swijeratne@mcic-ender-svr:~$
```

To access the folder, type:

```
1 basemount /export/NFS/Maria/BaseSpace/
2
```

To see the contents in the Projects:

```
1 ls -lsh /export/NFS/Saranga/BaseSpace/Projects/HiSeq\ 2500\ -\ v4\ reagents\:\
  ↪TruSeq\ PCR\ Free\ \(4\ replicates\ of\ NA12877\)/Samples/NA12877_*/Files/
```

```
/export/NFS/Saranga/BaseSpace/Projects/MiSeq v3: TruSeq Targeted RNA Expression (NFkB
↪& Cell Cycle: Human Brain-Liver-UHRR)/Samples/Brain10/Files/:
total 85M
85M -r--r--r-- 1 root root 85M Oct  5 14:09 Brain10_S22_L001_R1_001.fastq.gz

/export/NFS/Saranga/BaseSpace/Projects/MiSeq v3: TruSeq Targeted RNA Expression (NFkB
↪& Cell Cycle: Human Brain-Liver-UHRR)/Samples/Brain11/Files/:
total 62M
62M -r--r--r-- 1 root root 62M Oct  5 14:09 Brain11_S23_L001_R1_001.fastq.gz
```

6.3 Basic analysis of fastq files

You can get basic information about your `fastq` files without having to download them. For instance:

- View sequences inside Fastq files
- Get the number of reads for each `fastq` file
- Get basic statistics and read length distribution

Example: View your data

```
zcat /export/NFS/Saranga/BaseSpace/Projects/MiSeq\ v3\:\ TruSeq\ Targeted\ RNA\␣
↪Expression\ \(NFkB\ \&\ Cell\ Cycle\:\ Human\ Brain-Liver-UHRR\)/Samples/Brain1/
↪Files/Brain1_S13_L001_R1_001.fastq.gz | head -n 4
```

```
@M03438:48:000000000-AGGNU:1:1101:11792:1006 1:N:0:13
NTCAATCCCCAGCAGTGGAATAAGGCCTGTTGTTCCTTGCAGTGGATCCTG
+
#88ABFFGCFEEG<FF<FDFFEGGFGGFCFGFFGGEGGGGGGFGGFGGGGG
```

Example: Count the number of sequences using native Linux commands

```
zcat  /export/NFS/Saranga/BaseSpace/Projects/MiSeq\ v3\:\ TruSeq\ Targeted\ RNA\␣
↪Expression\ \(NFkB\ \&\ Cell\ Cycle\:\ Human\ Brain-Liver-UHRR\)/Samples/Brain1/
↪Files/Brain1_S13_L001_R1_001.fastq.gz | grep -c "@M03438:"
```

```
838876
```

FILE SIZE 85M

TIME TAKEN 1.327s

Example: Count the number of sequneces using using fastqutils .

```
fastqutils names /export/NFS/Saranga/BaseSpace/Projects/MiSeq\ v3\:\ TruSeq\␣
↪Targeted\ RNA\ Expression\ \(NFkB\ \&\ Cell\ Cycle\:\ Human\ Brain-Liver-UHRR\)/
↪Samples/Brain1/Files/Brain1_S13_L001_R1_001.fastq.gz | wc -l
```

```
838876
```

FILE SIZE 85M

TIME TAKEN 23.00s

Example: Get the Length distribution and other statistics

```
fastqutils stats /export/NFS/Saranga/BaseSpace/Projects/MiSeq\ v3\:\ TruSeq\␣
↪Targeted\ RNA\ Expression\ \(NFkB\ \&\ Cell\ Cycle\:\ Human\ Brain-Liver-UHRR\)/
↪Samples/Brain1/Files/Brain1_S13_L001_R1_001.fastq.gz
```

```
Space:   basespace
Pairing: Fragmented
Quality scale: Illumina
Number of reads:  838876

Length distribution
Mean: 51.0
StdDev:  0.0
Min:  51
25 percentile: 51
Median:  51
75 percentile: 51
Max:  51
```

(continues on next page)

(continued from previous page)

```
Total:   838876

Quality distribution
pos   mean   stdev min   25pct 50pct 75pct max   count
1  33.5948650337  2.15033983948  2  34 34 34 34 838876
2  33.6285434319  2.01137931152  12 34 34 34 34 838876
3  33.6910246568  1.81306178651  11 34 34 34 34 838876
4  33.6861812711  1.84720681841  11 34 34 34 34 838876
5  33.6998292954  1.73881480815  12 34 34 34 34 838876
6  37.2788564698  2.49830369426  2  38 38 38 38 838876
7  37.3219820331  2.39576158974  2  38 38 38 38 838876
8  37.1795640834  2.72499009837  2  38 38 38 38 838876
9  37.158373824   2.76769991544  10 38 38 38 38 838876
10 37.0991517221  2.93041515545  10 38 38 38 38 838876
11 37.080357526   3.00286915879  10 38 38 38 38 838876
12 37.0752506926  2.93530617165  10 38 38 38 38 838876
13 37.1372705859  2.85104291311  10 38 38 38 38 838876
14 37.0559641711  3.00820427859  10 38 38 38 38 838876
15 37.0753877808  2.99577677236  10 38 38 38 38 838876
16 37.0950426523  2.92821324956  10 38 38 38 38 838876
17 37.204973083   2.64727500649  10 38 38 38 38 838876
18 37.1618308308  2.75627448135  10 38 38 38 38 838876
19 37.0932426247  2.92692822638  10 38 38 38 38 838876
20 37.1103548081  2.89001817524  10 38 38 38 38 838876
21 37.117659821   2.90476888945  10 38 38 38 38 838876
22 37.1280034236  2.80218109494  10 38 38 38 38 838876
23 36.9527546383  3.18334971719  9  37 38 38 38 838876
24 37.1825096915  2.70092644993  10 38 38 38 38 838876
25 37.2478948021  2.58021368225  10 38 38 38 38 838876
26 37.1056830807  3.04029966339  10 38 38 38 38 838876
27 37.0417129588  3.21755695865  9  38 38 38 38 838876
28 36.9245287742  3.46922882082  9  38 38 38 38 838876
29 37.0233538687  3.22132395112  9  38 38 38 38 838876
30 36.9768440151  3.36846192959  9  38 38 38 38 838876
31 36.9019378311  3.51733823479  10 38 38 38 38 838876
32 37.0442532627  3.15497307231  10 38 38 38 38 838876
33 37.0763462061  3.05842992907  10 38 38 38 38 838876
34 36.9112836701  3.51048896466  10 38 38 38 38 838876
35 36.871104907   3.50954115324  10 37 38 38 38 838876
36 36.9835911386  3.29022069717  9  38 38 38 38 838876
37 36.9526103977  3.40894509478  9  38 38 38 38 838876
38 36.9730198504  3.35198104312  10 38 38 38 38 838876
39 36.925962836   3.42925499742  9  38 38 38 38 838876
40 36.9641508399  3.3749237703   10 38 38 38 38 838876
41 36.9701290775  3.37108719426  9  38 38 38 38 838876
42 36.925310773   3.46541098262  9  38 38 38 38 838876
43 36.4323821399  4.37591782904  9  37 38 38 38 838876
44 36.6849510536  3.84804558398  10 37 38 38 38 838876
45 36.8000574578  3.71757395575  10 37 38 38 38 838876
46 36.743696327   3.89392229051  9  37 38 38 38 838876
47 36.6721195981  4.06597933483  10 37 38 38 38 838876
48 36.7998965282  3.73484635736  9  37 38 38 38 838876
49 36.9068074423  3.46537970313  10 38 38 38 38 838876
50 36.828634983   3.67970695764  10 37 38 38 38 838876
51 36.7344208202  3.75309659394  9  37 38 38 38 838876

