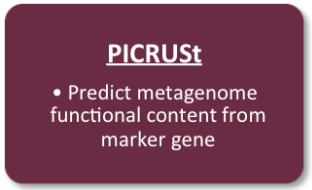


Workshop 11: Metagenomics Analysis

Shi, Baochen
Department of Pharmacology, UCLA

Outlines

The workshop: 2 hours per day over 3 days.

Day 1.
i) how to perform the 16S rRNA-based analysis using bioinformatics pipeline QIIME.

Day 2.
i) Introduce statistical analyses in QIIME

Day 3.

i) Functional analyses of the microbiome

ii) Open Q&A

Outlines

We will demonstrate 16S amplicon analysis using QIIME

a) Run QIIME on Hoffman2 or local installation

b) Sequence data preparation

c) Operational Taxonomic Units (OTU) picking, Taxonomic assignment & inferring phylogeny

d) microbiome diversity analyses

Useful UNIX Commands: Covered in Workshop 1 & 2

- Where am I? pwd
- Current directory ./
- Home directory ~/
- Change directory cd ~/data
- Move up one level cd ..
- List files in folder ls
- Look at a file less fileName
- Copy a file cp ~/data/file ~/otherdir/
- Delete a file rm fileName
- Delete a directory rmdir ~/dirName/
- Move a file mv ~/data/file ~/otherdir/file
- Secure copy scp user@host1:dir/file user@host2:dir/file
- Compress a file gzip –c file > file.gz
- Uncompress a file gunzip file.gz
- Make a new folder mkdir data2
- Count lines in a file wc –l fileName

Questions/In Doubt/Lost?

- Unix manual: man functionX
- Google is your best friend!
- Any of your friendly QCB fellows

a) Run on Hoffman2

idre UCLA Institute for Digital Research and Education

A shared account on mac for students:

login: workshop
password: NGS_Analysis

Login in hoffman2:

```
ssh biosbc@hoffman2.idre.ucla.edu
```

Qiime is in :

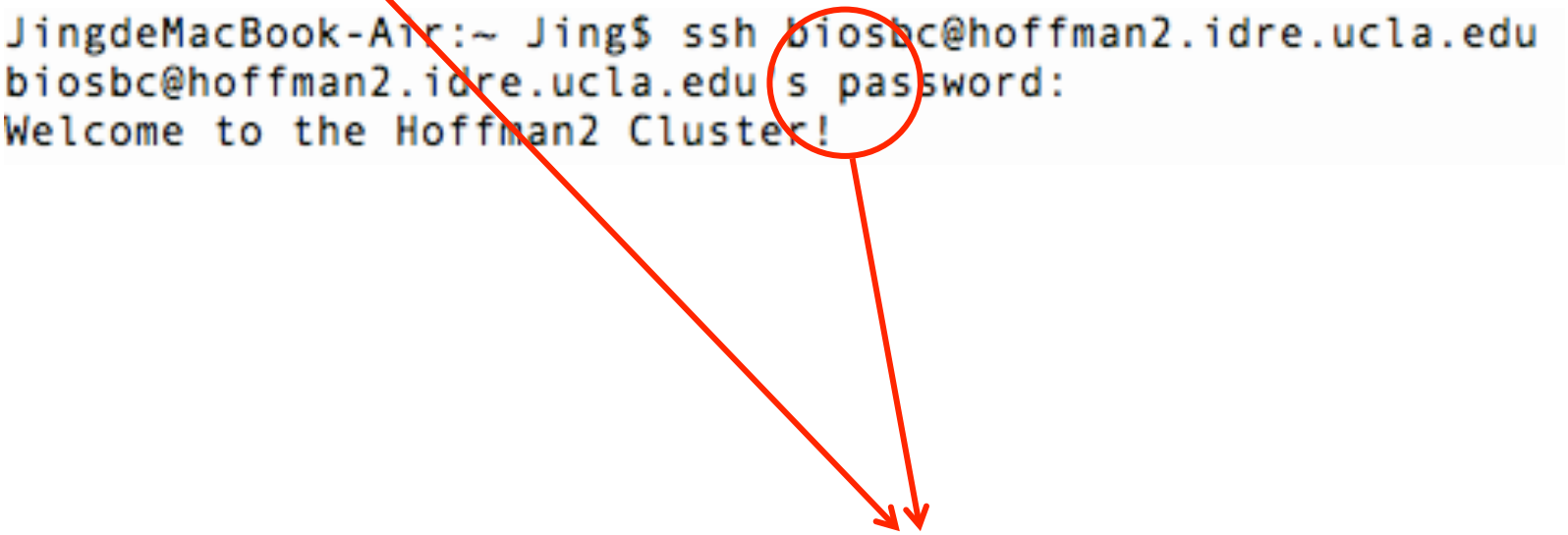
```
cd /u/local/apps
```

Load Qiime:

```
module load qiime
```

Logging into Hoffman2

ssh biosbc@hoffman2.idre.ucla.edu



change to your user ID

Useful Tools on Hoffman2

ls /u/local/apps/

```
abaqus              blcr              emacs             grads             libgtextutils     mumps             picard-tools      scilab            tvmet
abaqusdocs          blitz             espresso          graphicsmagick    libtool           muscle            plink             scons             uclust
accelrys            bmapuclatools     fastphase         graphviz          libxc             mygroup           pop-c++           sentaurus         udunits
Accelrys            boost             fasttree          gromacs           lmdb              mysql             povray            shapeit           usearch
activeperl          boost-jam         fastx_toolkit     gsl               loni_pipeline     namd              pplcer            shapelib          usr_lib_GLoverride
ActiveTcl           bowtie            fbat              hadoop            lumerical         ncl               preseq            skampi            valgrind
adina               bowtie2           fe-safe           handbrake         lynx              nco               proj.4            slots             vasp
Adobe               bwa               ffmpeg            harminv           mach              netbeans          protobuf          snappy            vcftools
affymetrix          caffe             fftw2             haskell           mafft             netcdf            pypy              solar             vegas
AFNI                casava            fftw3             hdf               manorm            newbler           pypy3             sortmerna         ViennaRNA
aida                cdbtools          flex              hdf5              maple             ngsplot           python            soxr              vim
amber               cd-hit            fltk              homer             maq               nlopt             qcachegrind       splicetrap        visit
ampliconnoise       cernlib           freebayes         hyperworks        maqview           numpy             qchem             spm               vmd
anaconda            cern_root         freesurfer        iaida             mathematica       nwchem            qgfe              stata             votca
annovar             clearcut          freetds           idl               matio             ocaml             qhull             stressapptest     vtk
ansys_inc           clhep             fribidi           idr               matlab            octave            qiime             structure         wannier90
ant                 cln               fsl               igraph            mats              omssa             qiime_data        subversion        weblogo
antlr               cmake             gamess            ilog              mecab             oommf             qrupdate          suitesparse       wolfram
armadillo           common            gatk              IM-IMA            meep              openbabel         qsub              sundials          x264
arpack              comsol            gatk-queue        impute            meld              opencv            qt                superlu_dist      x86_open64
atlas               conan             gaussian          imsl              merlin            openfoam          quantiSNP         swarm             xerces-c
atomeye             condor            gaussview         infernal          metis             openmotif         queue.logs        sysstat           xfdtd
atompaw             consed            gcta              inspect           mfold             opensees          R                 szip              xfig_OLD
autoconf            cp2k              gdal              installation      microbiomeutil    osiris            raxml             t500              xilinx-vivado
automake            cpmd              gdc               intel             migrate           osu-micro-benchmarks rdp_classifier tau               xmd
bamtools            ctffind           geant4            isight            minirosetta       papi              relion            tcl               xmedcon
bcftools            cufflinks         genetorrent       jags              mira              paraview          remcom            tcsh              xpdf_OLD
bcl2qfastq          cytoscape         geos              jam               molcas            parmetis          repeatmasker      tecplot360        yaml
beagle              ddd               gflags            java              molden            parsec            RepeatModeler     texlive           yasm
beaglecall          ddplot            gftp              jaxodraw          molpro            parsinsert        resmap            tmux              zlib
beagle_utilities    dejagnu           ginac             jmol              mono              pcre              rosetta           tophat
bedtools            dirac             git_OLD           julia             mopac             pdt               rstudio           toscastructure
bfast               diskusage         globalarrays      kepler            mothur            penncnv           rsync             totalview
bioscope            dl_poly           glog              lammps            mpc               perl_modules      rtax              trans-ABySS
blas                dx                gmp               lapack            mpfr              petsc             ruby              treemix
blast               eclipse           gnuplot           lapack++          mpiblast          pgsql             sage              trilinos
blast+              eigen             gpac              ldope             mplayer           phantompeakqualtools samtools       trinity
blat                eigensoft         grace             leveldb           mrt               phase             scalapack         tt
```

