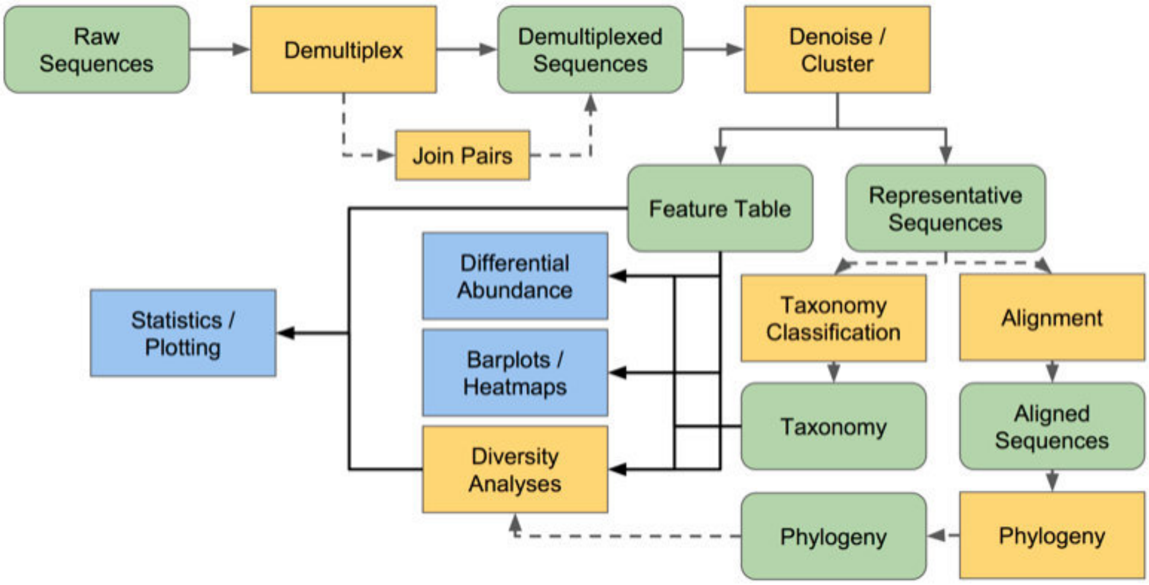
[qiime2][https://docs.qiime2.org/2019.7/]



屏显帮助信息:

`qiime --help`/`qiime demux --help`/`qiime demux emp-single --help`

Artifacts(.qza)/visualization(.qzv)文件为包含了一个或多个数据文件的zip压缩文件, 可通过`unzip`来解压缩查看, 但是更好方式是使用`qiime tools exprot`命令导出为特定格式文件([exproting tutorial][https://docs.qiime2.org/2019.7/tutorials/exporting/]).

1. 所有amplicon/metagenome 测序实验开始, 一般都是原始测序数据. 这可能是fastq数据, 包含DNA序列和对应质量值
2. 必须拆分这些数据, 知道这些reads来自哪些样本
3. 拆分后reads然后去噪音为amplicon sequence variats(ASVs)或聚类成为operational taxonomic units(OTUs), 以达成两个目的:
    - 减少测序错误
    - 去除重复序列
4. 得到的feature table和代表性序列是数据的关键信息. feature table为必要的样本观察矩阵, 例如, 数据集中每个样本的每个'feature'(OTUs, ASVs, etc)的出现次数
5. 针对feature table的分析包括:
    - Taxonomic classification of sequences(存在哪些species)
    - Alpha and beta diversity analysis, 或检测样本内和样本间的多阳性(样本的相似性)
    - 很多多样性分析是根据个体features之间的系统发育相似性进行的. 假如测序的是系统发育树标志物(16S rRNA 基因), 可将这些序列比对来评估得到的features之间的系统发育关系
    - 差异丰度测量可在不同的实验组内检测哪些features(OTUs, ASVs, taxa, etc)是显著性高/低表达的

更多统计检测: many other statistical tests and plotting methods are at your finger tips

[Common semantic types][https://docs.qiime2.org/2019.7/semantic-types/]

`FeatureTable[Frequency]`: feature表格, 其中值通过counts的形式描述一个OTU在对应样本中的表示频率

`FeatureTable[RelativeFrequency]`: feature表格, 其中值表示一个OUT在对应样本中的相对表达丰度, 其中每个样本中的值的和都为1.0

`FeatureTable[PresenceAbsence]`: feature表格, 其中值表示一个OUT在对应样本中的存在/不存在情况

`FeatureTable[Composition]`: feature表格, 每个值表明一个OUT在对应样本中的频率, 同时所有的频率都大于0

`Phylogeny[Rooted]`: rooted系统发育树

`Phylogeny[Unrooted]`: unrooted系统发育树

`DistanceMatrix`: 距离矩阵

`PCoAResults`: principal coordinate analysis(PCoA)结果

`SampleData[AlphaDiversity]`: alpha多样性值, 每个值关联一单个样本(identifier)

