

1. 文件导入、导出、数据预处理

- fasta格式文件

```
1 ##导入fasta格式, QIIME 2目前支持导入QIIME 1 seqs.fna文件格式, 该格式由一个fasta文件组成, 每条i
2 #每个序列必须正好一行, 不能拆分多行。每条序列的ID必须遵循格式_的要求。
3 #是序列所属样本的标识符, 是其样本中序列的标识符。
4 qiime tools import --input-path sequences.fna --output-path sequences.qza --type 'Feature'
```

- 根据样品清单导入fastq格式

```
1 ##已拆分好的fastq双端原始数据导入, Phred=32
2 time qiime tools import \
3   --type "SampleData[SequencesWithQuality]" \
4   --input-format SingleEndFastqManifestPhred33V2 \
5   --input-path ./manifest.tsv \
6   --output-path ./demux_seqs.qza
7 ##导入已拆分好的Phred33格式单端测序结果
8 qiime tools import --type 'SampleData[SequencesWithQuality]' --input-path manifest.valid
9   --output-path single-end-demux.valid.qza --input-format SingleEndFastqManifestPhred33V2
10 ##导入已拆分好的Phred33格式双端合并后的序列
11 qiime tools import --type SampleData[JoinedSequencesWithQuality] --input-path fj-joined/
12   --output-path fj-joined-demux.qza --input-format SingleEndFastqManifestPhred33V2
```

- 双端序列合并Joining reads

```
1 qiime vsearch join-pairs --i-demultiplexed-seqs paired-end-demux.qza --o-joined-sequences
```

- 导入物种注释文件

```
1 qiime tools import --type 'FeatureData[Taxonomy]' \
2   --input-format HeaderlessTSVTaxonomyFormat \      #HeaderlessTSVTaxonomyFormat没有表头的Tsv
3   --input-path 3xia.genus.taxonomy --output-path 3xia.genus.ref.taxonomy.qza
```

- 导出文件

```
1 ##导出Phylogeny[Unrooted]对象为newick格式
2 qiime tools export \
3   --input-path unrooted-tree.qza \
```

```

4  --output-path exported-tree
5  ##导出物种注释文件, 导出结果为tsv文件, 其实就是将qza文件重命名为zip后解压
6  qiime tools export --input-path taxonomy.qza --output-path taxa
7  ##导出代表序列文件, 导出结果为dna-sequences.fasta
8  qiime tools export --input-path filtered_rep_seq.qza --output-path filtered_rep_seq

```

2. BIOM文件常用命令+Qiime

• 文件导入

```

1  #BIOM v1.0.0
2  time qiime tools import --input-path feature-table-v100.biom --type 'FeatureTable[Frequency]'
3
4  #BIOM v2.1.0
5  qiime tools import --input-path feature-table-v210.biom --type 'FeatureTable[Frequency]'
6
7  ##导入taxonomy文件
8  qiime tools import --input-path taxa_table.biom --type 'FeatureData[Taxonomy]' --input-f

```

• 文件导出为Biom格式

```

1  # 导出FeatureTable[Frequency]对象为BIOM v2.1格式
2  qiime tools export --input-path feature-table.qza --output-path exported-feature-table
3

```

• Biom文件转换

```

1  #转换biom为tsv格式
2  biom convert -i feature-table.biom -o feature-table.tsv --to-tsv
3
4  #tsv转换为biom格式
5  biom convert -i feature-table.txt -o feature-table.biom --table-type="OTU table" --to-hd
6
7  #转换biom为经典格式, 并在最后列包括物种注释信息
8  biom convert -i table.biom -o table.from_biom_w_taxonomy.txt --to-tsv --header-key taxon
9
10 #转换biom为经典格式, 并在最后列包括物种注释信息, 并改名为ConsensusLineage, 此功能对于一些软件要
11 biom convert -i table.biom -o table.from_biom_w_consensuslineage.txt --to-tsv --header-ke
12
13

