**Installing QWRAP and other dependencies**

Minimum requirements: User should have a basic idea of how to install software on Linux operating system. These tools must be installed for proper functioning of the tool QWRAP. All these tools can be either installed on a Linux machine or a [QIIME virtual Box](http://qiime.org/install/virtual_box.html).

1. QIIME: There are several ways to install QIIME (<http://qiime.org/install/index.html>). Virtual box based installation seems to be the easiest for small data analysis or testing which can also be installed on a windows machine. This reference also has details about QIIME virtual box installation.
2. Usearch (<http://www.drive5.com/usearch/>). After downloading the 32bit linux binary, rename the file to usearch61
3. FASTQC (<http://www.bioinformatics.babraham.ac.uk/projects/fastqc/>)
4. FASTX (<http://hannonlab.cshl.edu/fastx_toolkit/>)
5. R (<http://www.r-project.org/>)

QWRAP is available for download at <https://github.com/QWRAP/QWRAP>. It can be installed using the following command on a Linux terminal:

git clone git://github.com/QWRAP/QWRAP.git

This command will download all the QWRAP scripts to the current directory in a folder named QWRAP. Now you need to add the location of this folder to the system PATH variable (so that you can run the program from any location on computer).

For example, if absolute path for the QWRAP folder is “/home/username/QWRAP” then you can add the following line to file “.bashrc” (/home/username/.bashrc) which is present in your home directory

export PATH=${PATH}:/home/username/QWRAP/

**Initiating a new QWRAP analysis**

QWRAP is designed for de-multiplexed dataset. So if your reads are multiplexed, please demultiplex them before using QWRAP.

Make sure all the raw data files (fastq format) are present in one folder. Usually, each fastq file represents one sample. Please rename the files so that it reflects a proper sample name. The QWRAP program works on all the files present in a given folder. Make sure the data files are compressed with “gzip” i.e. all files should have an extension “fastq.gz”. If the files are uncompressed and have an extension “.fastq”, then all files must be compressed before analysis. All files in a folder can be compressed by executing the following command inside the folder.

gzip \*.fastq

If the files don’t have a fastq extension, please rename them as fastq and then compress them as mentioned above.

For example, the QWRAP program folder containing a sample dataset called as “example\_rawdata.tar.gz” which has 8 samples from four different groups. We can uncompress the file example\_rawdata.tar.gz and the 8 samples looks like

tar xvf example\_rawdata.tar.gz

ls example\_rawdata/

**A1.fastq.gz A2.fastq.gz B1.fastq.gz B2.fastq.gz C1.fastq.gz C2.fastq.gz D1.fastq.gz D2.fastq.gz**

We will use this example dataset as an example to go through the analysis protocol.

Let us create a new directory for analysis and storing the results and name it as “ANALYSIS”. Open the directory “ANALYSIS”, we are going to run all the commands inside this directory to create a well-defined structure of different results, which is later useful for generating HTML report :

mkdir ANALYSIS

cd ANALYSIS

Copy the example\_rawdata folder into the ANALYSIS folder.

**Quality Checking of raw data**

Run the data quality check on all files present in the raw data folder (compressed fastq files). If the raw data is located in folder “example\_rawdata”, then run the program as

quality\_check\_rawdata.sh example\_rawdata

The following command calls a program FASTQC and runs it over all the files present in the “example\_rawdata” directory. This script creates a folders fastqc\_rawdata with FASTQ report for all samples. Inside the folder, a HTML file “FASTQC\_overview.html” is created which provides an combined overview of the quality statistics for all samples and also provide more detailed report for individual samples. Users are suggested to look at the report carefully and decide the parameter which might be useful for quality filtering. Please note the length of sequence reads present in the dataset as this parameter can be used later for trimming bad quality reads.

**Quality Filtering and quality checking for processed data**

Here quality filtering is performed in two steps.

1. All the reads are trimmed to a user defined sequence length.

2. Only those reads are kept which has more than 80% bases with a QScore >20.

The program “quality\_filter\_single.sh” requires the location of the folder containing raw data and the user defined length to trim the reads. If the user doesn’t want to trim the reads, they can provide the full length of the reads. In this case we want to trim the reads to sequence length 200. If user wants to modify the parameters for second step of quality filtering, they can edit the parameters (QC\_PERCENT, QC\_SCORE) in program file “quality\_filter\_single.sh” in QWRAP folder. We can start of quality filtering by executing the program as

quality\_filter\_single.sh example\_rawdata 200

The program does the quality filtering and produces the filtered fastq files in a directory called “filtered\_fastq”.

The program also creates the fasta file for all the samples in the current directory “ANALYSIS” which are used for subsequent analysis.