a) MacQIIME local installation

QIIME have a lot of dependencies .
The quickest way to get started using QIIME VirtualBox or MacQIIME
http://qiime.org/1.9.0/install/install.html

MacQIIME:
a1) Download MacQIIME (http://www.wernerlab.org/software/macqiime/macqiime-installation)

```
tar -xvf MacQIIME_*.tgz
```

a2) install MacQIIME
Copy macqiime folder to root, and then copy "macqiime" script to /usr/bin/

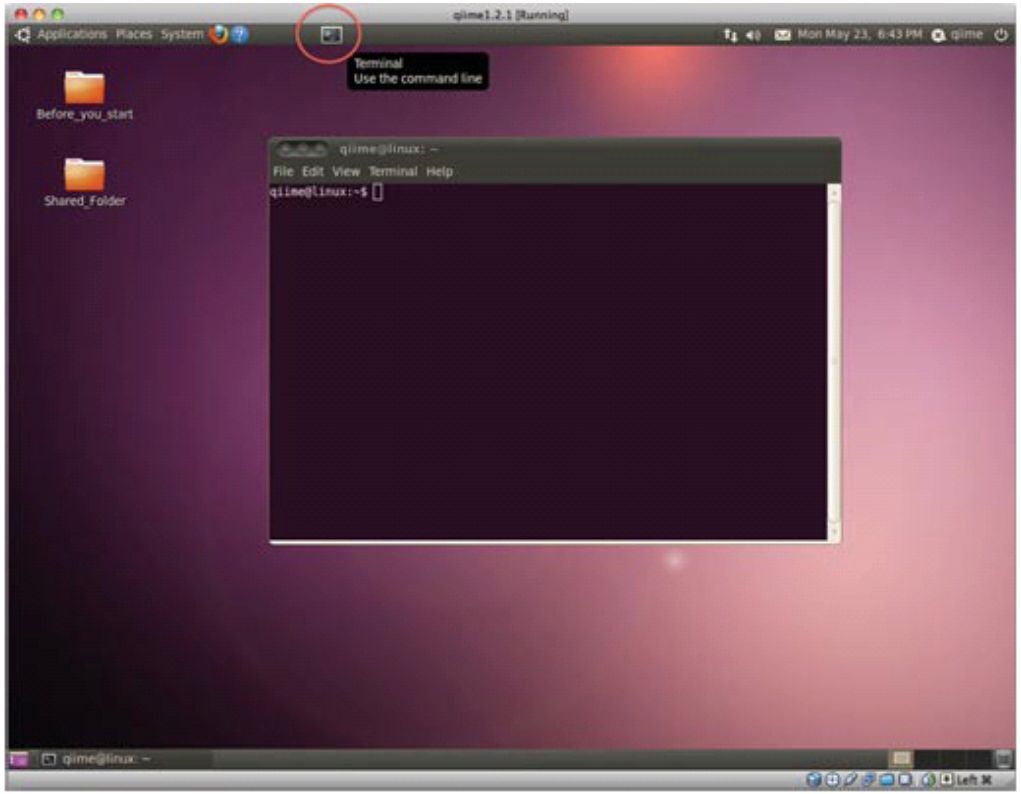
```
./install.s
```

source the environment variables

```
source /macqiime/configs/bash_profile.txt
```

Tutorial, test data & test script:
/macqiime/QIIME/

a) QIIME VirtualBox installation



QIIME VirtualBox (Linux, windows, Mac):

a1) Download and install oracle VirtualBox (http://qiime.org/1.9.0/install/virtual_box.html)