```

```

12
13 #带物种注释表格互转
14 biom convert -i table.biom -o table_tax.txt --to-tsv --header-key taxonomy
15 biom convert -i table_tax.txt -o new_table.biom --to-hdf5 --table-type="OTU table" --proc
16 biom convert -i table_tax.txt -o new_table.biom --to-json --table-type="OTU table" --proc

```

- 取子集亚组进行分析

```

1 biom subset-table -i otu_table.biom -a sample -s samples_list.txt -o otu_table_subset.biom

```

- Biom表统计

```

1 ##统计每个样品
2 biom summarize-table -i feature-table.biom -o feature-table.summarize.txt
3 biom summarize-table -i table.w_md.biom          #结果不输出为文件
4 ##
5 biom summarize-table -i table.w_md.biom --qualitative -o table.w_md_qual_summary.txt

```

3. 数据统计及可视化

- 元数据可视化

```

1 qiime metadata tabulate --m-input-file metadata.tsv --o-visualization metadata.qzv

```

- 物种注释可视化

```

1 qiime metadata tabulate --m-input-file taxonomy.qza --o-visualization taxonomy.qzv

```

- 特征表汇总及可视化

```

1 qiime feature-table summarize --i-table table.qza --o-visualization table.qzv \
2 --m-sample-metadata-file sample-metadata.tsv

```

- 特征序列汇总及可视化

```

1 qiime feature-table tabulate-seqs --i-data rep-seqs.qza --o-visualization rep-seqs.qzv

```

- 拆分文件统计及可视化

```

1 ##这使我们确定每个样本获得多少序列，并且还可以获得序列数据中每个位置处序列质量分布的摘要。
2 time qiime demux summarize \
3   --i-data ./demux_seqs.qza \
4   --o-visualization ./demux_seqs.qzv

```

4. 数据过滤 (OTU表过滤)

• Qiime1过滤Biom文件

```

1 # 按样品数据测序量过滤：选择counts>30000的样品
2 filter_samples_from_otu_table.py -i otu_table.biom -o otu_table1.biom -n 30000
3 # 查看过滤后结果：
4 biom summarize-table -i otu_table1.biom
5 # 按样品数据测序量过滤：选择counts<10000的样品
6 filter_samples_from_otu_table.py -i otu_table.biom -o otu_table_no_high_coverage_samples
7 # 按OTU丰度过滤：选择相对丰度均值大于十万分之一的OTU
8 filter_otus_from_otu_table.py --min_count_fraction 0.00001 -i otu_table.biom -o otu_table
9 # 按物种过滤OTU表：去除p__Chloroflexi菌门等
10 filter_taxa_from_otu_table.py -i otu_table3.biom -o otu_table4.biom -n p__Chloroflexi

```

• Qiime2过滤OTU_table

```

1 #过滤至少在2个样品中存在的Feature，去除偶然的Feature
2 qiime feature-table filter-features --i-table table.qza --p-min-samples 2 --o-filtered-table
3 #过滤低丰度，< 5的特征被过滤掉
4 qiime feature-table filter-features --i-table table.qza --p-min-frequency 5 --o-filtered-table
5 ##一次命令：筛选最小频率为50，至少在4个样品中出现的特征
6 time qiime feature-table filter-features \
7   --i-table ./table_2k.qza \
8   --p-min-frequency 50 \
9   --p-min-samples 4 \
10  --o-filtered-table ./table_2k_abund.qza
11

```

• Qiime2基于物种过滤过滤OTU_table

```

1 #过滤掉含Obelia的feature
2 qiime taxa filter-table --i-table table.qza --i-taxonomy taxonomy.qza --p-exclude mitochondria

```

- Qiime2基于OTU_Table过滤ref-seq序列

```
1 qiime feature-table filter-seqs --i-data rep-seqs.qza --i-table filtered-table.qza --o-
```

- Qiime2过滤掉比稀疏深度更少的序列的样本

```
1 time qiime feature-table filter-samples \  
2   --i-table ./dada2_table.qza \  
3   --p-min-frequency 2000 \  
4   --o-filtered-table ./table_2k.qza
```