The program “quality\_check\_filterdata.sh” uses the directory filtered\_fastq to generate the quality report for the filtered dataset.

quality\_check\_filterdata.sh filtered\_fastq

This script creates a folders fastqc\_filterdata with FASTQC report for all files of the filtered dataset. Inside the folder, thee HTML file “FASTQC\_overview.html” is created which provides an combined overview of the quality statistics for all samples and also provide more detailed report for individual samples.

**Generating QIIME input files and running QIIME scripts**

All the fasta files present in the current directory (ANALYSIS) will be included in the analysis. Based on the quality report of filtered dataset if the user wants to remove certain samples (sometime because of low sampling depth), they can just delete the corresponding fasta file from the “ANALYSIS” directory.

The program “microbiome-workflow1.sh” runs inside the ANALYSIS directory and creates several files which are required for any QIIME based analysis. You can run it as

microbiome-workflow1.sh

The program generates the following files:

a) seqs.fna: Here all fasta files present in a current folder are merged into a single fasta file called “seqs.fna”. The sample name becomes the header for the reads to distinguish different samples. A sample snapshot is given below. Since read headers are not very informative, they are renamed as 1,2… to save disk space.

>A1\_1

AACGTAGGTCACAAGCGTTGTCCGGAATTACTGGGTGTAAAGGGAGCGCA

>A1\_2

TACGTAGGGGGCAAGCGTTATCCGGATTTACTGGGTGTAAAGGGAGCGTA

…

>B1\_1

TACGGAGGATCCGAGCGTTATCCGGATTTATTGGGTTTAAAGGGAGCGTA

>B1\_2

AACGTAGGGTGCAAGCGTTGTCCGGAATTACTGGGTGTAAAGGGCGCGCA

…

b) mapping.txt: A default mapping file is created with a single column containing a header and sample names as shown below.

#SampleID

#Mapping file for the QIIME analysis

A1

A2

B1

B2

C1

C2

D1

D2

The mapping file can/should be edited to add grouping information about the sample. For example, since these 8 samples belong to four groups (A, B, C, D) a new column called “Group” can be added as shown below. This is a tab delimited file so all columns are separated by tab. In a similar fashion more attributed of samples can also be added.

#SampleID Group

#Mapping file for the QIIME analysis

A1 A

A2 A

B1 B

B2 B

C1 C

C2 C

D1 D

D2 D

c) sample\_order.txt: This file is used to determines how the samples are ordered in the OTU table. This is especially important as the same order is used for generating taxonomy bar charts. You can change the order of samples manually here to reorder them in OTU table.

A1

A2

B1

B2

C1

C2

D1

D2

**d) script.sh :** This file has all the commands required to generate the OTU table and taxonomy bar chart. Most of the commands are from the QIIME package and are arranged in a certain order. More details about the command can be found at <http://qiime.org/scripts/>.

Users can look into the file script.sh for exact command and their parameters. An example of “script.sh” file can be viewed [here](https://github.com/QWRAP/QWRAP/blob/master/sample_script.sh). Briefly, the script is going to perform the following functions in order

1. Chimera Filtering using program “Usearch”
2. OTU Clustering at 97% sequence similarity using program “uclust”.
3. Picking representative of OTUs based on abundance
4. Assigning taxonomy to OTUs using RDP classifier (threshold 0.8) using Greengenes database
5. Sorting OTU table based on file “sample\_order.txt”
6. Summarizing OTUs into taxonomic groups
7. Creating Normalized OTU table i.e. converting raw numbers from OTU table into proportion and also merging the taxa information in a single file.
8. A filtered list of top 10, top 25 and top 100 OTUs and taxa are generated.
9. A new filtered OTU table (where OTU abundance > 0.0005%) is created, with summarizing taxonomy and OTU table stats. The filtered file is used for all subsequent analysis.

Execute the file “script.sh” as

sh script.sh

This will execute all the commands present in the file and generate OTU table and the taxonomic charts.

To further advance the analysis with alpha and beta diversity calculations, another program “microbiome-workflow2.sh” is provided. This program generates a file called “script\_adv.sh” which contains QIIME commands to perform several jobs like sequence alignment, generating phylogenetic tree, calculating alpha diversity, beta diversity plots etc. An example of “script\_adv.sh” file can be viewed [here](https://github.com/QWRAP/QWRAP/blob/master/sample_script_adv.sh). The script requires one user defined parameter “sampling depth” (read depth). Since different samples may have different read depth, the read depth should be normalized across all samples. When a read depth is provided, a random sampling event is used to rarify the OTU table. If the sampling depth is 22986 (in our example dataset), you can run the script as

microbiome-workflow2.sh 22986

If no sampling depth is provided, the program will automatically calculate the minimum sampling depth from the file “otu\_table.stats”

Briefly the script is going to perform the following functions

1. Generating multiple sequence alignment of OTUs using program “PYNAST” and creating a phylogenetic tree using program “FASTTREE”. If the numbers of OTUs are large, this step can take long time.
2. Rarify OTU table based on minimum sampling depth
3. Calculate alpha diversity (using metrices : chao1, observed\_species ,PD\_whole\_tree, shannon, simpson) and generate plots
4. Caculate beta diversity (using metrices : bray\_curtis, unweighted\_unifrac, weighted\_unifrac) and generate plots
5. Generating UPGMA tree of samples.