Average quality string
```

(continues on next page)

(continued from previous page)

```
BBBBBFFFFFFFFFFFFFFFFFEFFFFFEFEEFFEEEEEEEEEEEEEEEEE
```

FILE SIZE 85M

TIME TAKEN 1.10m

6.4 Basic analysis on Alignment files (BAM)

Example: Check bam headers

```
samtools view -H /export/NFS/Saranga/BaseSpace/Projects/MiSeq\ v3\:\ TruSeq\␣
↪Targeted\ RNA\ Expression\ \(NFkB\ \&\ Cell\ Cycle\:\ Human\ Brain-Liver-UHRR\)/
↪AppSessions/TopHat\ Alignment\:\ No\ cSNP\ or\ RNA\ Editing\ found\ \(36\ Samples\)/
↪AppResults.26970091.Brain1/Files/alignments/Brain1.alignments.bam
```

```
@HD   VN:1.0   SO:coordinate
@RG   ID:19 SM:Brain1
@SQ   SN:chrM  LN:16571
@SQ   SN:chr1  LN:249250621
@SQ   SN:chr2  LN:243199373
@SQ   SN:chr3  LN:198022430
@SQ   SN:chr4  LN:191154276
@SQ   SN:chr5  LN:180915260
@SQ   SN:chr6  LN:171115067
@SQ   SN:chr7  LN:159138663
@SQ   SN:chr8  LN:146364022
@SQ   SN:chr9  LN:141213431
@SQ   SN:chr10 LN:135534747
@SQ   SN:chr11 LN:135006516
@SQ   SN:chr12 LN:133851895
@SQ   SN:chr13 LN:115169878
@SQ   SN:chr14 LN:107349540
@SQ   SN:chr15 LN:102531392
@SQ   SN:chr16 LN:90354753
@SQ   SN:chr17 LN:81195210
@SQ   SN:chr18 LN:78077248
@SQ   SN:chr19 LN:59128983
@SQ   SN:chr20 LN:63025520
@SQ   SN:chr21 LN:48129895
@SQ   SN:chr22 LN:51304566
@SQ   SN:chrX  LN:155270560
@SQ   SN:chrY  LN:59373566
@PG   ID:TopHat   VN:2.0.7 CL:/illumina/development/IsisRNA/2.4.19.5/IsisRNA_Tools/
↪bin/tophat --bowtie1 --read-realign-edit-dist 1 --segment-length 24 -o /data/input/
↪Alignment/samples/Brain1/replicates/Brain1/tophat_main -p 1 --GTF /illumina/
↪development/Genomes/Homo_sapiens/UCSC/hg19/Annotation/Genes/genes.gtf --rg-id 19 --
↪rg-sample Brain1 --library-type fr-firststrand --no-coverage-search --keep-fasta-
↪order /illumina/development/Genomes/Homo_sapiens/UCSC/hg19/Sequence/BowtieIndex/
↪genome /data/input/Alignment/samples/Brain1/replicates/Brain1/filtered/Brain1_S20_
↪L001_R1_001.fastq.gz
```

FILE SIZE 17M

TIME TAKEN 0.006s

Example: Check basic statistics on a Bam file

```
bamutils stats /export/NFS/Saranga/BaseSpace/Projects/MiSeq\ v3\:\ TruSeq\ Targeted\ RNA\ Expression\ \(NFkB\ \&\ Cell\ Cycle\:\ Human\ Brain-Liver-UHRR\)/AppSessions/TopHat\ Alignment\:\ No\ cSNP\ or\ RNA\ Editing\ found\ \(36\ Samples\)/AppResults.26970091.Brain1/Files/alignments/Brain1.alignments.bam
```

```
Reads:   766531
Mapped:  766531
Unmapped:   0

Flag distribution
[0x010] Reverse complimented  383242   50.00%
[0x100] Secondary alignment   47 0.01%


Reference counts  count
chr1  32252
chr10 6829
chr11 45127
chr12 74524
chr13 24336
chr14 81664
chr15 254
chr16 5704
chr17 46662
chr18 9922
chr19 25644
chr2  32527
chr20 10708
chr21 22
chr22 7805
chr3  83585
chr4  42487
chr5  47865
chr6  106512
chr7  18024
chr8  6921
chr9  25334
chrM  0
chrX  31823
chrY  0
```

6.5 Metadata

In each directory, BaseMount provides a number of hidden files with extra BaseSpace metadata. These are hidden files and their names start with a ".".