5. 物种注释+物种组成分析

- 提取参考序列（兼并引物，可能不行）

```
1 按测序的引物来提取参考序列中的一段  
2 time qiime feature-classifier extract-reads --i-sequences 85_otus.qza \  
3   --p-f-primer GGWACWGGWTGAACGWTWTAYCCYCC --p-r-primer TAIACYTCIGGRTGICCAARAAYCA \  
4   --p-min-length 300 --p-max-length 400 --o-reads ref-seqs.qza  
5 ##--p-trunc-len 120
```

- 过滤参考序列与注释信息

```
1 ## 仅保留细菌的代表序列命令  
2 qiime taxa filter-seqs  
3   --i-sequences ref-seqs.qza --i-taxonomy 99_otu_taxonomy.qza  
4   --p-include Bacteria --o-filtered-sequences ref-seqs-Bacteria.qza  
5  
6 ## 仅保留细菌的注释信息命令  
7 qiime rescript filter-taxa --i-taxonomy 99_otu_taxonomy.qza  
8   --m-ids-to-keep-file ref-seqs-Bacteria.qza --o-filtered-taxonomy ref-seqs-Bacteria-tax.q:
```

- 训练分类集

```
1 qiime feature-classifier fit-classifier-naive-bayes --i-reference-reads ref-seqs.qza  
2   --i-reference-taxonomy ref-taxonomy.qza --o-classifier classifier.qza
```

• 物种注释

```
1 qiime feature-classifier classify-sklearn --i-classifier 3xia.genus.classifier.qza --i
2 --o-classification taxonomy.3xia.genus.qza --p-n-jobs 16 #可设置线程数
```

• 选择合理算法进行注释

- 1 由于测序序列长度的限制，以及微生物种类的多样性，导致容易出现“一对多”（一条序列对应多个潜在物种注释）。
- 2 如果没有合理的注释方法，就会出现“选择困难症”。因此，在分析时，就需要对注释结果进行必要的取舍，既要保证注释的精度，又要尽可能不损失物种注释的精度（保证尽可能还原物种的精细组成）。
- 3

QIIME 2分析流程的常用注释算法 和对应数据库

扩增子数据类型	注释算法	对应参考数据库
原核16S rRNA基因	classify-sklearn	Greengenes或Silva
真核18S rRNA基因	classify-sklearn或brocc	Silva或NCBI
真菌ITS	classify-sklearn	UNITE
功能基因	brocc	NCBI-nt或nr

• 物种注释结果可视化

```
1 qiime metadata tabulate --m-input-file taxonomy.qza --o-visualization taxonomy.qzv
```

• 交互式条形图查看样本的分类组成（直接结果解压结果 barplot.qzv可得各分类水平数据）

```
1 qiime taxa barplot --i-table table.qza --i-taxonomy taxonomy.qza --m-metadata-file sample-metadata.tsv --o-barplot barplot.qzv
```

• （属）水平差异统计+可视化（3步）

- 1 #1. 各分类（11-16）水平合并并统计
- 2 qiime taxa collapse --i-table table-no-Cnidaria-unclassified.qza --i-taxonomy taxonomy.qza --p-level 6 --o-collapsed-table table-no-Cnidaria-unclassified-genus.stat.qza
- 3
- 4 循环语句多水平统计

```
5 for i in 2 6
```

```

6 do
7 qiime taxa collapse --i-table table_final.qza --i-taxonomy taxonomy.qza \
8   --p-level $i --o-collapsed-table table-l$i.qza
9 done
10 #导出 tsv
11 for file in ./table-l*.qza
12 do
13   base=$(basename $file .qza)
14   echo $base
15   qiime tools export --input-path $file --output-path $base
16   biom convert --to-tsv -i $base/feature-table.biom -o $base.tsv
17 done
18
19 #2. 格式化特征表，添加伪计数
20 qiime composition add-pseudocount --i-table genus.stat.qza --o-composition-table genus.s
21
22 #3. 计算差异属，指定分组类型
23 qiime composition ancom \
24   --i-table comp-gut-table-l6.qza --m-metadata-file sample-metadata.tsv \