The file “script\_adv.sh” can be executed as

sh script\_adv.sh

**Generating HTML report**

Although most of the results are generated in ANALYSIS folder and can be explored manually, we have developed a script which can put together most of the results into a user friendly HTML report. The HTML report is a static so the whole ANALYSIS folder can be moved or shared with another person or the results can also be shared using a webserver.

Run the following program to generate the HTML report.

report\_microbiome.sh

This creates an html file “microbiome\_report.html” in the ANALYSIS directory and can be opened using any web browser.

The most updated version of the protocol ([QWRAP-Readme.docx](https://github.com/QWRAP/QWRAP/blob/master/QWRAP-Readme.docx)) is also available at

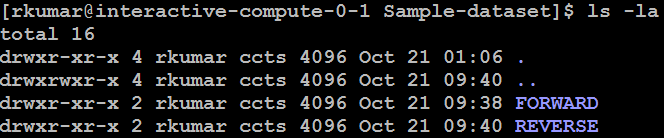
<https://github.com/QWRAP/QWRAP>. On the github page click the button “RAW” to view this file. This readme file also has instructions to perform paired end microbiome data analysis.

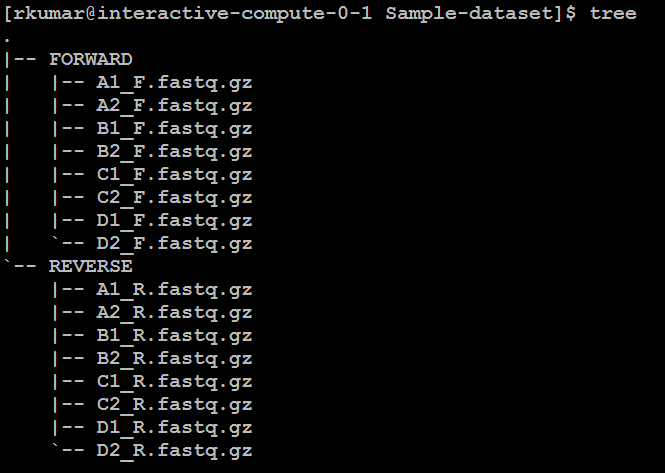
**PAIRED END READS ANALYSIS**

Locate the folders with forward and reverse reads.

Here in this example they are stored in separate folders FORWARD & REVERSE for forward and reverse end reads.

Just check the files.





**Step A: Create directory for analysis**: Create a directory to store analysis results and get (cd) into it.

mkdir ANALYSIS

cd ANALYSIS

**Step B: For Quality check before merging:**  run the program “quality\_check\_before.sh” with location of FWD reads and REV reads.

quality\_check\_before.sh ../FORWARD/ ../REVERSE/

This creates two folders fastqc\_beforef and fastqc\_beforer with FASTQC results for the forward and reverse files and stats.

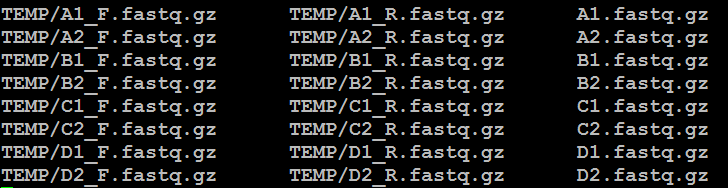
**Step C**: **Merge the forward and reverse files**: Run program: merge\_reads\_F\_R.sh with parameters containing the location of forward and reverse folder. This creates a TEMP folder which has all the reads for the analysis.

merge\_reads\_F\_R.sh ../FORWARD/ ../REVERSE/

**Step D:** **Prepare for merging reads:** The program “prepare\_merge\_fastq.sh” requires the location of TEMP folder (containing both forward and reverse files) as a command line argument. Run the program as

prepare\_merge\_fastq.sh TEMP/

This program creates a mapping file "Paired\_Filelist.txt" with each line containing the location of forward read and reverse reads and the predicted name of merged read.



You can edit the column 3 of the merged file in a text editor if required (especially if you want to rename the files. Make sure that the names in 3rd column should not include underscore (\_).

