Gray text: Custom name or part that changes with the user

Text on yellow background: Downloaded files, project-specific files or files I gave

General non-black characters Fixed format, should not be modified

Red text : Eye-catching, non-code

>> : at docker container Instructions entered within the order

$> : Instructions entered in the physical environment

PS Connection last introduced: [supercomputers](https://docs.google.com/document/d/1AUo0gO0j7083l-ulGyvCzKbmnkIi2CKtaWJ0DsoKQLI/edit?usp=sharing)

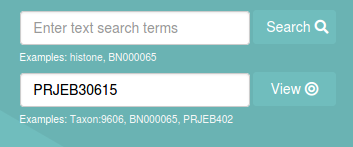
Start Archive:[Weis起始.zip](https://drive.google.com/file/d/1HqkD-o43cu1nX4RW7eyRJjcVEHB-vwml/view?usp=share_link)

**1、Download sequence**

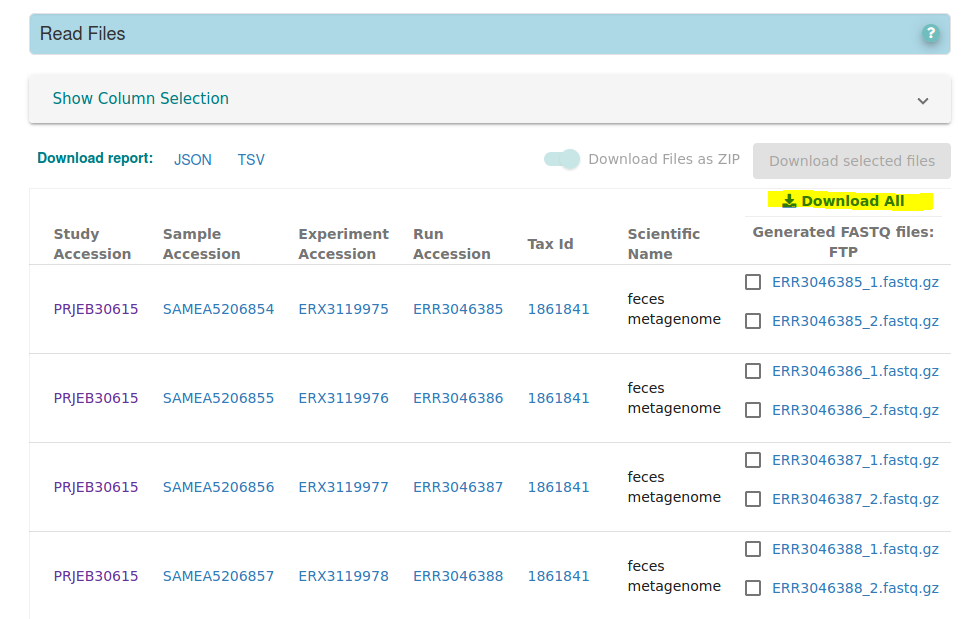
**1. Go to the ENA website**

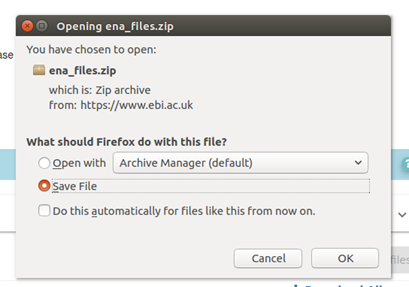
<https://www.ebi.ac.uk/ena/browser/home>

**2. Enter the item number in the upper right corner PRJEB30615**



**3. Download the FASTQ file (default folder /home/shenglab/Downloads)**



(Now sh file)

**4. Create a project folder in /data2/shenglab => weis\_qiime2 (All analysis is performed in data2.)**

$> docker run -it --name shenglab -v /:/tmp ubuntu bash # (The '/' to the left of the colon is the root directory, which means that the entire computer has entered the virtual environment, so you need to be careful with this operation)

>> cd /tmp # (Now the /tmp of the virtual environment corresponds to the root directory of the physical environment, so you should always pay attention to where your current working directory is)

>> mkdir /tmp/data2/shenglab1/weis\_1

At /data2/shenglab/weis\_1 establish gene folder fastq\_files

>> mkdir /tmp/data2/shenglab1/weis\_1/fastq\_files

**5. Once the data has been downloaded, Move to/data2/shenglab/weis\_qiime2/fastq\_files**

$> mkdir ena\_files (Please at Downloads and put sh file inside P.S. Pay attention to the working directory)

$> cd /ena\_files

$> chmod a+x ena.sh

$> sh ena.sh cd ./ena\_files sh

>> mv /tmp/home/shenglab1/Downloads/ena\_files /tmp/data2/shenglab1/weis\_1/fastq\_files/ena\_files

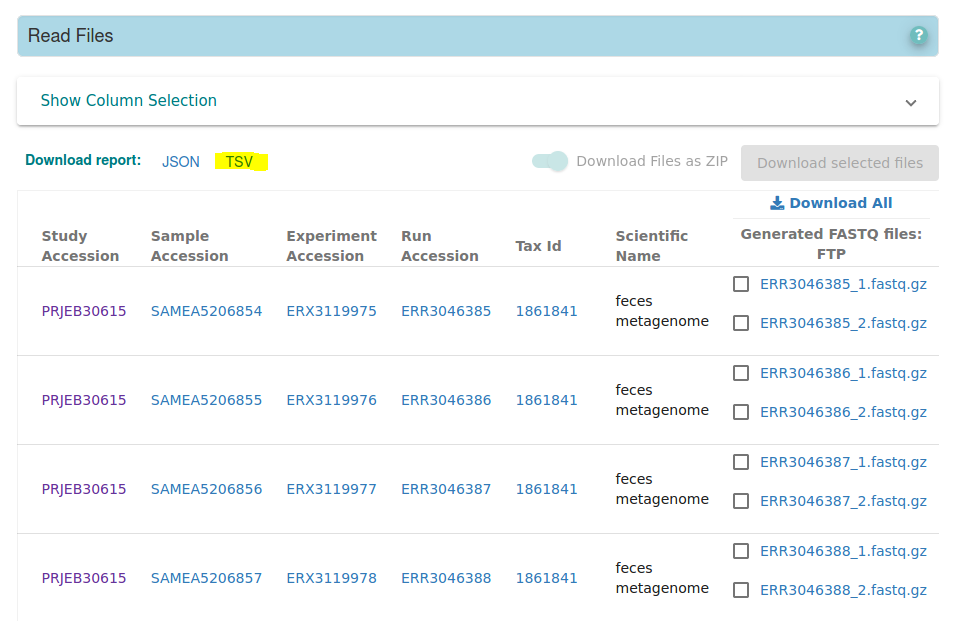
**2. Make metadata (on your own computer).**

**1.**

Create a folder (named: Weis).