`SampleData[SequencesWithQuality]`: 具有质量值的序列, 每套序列都和一个样本(identifier)关联(例如, 拆分后的序列)

`SampleData[PairedEndSequencesWithQuality]`: 具有质量值的双端序列, 每套序列都和一个样本(identifier)相关联(例如, 拆分后双端序列)

`FeatureData[Taxonomy]`: 关联feature identifier的taxonomic信息

`FeatureData[Sequence]`: 关联feature identifier的单个未比对序列(例如, 代表性序列)

`FeatureData[AlignedSequence]`: 关联feature identifier的单个比对的序列, 该比对是针对所有其他feature identifier(例如, 若超过一个feature identifier存在, 表明出现多重序列比对)

`FeatureData[PairedEndSequence]`: 关联feature identifier的双端测序序列

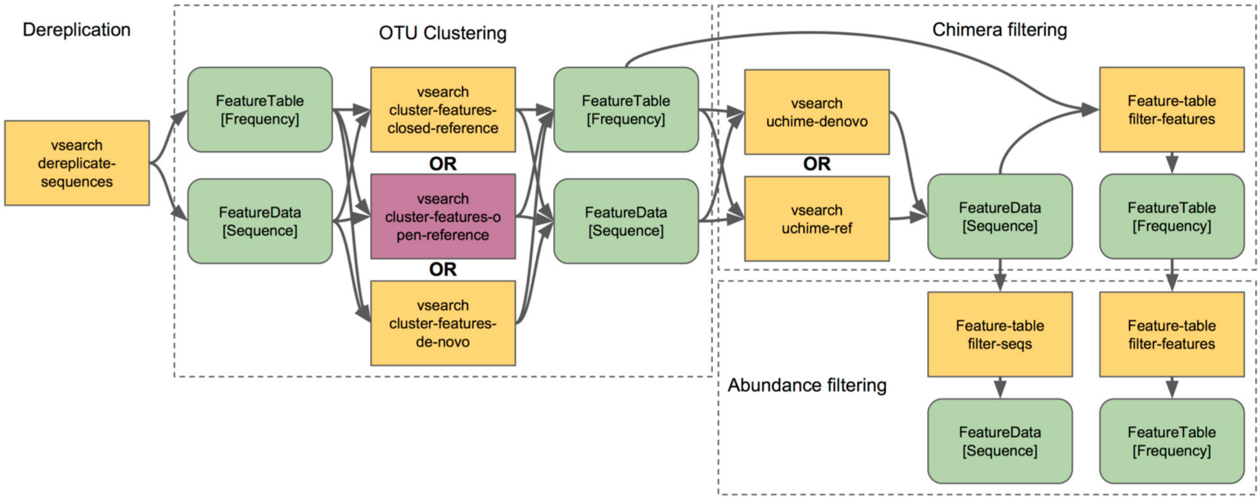
`EMPSingleEndsSequences`: 未拆分的单端测序序列数据, 根据[Earth Microbiom Project sequencing protocol][http://www.earthmicrobiome.org/protocols-and-standards/]生成

`EMPPairedEndSequences`: 同上, 未拆分的双端测序序列数据

`TaxonomicClassifier`: 训练好的classifer, 可用于执行序列的分类学比对

Clustering

`q2-vsearch`采用三种不同的[OTU clustering strategies][http://qiime.org/tutorials/otu_picking.html]: de novo, closed reference, the open reference. 所有输入输入数据都应经过[基本的质量过滤][https://www.nature.com/articles/nmeth.2276], 随后进行[chimera][https://docs.qiime2.org/2019.7/tutorials/chimera/]过滤和[aggressive OTU][https://www.nature.com/articles/nmeth.2276]过滤