a2) Download the QIIME Virtual Box

a3) Create a new virtual machine

Outlines

We will demonstrate steps for

a) Run QIIME on Hoffman2 or local installation

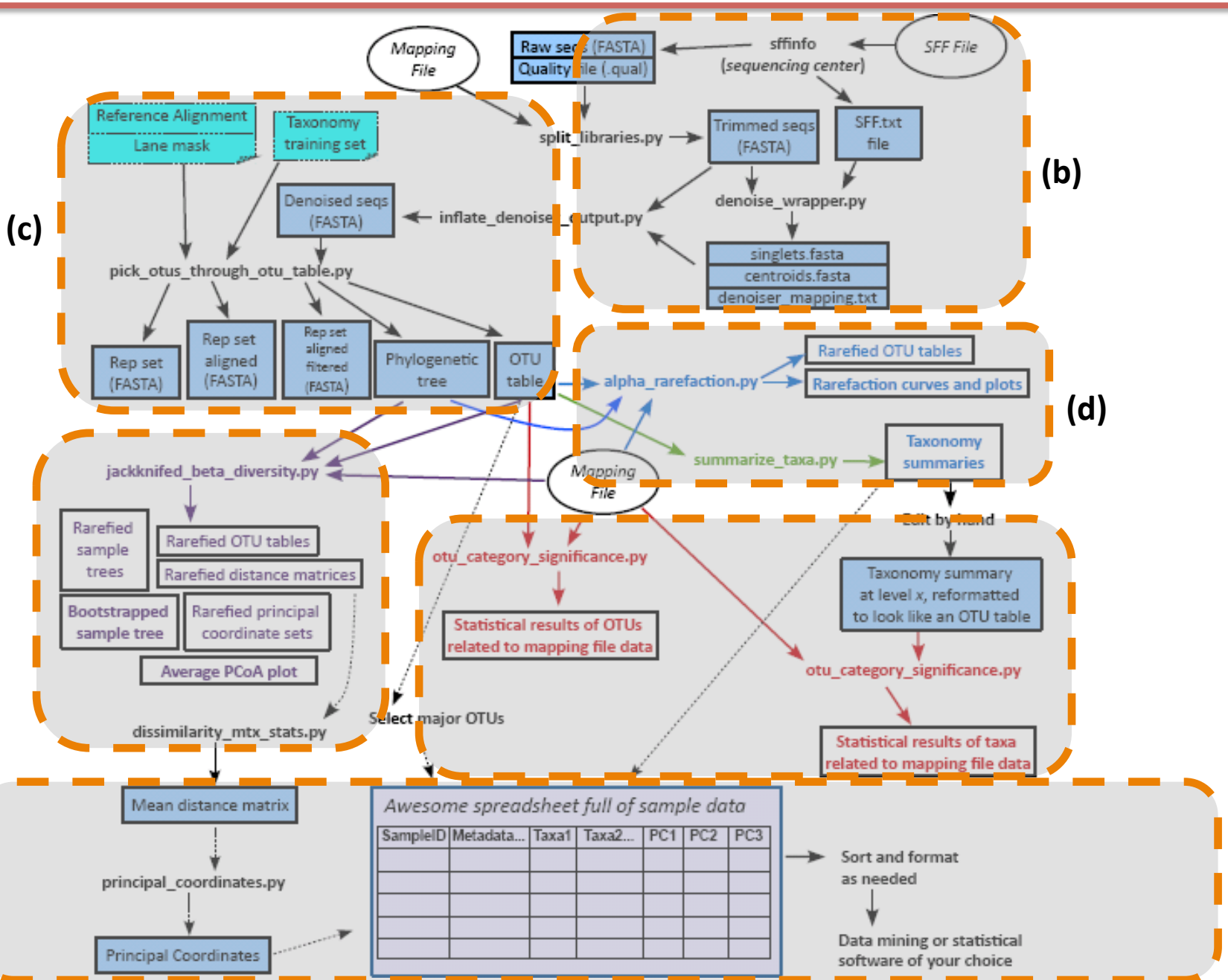
b) Sequence data preparation

c) Operational Taxonomic Units (OTU) picking, Taxonomic assignment & inferring phylogeny

d) microbiome diversity analyses

Quantitative Insights Into Microbial Ecology

Flowchart


Mapping File
Raw seqs (FASTA)
Quality file (.qual)
sffinfo
(sequencing center)
SFF File
(b)
split_libraries.py
Trimmed seqs (FASTA)
SFF.txt file
denoise_wrapper.py
singlets.fasta
centroids.fasta
denoiser_mapping.txt
(c)
Reference Alignment
Lane mask
Taxonomy training set
Denoised seqs (FASTA)
inflate_denoiser_output.py
pick_otus_through_otu_table.py
Rep set (FASTA)
Rep set aligned (FASTA)
Rep set aligned filtered (FASTA)
Phylogenetic tree
OTU table
(d)
Rarefied OTU tables
alpha_rarefaction.py
Rarefaction curves and plots
summarize_taxa.py
Taxonomy summaries
jackknifed_beta_diversity.py
Mapping File
Rarefied sample trees
Rarefied OTU tables
Rarefied distance matrices
Bootstrapped sample tree
Rarefied principal coordinate sets
Average PCoA plot
otu_category_significance.py
Statistical results of OTUs related to mapping file data
Edit by hand
Taxonomy summary at level x, reformatted to look like an OTU table
otu_category_significance.py
Statistical results of taxa related to mapping file data
dissimilarity_mtx_stats.py
Select major OTUs
Mean distance matrix
principal_coordinates.py
Principal Coordinates
Awesome spreadsheet full of sample data
SampleID
Metadata...
Taxa1
Taxa2...
PC1
PC2
PC3
Sort and format as needed
Data mining or statistical software of your choice

Flowchart

1. SFF (raw 454 data, optional)
2. fasta/qual files
3. demultiplexing/quality filtering

(b) Sequence data preparation

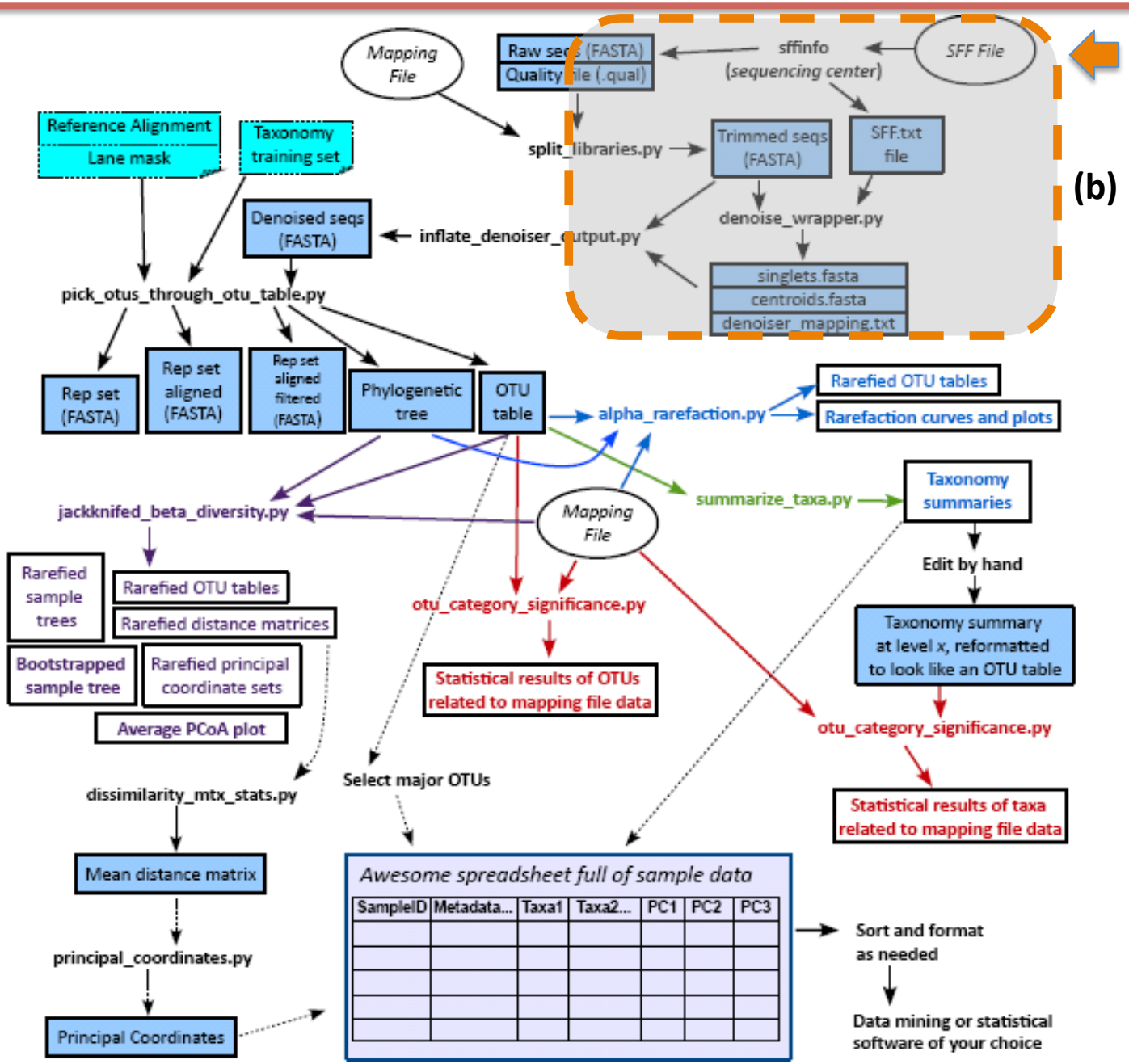
4. OTU picking
5. representative sequences
6. taxonomic assignments/tree building