**2.**

Go to the ENA website and download the report to Weis (File name: filereport\_read\_run\_PRJEB30615\_tsv.txt)



**3.**

Add the following files to Weis:

(1) metadata\_crowling.R

(2) supplementary data.xlsx

(3) Crawling\_KO\_number\_LEVEL23.R

(4) table1.R

(5) qiime2\_postanalysis.R

(6) qiime2\_alpha\_plot.R

(7) qiime2\_beta\_plot.R

(8) Wilcoxon\_and\_volcano\_plot.R

(9) before\_stamp.R

(10) STAMP\_2\_1\_3.exe

Open Rstudio (you need to install R, Rstudio first) and create the project

File => New Project => Existing Directory => Link to the Weis folder (After that, all from Weis. Rproj file into the project)

**4.**

Execute metadata\_crowling.R content，取得(1) weis\_metadata\_1210.csv , (2) id\_list.csv

**5.**

upload(1) weis\_metadata\_1210.csv , (2) id\_list.csv to Drive，and download to supercomputer (By default is saved /home/shenglab/Downloads)

**6. Move to /data2/shenglab/weis\_qiime2**

>> mv /tmp/home/shenglab1/Downloads/weis\_metadata\_1210.csv /tmp/data2/shenglab1/weis\_1/weis\_metadata\_1210.csv

>> mv /tmp/home/shenglab1/Downloads/id\_list.csv /tmp/data2/shenglab1/weis\_1/id\_list.csv

>> mv /tmp/home/shenglab1/Downloads/weis1.sh /tmp/data2/shenglab1/weis\_1/weis1.sh

>> mv /tmp/home/shenglab1/Downloads/weis2.sh /tmp/data2/shenglab1/weis\_1/weis2.sh

>> mv /tmp/home/shenglab1/Downloads/weis3.sh /tmp/data2/shenglab1/weis\_1/weis3.sh

**7. Remove container**

$> docker stop $(docker ps -a -q --filter="name=shenglab")

$> docker rm $(docker ps -a -q --filter="name=shenglab")

**3. Prepare materials**

**1.**

$> docker run -it --name shenglab -v /data2/shenglab1/weis\_1:/tmp ubuntu bash

>> cd /tmp

>> mkdir /tmp/fastq\_files/arr\_seq

**START OF WEIS1**

**2. Create the required folder to store the results of each sample 4 times**

>> i=1 n=0

while IFS=, read line

do

col1=$(echo ${line} | cut -d , -f 1 | tr -d "\r\n" | sed -e 's/^"//' -e 's/"$//')

((n >= i)) && \

mkdir /tmp/fastq\_files/arr\_seq/${col1} && \

mkdir /tmp/fastq\_files/arr\_seq/${col1}/${col1}\_1 && \

mkdir /tmp/fastq\_files/arr\_seq/${col1}/${col1}\_2

((n++))

done < weis\_metadata\_1210.csv

**3. Unzip to the corresponding folder**

>> while IFS=, read line

do

col1=$(echo ${line} | cut -d , -f 1 | tr -d "\r\n" | sed -e 's/^"//' -e 's/"$//')

col2=$(echo ${line} | cut -d , -f 2 | tr -d "\r\n" | sed -e 's/^"//' -e 's/"$//')

gzip /tmp/fastq\_files/ena\_files/${col1}\_1.fastq.gz --decompress --stdout > /tmp/fastq\_files/arr\_seq/${col2}/${col2}\_1/${col1}\_1.fastq

gzip /tmp/fastq\_files/ena\_files/${col1}\_2.fastq.gz --decompress --stdout > /tmp/fastq\_files/arr\_seq/${col2}/${col2}\_2/${col1}\_2.fastq

done < id\_list.csv

**4. Merge the results of the four experiments into a single .fastq file**

>> while IFS=, read line

do

col2=$(echo ${line} | cut -d , -f 2 | tr -d "\r\n" | sed -e 's/^"//' -e 's/"$//')

list1=$(eval "ls /tmp/fastq\_files/arr\_seq/${col2}/${col2}\_1/\*.fastq")

list2=$(eval "ls /tmp/fastq\_files/arr\_seq/${col2}/${col2}\_2/\*.fastq")

for i in ${list1}; do cat $i >> /tmp/fastq\_files/arr\_seq/${col2}/${col2}\_1.fastq ; done

for i in ${list2}; do cat $i >> /tmp/fastq\_files/arr\_seq/${col2}/${col2}\_2.fastq ; done

done < id\_list.csv

**5. Compress into .fastq.gz files (will run a little longer)**

>> i=1 n=0

while IFS=, read line

do

col1=$(echo ${line} | cut -d , -f 1 | tr -d "\r\n" | sed -e 's/^"//' -e 's/"$//')

((n >= i)) && \

gzip /tmp/fastq\_files/arr\_seq/${col1}/${col1}\_1.fastq && \

gzip /tmp/fastq\_files/arr\_seq/${col1}/${col1}\_2.fastq

((n++))

done < weis\_metadata\_1210.csv

**6. Delete materials that are no longer needed**

>> i=1 n=0

while IFS=, read line

do

col1=$(echo ${line} | cut -d , -f 1 | tr -d "\r\n" | sed -e 's/^"//' -e 's/"$//')

((n >= i)) && \

rm -r /tmp/fastq\_files/arr\_seq/${col1}/${col1}\_1 && \

rm -r /tmp/fastq\_files/arr\_seq/${col1}/${col1}\_2

((n++))

done < weis\_metadata\_1210.csv

**7. make manifest (qiime2 has a specified format, Read more** [**Importing data — QIIME 2 2022.2.0 documentation**](https://docs.qiime2.org/2022.2/tutorials/importing/)**)**

>> echo -e "sample-id\tforward-absolute-filepath\treverse-absolute-filepath" > manifest.tsv

>> i=1 n=0

while IFS=, read line

do

col1=$(echo ${line} | cut -d , -f 1 | tr -d "\r\n" | sed -e 's/^"//' -e 's/"$//')

((n >= i)) && \

echo -e "${col1}\t"'$PWD'"/fastq\_files/arr\_seq/${col1}/${col1}\_1.fastq.gz\t"'$PWD'"/fastq\_files/arr\_seq/${col1}/${col1}\_2.fastq.gz" >> manifest.tsv

((n++))

done < weis\_metadata\_1210.csv

END OF WEIS1

**8. Remove container**

$> docker stop $(docker ps -a -q --filter="name=shenglab")

$> docker rm $(docker ps -a -q --filter="name=shenglab")

**4. QIIME2**

**1. Install qiime2 (Only the first time you need to run)**

>> docker pull quay.io/qiime2/core:2022.2

**2.**

>> docker run -it --name shenglab -v /data2/shenglab1/weis\_1:/tmp quay.io/qiime2/core:2022.2 /bin/bash

