MCBL Documentation

Release 1.0

Saranga

January 13, 2017


MCBL


MCBL

CHAPTER 1

MCIC Computational Biology Lab

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 experience staff when needed, and we as a community move our bioinformatics knowledge forward.

1.2 MCBL Services

NGS Data Analysis	Contacts
• Handling large data sets • Quality control • Using standard and custome scripts to do data analysis • Using Big Standalone servers	• Dr. Vitor Pavinato • Saranga Wijeratne

Workshops & Training	Contacts
• Training on various NGS data analysis • Training on Linux enviornment and shell scripting • Training on Amazon Web Services and OSC	• Dr. Vitor Pavinato • Saranga Wijeratne

Next Generation Sequencing	Contacts
• High-troughput sequencing(Illumina Miseq platform)	• Dr. Tea Meulia • Dr. Fiorella Cisneros Carter • Maria Elena Hernandez-Gonzalez

See also:

MCIC main webpage for more details


MCBL

CHAPTER 2

MCBL Membership

2.1 Why you need a MCBL Membership

We are happy to help you to carry out your own analysis. This will include helping on your experiment design; discussion on the most effective way to carry out your data analysis; providing necessary computational infrastructure (Software,Scripts and Computers) and answering your questions you might come across along the way.

2.2 How to Apply to MCBL Membership

Step 1 Fill and submit complete MCBL application form: MCBL Application.

Step 2 Submit your membership fee to MCIC.

Step 3 Contact MCBL adminstrator (Saranga Wijeratne) for login credentials.

Note: Access to the MCBL, to the computers and software resources will not 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 Benefits

- Access to MCBL and most powerful MCBL computers 24/7.
- Free access to MCBL Workshops and Bioinformatic user group meetings.
- Access to 1 TB data storage space for the duraton of the membership.

2.4 MCBL Membership Duration

Membership is offered for 6 months or 1 year period. Minimum membership period is 6 months and membership request for shorter period(less than 6 months) won't be considered.

2.5 MCBL Membership Termination

MCBL Membership will be terminated after membership period over or upon written request from user or PI to terminate the membership.

2.6 Contacts

Person	Information
Dr. Vitor Pavinato	Questions regarding membership
Saranga Wijeratne	MCBL Server access and remote access
Jody Whittier	MCBL payments

CHAPTER 3

MCBL Servers and Computing Resources

3.1 Servers Overview

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 Overview

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 Overview

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 Overview

Following bioinformatics software are availabe through MCBL.Some of the commercial software are available for MCBL users. Please contact Saranga Wijeratne for availability of the software.

3.4.1 Commercial Software

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

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	3.2-2	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>
Trimfastq	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

3.4.3 Python Modules

CHAPTER 4

Genotyping by Sequencing (GBS) pipeline documentation

Note:

Required OS OS x or Linux.

Software Tassel 5

Documentation Tassel 5.0 Wiki

Author This document is created by Saranga Wijeratne

4.1 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)

4.2 Files You Need to Have

Following files need to be created or present before you start the pipeline:

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

Note: Fastq files should rename as follows, more on page 7

FLOWCELL_LANE_fastq.gz example: AL2P1XXX_2_fastq.gz

FLOWCELL_s_LANE_fastq.gz example: AL2P1XXX_s_2_fastq.gz

code_FLOWCELL_s_LANE_fastq.gz example: 00000000_AL2P1XXX_s_2_fastq.gz

---

```
#To rename original .fastq.gz file,
$ mv  AE_S1_L001_R1_001.fastq.gz AL2P1XXX_1_fastq.gz
```

2. GBSv2 Key File. Example key file, More
3. Referance Genome.

4.3 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 a 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