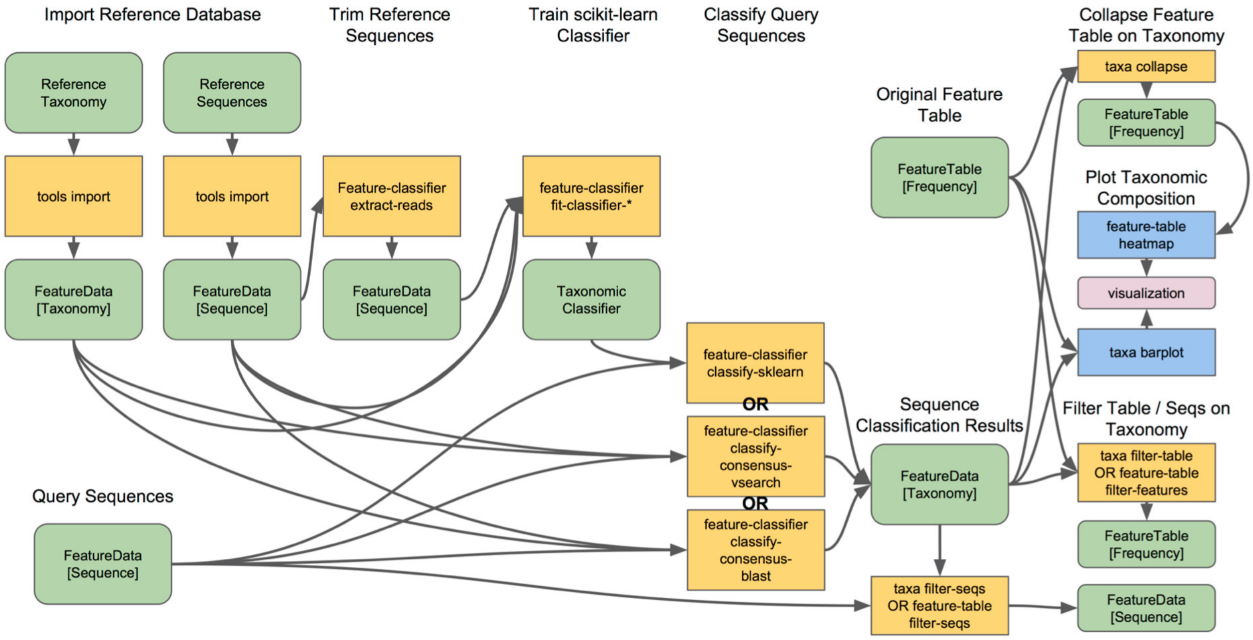
Taxonomy classification and taxonomic analyses

探究样本中所包含的生物, 通过比较query序列(features, ASVs, OTUs)和包含已知分类组成的参考数据库.

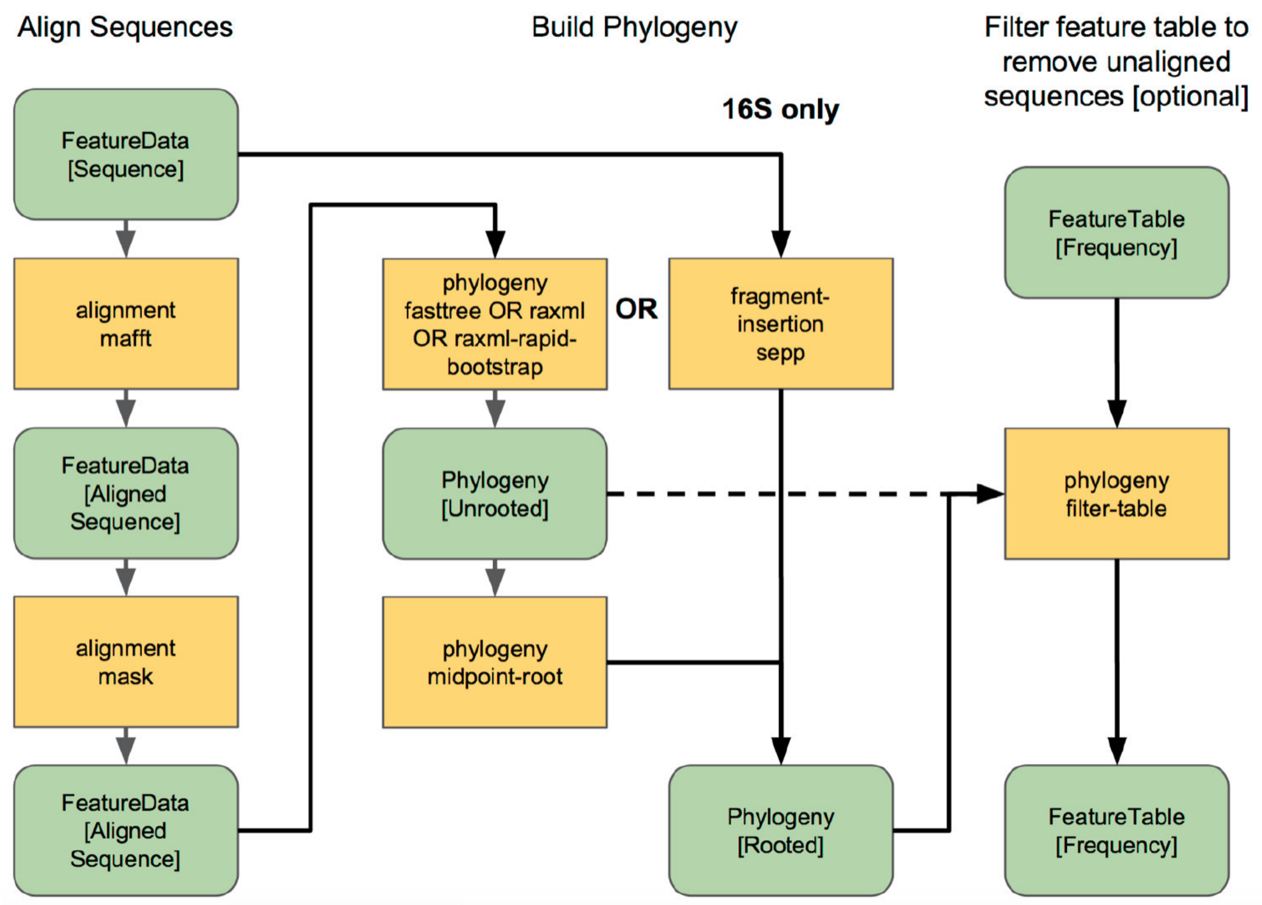
`q2-feature-classifier`包含三种不同的分类方法. `classify-consensus-blast`和`classify-consensus-vsearch`都是基于比对的方式, 在N个top hits中发现一致性比对, 这两个方法直接根据`FeatureData[Taxonomy]`和`FeatureData[Sequence]`文件比对, 无需提前训练.