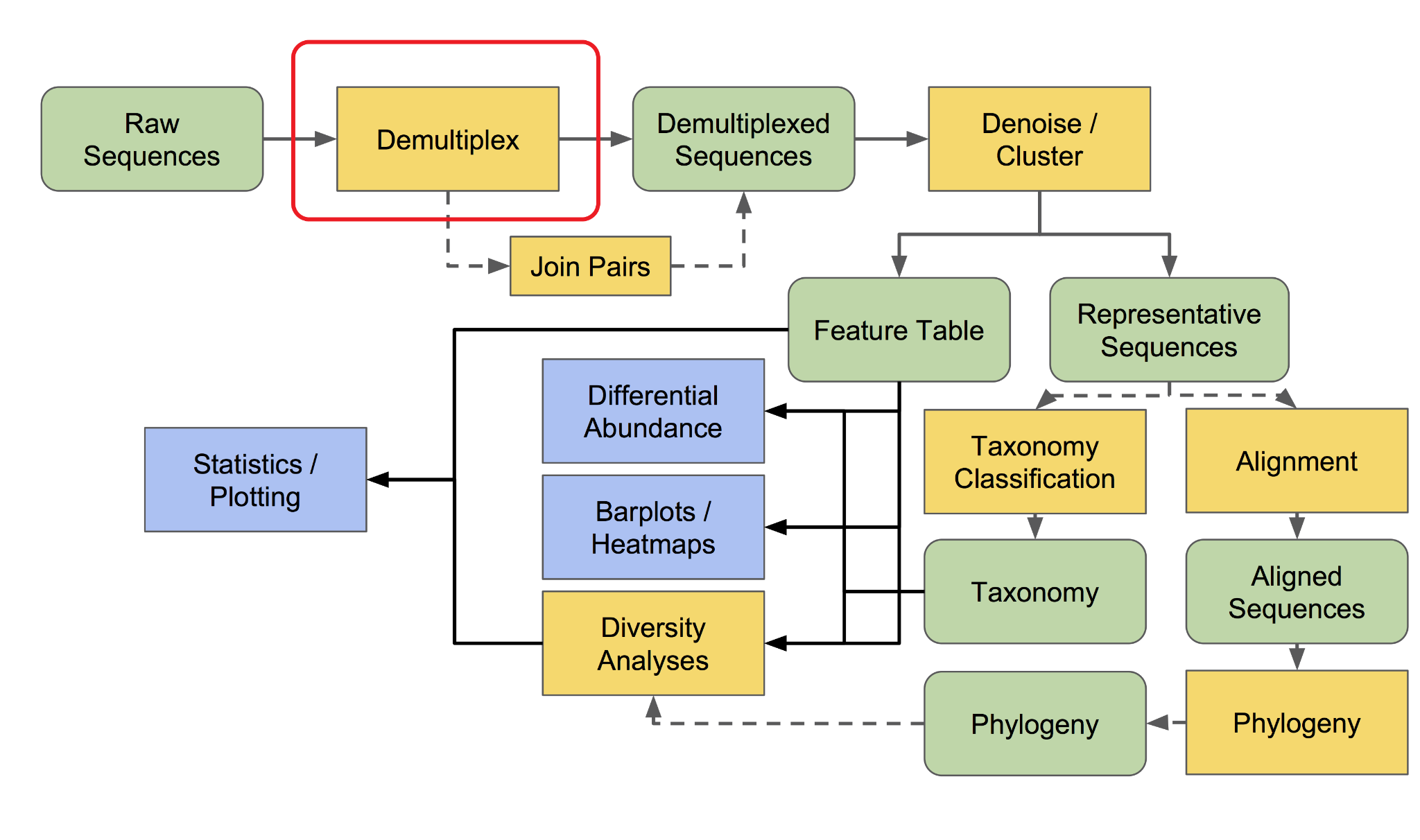
>> cd /tmp

**3. metadata turn csv to tsv**

>> pip install csvkit #(There may be warnings, but don't worry)

>> csvformat -T weis\_metadata\_1210.csv > weis\_metadata\_1210.tsv

**4. Import data into the QIIME2 common format .qza**

****

**START OF WEIS2**

>> qiime tools import \

--type 'SampleData[PairedEndSequencesWithQuality]' \

--input-path manifest.tsv \

--input-format PairedEndFastqManifestPhred33V2 \

--output-path demultiplexed-seqs.qza

**5. Remove adapter**

>> qiime cutadapt trim-paired \

--i-demultiplexed-sequences demultiplexed-seqs.qza \

--p-front-f AYTGGGYDTAAAGNG \

--p-front-r CCGTCAATTCMTTTRAGTTT \

--p-match-read-wildcards \

--p-match-adapter-wildcards \

--p-discard-untrimmed \

--o-trimmed-sequences trimmed-seqs.qza

**6. QIIME2 Visualization file format .qzv (Place the file in** [**QIIME 2 View**](https://view.qiime2.org/) **to display the visualization)**