(c) Operational Taxonomic Units (OTU) picking, Taxonomic assignment & inferring phylogeny

7. OTU table and downstream processing

(d) microbiome diversity analyses

Flowchart


Mapping File
Raw seqs (FASTA)
Quality file (.qual)
sffinfo
(sequencing center)
SFF File
Reference Alignment
Lane mask
Taxonomy training set
split_libraries.py
Trimmed seqs (FASTA)
SFF.txt file
(b)
Denoised seqs (FASTA)
inflate_denoiser_output.py
denoise_wrapper.py
singlets.fasta
centroids.fasta
denoiser_mapping.txt
pick_otus_through_otu_table.py
Rep set (FASTA)
Rep set aligned (FASTA)
Rep set aligned filtered (FASTA)
Phylogenetic tree
OTU table
alpha_rarefaction.py
Rarefied OTU tables
Rarefaction curves and plots
summarize_taxa.py
Taxonomy summaries
jackknifed_beta_diversity.py
Mapping File
Edit by hand
Rarefied sample trees
Rarefied OTU tables
Rarefied distance matrices
Bootstrapped sample tree
Rarefied principal coordinate sets
Average PCoA plot
otu_category_significance.py
Statistical results of OTUs related to mapping file data
Taxonomy summary at level x, reformatted to look like an OTU table
otu_category_significance.py
Statistical results of taxa related to mapping file data
Select major OTUs
dissimilarity_mtx_stats.py
Mean distance matrix
principal_coordinates.py
Principal Coordinates
Awesome spreadsheet full of sample data
SampleID
Metadata...
Taxa1
Taxa2...
PC1
PC2
PC3
Sort and format as needed
Data mining or statistical software of your choice

b) Data preparation

We will start out with raw sequencing data generated on 454

- Sequences (.fna): 454 generated FASTA file.
- Quality Scores (.qual): 454 generated quality score file
- experimental data about the samples (Mapping File) generated by user.

#SampleID	BarcodeSequence	LinkerPrimerSequence	Treatment	DOB	Description
#Example mapping file for the QIIME analysis package.					
PC.354	AGCACGAGCCTA	YATGCTGCCTCCCGTAGGAGT	Control	20061218	Control_mouse_I.D._354
PC.355	AACTCGTCGATG	YATGCTGCCTCCCGTAGGAGT	Control	20061218	Control_mouse_I.D._355
PC.356	ACAGACCACTCA	YATGCTGCCTCCCGTAGGAGT	Control	20061126	Control_mouse_I.D._356
PC.481	ACCAGCGACTAG	YATGCTGCCTCCCGTAGGAGT	Control	20070314	Control_mouse_I.D._481
PC.593	AGCAGCACTTGT	YATGCTGCCTCCCGTAGGAGT	Control	20071210	Control_mouse_I.D._593
PC.607	AACTGTGCGTAC	YATGCTGCCTCCCGTAGGAGT	Fast	20071112	Fasting_mouse_I.D._607
PC.634	ACAGAGTCGGCT	YATGCTGCCTCCCGTAGGAGT	Fast	20080116	Fasting_mouse_I.D._634
PC.635	ACCGCAGAGTCA	YATGCTGCCTCCCGTAGGAGT	Fast	20080116	Fasting_mouse_I.D._635
PC.636	ACGGTGAGTGTC	YATGCTGCCTCCCGTAGGAGT	Fast	20080116	Fasting_mouse_I.D._636

On Hoffman2, copy all the files you need to scratch folder
cd /u/scratch/f/xxx/ change to your ID
cd /u/local/apps/qiime/1.8.0/examples/qiime_tutorial/

cp /u/local/apps/qiime/1.8.0/examples/qiime_tutorial/file /u/scratch/f/xxx/

qiime
Quantitative Insights Into Microbial Ecology

b) Data preparation

Standard flowgram format (SFF) is a binary file format used to encode results of 454 pyrosequencing

```
process_sff.py -i sffs/ -f -o output_dir
```

- Sequences (.fna): 454 generated FASTA file.
- Quality Scores (.qual): 454 generated quality score file

b) Data preparation

Mapping File

#SampleID	BarcodeSequence	LinkerPrimerSequence	Treatment	DOB	Description
#Example mapping file for the QIIME analysis package.					
PC.354	AGCACGAGCCTA	YATGCTGCCTCCCGTAGGAGT	Control	20061218	Control_mouse_I.D._354
PC.355	AACTCGTCGATG	YATGCTGCCTCCCGTAGGAGT	Control	20061218	Control_mouse_I.D._355
PC.356	ACAGACCACTCA	YATGCTGCCTCCCGTAGGAGT	Control	20061126	Control_mouse_I.D._356
PC.481	ACCAGCGACTAG	YATGCTGCCTCCCGTAGGAGT	Control	20070314	Control_mouse_I.D._481
PC.593	AGCAGCACTTGT	YATGCTGCCTCCCGTAGGAGT	Control	20071210	Control_mouse_I.D._593
PC.607	AACTGTGCGTAC	YATGCTGCCTCCCGTAGGAGT	Fast	20071112	Fasting_mouse_I.D._607
PC.634	ACAGAGTCGGCT	YATGCTGCCTCCCGTAGGAGT	Fast	20080116	Fasting_mouse_I.D._634
PC.635	ACCGCAGAGTCA	YATGCTGCCTCCCGTAGGAGT	Fast	20080116	Fasting_mouse_I.D._635
PC.636	ACGGTGAGTGTC	YATGCTGCCTCCCGTAGGAGT	Fast	20080116	Fasting_mouse_I.D._636

assign the multiplexed reads to samples based on their nucleotide barcode (*demultiplexing*)

```
split_libraries.py -m Map.txt -f Example.fna -q Example.qual -o split_library_output
```

Quantitative Insights Into Microbial Ecology

b) Data preparation

- **split_library_log.txt** : summary of demultiplexing and quality filtering, including the number of reads detected for each sample and a brief summary of any reads that were removed due to quality considerations.
- **histograms.txt** : tab-delimited file shows the number of reads at regular size intervals before and after splitting.
- **seqs.fna** : fasta formatted

This step also performs quality filtering based on the characteristics of each sequence, removing any low quality or ambiguous reads.

```
split_libraries.py -m Map.txt -f Example.fna -q Example.qual -o split_library_output
```