`classify-sklearn`是基于机器学习的分类方法, 理论上可以采用[scikit-learn][http://scikit-learn.org/]中任何可行的方法. 这些classifier必须先训练. QIIME2也提供了多个提前训练好的[classifier][https://docs.qiime2.org/2019.7/data-resources/].

以上三个方法都很不错, 其中`classify-sklearn`使用Naive Bayes classifer表现稍微好些


Import Reference Database
Reference Taxonomy
Reference Sequences
tools import
tools import
FeatureData [Taxonomy]
FeatureData [Sequence]
Trim Reference Sequences
Feature-classifier extract-reads
FeatureData [Sequence]
Train scikit-learn Classifier
feature-classifier fit-classifier-*
Taxonomic Classifier
Classify Query Sequences
feature-classifier classify-sklearn
OR
feature-classifier classify-consensus-vsearch
OR
feature-classifier classify-consensus-blast
Query Sequences
FeatureData [Sequence]
Original Feature Table
FeatureTable [Frequency]
Sequence Classification Results
FeatureData [Taxonomy]
taxa filter-seqs OR feature-table filter-seqs
Collapse Feature Table on Taxonomy
taxa collapse
FeatureTable [Frequency]
Plot Taxonomic Composition
feature-table heatmap
visualization
taxa barplot
Filter Table / Seqs on Taxonomy
taxa filter-table OR feature-table filter-features
FeatureTable [Frequency]
FeatureData [Sequence]

Sequence alignment and phylogeny building



Diversity analysis

该分析可解决:

样本中存在多少不同的species/OTUs/ASVs

每个样本中存在什么程度的系统发育树多样性

个体样本或成组样本的相似性或差异性

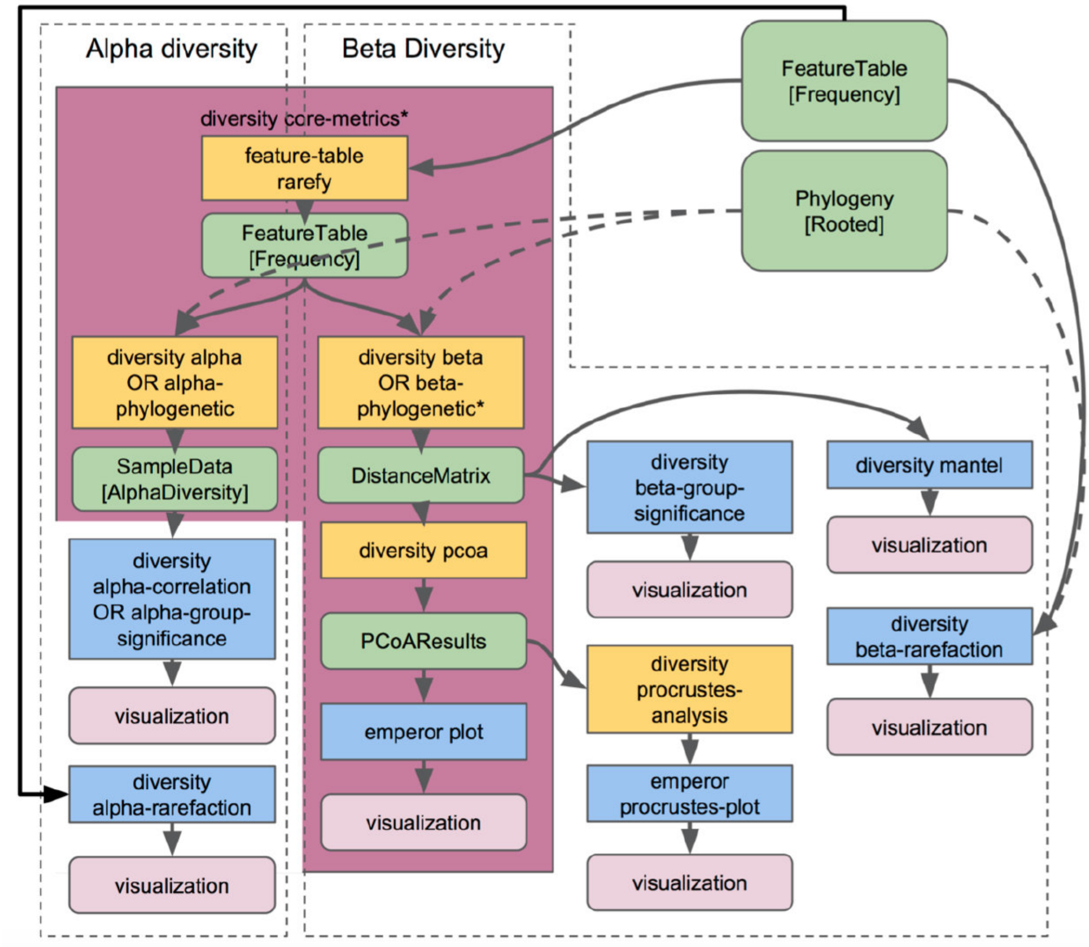
微生物组成的差异和哪些因素相关(PH, 海拔, 血压, 身体部位...)