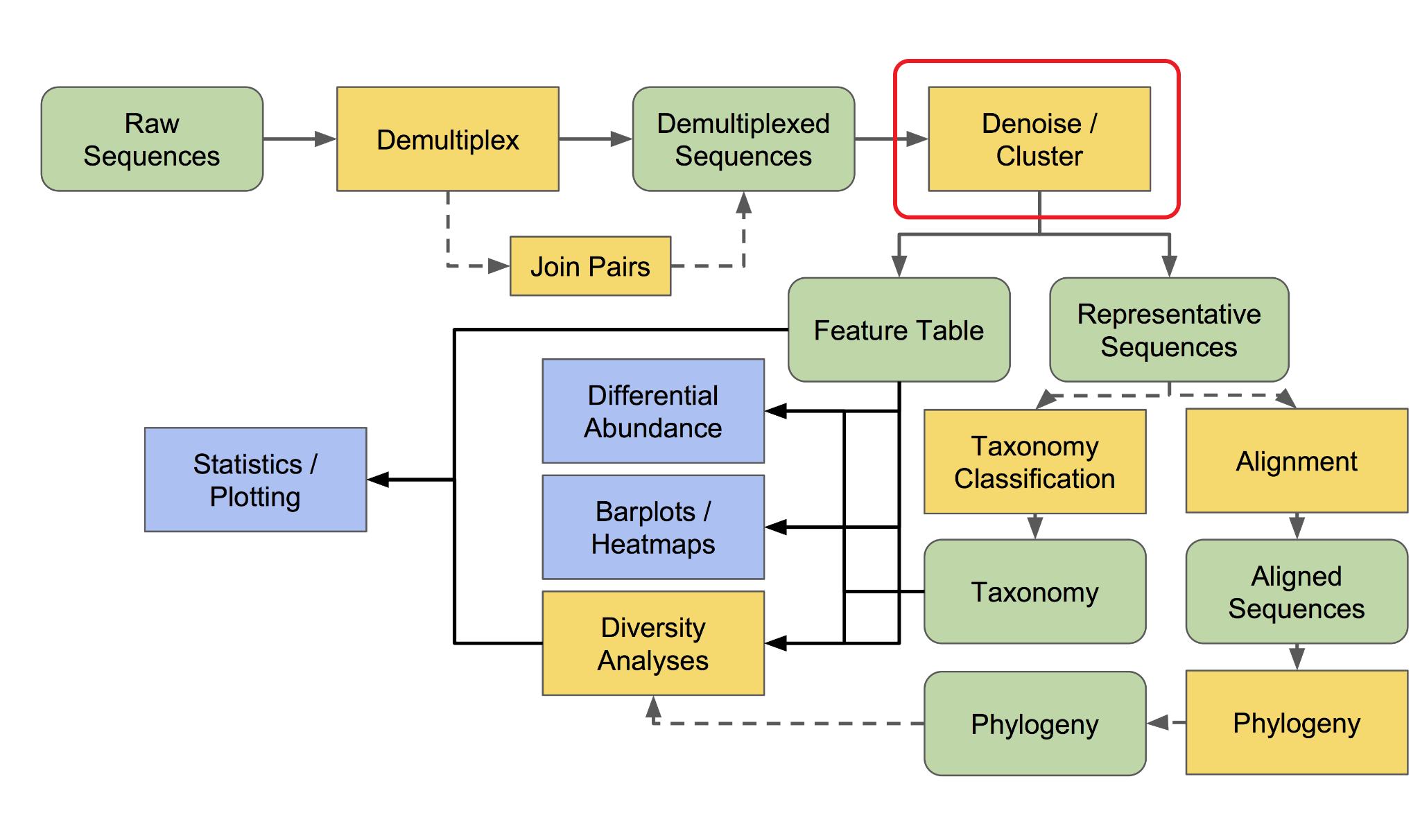
>> qiime demux summarize \

--i-data trimmed-seqs.qza \

--o-visualization trimmed-seqs.qzv

**7. dada2 (It will be a little long for a few hours; Observe trimmed-seqs.qzv to remove the poor-quality fractions)**

[**https://forum.qiime2.org/t/merging-quality-control-and-overlapping/12618/8**](https://forum.qiime2.org/t/merging-quality-control-and-overlapping/12618/8)

****

>> qiime dada2 denoise-paired \

--i-demultiplexed-seqs trimmed-seqs.qza \

--p-trim-left-f 0 \

--p-trim-left-r 0 \

--p-trunc-len-f 266 \

--p-trunc-len-r 174 \

--p-n-threads 0 \ # 0 stands for using all operating cores, note the computer memory capacity, and you can open Htop to view

--o-representative-sequences rep-seqs.qza \

--o-table table.qza \

--o-denoising-stats stats-dada2.qza

**8. DADa2 process information visualization**

>> qiime metadata tabulate \

--m-input-file stats-dada2.qza \

--o-visualization stats-dada2.qzv

**9. Visualization of DADA2 results**

>> qiime feature-table summarize \

--i-table table.qza \

--m-sample-metadata-file weis\_metadata\_1210.tsv \

--o-visualization table.qzv

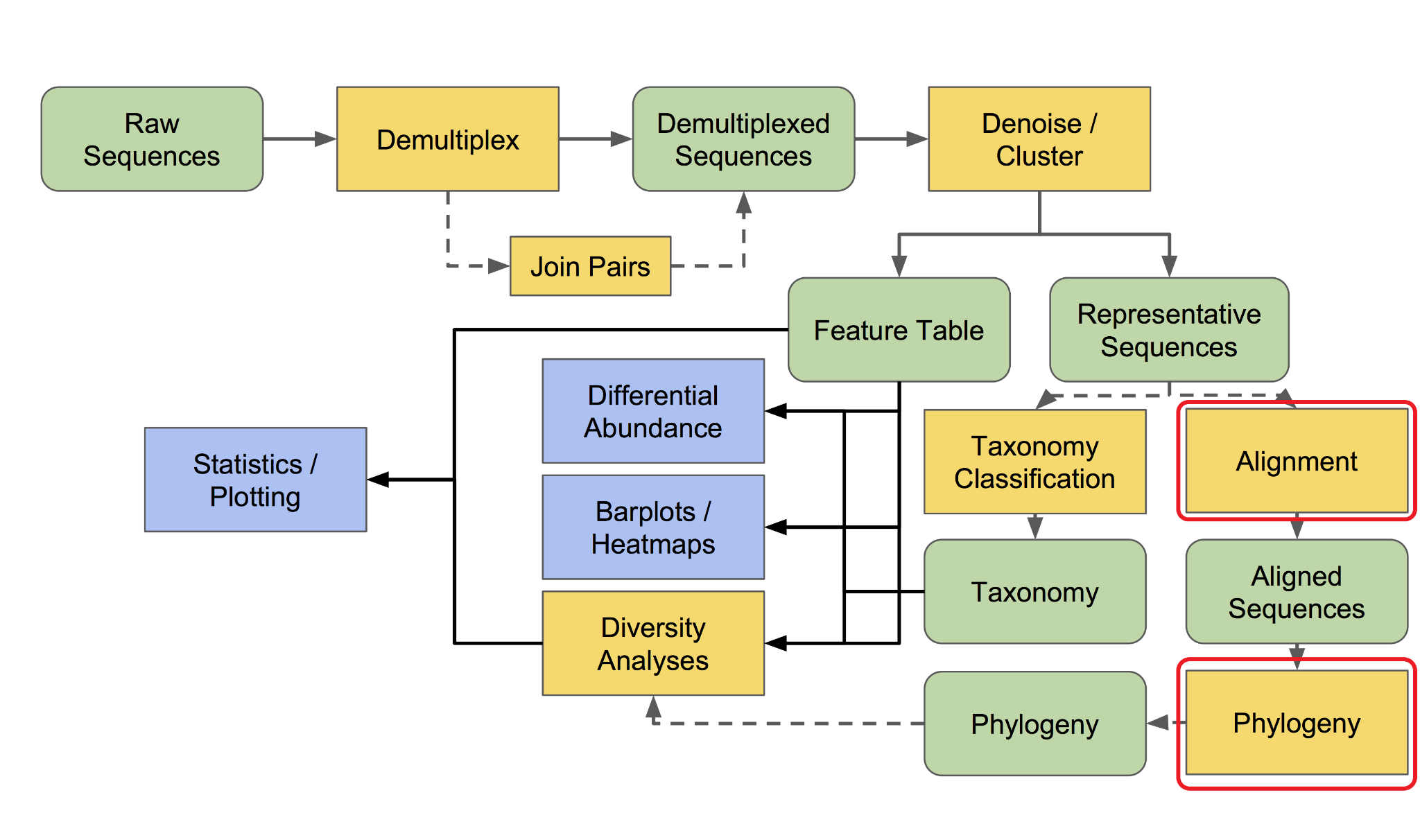
**10. Sequence visualization**

>> qiime feature-table tabulate-seqs \

--i-data rep-seqs.qza \

--o-visualization rep-seqs.qzv

**11. Generating a Phylogenetic Tree (Rooted/Rootless)**

****

>> qiime phylogeny align-to-tree-mafft-fasttree \

--i-sequences rep-seqs.qza \

--o-alignment aligned-rep-seqs.qza \

--o-masked-alignment masked-aligned-rep-seqs.qza \

--o-tree unrooted-tree.qza \

--o-rooted-tree rooted-tree.qza

**12. normalization**

The following (rarefy / relative-frequency) two methods are selected, due to relative-frequency, some methods are limited, we take rarefy as an example

