灰底文字 : 自訂的名稱 或是 隨使用者變動的部分

黃底文字 : 下載的檔案, 專案專屬的檔案 或是 我給的檔案

一般無底色文字 : 固定格式，應該不需要修改

紅色文字 : 醒目提示，非程式碼

>> : 在 docker container 內輸入的指令

$> : 在實體環境輸入的指令

PS 上次介紹的連結：[超級電腦](https://docs.google.com/document/d/1AUo0gO0j7083l-ulGyvCzKbmnkIi2CKtaWJ0DsoKQLI/edit?usp=sharing)

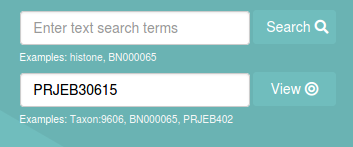
起始檔案：[Weis起始.zip](https://drive.google.com/file/d/1HqkD-o43cu1nX4RW7eyRJjcVEHB-vwml/view?usp=share_link)

**一、下載序列**

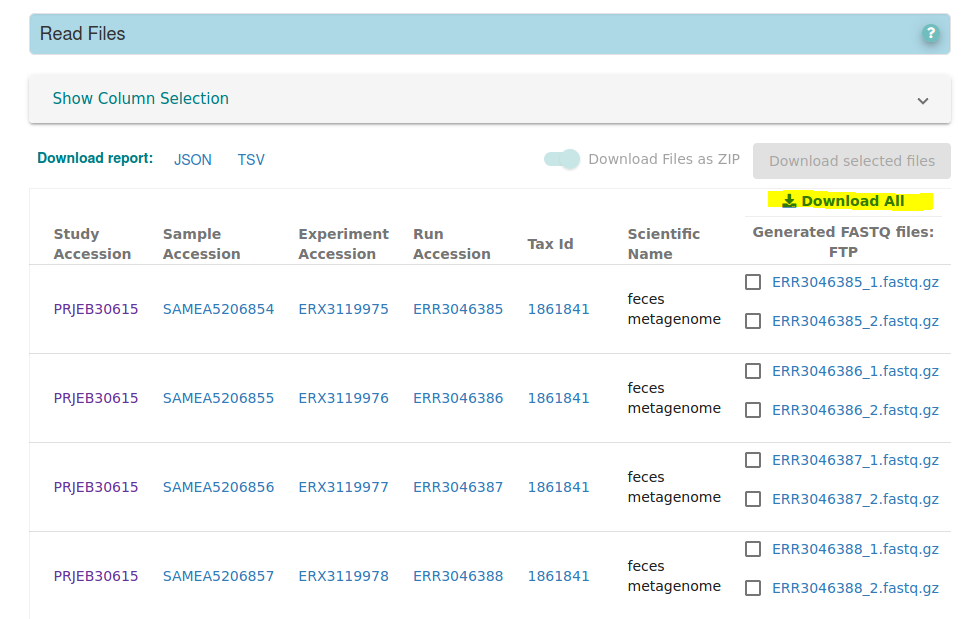
**1. 進入ENA網站**

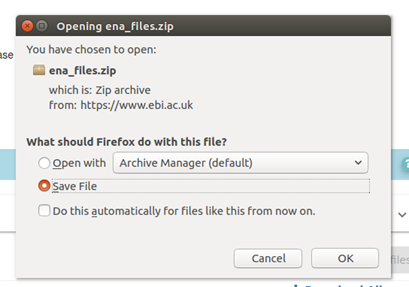
<https://www.ebi.ac.uk/ena/browser/home>

**2. 右上角輸入專案號碼 PRJEB30615**



**3. 下載fastq檔 (預設存到 /home/shenglab/Downloads)**



(現為sh file)

**4. 在/data2/shenglab建立專案資料夾 => weis\_qiime2 (所有分析都會在data2執行)**

$> docker run -it --name shenglab -v /:/tmp ubuntu bash # (冒號左邊的 '/' 是根目錄，代表整台電腦都進入虛擬環境，操作要特別小心)

>> cd /tmp # (現在虛擬環境的/tmp對應的是實體環境的根目錄，時時都要注意自己現在的工作目錄在哪)

>> mkdir /tmp/data2/shenglab1/weis\_PRJEB30615

在/data2/shenglab/weis\_qiime2 建立基因資料夾 fastq\_files

>> mkdir /tmp/data2/shenglab1/weis\_PRJEB30615/fastq\_files

**5. 將資料下載完後，移到/data2/shenglab/weis\_qiime2/fastq\_files**

$> mkdir ena\_files (請在Downloads中建立，並把sh file放入 P.S. 注意工作目錄)

$>

$> chmod a+x ena.sh

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

>> mv /tmp/home/shenglab1/Downloads/ena\_files /tmp/data2/shenglab1/weis\_PRJEB30615/fastq\_files/ena\_files