-l, --min_seq_length
Minimum sequence length, in nucleotides [default: 200]
-t, --trim_seq_length
Calculate sequence lengths after trimming primers and barcodes [default: False]
-s, --min_qual_score
Min average qual score allowed in read [default: 25]

qiime
Quantitative Insights Into Microbial Ecology

b) Data preparation

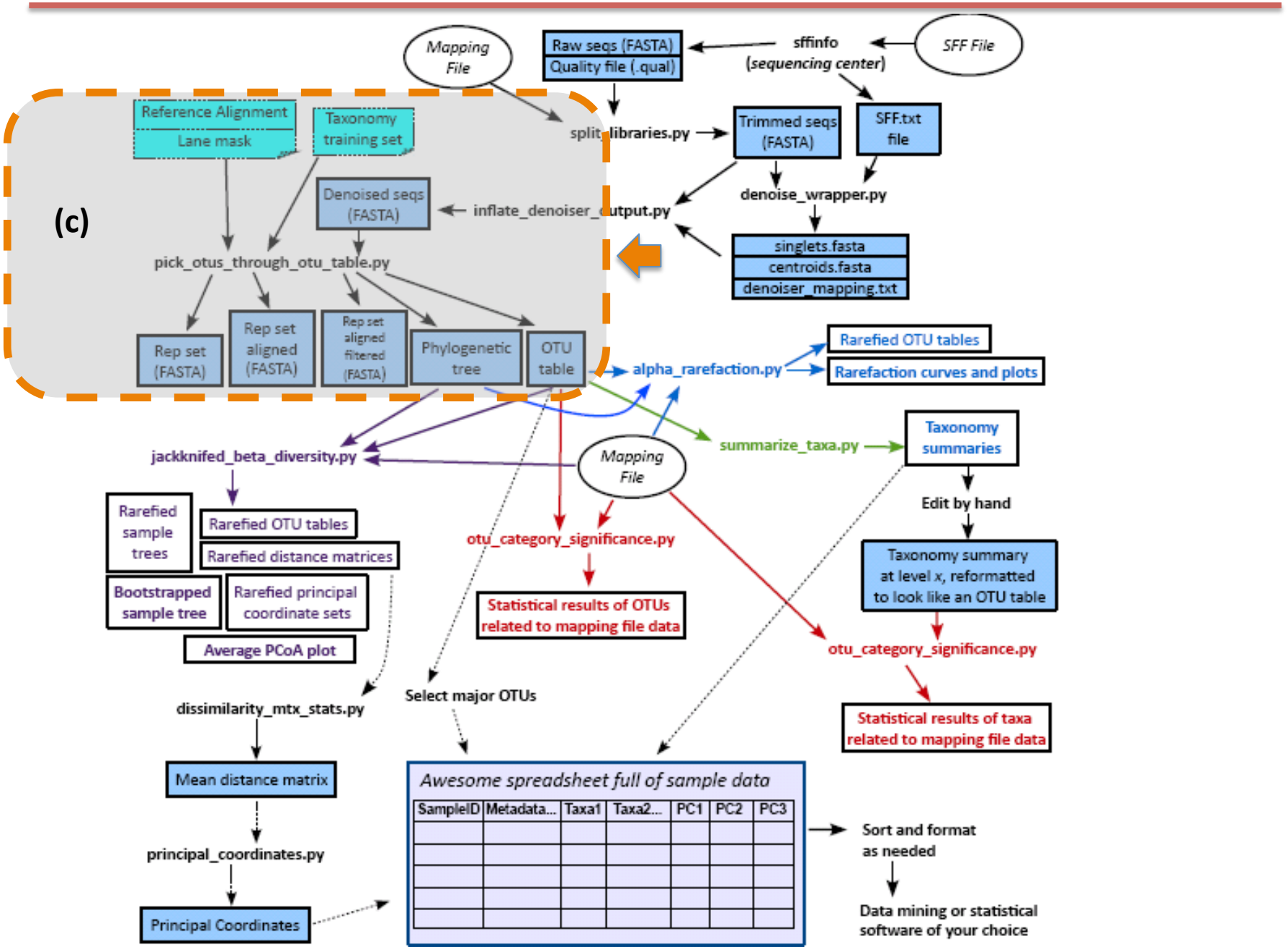
Illumina runs:
demultiplexing of Fastq sequence data where barcodes and sequences are contained in two separate fastq files.

Demultiplex and quality filter (at Phred >= Q20) one lane of Illumina fastq data

```
split_libraries_fastq.py –i  fastq.gz –b barcode.fastq.gz --rev_comp_mapping_barcodes –o out_dir/
-m map.txt -q 19
```

qiime
Quantitative Insights Into Microbial Ecology

Flowchart


Mapping File
Raw seqs (FASTA)
Quality file (.qual)
sffinfo
(sequencing center)
SFF File
(c)
Reference Alignment
Lane mask
Taxonomy training set
split_libraries.py
Trimmed seqs (FASTA)
SFF.txt file
Denoised seqs (FASTA)
inflate_denoiser_output.py
denoise_wrapper.py
singlets.fasta
centroids.fasta
denoiser_mapping.txt
pick_otus_through_otu_table.py
Rep set (FASTA)
Rep set aligned (FASTA)
Rep set aligned filtered (FASTA)
Phylogenetic tree
OTU table
alpha_rarefaction.py
Rarefied OTU tables
Rarefaction curves and plots
summarize_taxa.py
Taxonomy summaries
jackknifed_beta_diversity.py
Mapping File
Edit by hand
Rarefied sample trees
Rarefied OTU tables
Rarefied distance matrices
Bootstrapped sample tree
Rarefied principal coordinate sets
Average PCoA plot
otu_category_significance.py
Statistical results of OTUs related to mapping file data
Taxonomy summary at level x, reformatted to look like an OTU table
otu_category_significance.py
Statistical results of taxa related to mapping file data
dissimilarity_mtx_stats.py
Select major OTUs
Mean distance matrix
Awesome spreadsheet full of sample data
SampleID
Metadata...
Taxa1
Taxa2...
PC1
PC2
PC3
Sort and format as needed
principal_coordinates.py
Principal Coordinates
Data mining or statistical software of your choice

Flowchart

1. SFF (raw 454 data, optional)
2. fasta/qual files
3. demultiplexing/quality filtering

(b) Sequence data preparation

4. OTU picking
5. representative sequences
6. taxonomic assignments/tree building

(c) Operational Taxonomic Units (OTU) picking, Taxonomic assignment & inferring phylogeny

7. OTU table and downstream processing

(d) microbiome diversity analyses

c) OTU analysis

This workflow consists of the following steps:

OTU picking, Taxonomic assignment

c1) Pick OTUs based on sequence similarity within the reads (pick_otus.py)

c2) Pick a representative sequence for each OTU (pick_rep_set.py)

c3) Assign taxonomy to OTU representative sequences (assign_taxonomy.py)

Inferring phylogeny

c4) Align OTU representative sequences (align_seqs.py)

c5) Filter the alignment (filter_alignment.py)

c6) Build a phylogenetic tree (make_phylogeny.py)

Summarize the OTU table

c7) Make the OTU table (make_otu_table.py)

qiime
Quantitative Insights Into Microbial Ecology

c) OTU analysis

c1) Pick OTUs based on sequence similarity within the reads

The OTU picking step assigns similar sequences to operational taxonomic units (OTUs) by clustering sequences based on a user-defined similarity threshold. Sequences which are similar at or above the threshold level are taken to represent the presence of a taxonomic unit in the sequence collection.