```
cd /export/NFS/Saranga/BaseSpace/Projects/MiSeq\ v3\:\ TruSeq\ Targeted\ RNA\ Expression\ \(NFkB\ \&\ Cell\ Cycle\:\ Human\ Brain-Liver-UHRR\)/Samples/
ls -l .?* #List only the hidden files
```

```
-r-------- 1 swijeratne swijeratne        149 Nov 16 11:29 .curl
.
.
```

(continues on next page)

(continued from previous page)

```
.
-r--r--r-- 1 swijeratne swijeratne      40674 Nov 16 11:29 .json
```

Display content of the .json file

```
cat .json
```

Query through a .json file qith "jq"

```
cat .json | jq '.Response.Items[] | select(.NumReadsPF ) | {Name: .Name, PF : .
↪NumReadsPF}'
cat .json | jq '.Response.Items[] | select(.NumReadsPF ) | "\( .Name)\t\(.NumReadsPF)"
↪'
cat .json | jq '.Response.Items[] | select(.NumReadsPF > 747912) | "\( .Name)\t\(.
↪NumReadsPF)"'
```

6.6 Limitations of BaseMount

According to Illumina,

Every new directory access made by BaseMount relies on FUSE, the BaseSpace API and the user's credentials. This mechanism means that, as currently available, BaseMount does not support the following types of access:

- Cluster access, where many compute nodes can access the files. FUSE mounted file systems are per-host and cannot be accessed from many hosts without additional infrastructure.
- BaseMount also doesn't refresh files or directories. In order to reflect changes done via the Web GUI in your command line tree, you currently need to unmount (basemount –unmount) and restart BaseMount.
- The Runs Files directory is not mounted automatically for you as there can be 100k + files available in that mount which can take a couple minutes to load for really large runs. You can still mount this directory manually if needed.
- In general, lots of concurrent requests can cause stability issues on a resource constrained system. Keep in mind, this is an early release and stability will increase.

CHAPTER 7

Download from hudsonalpha.org

Note:

Required OS OS x or Linux. Windows users, please contact Maria Elena Hernandez-Gonzalez

Software wget / curl

Terminal emulator

- Terminal (OS x)
- Genome Terminal or Other Emulator (Linux)

Author This document was created by Saranga Wijeratne

7.1 Download

1. Create a Samples.txt file with your sample links (the links are provided in the Excelsheet) as follows:

```
#Content of the Samples.txt
http://mysample.download.org/dl/d4/Meulia/myprojectnumber/data_150522/C6V7FANXX_
↪s8_0_TruseqHTDual_D712-TruseqHTDual_D508_SL104628.fastq.gz
http://mysample.download.org/dl/d4/Meulia/myprojectnumber/data_150522/C6V7FANXX_
↪s3_0_TruseqHTDual_D703-TruseqHTDual_D501_SL104549.fastq.gz
http://mysample.download.org/dl/d4/Meulia/myprojectnumber/data_150522/C6V7FANXX_
↪s5_0_TruseqHTDual_D709-TruseqHTDual_D506_SL104602.fastq.gz
http://mysample.download.org/dl/d4/Meulia/myprojectnumber/data_150522/C6V7FANXX_
↪s8_0_TruseqHTDual_D705-TruseqHTDual_D501_SL104565.fastq.gz
```

2. Use the Terminal and navigate to the location where Samples.txt is saved.

```
#If your Samples.txt is saved under ~/Downloads
$ cd ~/Downloads
```

3. On OS x, issue the following command to download your files:

```
$ for f in $(cat Samples.txt ); do curl --progress-bar -O $f; done
```

4. On Linux, issue the following command to download your files,

```
$ for f in $(cat Samples.txt ); do wget -v $f; done
```

7.2 Check *checksum*

To detect errors which may have been introduced during the downloading, you have to run checksum on your downloaded files.

1. Navigate to the location where you have downloaded your files.

```
#If your files are saved under ~/Downloads
$ cd ~/Downloads
```

2. Then, if you're on OS x Terminal, type in the following command:

```
$ md5 *
```

```
MD5 (C6V7FANXX_s3_0_TruseqHTDual_D703-TruseqHTDual_D501_SL104549.fastq.gz) =␣
↪d41d8cd428f00b204e9800998ecf8427e
MD5 (C6V7FANXX_s5_0_TruseqHTDual_D709-TruseqHTDual_D506_SL104602.fastq.gz) =␣
↪d49d8cdf00j204e9800998ecf8427e
MD5 (C6V7FANXX_s8_0_TruseqHTDual_D705-TruseqHTDual_D501_SL104565.fastq.gz) =␣
↪d47d8cd98dfds0b204e9800998ecf8427e
MD5 (C6V7FANXX_s8_0_TruseqHTDual_D712-TruseqHTDual_D508_SL104628.fastq.gz) =␣
↪d42d8cd98f00bdfse9800998ecf8427e
```

If you're on Linux terminal, type in the following commmand:

```
$ md5sum *
```

```
d41d8cd428f00b204e9800998ecf8427e   C6V7FANXX_s3_0_TruseqHTDual_D703-TruseqHTDual_
↪D501_SL104549.fastq.gz
d49d8cdf00j204e9800998ecf8427ed56   C6V7FANXX_s5_0_TruseqHTDual_D709-TruseqHTDual_
↪D506_SL104602.fastq.gz
d47d8cd98dfds0b204e9800998ecf8427e  C6V7FANXX_s8_0_TruseqHTDual_D705-TruseqHTDual_
↪D501_SL104565.fastq.gz
d47d8cd98dfds0b204e9800998ecf8427e  C6V7FANXX_s8_0_TruseqHTDual_D712-TruseqHTDual_
↪D508_SL104628.fastq.gz
```

Tip: Match these checksum values with the values provided in the Excelsheet. For any samples with mismatching checksum, you have to re-download the samples.

CHAPTER 8

Filter a CASAVA-generated fastq file

Note:

Required OS OS x or Linux. Windows users, please contact Saranga Wijeratne

Software Illumina CASAVA-1.8 FASTQ Filter

Purpose This document provides instructions about how to remove Passing Filter (PF) failed reads from a Fastq file

More Read more about PF here: and here

Author This document was created by Saranga Wijeratne

8.1 Software installation

Note: If you are runing this on MCBL *mcic-ender-svr*, please skip the installation. Following command will load the software module to your environment.