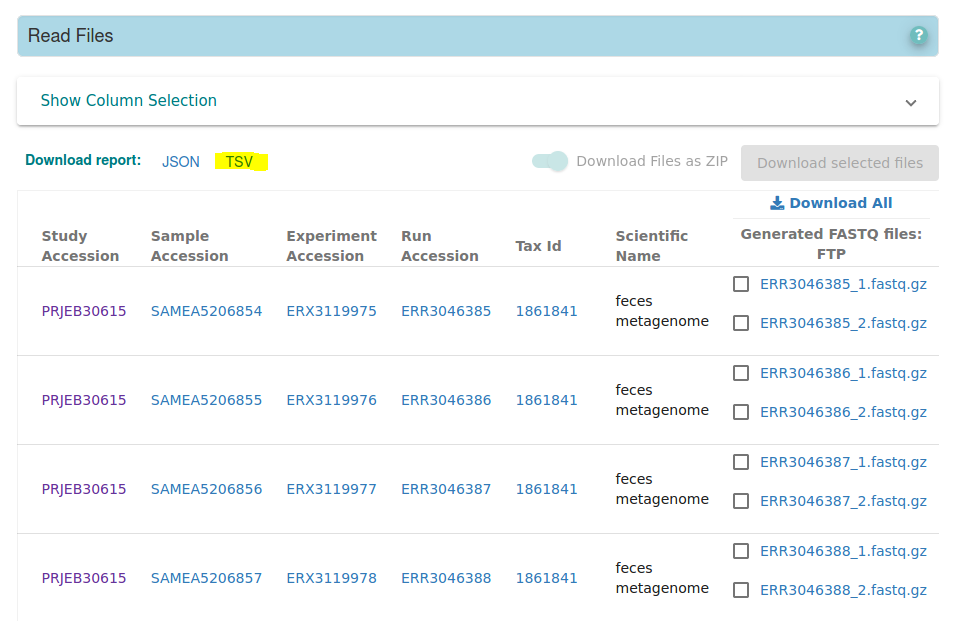
**二、製作metadata (在自己電腦)**

**1.**

建立資料夾 (名為: Weis)

**2.**

一樣至ENA網站，下載report到Weis (檔名: filereport\_read\_run\_PRJEB30615\_tsv.txt)



**3.**

添加以下檔案到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

開啟Rstudio (需先安裝R, Rstudio)，並建立專案

File => New Project => Existing Directory => 連結Weis資料夾 (之後都從Weis.Rproj檔進入專案)

**4.**

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

**5.**

上傳(1) weis\_metadata\_1210.csv , (2) id\_list.csv至雲端硬碟，並下載到超級電腦 (預設存到 /home/shenglab/Downloads)

**6. 移至/data2/shenglab/weis\_qiime2**

>> mv /tmp/home/shenglab1/Downloads/weis\_metadata\_1210.csv /tmp/data2/shenglab1/weis\_PRJEB30615/weis\_metadata\_1210.csv

>> mv /tmp/home/shenglab1/Downloads/id\_list.csv /tmp/data2/shenglab1/weis\_PRJEB30615/id\_list.csv

**7. 移除 container**

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

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

**三、準備資料**

**1.**

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

>> cd /tmp

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

**2. 建立所需資料夾，存放每個樣本4次的實驗結果**

>> 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. 解壓縮到對應的資料夾**

>> 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. 將四次實驗結果合併成一個.fastq檔**

>> 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. 壓縮成.fastq.gz檔 (會跑久一點)**

>> 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. 刪掉已經不需要的資料**

>> 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. 製作 manifest (qiime2有指定格式，詳閱**[**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

**8. 移除 container**

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

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

**四、QIIME2**

**1. 安裝qiime2 (只有第一次需要跑)**

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

**2.**

>> docker run -it --name shenglab -v /data2/shenglab/weis\_qiime2:/tmp quay.io/qiime2/core:2022.2 /bin/bash