De novo OTU picking based on sequence similarity within the reads (pick_otus.py)

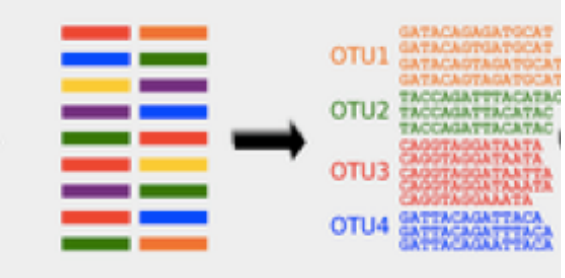
```
pick_otus.py -i seqs.fna -o picked_otus_90_percent_rev/ -s 0.97 -z
```

-s 97% sequence similarity & –z enable reverse strand matching

e.g., a genus, when the similarity threshold is set at 0.94; or a species, when the similarity threshold is set at 0.97

The output file:

0	seq1	seq5	
1	seq2		
2	seq3		
3	seq4	seq6	seq7



qiime
Quantitative Insights Into Microbial Ecology

OTU picking strategies in QIIME

Closed-reference OTU picking

If the user provides taxonomic assignments for sequences in the reference database, those are assigned to OTUs.

```
pick_closed_reference_otus.py –i seqs.fna –r  ref/refseqs.fna –o dir-out/ -t ref/taxa.txt
```

-s, --assign_taxonomy
Assign taxonomy to each sequence
-a, --parallel
Run in parallel where available

OTU picking strategies in QIIME

Open-reference OTU picking

preferred strategy for OTU picking:

```
pick_open_reference_otus.py –i seqs.fna –r refseqs.fna –o dir/
```

-a, --parallel
Run in parallel where available

Quantitative Insights Into Microbial Ecology

c) OTU analysis

This workflow consists of the following steps:

c1) Pick OTUs based on sequence similarity within the reads (pick_otus.py)

c2) Pick a representative sequence for each OTU (pick_rep_set.py)

After picking OTUs, you can pick a representative set of sequences. For each OTU, you will end up with one sequence that can be used in subsequent analyses.

```
pick_rep_set.py -i seqs_otus.txt -f seqs.fna -o rep_set.fna
```

Quantitative Insights Into Microbial Ecology

c) OTU analysis

This workflow consists of the following steps:

c1) Pick OTUs based on sequence similarity within the reads (pick_otus.py)

c2) Pick a representative sequence for each OTU (pick_rep_set.py)

c3) Assign taxonomy to OTU representative sequences (assign_taxonomy.py)

Assignment with the consensus taxonomy assigner (default, uclust)

```
assign_taxonomy.py -i repr_set.fna -r ref_seq_set.fna -t id_to_taxonomy.txt
```

Reference and id-to-taxonomy for 16S rRNA sequences from Greengenes (gg_13_5.fasta & gg_13_5_taxonomy)
(http://greengenes.secondgenome.com/downloads/database/13_5)

Assignment with the RDP Classifier: The RDP Classifier assigns taxonomies using Naive Bayes classification.

```
assign_taxonomy.py -i repr_set.fna -m rdp -c 0.80
```

-c, --confidence
Minimum confidence to record an assignment

The output: sequence id (1st column), taxonomy (2nd column) and quality score (3rd column).

```
denovo367 k__Bacteria; p__Bacteroidetes; c__Bacteroidia; o__Bacteroidales; f__S24-7; g__; s__ 1.00
```

qiime
Quantitative Insights Into Microbial Ecology

c) OTU analysis

This workflow consists of the following steps:

c1) Pick OTUs based on sequence similarity within the reads (pick_otus.py)
c2) Pick a representative sequence for each OTU (pick_rep_set.py)
c3) Assign taxonomy to OTU representative sequences (assign_taxonomy.py)

Alignment of the OTU representative sequences and phylogeny inference is necessary if phylogenetic metrics such as UniFrac will be used in microbiome diversity analyses.

c4) Align OTU representative sequences (align_seqs.py)
c5) Filter the alignment (filter_alignment.py)
c6) Build a phylogenetic tree (make_phylogeny.py)

c7) Make the OTU table (make_otu_table.py)

qiime
Quantitative Insights Into Microbial Ecology

c) OTU analysis

This workflow consists of the following steps:

c1) Pick OTUs based on sequence similarity within the reads (pick_otus.py)
c2) Pick a representative sequence for each OTU (pick_rep_set.py)
c3) Assign taxonomy to OTU representative sequences (assign_taxonomy.py)
c4) Align OTU representative sequences (align_seqs.py)

PyNAST - The default alignment, implementation of the NAST alignment algorithm. The NAST aligns each candidate sequence to the best-matching sequence in a pre-aligned database of sequences ("template" sequence). Candidate sequences are not permitted to introduce new gap into template database, the algorithm introduces local mis-alignments to preserve the existing template sequence.

```
align_seqs.py -i rep_set.fna
```

Quantitative Insights Into Microbial Ecology

Chimera checking sequences with QIIME

Applying ChimeraSlayer:

```
perl /macqiime/microbiomeutil_2010-04-29/ChimeraSlayer/ChimeraSlayer.pl --query rep_set_aligned.fasta
```

or

```
identify_chimeric_seqs.py -m ChimeraSlayer -i rep_set_aligned.fasta -a /macqiime/microbiomeutil_2010-04-29/RESOURCES/
rRNA16S.gold.NAST_ALIGNED.fasta -o chimeric_seqs.txt
```

remove chimeric sequences from your alignment using your chimeric sequence list

```
filter_fasta.py -f rep_set_aligned.fasta -o non_chimeric_rep_set_aligned.fasta -s chimeric_seqs.txt -n
```

qiime
Quantitative Insights Into Microbial Ecology

c) OTU analysis

This workflow consists of the following steps:

c1) Pick OTUs based on sequence similarity within the reads (pick_otus.py)
c2) Pick a representative sequence for each OTU (pick_rep_set.py)
c3) Assign taxonomy to OTU representative sequences (assign_taxonomy.py)
c4) Align OTU representative sequences (align_seqs.py)
c5) Filter the alignment (filter_alignment.py)

This script will remove positions which are gaps in every sequence (not covered by amplicon). Additionally, this will remove non-conserved positions, which are uninformative for tree building.

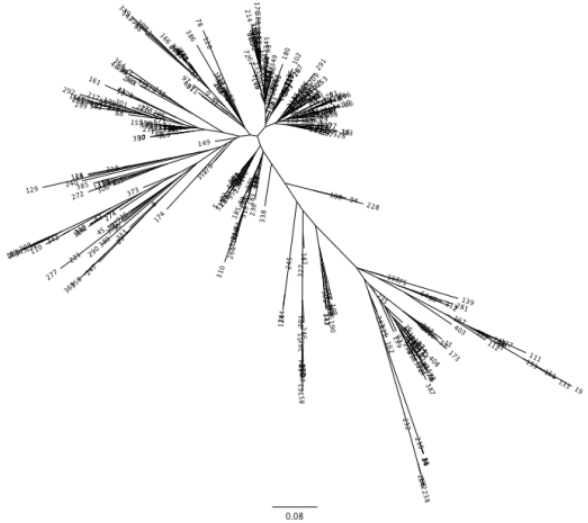
```
filter_alignment.py -i seqs_rep_set_aligned.fasta -o filtered_alignment/
```

c) OTU analysis

This workflow consists of the following steps:

c1) Pick OTUs based on sequence similarity within the reads (pick_otus.py)
c2) Pick a representative sequence for each OTU (pick_rep_set.py)
c3) Assign taxonomy to OTU representative sequences (assign_taxonomy.py)
c4) Align OTU representative sequences (align_seqs.py)
c5) Filter the alignment (filter_alignment.py)
c6) Build a phylogenetic tree (make_phylogeny.py)

```
make_phylogeny.py –i rep_set_aligned_pfiltered.fasta -o rep_phylo.tre
```



qiime
Quantitative Insights Into Microbial Ecology

c) OTU analysis

This workflow consists of the following steps:

c1) Pick OTUs based on sequence similarity within the reads (pick_otus.py)
c2) Pick a representative sequence for each OTU (pick_rep_set.py)
c3) Assign taxonomy to OTU representative sequences (assign_taxonomy.py)