4.4 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. 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 d
$ run_pipeline.pl -fork1 -TagExportToFastqPlugin  -db db/Tomato.db -o tagsForAlign/tagsForAlign.fa.gz
$ 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/tagsFor
$ run_pipeline.pl -fork1 -SAMToGBSdbPlugin -i tagsForAlign/tagsForAlign.sam  -db db/Tomato.db  -aProp
$ run_pipeline.pl -fork1 -DiscoverySNPCallerPluginV2 -db db/Tomato.db  -sC "chr00" -eC "chr12" -mnLCo
$ run_pipeline.pl -fork1 -ProductionSNPCallerPluginV2 -db db/Tomato.db  -e ApeKI -i fastq -k key/Toma
```

CHAPTER 5

Basic Microbiome Analysis

5.1 QIIME

Note:

Required OS OS x or Linux.

Software Qiime 1.9

Documentation Qiime Tutorial

Author This document is created by Saranga Wijeratne

5.1.1 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: Metadata mapping file example is given here. Read the section carefully. If you are planing to creat the mapping file by hand read this section.

Biom File: OTU observation file. Read more here

5.1.2 Files You Need to Have

Following files need to be downloaded or presented before you start the pipeline. For this tutorial, Mothur tutorial data published in Schloss Wiki will use. These data are 16s rRRNA Amplicons sequenced with MiSeq technology.

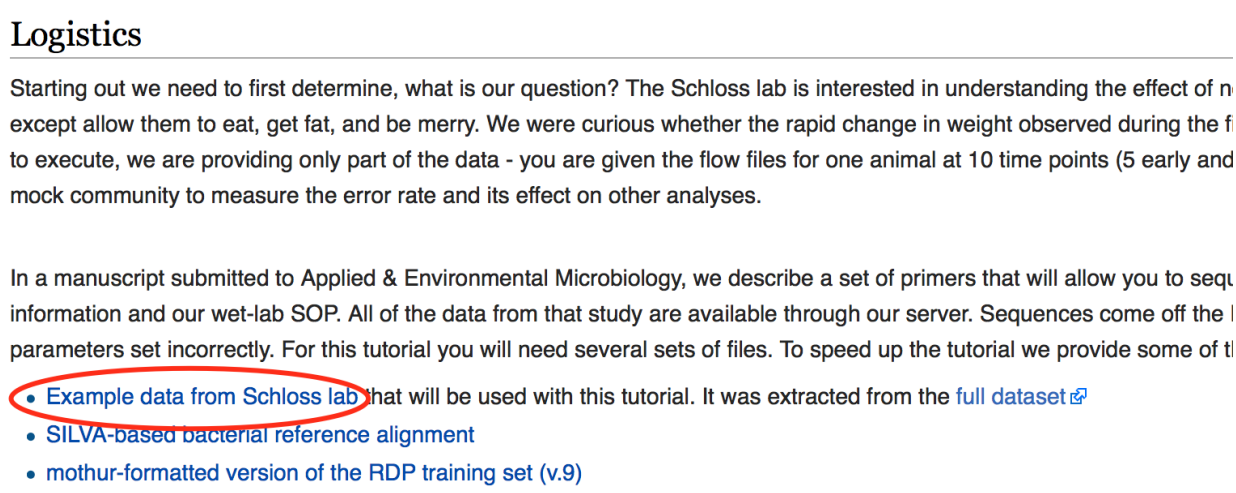
1. Create Folders

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

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

2. 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 I
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`, issue following command to get the data. Data is archived. `unzip -j` will extract all the files to same directory where you are on right now.

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

Rename the filenames to make it easy to downstream analysis.

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

Command explanation.

- `for f in *.fastq;` reads anyfile ends wiht *.fastq* one at a time
- `do` start 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` finish the loop

3. Data Explanation

Files and experiment are discribed as follows in 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.

5.1.3 GBSv2 Pipeline Plugins

Plugin	Description
GBSSeqToTagDB-Plugin	Executed to pull distinct tags from the database and export them in the fastq format. More
TagExportTo-FastqPlugin	Retrieves distinct tags stored in the database and reformats them to a FASTQ file. More
SAMToGBSdb-Plugin	Used to identify SNPs from aligned tags using the GBS DB. More
DiscoveryS-NPCallerPluginV2	Takes a GBSv2 database file as input and identifies SNPs from the aligned tags. More
SNPQualityProfil-erPlugin	Scores all discovered SNPs for various coverage depth and genotypic statistics for a given set of taxa. More
UpdateSNPPosi-tionQualityPlugin	Reads a quality score file to obtain quality score data for positions stored in the snpposition table. More
SNPCut-PosTagVerifica-tionPlugin	Allows a user to specify a Cut or SNP position for which they would like data printed. More
GetTagSequence-FromDBPlugin	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
Production-SNPCallerPlug-inV2	Converts data from fastq and keyfile to genotypes then adds these to a genotype file in VCF or HDF5 format. More

5.1.4 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. 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 d
$ run_pipeline.pl -fork1 -TagExportToFastqPlugin  -db db/Tomato.db -o tagsForAlign/tagsForAlign.fa.gz
$ 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/tagsFor
```

```
$ run_pipeline.pl -fork1 -SAMToGBSdbPlugin -i tagsForAlign/tagsForAlign.sam  -db db/Tomato.db  -aProp
$ run_pipeline.pl -fork1 -DiscoverySNPCallerPluginV2 -db db/Tomato.db  -sC "chr00" -eC "chr12" -mnLC
$ run_pipeline.pl -fork1 -ProductionSNPCallerPluginV2 -db db/Tomato.db  -e ApeKI -i fastq -k key/Toma
```

5.2 Mothur

CHAPTER 6

Data Downloading from Cloud Services

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 is created by Saranga Wijeratne

6.1 Downloading from hudsonalpha.org

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_TruseqHTDua
http://mysample.download.org/dl/d4/Meulia/myprojectnumber/data_150522/C6V7FANXX_s3_0_TruseqHTDua
http://mysample.download.org/dl/d4/Meulia/myprojectnumber/data_150522/C6V7FANXX_s5_0_TruseqHTDua
http://mysample.download.org/dl/d4/Meulia/myprojectnumber/data_150522/C6V7FANXX_s8_0_TruseqHTDua
```

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
```