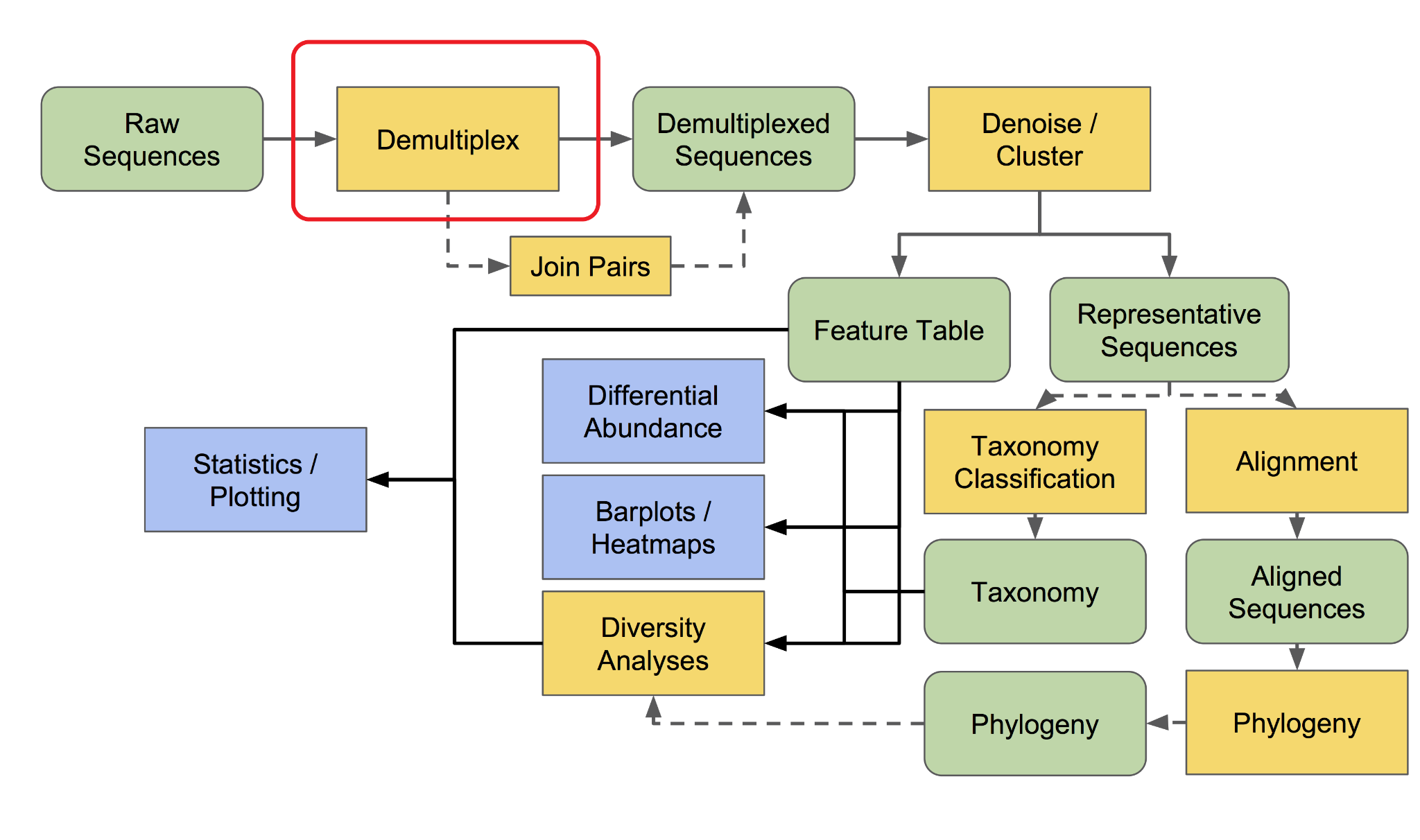
>> cd /tmp

**3. metadata 轉檔 csv to tsv**

>> pip install csvkit #(可能會有warning但不須擔心)

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

**4. 匯入資料成QIIME2通用格式 .qza**

****

>> qiime tools import \

--type 'SampleData[PairedEndSequencesWithQuality]' \

--input-path manifest.tsv \

--input-format PairedEndFastqManifestPhred33V2 \

--output-path demultiplexed-seqs.qza

**5. 移除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 視覺化檔案格式 .qzv (將檔案放到** [**QIIME 2 View**](https://view.qiime2.org/) **顯示視覺化結果)**