Alignment of the OTU representative sequences and phylogeny inference is necessary if phylogenetic metrics such as UniFrac will be used in microbiome diversity analyses.

c4) Align OTU representative sequences (align_seqs.py)
c5) Filter the alignment (filter_alignment.py)
c6) Build a phylogenetic tree (make_phylogeny.py)

Summarize the OTU table
c7) Make the OTU table (make_otu_table.py)

qiime
Quantitative Insights Into Microbial Ecology

c) OTU analysis

This workflow consists of the following steps:

c1) Pick OTUs based on sequence similarity within the reads (pick_otus.py)
c2) Pick a representative sequence for each OTU (pick_rep_set.py)
c3) Assign taxonomy to OTU representative sequences (assign_taxonomy.py)
c4) Align OTU representative sequences (align_seqs.py)
c5) Filter the alignment (filter_alignment.py)
c6) Build a phylogenetic tree (make_phylogeny.py)
c7) Make the OTU table (make_otu_table.py)

The script tabulates the number of times an OTU is found in each sample, and adds the taxonomic predictions for each OTU

```
make_otu_table.py –i seqs_otus.txt –t rep_set_tax_assignments.txt –o otu_table.biom -e chimeric_seqs.txt
```

-e remove chimeric OTU

Quantitative Insights Into Microbial Ecology

c) OTU analysis

Summarize the OTU table

```
biom summarize-table –i otu_table.biom
```

```
Num samples: 9
Total count: 1337
Counts/sample summary:
Min: 146.0
Max: 150.0
Median: 149.000
Mean: 148.556
Std. dev.: 1.257
Counts/sample detail:
PC.481: 146.0
PC.355: 147.0
PC.636: 148.0
.....
```

Convert table from BIOM to tab-separated text format

```
biom convert –i otu_table.biom -o otu_table.txt -b
```

qiime
Quantitative Insights Into Microbial Ecology

c) OTU analysis

Summarize communities by taxonomic composition

```
summarize_taxa_through_plots.py -i otu_table.biom -o taxa_summary –m Fasting_Map.txt
```

the relative abundances of taxa (at the different level) within each sample

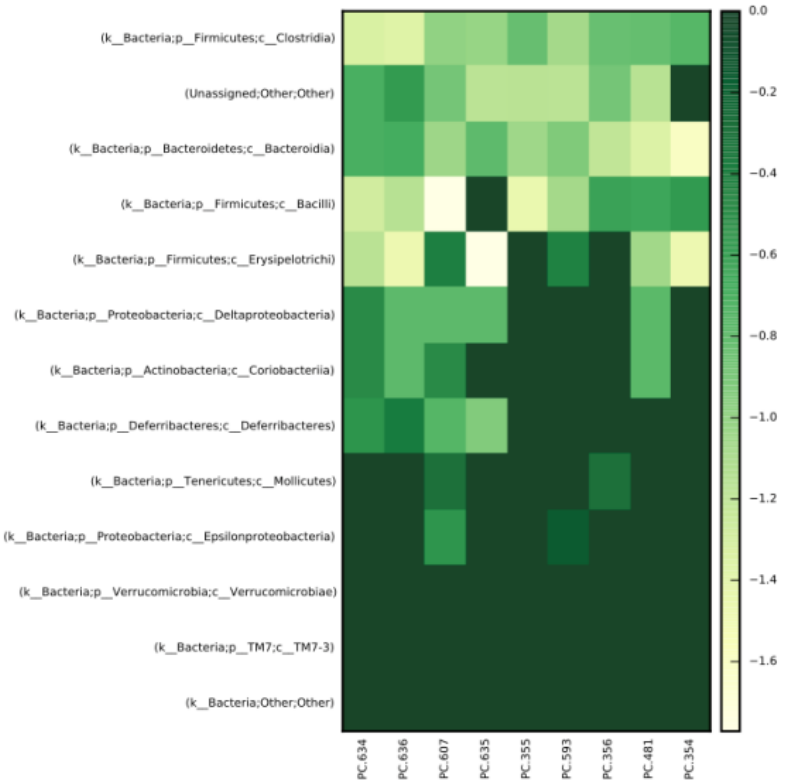
```
#OTU ID PC.636 PC.635 PC.356 PC.481 PC.354 PC.593 PC.355 PC.607 PC.634
k__Bacteria;Other;Other 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.00671140939597 0.0
k__Bacteria;p__Actinobacteria;c__Coriobacteriia 0.00675675675676 0.0 0.0 0.00684931506849 0.0 0.0 0.0 0.0134228187919 0.013333333333
k__Bacteria;p__Bacteroidetes;c__Bacteroidia 0.675675675676 0.530201342282 0.2 0.143835616438 0.0805369
```

qiime
Quantitative Insights Into Microbial Ecology

c) OTU analysis

Make a taxonomy heatmap

```
make_otu_heatmap.py -m Fasting_Map.txt -i otu_table.biom -o otu_table_heatmap.pdf
```

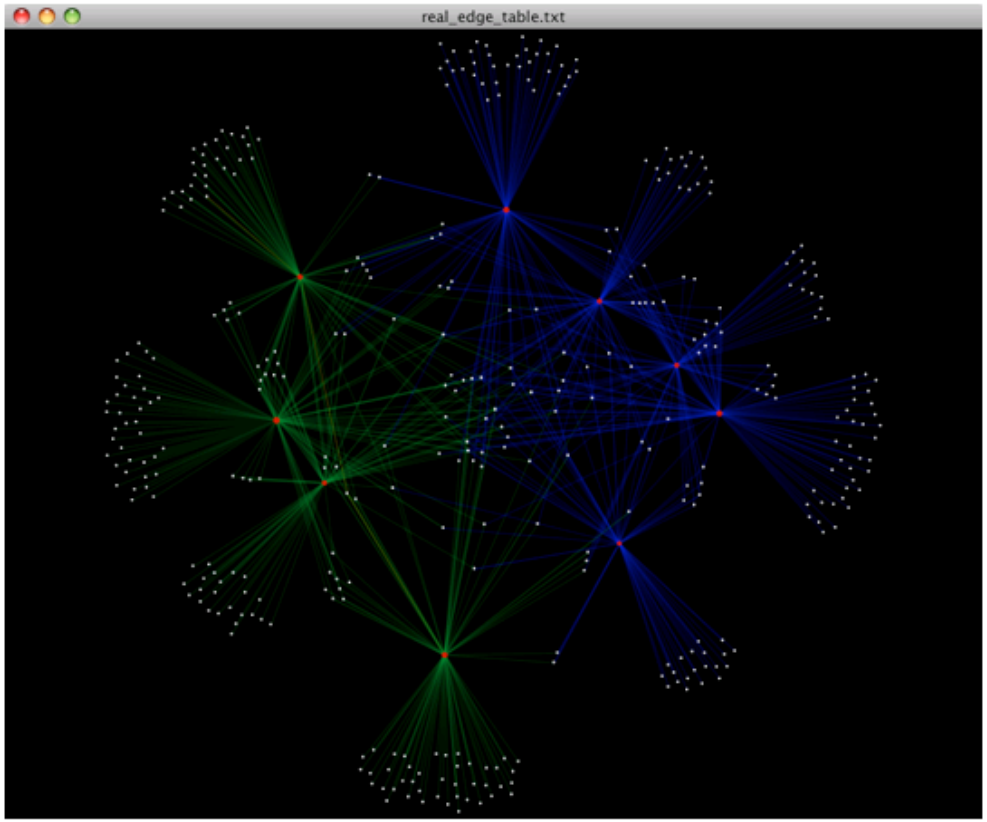


Quantitative Insights Into Microbial Ecology

c) OTU analysis

Make an OTU network

```
make_otu_network.py -m Fasting_Map.txt -i otu_table.biom -o network
```