这些问题可通过alpha-/beta-divesity分析解决. Alpha多样性检测样本内的多样性. Beta多样性检测样本间的多样性或差异性. 可用这些信息来检测组内样本的alpha diversity, 表明哪些组是拥有更高/更低的species richness, 组间的beta diversity是否更大, 例如, 组内样本间相对于其他组内的样本相似度更高, 表明这些组成员主导了这些样本的微生物组成差异

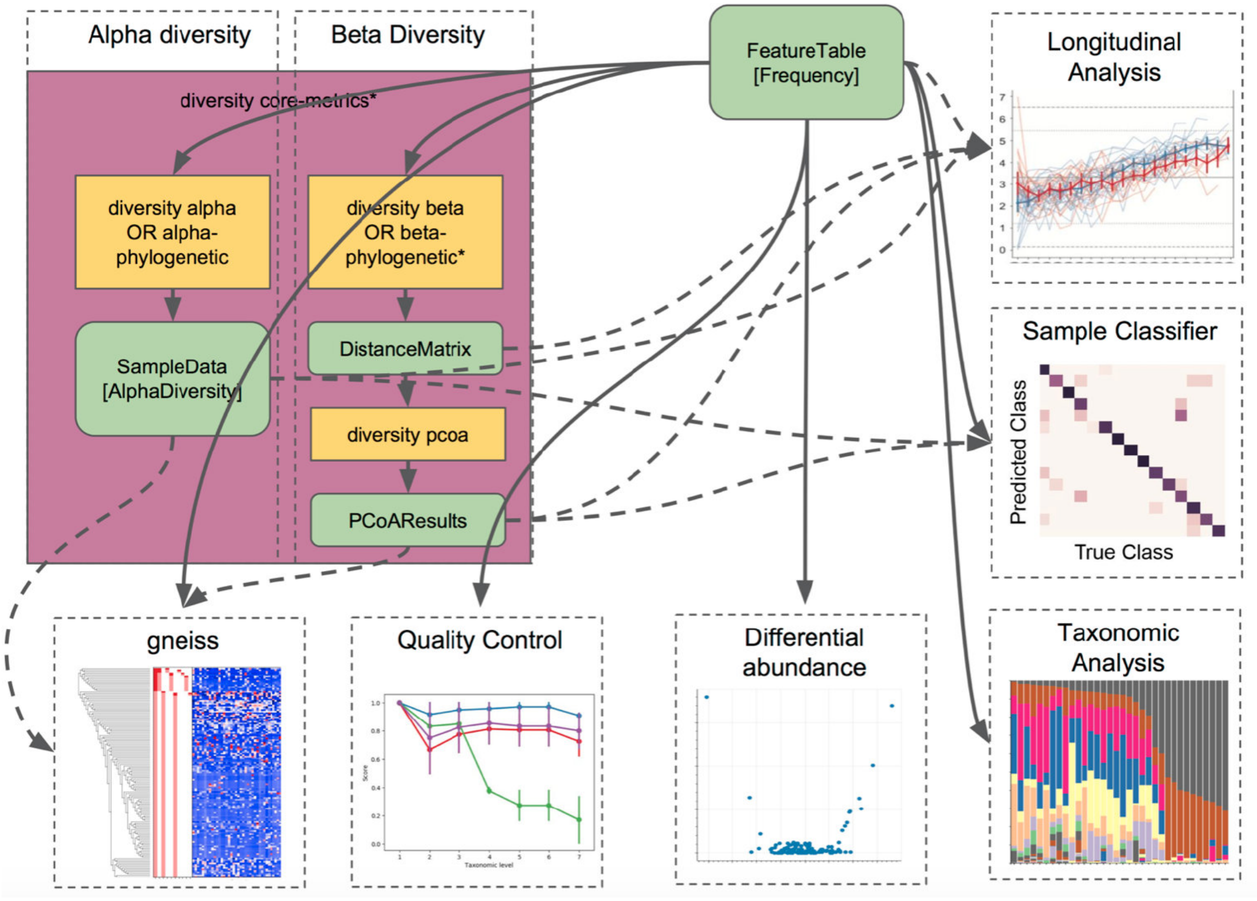
`SampleData[AlphaDiversity]` artifacts, 包含feature 表格中的每个样本的alpha多样性评估, 是用于alpha多样性分析的主要文件.