```

- 相对丰度计算

```

1 qiime feature-table relative-frequency --i-table filtered-table.qza \
2   --o-relative-frequency-table filtered-relative-frequency-table.qza
3 ##导出为biom格式或可视化

```

6. 多样性分析

- 构建进化树用于多样性分析

```

1 time qiime phylogeny align-to-tree-mafft-fasttree --i-sequences rep-seqs.qza --o-alignment
2 ##创建一个片段插入树
3 time qiime fragment-insertion sepp \
4   --i-representative-sequences ./dada2_rep_set.qza \
5   --i-reference-database sepp-refs-gg-13-8.qza \
6   --o-tree ./tree.qza \
7   --o-placements ./tree_placements.qza \
8   --p-threads 1

```

• Alpha稀疏曲线--注意选择测序深度

```
1 #基于进化树
2 qiime diversity alpha-rarefaction \
3   --i-table table.qza \
4   --i-phylogeny rooted-tree.qza \
5   --p-max-depth 4000 \
6   --m-metadata-file sample-metadata.tsv \
7   --o-visualization alpha-rarefaction.qzv
8 #不基于进化树
9 qiime diversity alpha-rarefaction --i-table ./dada2_table.qza --m-metadata-file ./metada
```

• 计算核心多样性

```
1 qiime diversity core-metrics-phylogenetic --i-phylogeny rooted-tree.qza --i-table t
```

• Alpha多样性组间显著性分析和可视化

```
1 ##单因素对Alpha多样性的影响
2 #分别计算shannon、faith_pd、observed_features、evenness4个指数，注意${Alpha}_vector.qza这种
3 alpha=shannon      #指定变量，后面只需要修改指数名即可快速执行命令
4 for i in {shannon,faith_pd,observed_features,evenness};do qiime diversity alpha-group-sig
5   --m-metadata-file ./42sample.metadata2.tsv --o-visualization core-metrics-results/${i}-g
6
7 ##多因素对Alpha多样性的影响（如时间、空间）
8 ##在一些实验中，多个相互作用的因子可能共同影响α多样性。  如果我们的α多样性估计遵循[正态分布]，我们
9 qiime longitudinal anova \
10   --m-metadata-file ./core-metrics-results/faith_pd_vector.qza \
11   --m-metadata-file ./metadata.tsv \
12   --p-formula 'faith_pd ~ genotype * donor_status' \
13   --o-visualization ./core-metrics-results/faiths_pd_anova.qzv
```

• Beta多样性

```
1 ##分别计算bray_curtis、jaccard、unweighted_unifrac、weighted_unifrac4个指数，注意${index}_c
2 beta=weighted_unifrac  #赋值
3 column=body-site
```



```

4 time qiime diversity beta-group-significance \
5   --i-distance-matrix core-metrics-results/${beta}_distance_matrix.qza \
6   --m-metadata-file sample-metadata.tsv \
7   --m-metadata-column ${column} \
8   --o-visualization core-metrics-results/${beta}-${column}-significance.qzv \
9   --p-pairwise
10
11 ##使用adonis动作来查看多变量模型。 adonis动作使用PERMANOVA检验，但是允许同时检验多种效应（类似于ANOVA）
12 time qiime diversity adonis \
13   --i-distance-matrix core-metrics-results/unweighted_unifrac_distance_matrix.qza \
14   --m-metadata-file metadata.tsv \
15   --o-visualization core-metrics-results/unweighted_adonis.qzv \
16   --p-formula genotype+donor          #genotype+donor两个因素

```

7. 纵向分析（时间序列，Beta多样性）

- 教程：

<https://github.com/BANZUIGE/QIIME2ChineseManual/tree/master/docs>

- > QIIME 2教程. 07Cell帕金森小鼠Parkinson's Mouse(2021.2, 最佳实战).md

```

1 ##绘制波动率图，输入元数据，非权重unifrac的pcoa结果，指定状态列为时间，个体列为小鼠，分组列为供体
2 time qiime longitudinal volatility \
3   --m-metadata-file ./metadata.tsv \
4   --m-metadata-file ./core-metrics-results/unweighted_unifrac_pcoa_results.qza \
5   --p-state-column days_post_transplant \
6   --p-individual-id-column mouse_id \
7   --p-default-group-column 'donor_status' \
8   --p-default-metric 'Axis 2' \
9   --o-visualization ./pc_vol.qzv