```
$ module load fasq_filter/0.1
```

On your own server,

Warning: If you don't have administrator privileges on the machine, you wouldn't be able run `sudo` (last command in the following code block) commands.

```
$ wget http://cancan.cshl.edu/labmembers/gordon/fastq_illumina_filter/fastq_illumina_
↪filter-0.1.tar.gz
$ tar -xzf fastq_illumina_filter-0.1.tar.gz
$ cd fastq_illumina_filter-0.1
$ make
$ sudo cp fastq_illumina_filter /usr/local/bin
```

Tip: Put your executables in `~/bin` or full-path to executables in `$PATH` in the absence of `sudo` privilages.

8.2 Filter a fastq file

Input File C8EC8ANXX_s2_1_illumina12index_1_SL143785.fastq.gz

Output File C8EC8ANXX_s2_1_illumina12index_1_SL143785.filtered.fastq.gz

```
$ zcat C8EC8ANXX_s2_1_illumina12index_1_SL143785.fastq.gz | fastq_illumina_filter -
↪vvN | gzip > C8EC8ANXX_s2_1_illumina12index_1_SL143785.filtered.fastq.gz
```

8.3 Filter multiple fastq files

Input File Fastq_filenames.txt

Output Files Individual Fastq files

1. Create a Fastq_filenames.txt file with your Fastq filenames in seperate lines as follows:

```
#Content of Samples.txt
C6V7FANXX_s8_0_TruseqHTDual_D712-TruseqHTDual_D508_SL104628.fastq.gz
C6V7FANXX_s3_0_TruseqHTDual_D703-TruseqHTDual_D501_SL104549.fastq.gz
C6V7FANXX_s5_0_TruseqHTDual_D709-TruseqHTDual_D506_SL104602.fastq.gz
C6V7FANXX_s8_0_TruseqHTDual_D705-TruseqHTDual_D501_SL104565.fastq.gz
```

2. Save the above file in the same folder with your Fastq files.
3. Use the Terminal and navigate to the location where Fastq_filenames.txt is saved.

```
#If your Fastq_filenames.txt is saved under ~/Downloads
$ cd ~/Downloads
```

4. Type in the following command to filter Fastqs in the Fastq_filenames.txt.

```
$ for f in $(cat Fastq_filenames.txt); do zcat $f | fastq_illumina_filter -vvN |␣
↪gzip > ${f%.*.fastq.gz}.filtered.fastq.gz;done
```

CHAPTER 9

Fastq adapter removal and QC

Note:

Required OS OS x or Linux. Windows users, please contact Saranga Wijeratne

Software Trimmomatic

Purpose This document provides instructions about how to remove adapters and filter low quality bases from a Fastq file

More Read more about Read trimming adapter removing here:

Author This document is created by Saranga Wijeratne

9.1 Load the software

Note: If you are runing this on MCBL *mcic-ender-svr* following command will load the software module to your environment.

```
$ module load Trimmomatic/3.2.2
```

then you can get the help how to run Trimmomatic,

```
$ java -jar $TRIMHOME/trimmomatic-0.33.jar
```

9.2 Files needed

Input Files In put files should be in fastq format/compressed fastq(.fq, .fastq, .fq.gz, .fastq.gz). Read Introduction e.g :C8EC8ANXX_s2_1_illumina12index_1_SL143785.fastq,

C8EC8ANXX_s2_1_illumina12index_1_SL143785.fastq.gz, s_1_1_sequence.txt.gz lane1_forward.fq.gz

Adapter File Currently, following Adapter sequence files are hosted in MCBL server.

- TruSeq2-PE.fa
- TruSeq2-SE.fa
- TruSeq3-PE.fa
- TruSeq3-SE.fa
- NexteraPE-PE.fa

Warning: If you want to make your own adapter sequence file, please read the The Adapter Fasta section and Making cutome clipping files here before you make your Adapter sequence file.

9.3 Code examples

Single-end fastq files

```
$java -jar $TRIMHOME/trimmomatic-0.33.jar SE -threads 12 s_1_1_sequence.txt.gz lane1_
↪forward.fq.gz ILLUMINACLIP:$TRIMHOME/adapters/TruSeq3-SE.fa:2:30:10 LEADING:3␣
↪TRAILING:3 SLIDINGWINDOW:4:15 MINLEN:36
```

Paired End Fastq Files

```
$java -jar $TRIMHOME/trimmomatic-0.33.jar PE -threads 12 C8EC8ANXX_s2_1_
↪illumina12index_1_SL143785.fastq.gz C8EC8ANXX_s2_2_illumina12index_1_SL143785.fastq.
↪gz C8EC8ANXX_s2_1_Trimmed_1P.fastq.gz C8EC8ANXX_s2_1_Trimmed_1U.fastq.gz C8EC8ANXX_
↪s2_2_Trimmed_1P.fastq.gz C8EC8ANXX_s2_2_Trimmed_1U.fastq.gz ILLUMINACLIP:$TRIMHOME/
↪adapters/TruSeq3-PE.fa:2:30:10 LEADING:3 TRAILING:3 SLIDINGWINDOW:4:15 MINLEN:36
```

Multiple fastq files

Tip: Assumption has been made that your data in "Raw_Data" folder

Input Files C6EF7ANXX_s3_1_illumina12index_10_SL100996.fastq.gz
C6EF7ANXX_s3_1_illumina12index_19_SL100997.fastq.gz C6EF7ANXX_s3_1_illumina12index_22_SL100998.fastq.gz
C6EF7ANXX_s3_1_illumina12index_25_SL100999.fastq.gz C6EF7ANXX_s3_1_illumina12index_27_SL101000.fastq.gz
C6EF7ANXX_s3_1_illumina12index_3_SL100994.fastq.gz C6EF7ANXX_s3_1_illumina12index_5_SL100995.fastq.gz
C6EF7ANXX_s3_2_illumina12index_10_SL100996.fastq.gz C6EF7ANXX_s3_2_illumina12index_19_SL100997.fastq.gz
C6EF7ANXX_s3_2_illumina12index_22_SL100998.fastq.gz C6EF7ANXX_s3_2_illumina12index_25_SL100999.fastq.gz
C6EF7ANXX_s3_2_illumina12index_27_SL101000.fastq.gz C6EF7ANXX_s3_2_illumina12index_3_SL100994.fastq.gz
C6EF7ANXX_s3_2_illumina12index_5_SL100995.fastq.gz

These are paired-end fastq files. **e.g** *C6EF7ANXX_s3_1_illumina12index_10_SL100996.fastq.gz and C6EF7ANXX_s3_2_illumina12index_10_SL100996.fastq.gz* belongs to single sample.

Adapter File $TRIMHOME/adapters/TruSeq3-PE.fa (Make sure you change this accordingly)

Output Files Each paired-end read (**e.g** *C6EF7ANXX_s3_1_illumina12index_10_SL100996.fastq.gz and C6EF7ANXX_s3_2_illumina12index_10_SL100996.fastq.gz*) will give 4 outputs:

- Q_trimmed_6EF7ANXX_s3_1_illumina12index_10_SL100996_1P.fastq.gz - for paired forwad reads
- Q_trimmed_6EF7ANXX_s3_1_illumina12index_10_SL100996_1U.fastq.gz - for unpaired forward reads
- Q_trimmed_6EF7ANXX_s3_2_illumina12index_10_SL100996_1P.fastq.gz - for paired reverse reads
- Q_trimmed_6EF7ANXX_s3_2_illumina12index_10_SL100996_1U.fastq.gz - for unpaired reverse reads