rarefaction To observe table.qzv determines the sampling depth

>> qiime feature-table rarefy \

--i-table table.qza \

--p-sampling-depth 120108 \

--p-with-replacement \

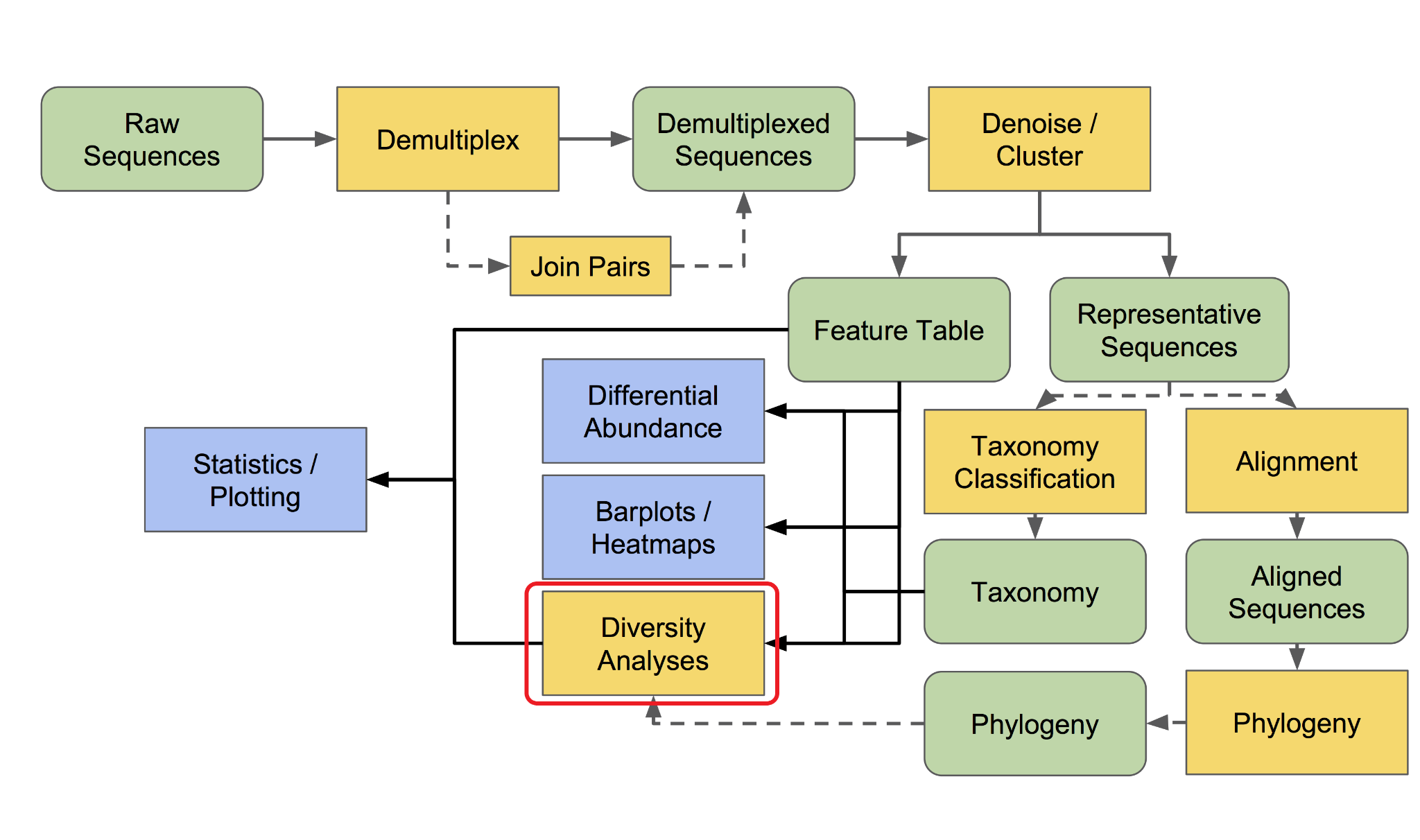
--o-rarefied-table rarefied\_table.qza

>> qiime feature-table relative-frequency \

--i-table table.qza \

--o-relative-frequency-table rel-feature-table.qza

**13. alpha diversity (All available metric references** [**Alpha and Beta Diversity Explanations and Commands**](https://forum.qiime2.org/t/alpha-and-beta-diversity-explanations-and-commands/2282)**)**

****

>> mkdir metrics-results

>> for METRIC in observed\_features chao1 shannon simpson

do

qiime diversity alpha \

--i-table rarefied\_table.qza \

--p-metric ${METRIC} \

--o-alpha-diversity metrics-results/${METRIC}\_vector.qza

done

**14. alpha diversity visualization**

>> mkdir alpha-diversity-visualization

>> for METRIC in observed\_features chao1 shannon simpson

do

qiime diversity alpha-group-significance \

--i-alpha-diversity metrics-results/${METRIC}\_vector.qza \

--m-metadata-file weis\_metadata\_1210.tsv \

--o-visualization alpha-diversity-visualization/${METRIC}-group-significance.qzv

done

**15. beta diversity distance matrix (phylogenetic)**

>> for METRIC in unweighted\_unifrac weighted\_unifrac

do

qiime diversity beta-phylogenetic \

--i-table rarefied\_table.qza \

--i-phylogeny rooted-tree.qza \

--p-metric ${METRIC} \

--o-distance-matrix metrics-results/${METRIC}\_distance\_matrix.qza

done

**16. beta diversity distance matrix (non-phylogenetic)**

>> for METRIC in canberra

do

qiime diversity beta \

--i-table rarefied\_table.qza \

--p-metric ${METRIC} \

--o-distance-matrix metrics-results/${METRIC}\_distance\_matrix.qza

done

**17. beta diversity statistical verification**

>> mkdir beta-diversity-visualization

>> for METRIC in unweighted\_unifrac weighted\_unifrac canberra

do

qiime diversity beta-group-significance \

--i-distance-matrix metrics-results/${METRIC}\_distance\_matrix.qza \

--m-metadata-file weis\_metadata\_1210.tsv \

--m-metadata-column Entacapone \

--p-method permanova \

--p-pairwise \

--o-visualization beta-diversity-visualization/${METRIC}-disease-status-significance.qzv

done

**18. beta diversity PCoA matrix**

>> for METRIC in unweighted\_unifrac weighted\_unifrac canberra

do

qiime diversity pcoa \

--i-distance-matrix metrics-results/${METRIC}\_distance\_matrix.qza \

--o-pcoa metrics-results/${METRIC}\_pcoa\_matrix.qza

done

**19. beta diversity PCoA visualization**

>> for METRIC in unweighted\_unifrac weighted\_unifrac canberra

do

qiime emperor plot \

--i-pcoa metrics-results/${METRIC}\_pcoa\_matrix.qza \

--m-metadata-file weis\_metadata\_1210.tsv \

--o-visualization beta-diversity-visualization/${METRIC}\_emperor.qzv

done

**20. Observe table.qzv to determine the maximum depth of the alpha sparse curve (I do it around the median of frequency)**