`DistanceMatrix` artifacts, 包含feature 表格中成对样本的距离/差异性值, 是用于beta多样性分析的主要文件.

`PCoAResults` artifacts, 包含每个距离/差异性 metric的principal coordinates ordination结果, [principal coordinates analysis][https://mb3is.megx.net/gustame/dissimilarity-based-methods/principal-coordinates-analysis]是一个维度所见技术, 用于在2/3维空间查看样本差异性比较



Fun with feature tables



Analyze longitudinal data: `q2-longitudinal` is a plugin for performing statistical analyses of [longitudinal experiments][https://en.wikipedia.org/wiki/Longitudinal_study], 例如, where samples are collected from individual patients/subjects/sites repeatedly over time. This include longitudinal studies of alpha and beta diversity, and some really awesome, interactive plots.

Predict the future/the past: `q2-sample-classifer`用于根据机器学习分析feature数据. 支持分类和回归模型. 通过该分析可以:

- predict sample metadata as a function of feature data(例如, 使用粪便样本预测肿瘤易感性, 或在发酵前基于葡萄的微生物组成预测红酒质量)
- identify features that are predictive of different sample characteristics
- quantity rates of microbial maturation(在胎儿肠道中追踪微生物发展, 以及长期影响不良, 抗生素, 饮食, 和分娩方式的对微生物发展的影响)
- perdict outliers and mislabeled sampels

Differential abundance: 判断不同的分组的样本中哪些features是显著性多或少, 当前QIIME2支持多种不同差异丰度检测, 包括[ANCOM][https://docs.qiime2.org/2019.7/tutorials/moving-pictures/#ancom]和[q2-gnesis][https://docs.qiime2.org/2019.7/tutorials/gneiss/]

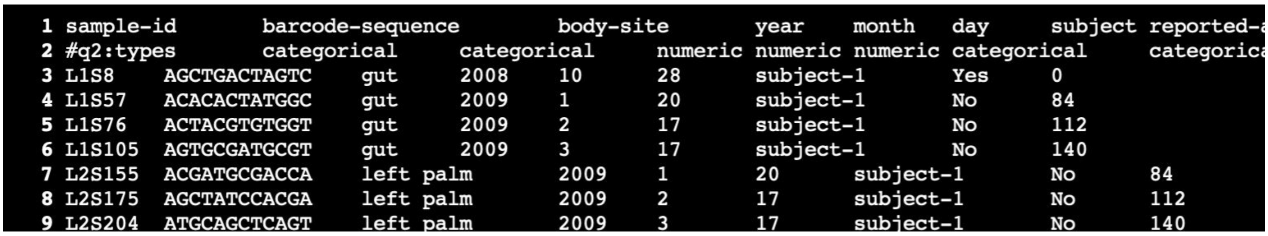
Evaluate and control data quality: `q2-quality-control`用于评估和控制测序数据质量, 包含:

- 检测不同生物信息学或分子方法多准确性, 或run与run之间多质量变化. 典型用于已知样本组成的分析, 例如, [mock communities][http://mockrobiota.caporasolab.us/]
- 根据比对到参考数据的情况来过滤序列, 或一些包含了DNA的短的部分的参考序列(例如, primer sequences). 用于去除匹配特殊物种的序列, 非靶向序列, 或其他无意义序列.

Moving picture tutorial

1. Obtaining and importing data

sample metadata: https://data.qiime2.org/2019.7/tutorials/moving-pictures/sample_metadata.tsv



```
sample-id       barcode-sequence        body-site       year    month   day     subject reported-a
#q2:types       categorical     categorical     numeric numeric numeric categorical     categorica
L1S8    AGCTGACTAGTC    gut     2008    10      28      subject-1       Yes     0
L1S57   ACACACTATGGC    gut     2009    1       20      subject-1       No      84
L1S76   ACTACGTGTGGT    gut     2009    2       17      subject-1       No      112
L1S105  AGTGCGATGCGT    gut     2009    3       17      subject-1       No      140
L2S155  ACGATGCGACCA    left palm       2009    1       20      subject-1       No      84
L2S175  AGCTATCCACGA    left palm       2009    2       17      subject-1       No      112
L2S204  ATGCAGCTCAGT    left palm       2009    3       17      subject-1       No      140
```

barcode reads:https://data.qiime2.org/2019.7/tutorials/moving-pictures/emp-single-end-sequences/barcodes.fastq.gz

sequences reads:https://data.qiime2.org/2019.7/tutorials/moving-pictures/emp-single-end-sequences/sequences.fastq.gz