```
$cd Raw_Data #make sure you change the folder name accordingly
$mkdir Trimmed_Data # Output will be saved here
$files_1=(*_s3_1_*.fastq.gz);files_2=(*_s3_2_*.fastq.gz);sorted_files_1=($(printf "
↪%s\n" "${files_1[@]}" | sort -u));sorted_files_2=($(printf "%s\n" "${files_2[@]}" |␣
↪sort -u));for ((i=0; i<${#sorted_files_1[@]}; i+=1));do java -jar $TRIMHOME/
↪trimmomatic-0.33.jar PE -threads 12  -trimlog Trimmed_Data/log-j3.stat -phred33   $
↪{sorted_files_1[i]} ${sorted_files_2[i]} Trimmed_Data/Q_trimmed_${sorted_files_1[i]%
↪%.*}_1P.fastq.gz Trimmed_Data/Q_trimmed_${sorted_files_1[i]%%.*}-U.fastq.gz Trimmed_
↪Data/Q_trimmed_${sorted_files_2[i]%%.*}_1P.fastq.gz Trimmed_Data/Q_trimmed_${sorted_
↪files_1[i]%%.*}-U.fastq.gz ILLUMINACLIP:$TRIMHOME/adapters/TruSeq3-PE:2:30:10␣
↪LEADING:3 TRAILING:3 SLIDINGWINDOW:4:20 MINLEN:40  &>Trimmed_Data/stat.txt; done
```

CHAPTER 10

DESeq2 with phyloseq

Note:

Required OS OS x or Linux.

Software R, phyloseq R library

Purpose This document provides instructions about how to find differentially abundant OTUs for microbiome data.

More Read more about phyloseq DEseq2 here and here.

Author This document was created by Saranga Wijeratne.

10.1 Software installation

Note: If you are runing this on MCBL *mcic-sel019-d*, please skip the installation. The following command in R-Studio will load the software module to your environment.

```
> library("phyloseq"); packageVersion("phyloseq")
[1] '1.16.2' # version
```

Install the `phyloseq` package as follows:

```
> source('http://bioconductor.org/biocLite.R')
> biocLite('phyloseq')
```

10.2 Import data with phyloseq

For this step, you need Biom and mapping files generated by the Qiime pipeline.

Input biom file *otu_table_mc10_w_tax.biom*

Qiime mapping file *mapping.txt*

Output file *DESeq2_Out*

Copy all the input files to your "Working Directory" before you execute the following commands.

```
# Filenames:
biom_file <- "otu_table_mc10_w_tax.biom"
mapping_file <- "mapping.txt"

# Import the biom file with phyloseq:
biom_otu_tax <- import_biom(biom_file)

# Import the mapping file with phyloseq:
mapping_file <- import_qiime_sample_data(mapping_file)

# Merge biom and mapping files with phyloseq:
merged_mapping_biom <- merge_phyloseq(biom_otu_tax,mapping_file)

# Set column names in the taxa table:
colnames(tax_table(merged_mapping_biom)) <- c("kingdom", "Phylum", "Class", "Order",
↪"Family", "Genus", "species")
```

Now, your merged mapping and Biom output should look as follows:

```
merged_mapping_biom

# phyloseq-class experiment-level object
# otu_table()   OTU Table:         [ 315 taxa and 9 samples ]
# sample_data() Sample Data:       [ 9 samples by 8 sample variables ]
# tax_table()   Taxonomy Table:    [ 315 taxa by 7 taxonomic ranks ]
```

The mapping file should look like this:

```
head(mapping_file)

# Sample Data:        [40 samples by 7 sample variables]:
# X.SampleID BarcodeSequence LinkerPrimerSequence InputFileName IncubationDate␣
↪Treatment Description
# S1           S1              NA                   NA      S1.fasta              0  ␣
↪    CO         CO1
# S2           S2              NA                   NA      S2.fasta              0  ␣
↪    CO         CO2
# S3           S3              NA                   NA      S3.fasta              0  ␣
↪    CO         CO3
# S4           S4              NA                   NA      S4.fasta             15  ␣
↪    CO         CO4
# S5           S5              NA                   NA      S5.fasta             15  ␣
↪    CO         CO5
```

To remove taxonomy level tags assigned to each level (**k__**, **p__**, etc..), issue the following commands:

```
tax_table(merged_mapping_biom) <- gsub("k__([[:alpha:]])", "\\1", tax_table( merged_
↪mapping_biom))
tax_table(merged_mapping_biom) <- gsub("p__([[:alpha:]])", "\\1", tax_table(merged_
↪mapping_biom))
tax_table(merged_mapping_biom) <- gsub("c__([[:alpha:]])", "\\1", tax_table(merged_
↪mapping_biom))
tax_table(merged_mapping_biom) <- gsub("o__([[:alpha:]])", "\\1", tax_table(merged_
↪mapping_biom))
tax_table(merged_mapping_biom) <- gsub("f__([[:alpha:]])", "\\1", tax_table(merged_
↪mapping_biom))
tax_table(merged_mapping_biom) <- gsub("g__([[:alpha:]])", "\\1", tax_table(merged_
↪mapping_biom))
tax_table(merged_mapping_biom) <- gsub("s__([[:alpha:]])", "\\1", tax_table(merged_
↪mapping_biom))
tax_table(merged_mapping_biom) <- gsub("p__(\\[)","\\1", tax_table(merged_mapping_
↪biom))
tax_table(merged_mapping_biom) <- gsub("c__(\\[)","\\1", tax_table(merged_mapping_
↪biom))
tax_table(merged_mapping_biom) <- gsub("o__(\\[)","\\1", tax_table(merged_mapping_
↪biom))
tax_table(merged_mapping_biom) <- gsub("f__(\\[)","\\1", tax_table(merged_mapping_
↪biom))
tax_table(merged_mapping_biom) <- gsub("g__(\\[)","\\1", tax_table(merged_mapping_
↪biom))
tax_table(merged_mapping_biom) <- gsub("s__(\\[)","\\1", tax_table(merged_mapping_
↪biom))
```

10.3 Testing for differential abundance among OTUs

Input file *merged_mapping_biom*

Output files *DESeq2_Out.txt*

1. Load the DESeq2 package into your R environment

```
library("DESeq2")
packageVersion("DESeq2")
    # [1] '1.12.4'
```

2. Assign DESeq2 output name and padj-cutoff

```
filename_out <- "DESeq2_Out.txt"
alpha <- 0.01
```

3. Convert to DESeqDataSet format The `phyloseq_to_deseq2()` function converts the phyloseq-format microbiome data (i.e merged_mapping_biom) to a `DESeqDataSet` with dispersion estimated, using the experimental design formula (i.e ~ Treatment):

```
diagdds <- phyloseq_to_deseq2(merged_mapping_biom, ~ Treatment)
```

4. Run DESeq

```
diagdds <- DESeq(diagdds, test="Wald", fitType="parametric")

## estimating size factors
```

(continues on next page)

(continued from previous page)

```
## estimating dispersions
## gene-wise dispersion estimates
## mean-dispersion relationship
## final dispersion estimates
## fitting model and testing
```

Warning: If you are getting the following error:

```
Error in estimateSizeFactorsForMatrix(counts(object), locfunc, geoMeans =␣
↪geoMeans) : every gene contains at least one zero, cannot compute log␣
↪geometric means
Calls: estimateSizeFactors ... estimateSizeFactors -> .local ->␣
↪estimateSizeFactorsForMatrix
```

Then please execute the following code (see here for more info):

```
gm_mean <- function(x, na.rm=TRUE){ exp(sum(log(x[x > 0]), na.rm=na.rm) /␣
↪length(x))}
geoMeans <- apply(counts(diagdds), 1, gm_mean)
diagdds <- estimateSizeFactors(diagdds, geoMeans = geoMeans)
diagdds <- DESeq(diagdds, test="Wald", fitType="parametric")
```

5. Process the results

The `results` function creates a table of results. Then the `res` table is filtered by `padj < alpha`.

```
res <- results(diagdds, cooksCutoff = FALSE)
sigtab <- res[which(res$padj < alpha), ]
sigtab <- cbind(as(sigtab, "data.frame"), as(tax_table(merged_mapping_
↪biom)[rownames(sigtab), ], "matrix")) # Bind taxonomic info to final results␣
↪table
write.csv(sigtab, as.character(filename_out)) # Write `sigtab` to file
```

CHAPTER 11

GBS

This documents a pipeline for the analysis of GBS (Genotyping-By-Sequencing) data.

Note:

Required OS OS x or Linux.

Software Tassel 5

Documentation Tassel 5.0 Wiki

Author This document is created by Saranga Wijeratne

11.1 File formats

1. File formats that will be using in this analysis:
    - HDF5
    - VCF
    - Hapmap
    - Plink
    - Projection Alignment
    - Phylip
    - FASTA, more
    - Fastq
    - **Numerical Data**
        - Phenotype Format

  - Trait Format
  - Covariate Format
  - Marker Values as Numerical Co-variates
- Square Numerical Matrix
- Table Report
- TOPM (Tags on Physical Map)

11.2 Files you need to have

The following files need to be present before you start the pipeline:

1. Sequencing data files (.fastq or .fastq.gz)

Note: Fastq files should follow this naming convention: (more on page 7 here) - FLOWCELL_LANE_fastq.gz (e.g. AL2P1XXX_2_fastq.gz) - FLOWCELL_s_LANE_fastq.gz (e.g. AL2P1XXX_s_2_fastq.gz) - code_FLOWCELL_s_LANE_fastq.gz (e.g.: 00000000_AL2P1XXX_s_2_fastq.gz)

```
# To rename a .fastq.gz file:
$ mv  AE_S1_L001_R1_001.fastq.gz AL2P1XXX_1_fastq.gz
```

2. GBSv2 key file (example key file, more information).
3. A reference genome.

11.3 GBSv2 pipeline plugins

Plugin	Description
GBSSeqToTagDBPlugin	Executed to pull distinct tags from the database and export them in the fastq format. More
TagExportToFastqPlugin	Retrieves distinct tags stored in the database and reformats them to a FASTQ file. More
SAMToGBSdbPlugin	Used to identify SNPs from aligned tags using the GBS DB. More
DiscoverySNPCallerPluginV2	Takes a GBSv2 database file as input and identifies SNPs from the aligned tags. More
SNPQualityProfilerPlugin	Scores all discovered SNPs for various coverage depth and genotypic statistics for a given set of taxa. More
UpdateSNPPositionQualityPlugin	Reads a quality score file to obtain quality score data for positions stored in the snpposition table. More
SNPCutPosTagVerificationPlugin	Allows user to specify a Cut or SNP position for which they would like data printed. More
GetTagSequenceFromDBPlugin	Takes an existing GBSv2 SQLite database file as input and returns a tab-delimited file containing a list of Tag Sequences stored in the specified database file. More
ProductionSNPCallerPluginV2	Converts data from fastq and keyfile to genotypes then adds these to a genotype file in VCF or HDF5 format. More

11.4 GBSv2 pipeline

1. Load Tassel 5.0 module