```

8. Qiime1用法

- 安装

```

1 #可能需提前安装biom-format和bioconductor-microbiome
2 conda install biom-format

```

```

3 conda install bioconductor-microbiome
4 conda install --yes r-essentials blast-legacy cd-hit cdbtools cogent ea-utils \
5 infernal microbiomeutil muscle mothur r-vegan sourcetracker sortmerna sumacust rdp-classifier
6 #创建环境qiime1并安装qiime1
7 conda create -n qiime1 python=2.7 qiime matplotlib=1.4.3 mock nose -c bioconda
8 print_qiime_config.py -t #检查安装

```

• Classifier文件安装

```

1 rdp-classifier安装
2 echo "export /home/data/gmb14/qiime/Zooplankton/qiime1/rdb/rdp_classifier_2.2/rdp_classifier
3

```

• 物种注释

```

1 (1) vsearch运行 (生成biom文件, 不可读)
2 vsearch --usearch_global ASVs.fa --db rdp_16s_v16.fa --biomout out_tax.txt --id 0.97
3
4 (2) qiime1运行 (主流)
5 #bacterial
6 assign_taxonomy.py -i usearch_analysis/ASVs.fa -r 99_otus.fasta -t 99_otu_taxonomy.txt
7 --confidence=0.5 --min_consensus_fraction=0.51 --similarity=0.8 -o usearch_analysis/
8 --blast_e_value / #使用blast方法时
9 # fungal:
10 assign_taxonomy.py -i ASVs.fa -r 99_otus.fasta -t 99_otu_taxonomy.txt \
11 -m rdp --confidence=0.5 --min_consensus_fraction=0.51 --similarity=0.9 -o usearch_analysis/
12 --blast_e_value=0.001\
13 #blast方法注释, 不要用seqkit去重复序列
14 assign_taxonomy.py -m uclust -i otu_rep.fasta -r 3xia.genus.Eu.fasta \
15 -t 3xia.genus.Eu.acc.tax -o 3xia_taxonomy
16
17 (3) usearch运行
18 usearch -sintax ASVs.fa -db rdp_16s_v16.fa -tabbedout ASV_tax_raw.txt -strand both -similarity 0.97
19 fungal:
20 usearch -sintax ASVs.fa -db silva_18s_v123.fa -tabbedout ASVs_tax_assignments.txt -strand both

```

```
/taxonomy_7_levels.txt --assignment_method uclust --similarity 0.8
```

• 结合qiime2获取相对丰度

```
1 1. 导出物种分类信息和置信度
2 qiime tools export --input-path taxonomy.qza --output-path taxa
3
4 2. 导出 BIOM 表，并加入将物种分类注释信息，就是处理下表头，让他兼容biom格式
5 #处理表头：替换操作
6 sed -i -e '1 s/Feature/#Feature/' -e '1 s/Taxon/taxonomy/' taxa/taxonomy.tsv
7 #导出otu(feature)表
8 qiime tools export --input-path filtered-table.qza --output-path qiime1/table_exported
9 #添加物种注释信息
10 biom add-metadata -i table_exported/feature-table.biom --observation-metadata-fp taxa/taxonomy.tsv
11 -o feature_taxa/feature_taxa.biom --sc-separated taxonomy
12 #biom转换成txt格式
13 biom convert -i feature_taxa.biom -o feature_taxa.txt --to-tsv --header-key taxonomy
14
15 3. qiime1获利各级分类结果--biom格式（相对丰度形式）
16 #结果按门、纲、目、科、属五个级别进行分类汇总，默认结果L2-L6
17 summarize_taxa.py -i feature_taxa.biom -o sum_taxa # summary each level percentage
18 统计L7种水平
19 summarize_taxa.py -L 7 -i feature_taxa.biom -o sum_taxa
```

9. Usearch11 用法

10. 分析流程