将barcodes和sequences文件导入为QIIME2 artifact, 这里为`EMPSingleEndSequences`, 包含未拆分的序列信息和barcode信息

`qiime tools import \ --type EMPSingleEndSequences \ --input-path emp-single-end-sequences \ --output-path emp-single-end-sequences.qza`

2. Demultiplexing sequences

拆分序列需要知道barcode序列所应对的样本. 该信息包含在sample metadata文件.

```
qiime demux emp-single \ --i-seqs emp-single-end-sequences.qza \ --m-barcodes-file sample-metadata.tsv \ --m-barcodes-column barcode-sequence \ --o-per-sample-sequences demux.qza \ --o-error-correction-details demux-details.qza
```

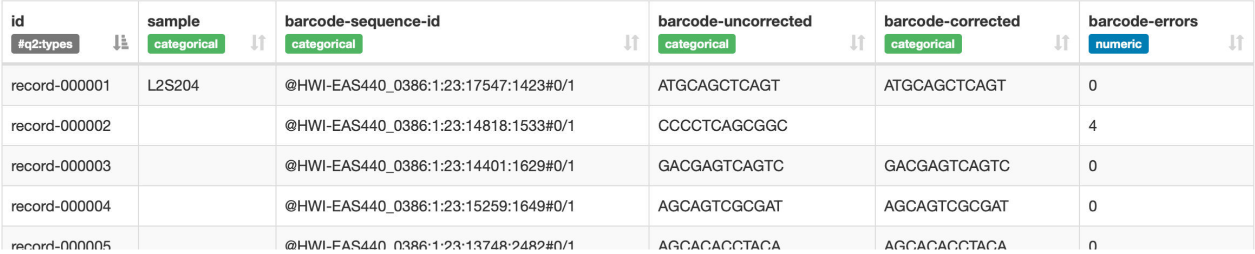
可使用命令`qiime metadata tabulate`查看`demux-details.qza`

```
qiime metadata tabulate \
```

```
--m-input-file demux-details.qza \
```

```
--o-visualization demux-details.visual
```

```
qiime tools view demux-details.visual.qzv
```



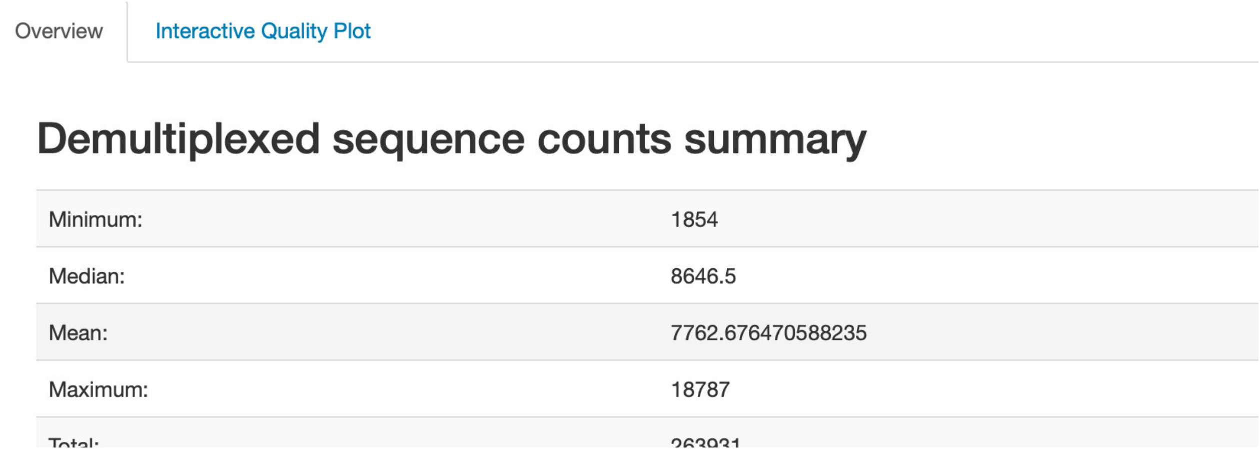
id #q2:types	sample categorical	barcode-sequence-id categorical	barcode-uncorrected categorical	barcode-corrected categorical	barcode-errors numeric
record-000001	L2S204	@HWI-EAS440_0386:1:23:17547:1423#0/1	ATGCAGCTCAGT	ATGCAGCTCAGT	0
record-000002		@HWI-EAS440_0386:1:23:14818:1533#0/1	CCCCTCAGCGGC		4
record-000003		@HWI-EAS440_0386:1:23:14401:1629#0/1	GACGAGTCAGTC	GACGAGTCAGTC	0
record-000004		@HWI-EAS440_0386:1:23:15259:1649#0/1	AGCAGTCGCGAT	AGCAGTCGCGAT	0
record-000005		@HWI-EAS440_0386:1:23:13748:2482#0/1	AGCACACCTACA	AGCACACCTACA	0