```
$ module load Tassel/5.0
```

2. Useful commands

To check all the plugins available, type:

```
$ run_pipeline.pl -Xmx200g -ListPlugins
```

To check all the parameters for given Plugin, *Ex: GBSSeqToTagDBPlugin*, type:

```
$ run_pipeline.pl -fork1 -GBSSeqToTagDBPlugin   -endPlugin -runfork1
```

Tip: Users are recommended to read more about GBS command line options in here. Page 1-2

3. File preparation

Create necessary folders and copy your raw data (fastqs), reference file and key file to appropriate folder:

```
$ mkdir fastq ref key db tagsForAlign hd5
```

4. Execute the pipeline

```
$ run_pipeline.pl -Xmx200g -fork1 -GBSSeqToTagDBPlugin -i fastq  -k key/Tomato_key.
↪txt -e ApeKI -db db/Tomato.db  -kmerLength 85 -mnQS 20  -endPlugin -runfork1
$ run_pipeline.pl -fork1 -TagExportToFastqPlugin  -db db/Tomato.db -o tagsForAlign/
↪tagsForAlign.fa.gz -c 5  -endPlugin -runfork1
$ cd ref
$ bwa index -a is S_lycopersicum_chromosomes.2.50.fa
$ cd ../
$ bwa samse ref/S_lycopersicum_chromosomes.2.50.fa tagsForAlign/tagsForAlign.sai␣
↪tagsForAlign/tagsForAlign.fa.gz > tagsForAlign/tagsForAlign.sam
$ run_pipeline.pl -fork1 -SAMToGBSdbPlugin -i tagsForAlign/tagsForAlign.sam  -db db/
↪Tomato.db  -aProp 0.0 -aLen 0 -endPlugin -runfork1
$ run_pipeline.pl -fork1 -DiscoverySNPCallerPluginV2 -db db/Tomato.db  -sC "chr00" -
↪eC "chr12" -mnLCov 0.1 -mnMAF 0.01  -endPlugin -runfork1
$ run_pipeline.pl -fork1 -ProductionSNPCallerPluginV2 -db db/Tomato.db  -e ApeKI -i␣
↪fastq -k key/Tomato_key2.txt  -kmerLength 85 -mnQS 20 -o hd5/HapMap_tomato.h5 -
↪endPlugin -runfork1
```

CHAPTER 12

A basic microbiome analysis

12.1 QIIME

Note:

Required OS OS x or Linux

Software Qiime 1.9

Documentation Qiime tutorial

Author This document was created by Saranga Wijeratne

12.1.1 File formats

This section includes descrption of varies file formats, including Qiime scripts, and parameters files. Read more here

Qiime Script index: Index of all the scripts used in Qiime.

Metadata mapping files: Metadata mapping files provide per-sample metadata.

Tip: A metadata mapping file example is given here. Read the section carefully. If you are planning to create the mapping file by hand, read this section.

Biom file: OTU observation file. Read more here.

12.1.2 Download the files

For this tutorial, we will use Mothur tutorial data from Schloss Wiki. These data are 16s rRRNA Amplicons sequenced with Illumina MiSeq.

1. Create folders

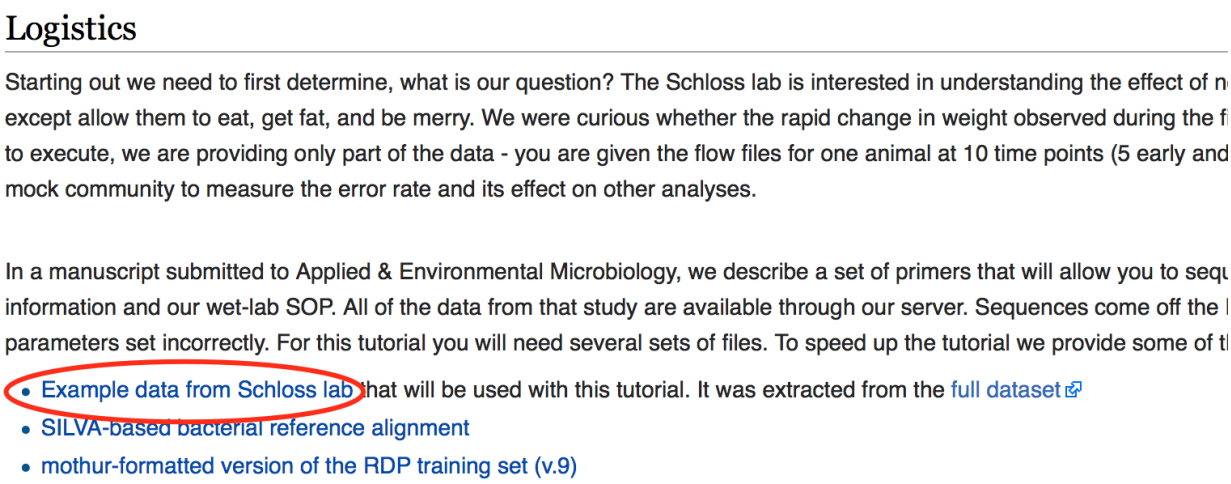
Make a new directory `MCICQiime` and then *cd* to move into the dirctory.

```
$ mkdir  MCICQiime
$ cd MCICQiime
```

2. Download data

Download data from Schloss Wiki

For this tutorial download only dataset shown in the image below (i.e Example data from Scholoss lab).



Logistics

Starting out we need to first determine, what is our question? The Schloss lab is interested in understanding the effect of n
except allow them to eat, get fat, and be merry. We were curious whether the rapid change in weight observed during the f
to execute, we are providing only part of the data - you are given the flow files for one animal at 10 time points (5 early and
mock community to measure the error rate and its effect on other analyses.

In a manuscript submitted to Applied & Environmental Microbiology, we describe a set of primers that will allow you to sequ
information and our wet-lab SOP. All of the data from that study are available through our server. Sequences come off the
parameters set incorrectly. For this tutorial you will need several sets of files. To speed up the tutorial we provide some of t

- Example data from Schloss lab that will be used with this tutorial. It was extracted from the full dataset
- SILVA-based bacterial reference alignment
- mothur-formatted version of the RDP training set (v.9)

Inside the `MCICQiime` dir, issue the following command to get the data. The data has been zipped, and we will use `unzip -j` to extract all the files to same directory we are in right now.