6.1.1 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) = d41d8cd428f00b204e9
MD5 (C6V7FANXX_s5_0_TruseqHTDual_D709-TruseqHTDual_D506_SL104602.fastq.gz) = d49d8cdf00j204e9800
MD5 (C6V7FANXX_s8_0_TruseqHTDual_D705-TruseqHTDual_D501_SL104565.fastq.gz) = d47d8cd98dfds0b204e
MD5 (C6V7FANXX_s8_0_TruseqHTDual_D712-TruseqHTDual_D508_SL104628.fastq.gz) = d42d8cd98f00bdfse98
```

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

```
$ md5sum *
```

```
d41d8cd428f00b204e9800998ecf8427e   C6V7FANXX_s3_0_TruseqHTDual_D703-TruseqHTDual_D501_SL104549.
d49d8cdf00j204e9800998ecf8427ed56   C6V7FANXX_s5_0_TruseqHTDual_D709-TruseqHTDual_D506_SL104602.
d47d8cd98dfds0b204e9800998ecf8427e  C6V7FANXX_s8_0_TruseqHTDual_D705-TruseqHTDual_D501_SL104565.
d47d8cd98dfds0b204e9800998ecf8427e  C6V7FANXX_s8_0_TruseqHTDual_D712-TruseqHTDual_D508_SL104628.
```

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 7

Filter a Fastq File (CASAVA generated)

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 is created by Saranga Wijeratne

7.1 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.g
$ 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.

7.2 Filter a Fastq

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 > C8EC8
```

7.3 Filter Multiple Fastqs

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 the 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%.*.f
```

CHAPTER 8

Adapter Removing and Quality Filtering

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

8.1 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
```

8.2 File 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.

8.3 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 I
```

Paired End Fastq Files

```
$java -jar $TRIMHOME/trimmomatic-0.33.jar PE -threads 12 C8EC8ANXX_s2_1_illumina12index_1_SL143785.fa
```

Multiple Fastqs

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 Fasq 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 accrodingly)

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 staved here
$files_1=(*_s3_1_*.fastq.gz);files_2=(*_s3_2_*.fastq.gz);sorted_files_1=($(printf "%s\n" "${files_1[(
```


MCBL

CHAPTER 9

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: and here

Author This document is created by Saranga Wijeratne

9.1 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")
```

Version:

```
[1] ‘1.16.2’
```

On your own server,

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

9.2 Import data with phyloseq

For this step you need Biom and mapping file generated from 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 following commands.

```
> biom_file<-"otu_table_mc10_w_tax.biom"
> mapping_file<-"mapping.txt"
> biom_otu_tax<-import_biom(biom_file) #Importing biomfile with phyloseq
> mapping_file<-import_qiime_sample_data(mapping_file)  #Importing mapping file with phyloseq
> merged_mapping_biom<-merge_phyloseq(biom_otu_tax,mapping_file) #Merging Biom and mapping file with
> colnames(tax_table(merged_mapping_biom))<-c("kingdom", "Phylum","Class", "Order", "Family", "Genus"
```

```
> merged_mapping_biom
```

Merged mapping and Biom output:

```
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 ]
```

Mapping file should be looked like this:

```
    Sample Data:        [40 samples by 7 sample variables]:
X.SampleID BarcodeSequence LinkerPrimerSequence InputFileName IncubationDate Treatment Description
    S1          S1               NA                NA      S1.fasta              0        CO
    S2          S2               NA                NA      S2.fasta              0        CO
    S3          S3               NA                NA      S3.fasta              0        CO
    S4          S4               NA                NA      S4.fasta             15        CO
    S5          S5               NA                NA      S5.fasta             15        CO
    S6          S6               NA                NA      S6.fasta             15        CO
    S7          S7               NA                NA      S7.fasta             30        CO
    S23        S23               NA                NA     S23.fasta             15        RE
    S24        S24               NA                NA     S24.fasta             15        RE
    S25        S25               NA                NA     S25.fasta             15        RE
    S26        S26               NA                NA     S26.fasta             30        RE
    S27        S27               NA                NA     S27.fasta             30        RE
    S28        S28               NA                NA     S28.fasta             30        RE
    S29        S29               NA                NA     S29.fasta             45        RE
```

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

```
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))
```

9.3 Differential Abundance OTU call

Input File merged_mapping_biom

Output Files DESeq2_Out.txt

1. Load the DESeq2 into your R enviornment:

```
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. *phyloseq_to_deseq2* function in the following lines converts phyloseq-format microbiom data (i.e merged_mapping_biom) into a *DESeqDataSet* with dispersion estimated, using experimental design formula (i.e ~ Treatment).

```
diagdds <- phyloseq_to_deseq2(merged_mapping_biom, ~ Treatment)
diagdds <- DESeq(diagdds, test="Wald", fitType="parametric")
```

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

Warning: If you are getting the following error, please execute the code block below. More...
Error in estimateSizeFactorsForMatrix(counts(object), locfunc, geoMeans = geoMeans) :
Calls: estimateSizeFactors ... estimateSizeFactors -> .local -> estimateSizeFactorsFo

```
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")
```

4. The `results` function creats 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), ],
write.csv(sigtab, as.character(filename_out)) #Writing `sigtab` to
```

CHAPTER 10

Indices and tables

- genindex
- modindex
- search

d

h

i

m

D

H

I

M