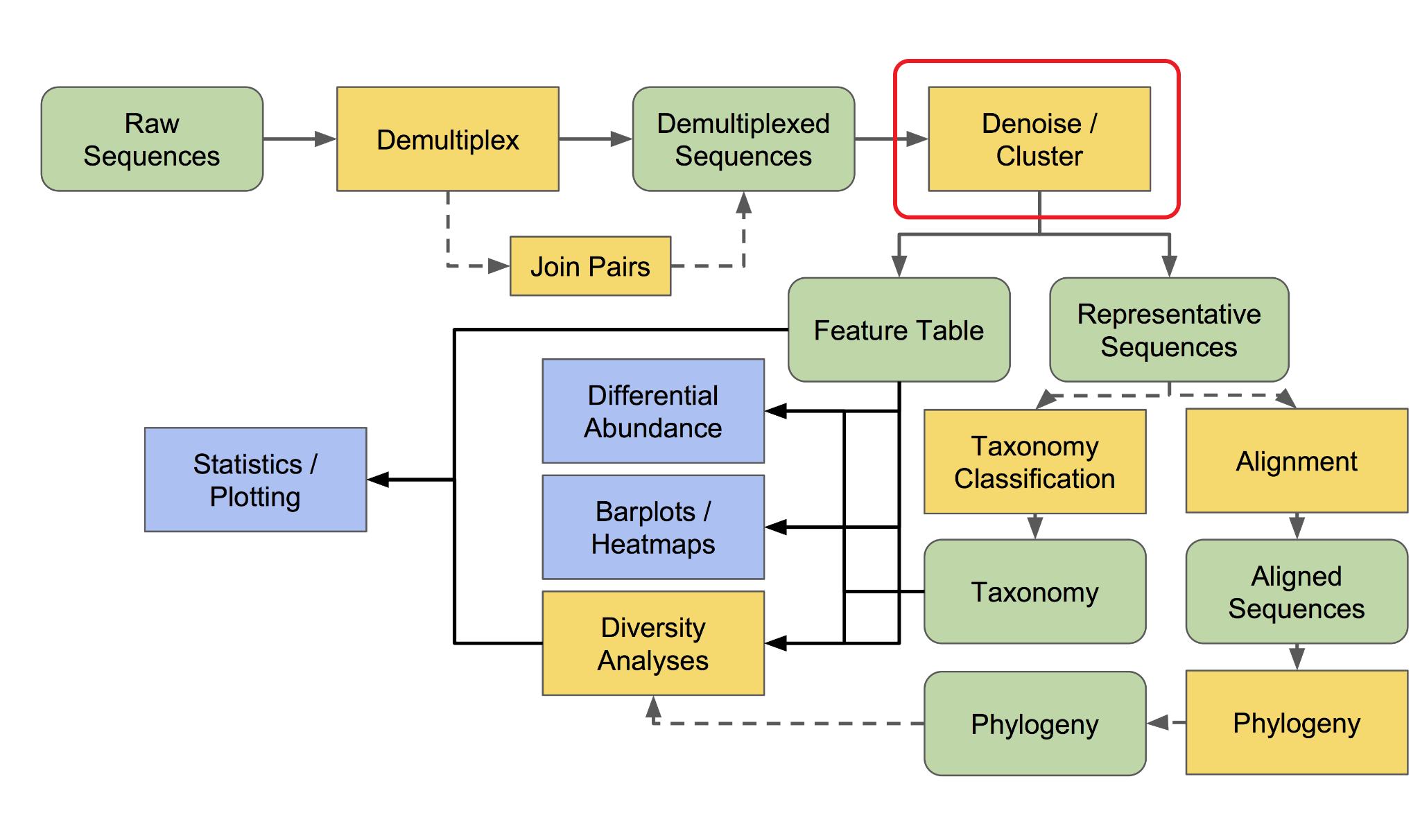
>> qiime demux summarize \

--i-data trimmed-seqs.qza \

--o-visualization trimmed-seqs.qzv

**7. dada2 (會有點久要幾個小時; 觀察trimmed-seqs.qzv移除品質不佳的部分)**

[**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代表使用所有運算核心，注意電腦記憶體容量，可開Htop來看