red circle represents a sample and white square represents an OTU. The lines represent the OTUs present in a particular sample

Quantitative Insights Into Microbial Ecology

c) OTU analysis

This workflow consists of the following steps:

OTU picking, Taxonomic assignment

c1) Pick OTUs based on sequence similarity within the reads (pick_otus.py)

c2) Pick a representative sequence for each OTU (pick_rep_set.py)

c3) Assign taxonomy to OTU representative sequences (assign_taxonomy.py)

Inferring phylogeny

c4) Align OTU representative sequences (align_seqs.py)

c5) Filter the alignment (filter_alignment.py)

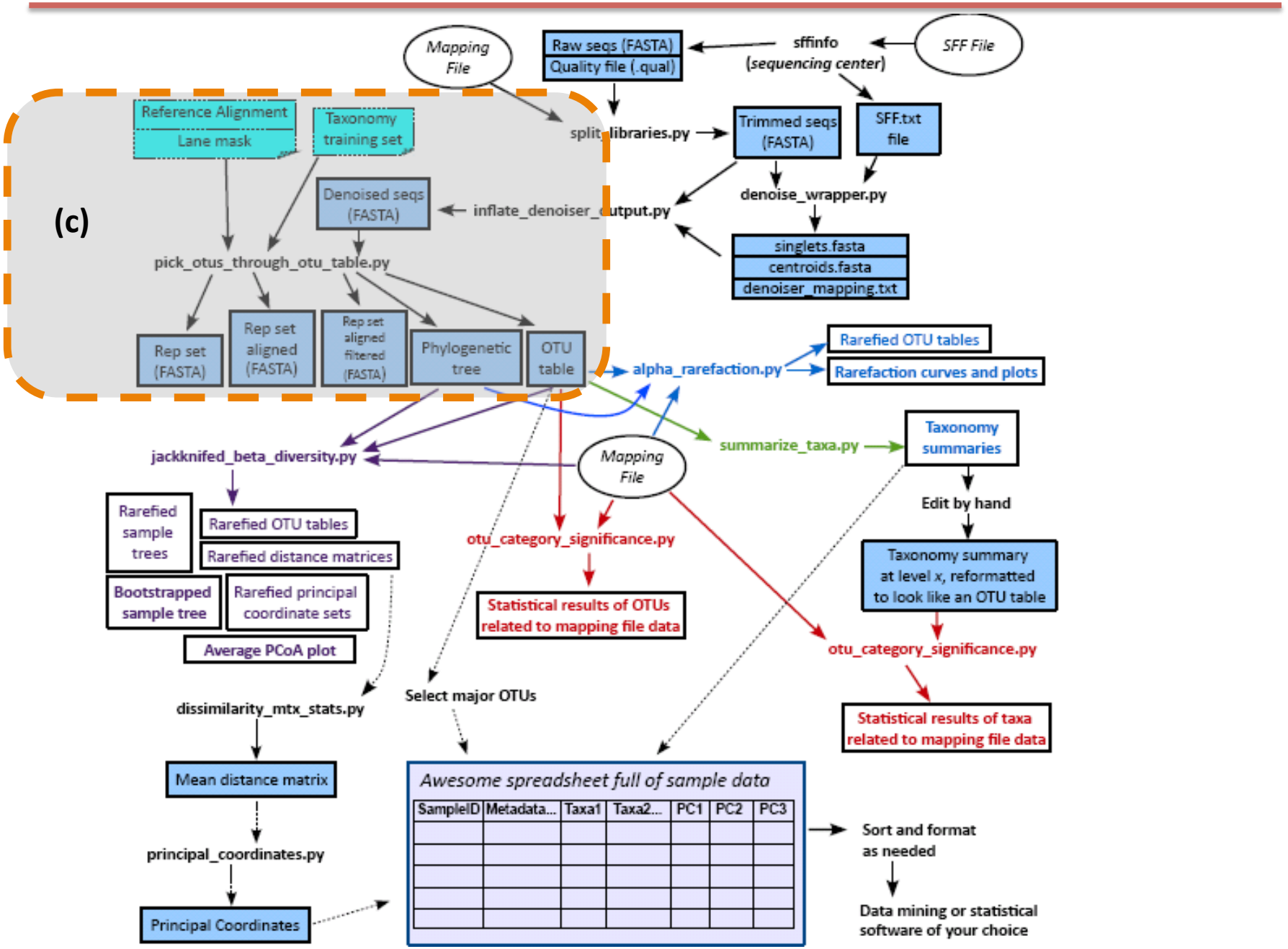
c6) Build a phylogenetic tree (make_phylogeny.py)

Summarize the OTU table

c7) Make the OTU table (make_otu_table.py)

qiime
Quantitative Insights Into Microbial Ecology

Flowchart


Mapping File
Raw seqs (FASTA)
Quality file (.qual)
sffinfo
(sequencing center)
SFF File
Reference Alignment
Lane mask
Taxonomy training set
split_libraries.py
Trimmed seqs (FASTA)
SFF.txt file
(c)
Denoised seqs (FASTA)
inflate_denoiser_output.py
denoise_wrapper.py
singlets.fasta
centroids.fasta
denoiser_mapping.txt
pick_otus_through_otu_table.py
Rep set (FASTA)
Rep set aligned (FASTA)
Rep set aligned filtered (FASTA)
Phylogenetic tree
OTU table
alpha_rarefaction.py
Rarefied OTU tables
Rarefaction curves and plots
summarize_taxa.py
Taxonomy summaries
jackknifed_beta_diversity.py
Mapping File
Edit by hand
Rarefied sample trees
Rarefied OTU tables
Rarefied distance matrices
Bootstrapped sample tree
Rarefied principal coordinate sets
Average PCoA plot
otu_category_significance.py
Statistical results of OTUs related to mapping file data
Taxonomy summary at level x, reformatted to look like an OTU table
otu_category_significance.py
Statistical results of taxa related to mapping file data
Select major OTUs
dissimilarity_mtx_stats.py
Mean distance matrix
principal_coordinates.py
Principal Coordinates
Awesome spreadsheet full of sample data
SampleID
Metadata...
Taxa1
Taxa2...
PC1
PC2
PC3
Sort and format as needed
Data mining or statistical software of your choice