>> qiime diversity alpha-rarefaction \

--i-table table.qza \

--i-phylogeny rooted-tree.qza \

--p-min-depth 1 \

--p-max-depth 440000 \

--p-steps 40 \

--p-metrics observed\_features chao1 shannon simpson \

--m-metadata-file weis\_metadata\_1210.tsv \

--o-visualization alpha-rarefaction.qzv

**21. Training Species Classifier V4-V5 (520F, 926R Primers) (If there is a ready-made one on the network and the same Primers, you can catch it directly, you don't need to train it yourself, because it takes a long time, and not every computer has that much resources).**

**Related Teaching:**

<https://docs.qiime2.org/2022.2/tutorials/feature-classifier/>

<https://docs.qiime2.org/2022.2/data-resources/>

<https://forum.qiime2.org/t/processing-filtering-and-evaluating-the-silva-database-and-other-reference-sequence-data-with-rescript/15494>

<https://forum.qiime2.org/t/pre-trained-silva-classifier-v3-v4-qiime-2021-4/20470>

>> wget 'https://data.qiime2.org/2022.2/common/silva-138-99-seqs.qza'

>> wget 'https://data.qiime2.org/2022.2/common/silva-138-99-tax.qza'

>> qiime feature-classifier extract-reads \

--i-sequences silva-138-99-seqs.qza \

--p-f-primer AYTGGGYDTAAAGNG \

--p-r-primer CCGTCAATTCMTTTRAGTTT \

--o-reads ref-seqs-520f-926r.qza

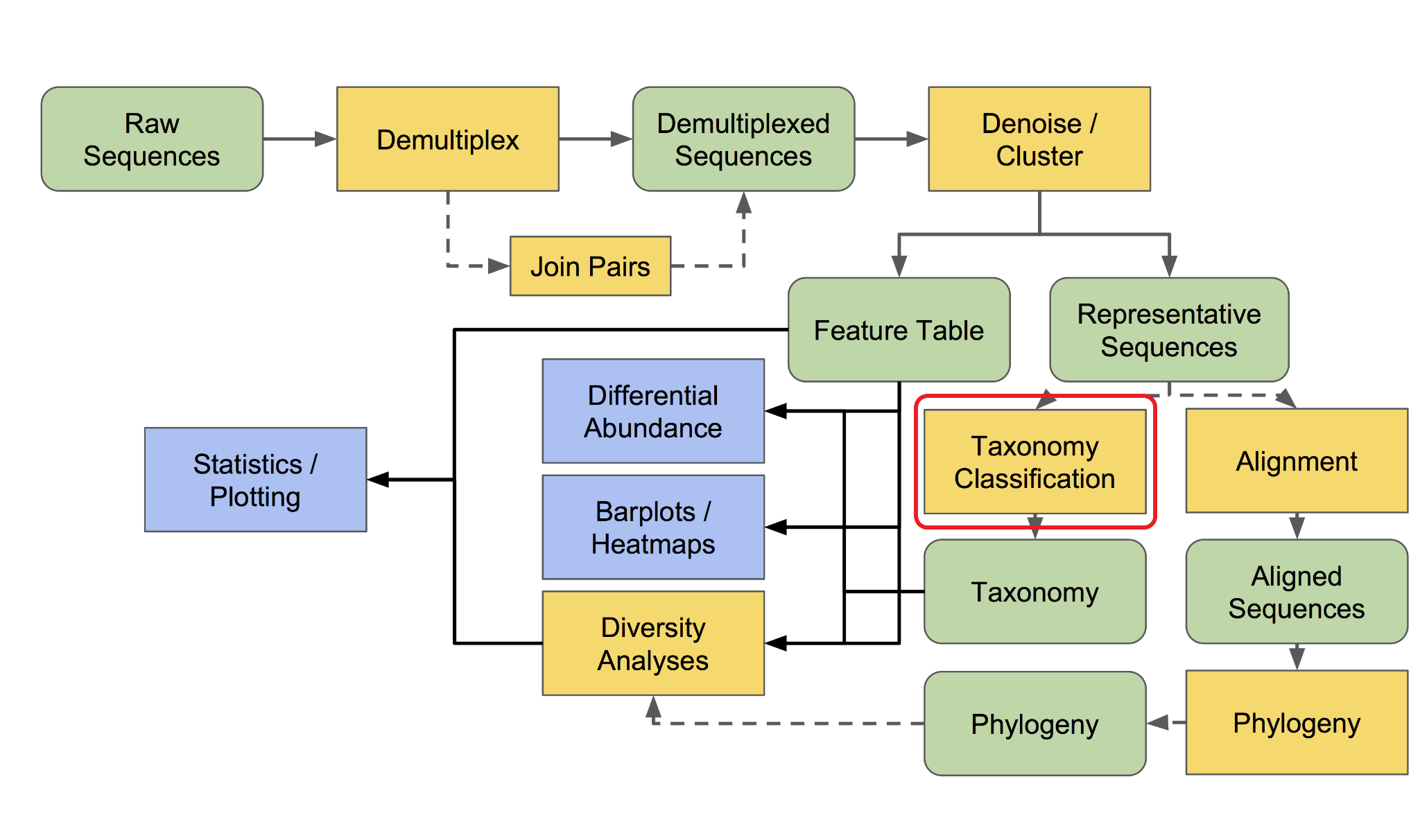
>> qiime feature-classifier fit-classifier-naive-bayes \