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

--o-table table.qza \

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

**8. dada2過程資訊視覺化**

>> qiime metadata tabulate \

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

--o-visualization stats-dada2.qzv

**9. dada2結果視覺化**

>> qiime feature-table summarize \

--i-table table.qza \

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

--o-visualization table.qzv

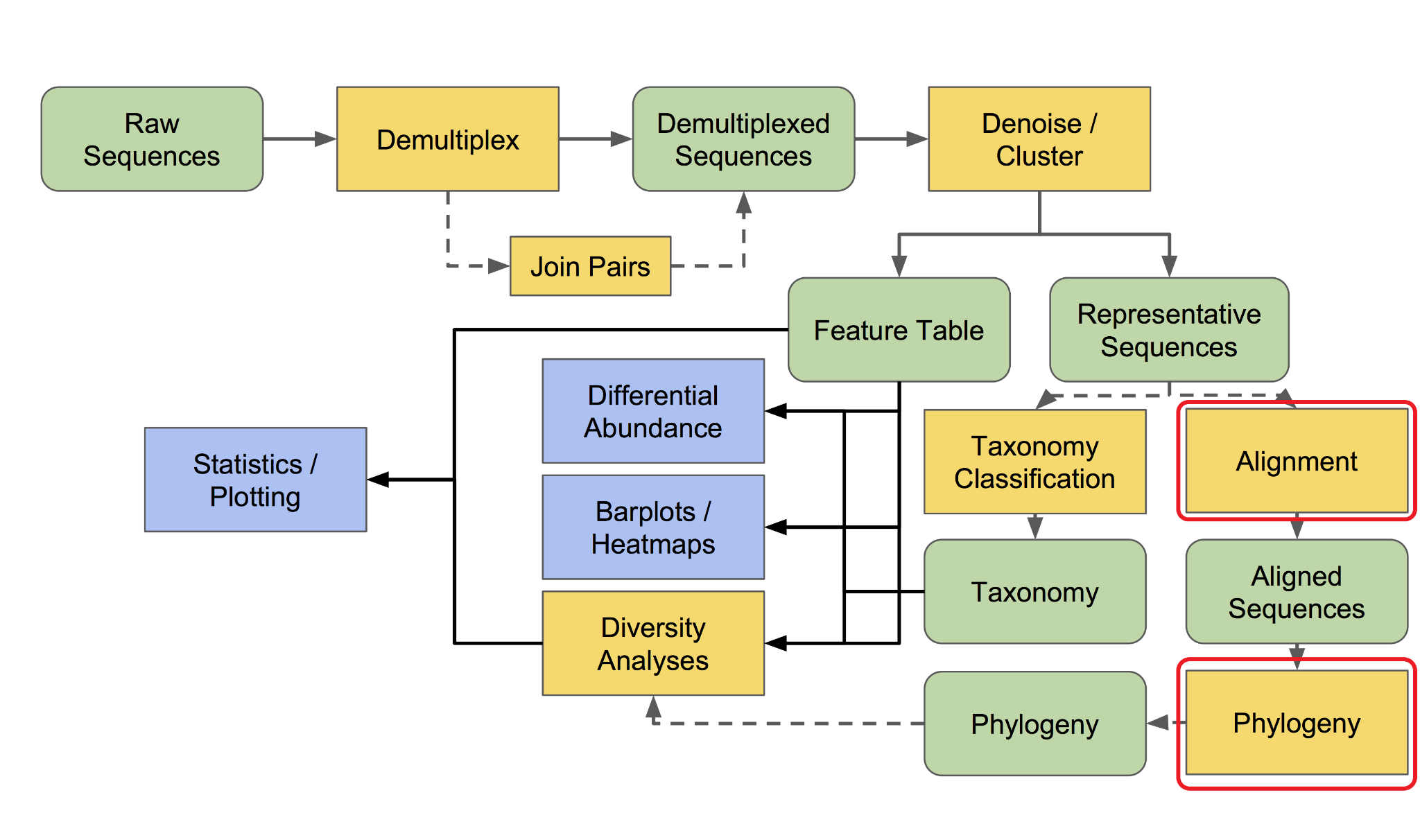
**10. 序列視覺化**

>> qiime feature-table tabulate-seqs \

--i-data rep-seqs.qza \

--o-visualization rep-seqs.qzv

**11. 生成系統演化樹 (分成有根/無根兩種方法)**

****

>> 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**

以下 (rarefy / relative-frequency) 兩個方法擇一，基於relative-frequency在某些方法受限，我們這裡以rarefy為例