```
$ wget http://www.mothur.org/w/images/d/d6/MiSeqSOPData.zip
$ unzip -j  MiSeqSOPData.zip
```

Rename the files for downstream analyses:

```
$ for f in *.fastq; do mv $f  ${f%%_L*}.fastq;done
```

And explanation of the preceding commands:

- `for f in *.fastq;` reads any file that ends with *.fastq*, one at a time.
- `do` starts the body of the *for* loop.
- `mv $f do mv $f ${f%%_L*}.fastq;` rename $f (i.e F3D0_S188_L001_R1_001.fastq) to ${f%%_L*}.fastq (i.e F3D0_S188.fastq)
- `done` finishes the loop.

3. An explanation of the data

The files and experiment are described in the Schloss Wiki.

> Because of the large size of the original dataset (3.9 GB) we are giving you 21 of the 362 pairs of fastq files. For example, you will see two files: `F3D0_S188_L001_R1_001.fastq` and `F3D0_S188_L001_R2_001.fastq`. These two files correspond to Female 3 on Day 0 (i.e. the day of weaning). The first and all those with R1 correspond to read 1 while the second and all those with R2 correspond to the second or reverse read. These sequences are 250 bp and overlap in the V4 region of

the 16S rRNA gene; this region is about 253 bp long. So looking at the files in the `MiSeq_SOP` folder that you've downloaded you will see 22 fastq files representing 10 time points from Female 3 and 1 mock community. You will also see `HMP_MOCK.v35.fasta` which contains the sequences used in the mock community that we sequenced in fasta format.

12.1.3 GBSv2 pipeline plugins

Plugin	Description
GBSSeqToTagDBPlugin	Executed to pull distinct tags from the database and export them in the fastq format. More
TagExportToFastqPlugin	Retrieves distinct tags stored in the database and reformats them to a FASTQ file. More
SAMToGBSdbPlugin	Used to identify SNPs from aligned tags using the GBS DB. More
DiscoverySNPCallerPluginV2	Takes a GBSv2 database file as input and identifies SNPs from the aligned tags. More
SNPQualityProfilerPlugin	Scores all discovered SNPs for various coverage depth and genotypic statistics for a given set of taxa. More
UpdateSNPPositionQualityPlugin	Reads quality score file to obtain quality score data for positions stored in the snpposition table. More
SNPCutPosTagVerificationPlugin	Allows user to specify a Cut or SNP position for which they would like data printed. More
GetTagSequenceFromDBPlugin	Takes an existing GBSv2 SQLite database file as input and returns a tab-delimited file containing a list of Tag Sequences stored in the specified database file. More
ProductionSNPCallerPluginV2	Imports data from fastq and keyfile to genotypes then adds these to a genotype file in VCF or HDF5 format. More

12.1.4 GBSv2 pipeline

1. Load the Tassel 5.0 module

```
$ module load Tassel/5.0
```

2. Useful commands

To check all the available plugins, type:

```
$ run_pipeline.pl -Xmx200g -ListPlugins
```

To check all the parameters for a given plugin, e.g. `GBSSeqToTagDBPlugin`, type:

```
$ run_pipeline.pl -fork1 -GBSSeqToTagDBPlugin   -endPlugin -runfork1
```

Tip: Users are recommended to read more about GBS command line options here. Page 1-2

3. File preparation

Create necessary folders and copy your raw data (fastqs), reference file and key file to appropriate folder:

```
$ mkdir fastq ref key db tagsForAlign hd5
```

4. Commands for the pipeline

```
$ run_pipeline.pl -Xmx200g -fork1 -GBSSeqToTagDBPlugin -i fastq  -k key/Tomato_key.
↪txt -e ApeKI -db db/Tomato.db  -kmerLength 85 -mnQS 20  -endPlugin -runfork1
$ run_pipeline.pl -fork1 -TagExportToFastqPlugin  -db db/Tomato.db -o tagsForAlign/
↪tagsForAlign.fa.gz -c 5  -endPlugin -runfork1
```

(continues on next page)

(continued from previous page)

```
$ cd ref
$ bwa index -a is S_lycopersicum_chromosomes.2.50.fa
$ cd ../
$ bwa samse ref/S_lycopersicum_chromosomes.2.50.fa tagsForAlign/tagsForAlign.sai␣
↪tagsForAlign/tagsForAlign.fa.gz > tagsForAlign/tagsForAlign.sam
$ run_pipeline.pl -fork1 -SAMToGBSdbPlugin -i tagsForAlign/tagsForAlign.sam  -db db/
↪Tomato.db  -aProp 0.0 -aLen 0 -endPlugin -runfork1
$ run_pipeline.pl -fork1 -DiscoverySNPCallerPluginV2 -db db/Tomato.db  -sC "chr00" -
↪eC "chr12" -mnLCov 0.1 -mnMAF 0.01  -endPlugin -runfork1
$ run_pipeline.pl -fork1 -ProductionSNPCallerPluginV2 -db db/Tomato.db  -e ApeKI -i␣
↪fastq -k key/Tomato_key2.txt  -kmerLength 85 -mnQS 20 -o hd5/HapMap_tomato.h5 -
↪endPlugin -runfork1
```

CHAPTER 13

Indices and tables

- genindex
- modindex
- search

Python Module Index

a

About, 1

b

basemount, 21
basespace, 9
basespace-share, 15

c

ComputingResources, 6

d

DataDownloading, 30
DESeq2-phyloseq, 37

f

fastq_qc, 34
FilterFastq, 32

g

GBS, 42

h

Home, 1

m

Membership, 4
Microbiome, 45

Index

A

About (*module*), 1

B

basemount (*module*), 21
basespace (*module*), 9
basespace-share (*module*), 15

C

ComputingResources (*module*), 6

D

DataDownloading (*module*), 30
DESeq2-phyloseq (*module*), 37

F

fastq_qc (*module*), 34
FilterFastq (*module*), 32

G

GBS (*module*), 42

H

Home (*module*), 1

M

Membership (*module*), 4
Microbiome (*module*), 45