--i-reference-reads ref-seqs-520f-926r.qza \

--i-reference-taxonomy silva-138-99-tax.qza \

--o-classifier silva-138-99-520f-926r-classifier.qza

**22. taxonomic analysis**

****

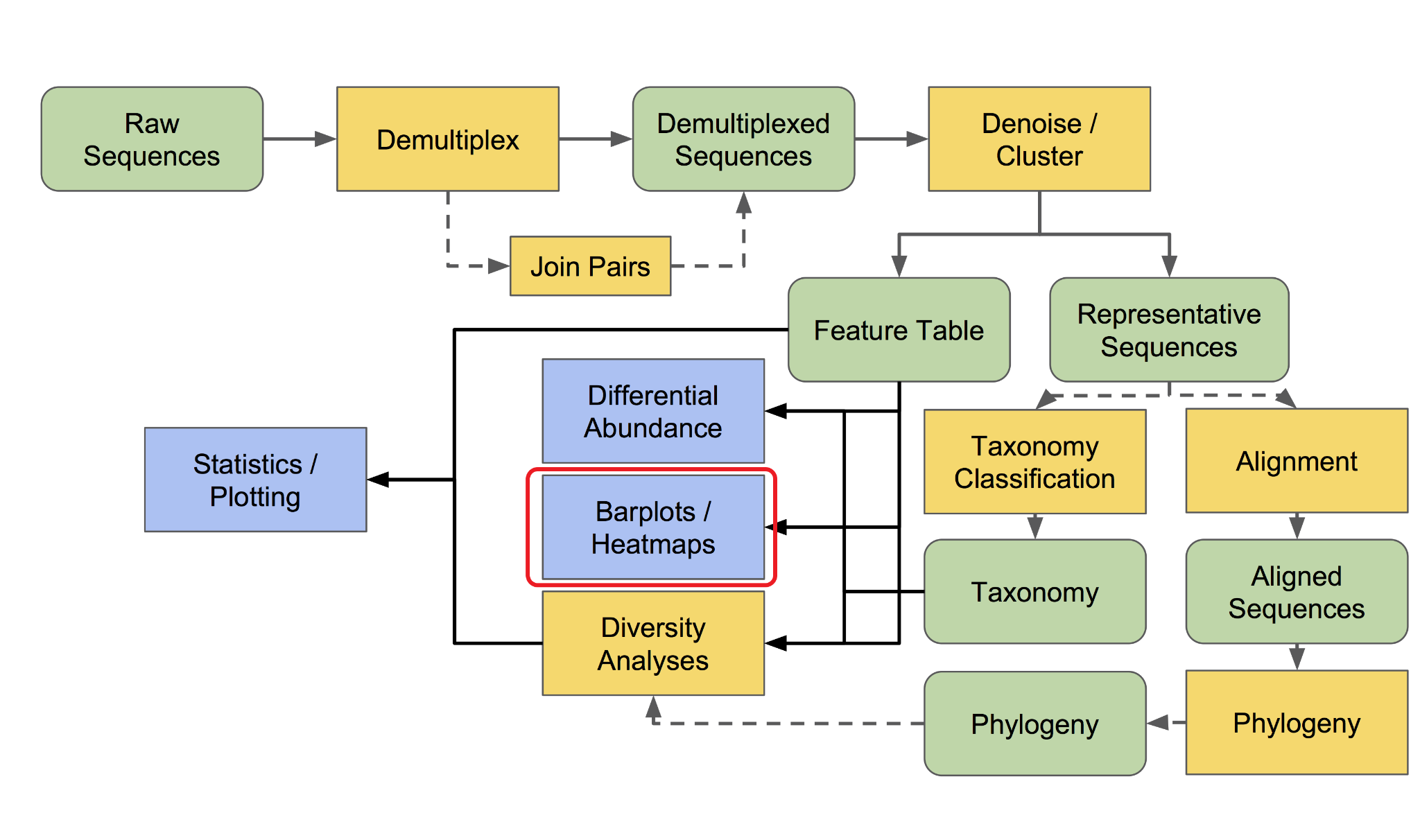
>> qiime feature-classifier classify-sklearn \

--i-classifier silva-138-99-520f-926r-classifier.qza \

--i-reads rep-seqs.qza \

--o-classification taxonomy.qza

**23. taxonomy barplot visualization**

****

>> qiime taxa barplot \

--i-table table.qza \

--i-taxonomy taxonomy.qza \

--m-metadata-file weis\_metadata\_1210.tsv \

--o-visualization taxa-bar-plots.qzv

**24. make taxonomy frequency form**

>> mkdir phyloseq

>> taxon=( domain-kingdom phylum class order family genus species )

for i in {1..7}

do

qiime taxa collapse \

--i-table table.qza \

--i-taxonomy taxonomy.qza \

--p-level ${i} \

--o-collapsed-table phyloseq/${taxon[${i}-1]}-table.qza

done

**25. make taxonomy relative frequency form**

>> for taxon in domain-kingdom phylum class order family genus species

do

qiime feature-table relative-frequency \

--i-table phyloseq/${taxon}-table.qza \

--o-relative-frequency-table phyloseq/rel-${taxon}-table.qza

done

**26. taxonomy frequency Heatmap visualization**

>> for taxon in order family genus

do

qiime feature-table heatmap \

--i-table phyloseq/${taxon}-table.qza \

--o-visualization heatmap\_${taxon}.qzv \

--p-color-scheme YlOrRd \

--p-cluster both

done

**27. export feature corresponding taxonomy of the table**

>> qiime tools export \

--input-path taxonomy.qza \

--output-path phyloseq/taxonomy

>> mv phyloseq/taxonomy/taxonomy.tsv phyloseq/taxonomy.tsv

>> rm -r phyloseq/taxonomy

**28. export feature frequency form**

>> qiime tools export \

--input-path table.qza \

--output-path phyloseq/feature-table

>> biom convert -i phyloseq/feature-table/feature-table.biom -o phyloseq/feature-table.tsv --to-tsv

>> cp phyloseq/feature-table/feature-table.biom phyloseq/feature-table.biom

>> rm -r phyloseq/feature-table

**29. export feature relative frequency form**

>> qiime tools export \

--input-path rel-feature-table.qza \

--output-path phyloseq/rel-feature-table

>> biom convert -i phyloseq/rel-feature-table/feature-table.biom -o phyloseq/rel-feature-table.tsv --to-tsv

>> rm -r phyloseq/rel-feature-table

**30. export taxonomy frequency form**

>> for taxon in domain-kingdom phylum class order family genus species

do

qiime tools export \

--input-path phyloseq/${taxon}-table.qza \

--output-path phyloseq/${taxon}-table

biom convert -i phyloseq/${taxon}-table/feature-table.biom -o phyloseq/${taxon}-table.tsv --to-tsv

rm -r phyloseq/${taxon}-table

done

**31. export taxonomy relative frequency form**

>> for taxon in domain-kingdom phylum class order family genus species

do

qiime tools export \

--input-path phyloseq/rel-${taxon}-table.qza \

--output-path phyloseq/rel-${taxon}-table

biom convert -i phyloseq/rel-${taxon}-table/feature-table.biom -o phyloseq/rel-${taxon}-table.tsv --to-tsv

rm -r phyloseq/rel-${taxon}-table

done

**32. Export Order**

>> qiime tools export \

--input-path rep-seqs.qza \

--output-path phyloseq/rep-seqs

>> mv phyloseq/rep-seqs/dna-sequences.fasta phyloseq/dna-sequences.fna

>> rm -r phyloseq/rep-seqs

**33. Export rootless trees**

>> qiime tools export \

--input-path unrooted-tree.qza \

--output-path phyloseq

>> mv phyloseq/tree.nwk phyloseq/unrooted\_tree.nwk

**34. Export Rooted Trees**

>> qiime tools export \

--input-path rooted-tree.qza \

--output-path phyloseq

>> mv phyloseq/tree.nwk phyloseq/rooted\_tree.nwk

**35. Prepare the information into the phyloseq folder**

>> cp weis\_metadata\_1210.tsv phyloseq/weis\_metadata\_1210.tsv

>> cp rep-seqs.qza phyloseq/rep-seqs.qza

>> cp table.qza phyloseq/feature-table.qza

>> cp rel-feature-table.qza phyloseq/rel-feature-table.qza

>> cp unrooted-tree.qza phyloseq/unrooted-tree.qza

>> cp rooted-tree.qza phyloseq/rooted-tree.qza

>> cp taxonomy.qza phyloseq/taxonomy.qza

>> cp rarefied\_table.qza phyloseq/rarefied\_table.qza

# Place two folders /data2/shenglab/weis\_qiime2/phyloseq and /data2/shenglab/weis\_qiime2/metrics-results copy to your computer's Weis folder