rarefaction 要觀察 table.qzv 決定抽樣深度

>> qiime feature-table rarefy \

--i-table table.qza \

--p-sampling-depth 120108 \

--p-with-replacement \

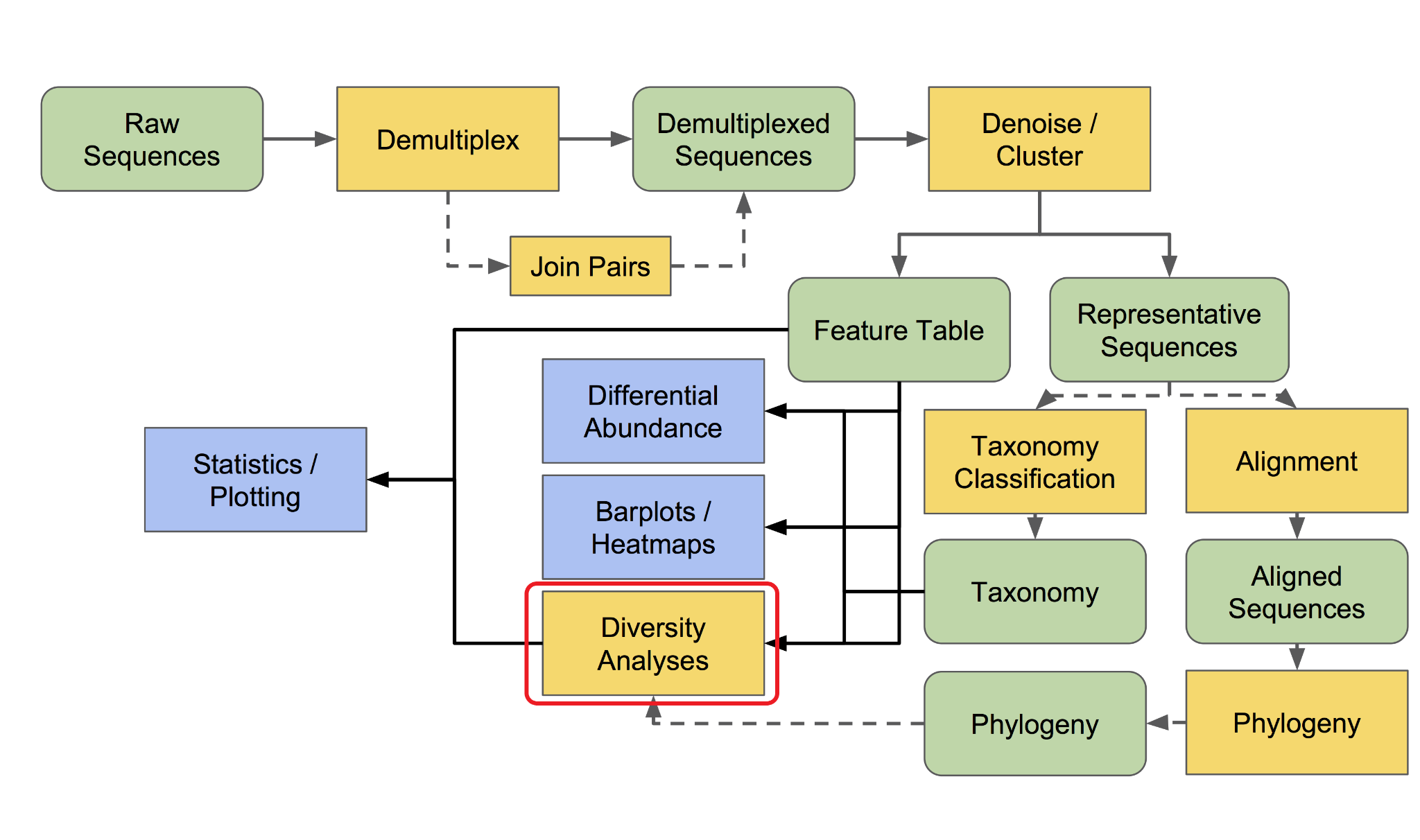
--o-rarefied-table rarefied\_table.qza

>> qiime feature-table relative-frequency \

--i-table table.qza \

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

**13. alpha diversity (所有可用metric參考[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 視覺化**

>> 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 距離矩陣 (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 距離矩陣 (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 統計檢定**

>> 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 矩陣**

>> 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 視覺化**

>> 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. 觀察 table.qzv 決定alpha稀疏曲線最大深度(我做到frequency的中位數左右)**

>> 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. 訓練物種分類器V4-V5 (520F, 926R Primers) (如果網路上有現成而且Primers相同的就直接抓來用，不需要自己訓練，因為要等很久，也不是每台電腦都有那麼大的資源)**

**相關教學:**

<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

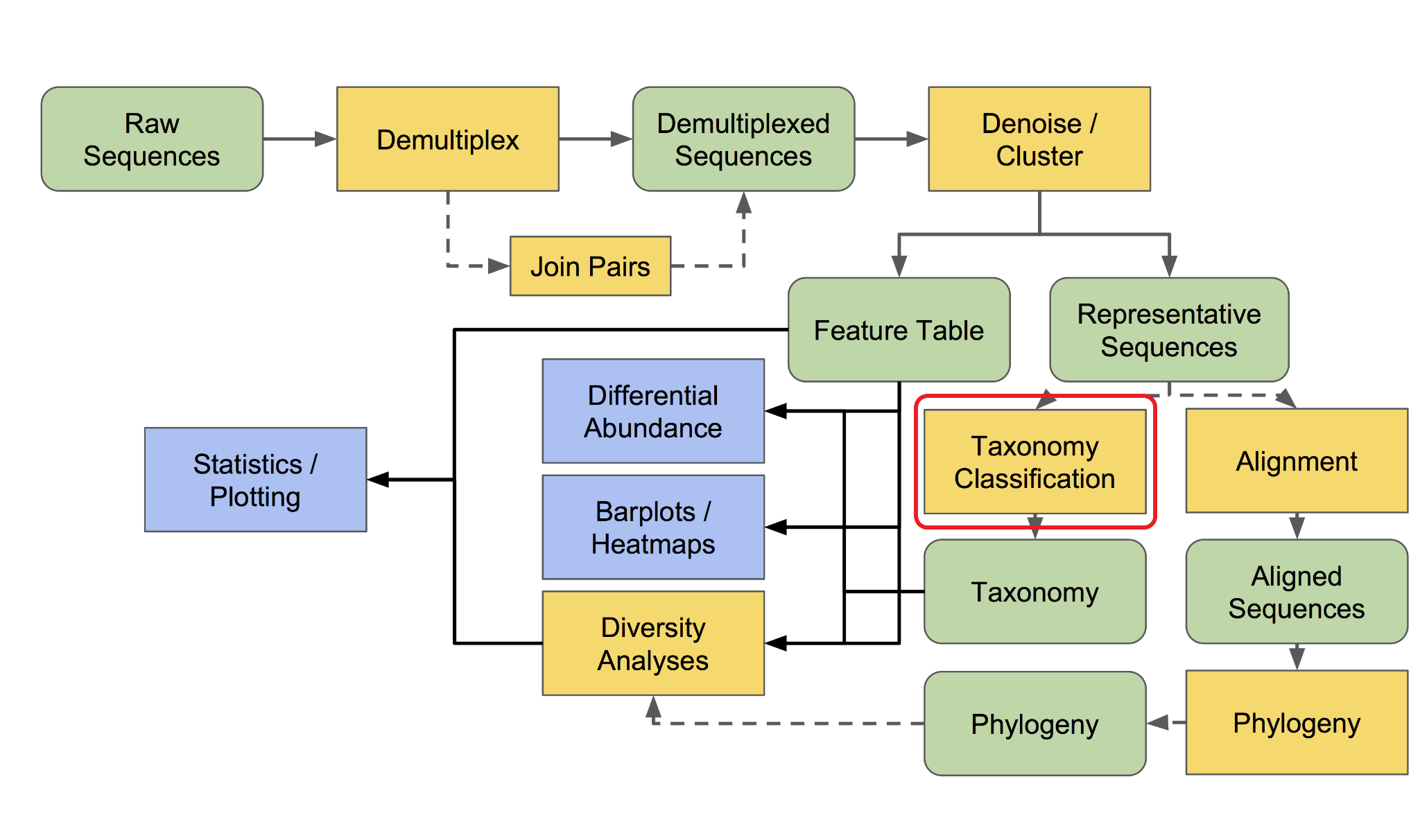
>> qiime feature-classifier fit-classifier-naive-bayes \