**Step E: Merge the reads:** This is done using program merge\_fastq.sh. This script uses program USEARCH for merging reads. The program needs 5 parameters which includes the quality control parameters for merging.

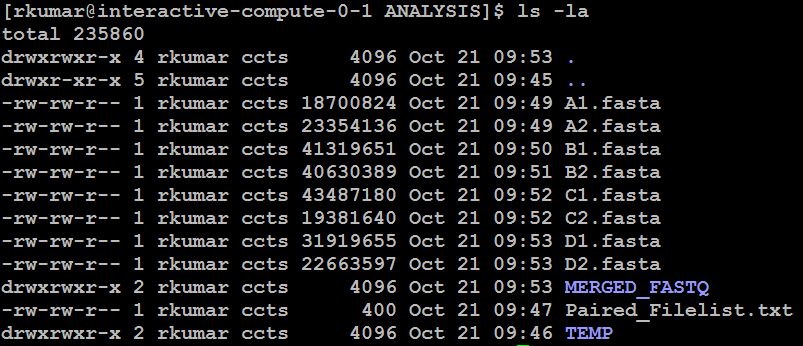
1. Name of mapping file (Paired\_Filelist.txt)
2. Length for trimming forward reads. Provide reads full length if no trimming is required.
3. Length for trimming reverse reads. Provide reads full length if no trimming is required.
4. Max mismatch allowed between forward and reverse reads when aligned.
5. Minimum overlap required between forward and reverse reads when aligned.

In this examples since we had reads of length 251 and decided not to trim them, we used following parameters.

merge\_fastq.sh Paired\_Filelist.txt 251 251 10 200

This script after merging creates all fasta files in the current directory. It also stores the FASTQ files in a folder MERGED\_FASTQ.

So the current ANALYIS directory will look something like this



**Step F: For Quality check after merging:**  run the program “quality\_check\_after.sh” with location of merged FASTQ reads.

quality\_check\_after.sh MERGED\_FASTQ/

This creates two folders fastqc\_afterf and fastqc\_afterr with FASTQC results and stats.

**Step G: Run the microbiome workflow1**.

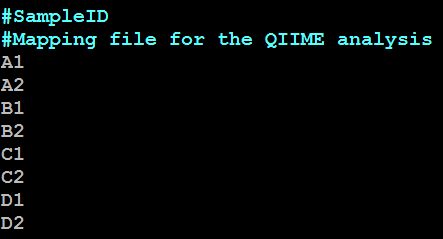
microbiome-workflow1.sh

This script runs inside the ANALYSIS directory and creates many files which are required for QIIME analysis.

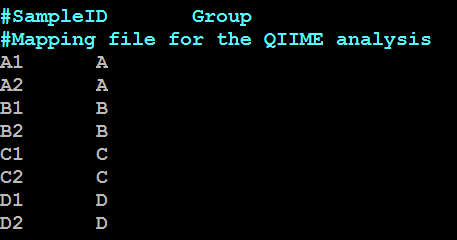
Seqs.fna: Here all fasta files present in a current folder are merged into a single fasta file called seqs.fna. The sample name becomes the header for the reads to distinguish different samples. A snapshot looks as below. Since read headers are not required they are renamed as 1,2… to save space.



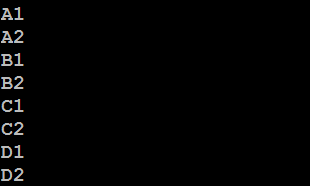
Mapping.txt: The default mapping file has single column containing a header and sample names as shown below



The mapping file can/should be edited to add grouping information about the sample. For example a column can be added as shown below (separated by tab). Similarly more attributed of samples can be added.



Sample\_order.txt: This determines how the samples are ordered in OTU table. This is especially important as the same order is used for taxonomy bas charts. You can change the order of samples here to reorder them.



**Step H: Run the QIIME script:**

The script microbiome-workflow1.sh has created a script “script.sh” which contains a set of QIIME commands to generate taxonomy bar charts. You can change/edit/add the commands if needed and run the script as

sh script.sh

**Step I:** **Run the microbiome workflow2**. This workflow generates script “script\_adv.sh” to do sequence alignment, generate tree, alpha diversity, beta diversity, plots etc. It requires a parameter as even sampling depth. Since all samples have different sampling depth, a minimum sampleing depth should be choosen to normalize all samples to same depth. It the number is 50000, you can run the script as

microbiome-workflow2.sh 5000

This creates a script named “script\_adv.sh”

**Step J: Run the QIIME script: You can edit this file as per your requirements and run it as**

sh script\_adv.sh

**Step K: Generate the whole report:** Run the following script to generate the web report.

report\_microbiome.sh

This creates a html report which can be browsed using file microbiome\_report.html