END OF WEIS 2

**36. Remove container**

$> docker stop $(docker ps -a -q --filter="name=shenglab")

$> docker rm $(docker ps -a -q --filter="name=shenglab")

**5、PICRUSt2**

**1.**

$> wget https://github.com/picrust/picrust2/archive/v2.4.2.tar.gz (請在Downloads中下載 P.S. 注意工作目錄)

移至/data2/shenglab1/weis\_ PRJEB30615

$> docker run -it --name shenglab\_ubt -v /:/tmp ubuntu bash

>> cd /tmp

>> mv /tmp/home/shenglab/Downloads/v2.4.2.tar.gz /tmp/data2/shenglab/weis\_qiime2/v2.4.2.tar.gz

$> docker stop $(docker ps -a -q --filter="name=shenglab\_ubt")

$> docker rm $(docker ps -a -q --filter="name=shenglab\_ubt")

$> docker run -it --name shenglab\_pc2 -v /data2/shenglab/weis\_qiime2:/tmp continuumio/anaconda3 bash

>> tar xvzf v2.4.2.tar.gz

>> cd picrust2-2.4.2/

>> conda env create -f picrust2-env.yaml

>> conda activate picrust2

>> pip install --editable .

The above method is PICRUSt2 method 1 (**but you need to pay attention to the location of the map input files later).**

**1.**

conda activate qiime2-2023.2

**2.**

conda install q2-picrust2=2023.2 \

-c conda-forge \

-c bioconda \

-c gavinmdouglas

qiime picrust2 full-pipeline --help

The above method is PICRUSt2 method 2 (qiime2-2023.2 must be used**).**

**1. (Do this for each method)**

Go back to your computer and execute Crawling\_KO\_number\_LEVEL23.R contents, acquisition KO\_LEVEL23\_2022\_1214.tsv

Deliver to a supercomputer in the cloud

Move to /data2/shenglab/weis\_qiime2

$> docker run -it --name shenglab\_ubt -v /:/tmp ubuntu bash

>> cd /tmp

>> mv /tmp/home/shenglab/Downloads/KO\_LEVEL23\_2022\_1214.tsv /tmp/data2/shenglab/weis\_qiime2/KO\_LEVEL23\_2022\_1214.tsv

$> docker stop $(docker ps -a -q --filter="name=shenglab\_ubt")

$> docker rm $(docker ps -a -q --filter="name=shenglab\_ubt")

**2. Install anaconda3 (only the first time you need to run).**

$> docker pull continuumio/anaconda3

**3. Establish PICRUSt2**

$> docker run -it --name shenglab\_pc2 -v /data2/shenglab/weis\_qiime2:/tmp continuumio/anaconda3 bash

>> conda create -n picrust2 -c bioconda -c conda-forge picrust2=2.4.1 # This line can take a lot of time

**4. Start PICRUSt2 (if there is a pop-out container, just start this step when you come back)**

>> conda activate picrust2

>> cd /tmp

Points 2~4 are method 3 for creating PICRUSt2 (**not recommended).**

START OF WEIS3

**5. Import data**

>> picrust2\_pipeline.py \

-s phyloseq/dna-sequences.fna \

-i phyloseq/feature-table.biom \

-o picrust2\_output\_K

**6.**

>> cd ./picrust2\_output\_K

>> hsp.py \

-i 16S \

-t out.tre -o marker\_predicted\_and\_nsti.tsv.gz \

-p 1 \

-n

**7.**

>> hsp.py \

-i KO -t out.tre \

-o KO\_predicted.tsv.gz \

-p 1

**8.**

>> metagenome\_pipeline.py \

-i ../phyloseq/feature-table.biom \

-m marker\_predicted\_and\_nsti.tsv.gz \

-f KO\_predicted.tsv.gz \

-o KO\_metagenome\_out \

--strat\_out

**9.**

>> pathway\_pipeline.py \

-i KO\_metagenome\_out/pred\_metagenome\_contrib.tsv.gz \

-o KEGG-Pathways \

--no\_regroup \

--map /tmp/picrust2-2.4.2/picrust2/default\_files/pathway\_mapfiles/KEGG\_pathways\_to\_KO.tsv && \

**10.**

>> add\_descriptions.py \

-i KEGG-Pathways/path\_abun\_unstrat.tsv.gz \

--custom\_map\_table /tmp/picrust2-2.4.2/picrust2/default\_files/description\_mapfiles/KEGG\_pathways\_info.tsv.gz \

-o KEGG-Pathways/path\_abun\_unstrat\_descrip.tsv.gz && \

**11. Add a description of KEGG's data**

>> add\_descriptions.py \

-i KEGG-Pathways/path\_abun\_unstrat.tsv.gz \

--custom\_map\_table /tmp/KO\_LEVEL23\_2022\_1214.tsv \

-o KEGG-Pathways/L2\_3\_path\_abun\_unstrat\_descrip.tsv.gz

**12. Unzip the final result**

>> gzip KEGG-Pathways/L2\_3\_path\_abun\_unstrat\_descrip.tsv.gz --decompress --stdout > KEGG-Pathways/L2\_3\_path\_abun\_unstrat\_descrip.tsv

# data2/shenglab/weis\_qiime2/picrust2\_output\_K/KEGG-Pathways/L2\_3\_path\_abun\_unstrat\_descrip.tsv Copy to your computer's Weis folder

END OF WEIS3

**13. Remove container (Make sure you don't need it anymore before removing it, otherwise you have to run it again in step 3)**

$> docker stop $(docker ps -a -q --filter="name=shenglab\_pc2")

$> docker rm $(docker ps -a -q --filter="name=shenglab\_pc2")

**6. R Drawing & Extraneous Analysis**

**1. Table 1 Narrative statistics**

execute table1.R

**2. alpha and beta diversity analysis The drawing results are reproduced on R, adding a self-style drawing style**

execute qiime2\_alpha\_plot.R and qiime2\_beta\_plot.R

**3. You can also use the DADA2 file to build a phyloseq format on R to do your own analysis, and here you will demonstrate diversity analysis (not necessary).**

# It should be noted that different software has different methods of normalization when doing diversity analysis, so the results will be somewhat different

execute qiime2\_postanalysis.R

**4. Group verification**

execute Wilcoxon\_and\_volcano\_plot.R

**7、STAMP**

**1. Change the pathway data to a STAMP compliant format**

execute before\_stamp.R acquisition pathway\_stamp.tsv

**2. Install STAMP**

STAMP\_2\_1\_3.exe

**3. STAMPanalysis**

File > load data > pathway\_stamp.tsv and phyloseq/weis\_metadata\_1210.tsv Throw it in

Select Variables in the upper right corner Entacapone

Select on the left Two groups

Statistics test: White’s nonparametric t-test

Multiple test correction: Storey FDR

q-value filter (Ticked together will only show significant variables)

Select below Extended error bar