--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**

****

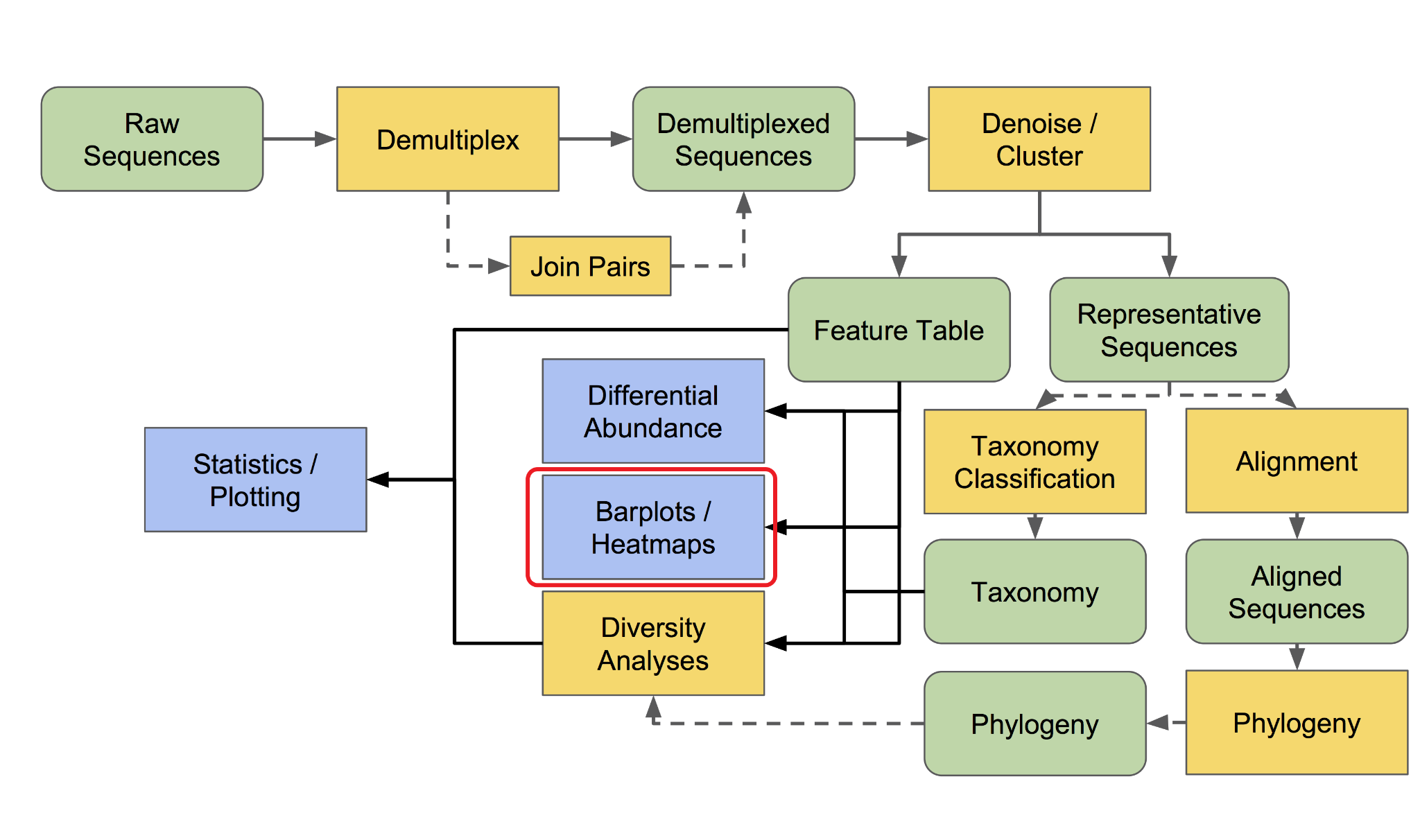
>> qiime feature-classifier classify-sklearn \

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

--i-reads rep-seqs.qza \

--o-classification taxonomy.qza

**23. taxonomy barplot 視覺化**

****

>> 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. 製作taxonomy frequency表格**

>> 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. 製作taxonomy relative frequency表格**

>> 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 熱圖視覺化**

>> 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. 輸出feature對應taxonomy的表格**

>> qiime tools export \

--input-path taxonomy.qza \

--output-path phyloseq/taxonomy

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

>> rm -r phyloseq/taxonomy

**28. 輸出feature frequency表格**

>> 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. 輸出feature relative frequency表格**

>> 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. 輸出taxonomy frequency表格**

>> 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. 輸出taxonomy relative frequency表格**

>> 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. 輸出序列**

>> 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. 輸出無根樹**

>> qiime tools export \

--input-path unrooted-tree.qza \

--output-path phyloseq

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

**34. 輸出有根樹**

>> qiime tools export \

--input-path rooted-tree.qza \

--output-path phyloseq

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

**35. 備齊資料進phyloseq資料夾**

>> 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

# 將兩個資料夾 /data2/shenglab/weis\_qiime2/phyloseq 以及 /data2/shenglab/weis\_qiime2/metrics-results 複製到自己電腦的Weis資料夾

**36. 移除 container**

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

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

**五、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 .

上述方法為**PICRUSt2方法1(但後須操作需要注意map input files 的位置)**

**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

上述方法為**PICRUSt2方法2(須使用**qiime2-2023.2**)**

**1. (每種方法都要執行這個)**

回自己電腦執行Crawling\_KO\_number\_LEVEL23.R內容，取得KO\_LEVEL23\_2022\_1214.tsv

用雲端傳到超級電腦

移至/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. 安裝anaconda3 (只有第一次需要跑)**

$> docker pull continuumio/anaconda3

**3. 建立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 # 這行可能花上非常多時間

**4. 啟動PICRUSt2 (如果有跳出container，回來後只要從這步開始)**

>> conda activate picrust2

>> cd /tmp

2~4點為創建**PICRUSt2方法3(不推薦)**

**5. 匯入資料**

>> 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 //opt/conda/envs/picrust2/lib/python3.6/site-packages/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 //opt/conda/envs/picrust2/lib/python3.6/site-packages/picrust2/default\_files/description\_mapfiles/KEGG\_pathways\_info.tsv.gz \

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

**11. 添加 KEGG 的資料描述**

>> 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. 解壓縮最後結果**

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

# 將 /data2/shenglab/weis\_qiime2/picrust2\_output\_K/KEGG-Pathways/L2\_3\_path\_abun\_unstrat\_descrip.tsv 複製到自己電腦的Weis資料夾

**13. 移除 container (確定不再用到才移除，不然第3步又要重跑)**

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

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

**六、R繪圖&額外的分析**

**1. Table 1 敘述統計**

執行 table1.R

**2. 將 alpha 與 beta diversity analysis 的繪圖結果在R上重現，增加自我風格的繪圖樣式**

執行 qiime2\_alpha\_plot.R 與 qiime2\_beta\_plot.R

**3. 也可以自己利用做完DADA2的檔案在R上建構成phyloseq格式自己做分析，這裡示範diversity analysis (非必要)**

# 需要注意的是不同軟體在diversity analysis時，做normalization的方法不盡相同，所以結果會有些差異

執行 qiime2\_postanalysis.R

**4. 分組檢定**

執行 Wilcoxon\_and\_volcano\_plot.R

**七、STAMP**

**1. 將pathway資料改成符合STAMP要求的格式**

執行 before\_stamp.R 取得 pathway\_stamp.tsv

**2. 安裝STAMP**

STAMP\_2\_1\_3.exe

**3. STAMP分析**

File > load data > 將 pathway\_stamp.tsv 以及 phyloseq/weis\_metadata\_1210.tsv 丟入

右上角選擇變數 Entacapone

左邊選擇 Two groups

Statistics test: White’s nonparametric t-test

Multiple test correction: Storey FDR

q-value filter (勾起來就只會顯示顯著的變數)

下方選擇 Extended error bar