拆分完序列后, 生成拆分summary

```
qiime demux summarize \ --i-data demux.qza \ --o-visualization demux.qzv
```

使用命令查看summary文件

```
qiime tools view demux.qzv
```



Demultiplexed sequence counts summary

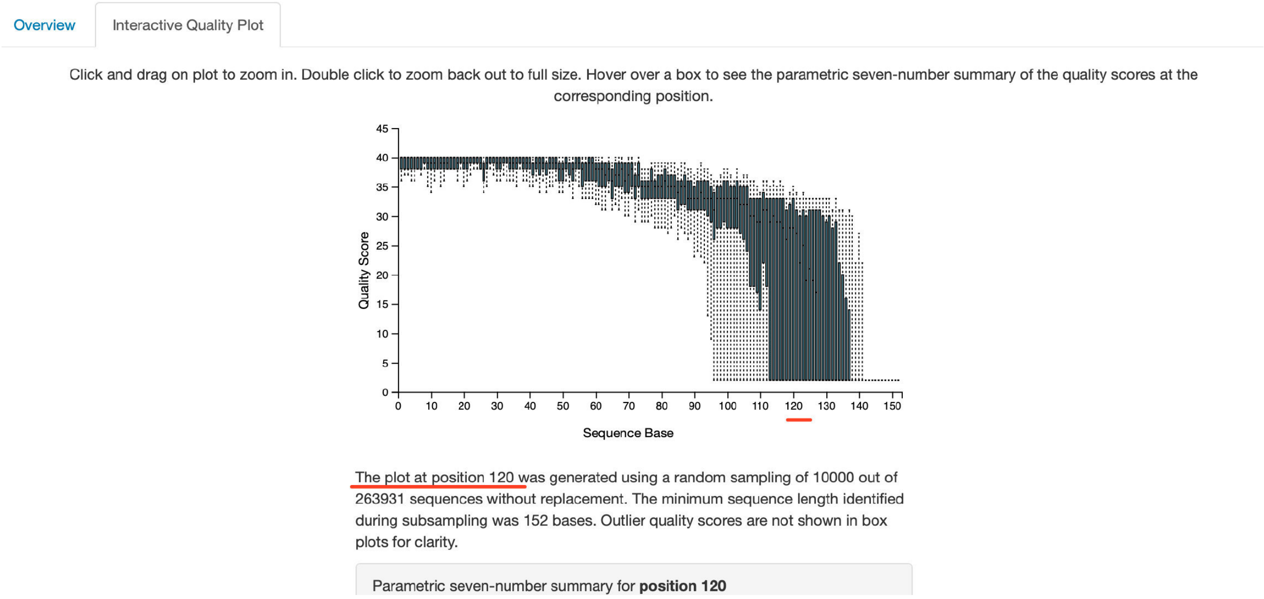
Minimum:	1854
Median:	8646.5
Mean:	7762.676470588235
Maximum:	18787
Total:	263931

3. Sequence quality control and feature table construction

QIIME2插件包括多个可行的指控方式, 包括DADA2/Deblur/basic quality-score-based filtering. 这些结果都将为一个`FeatureTable[Frequency]` QIIME2 artifact 文件, 包含数据中每个样本的唯一序列的counts(frequencies), 同时一个`FeatureData[Sequence]` QIIME2 artifact, 包含`FeatureTable`中的feature indentifiers和它们所代表序列的联系.

Option 1: DADA2

DADA2是一个可以检测并矫正Illumina amplicon sequence data的流程. 由插件`q2-dada2`完成, 该质控过程将会额外过滤序列数据中额外的phiX reads, 同时还有chimeric序列. 根据`qiime demux summarize`生成的`demux.qzv`文件, 设置对应的`--p-trim-left m`(删除每个序列前m个碱基)和`--p-trunc-len n`(在位置n截短序列), 这两个参数允许用户删除序列的低质量区域.



Click and drag on plot to zoom in. Double click to zoom back out to full size. Hover over a box to see the parametric seven-number summary of the quality scores at the corresponding position.

The plot at position 120 was generated using a random sampling of 10000 out of 263931 sequences without replacement. The minimum sequence length identified during subsampling was 152 bases. Outlier quality scores are not shown in box plots for clarity.

Parametric seven-number summary for **position 120**

可见序列初始处质量高, 在120位置质量下降厉害

```
qiime dada2 denoise-single \ --i-demultiplexed-seqs demux.qza \ --p-trim-left 0 \ --p-trunc-len 120 \ --o-representative-sequences rep-seqs-dada2.qza \ --o-table table-dada2.qza \ --o-denoising-stats stats-dada2.qza
```

```
qiime metadata tabulate \ --m-input-file stats-dada2.qza \ --o-visualization stats-dada2.qzv
```

Option 2: Deblur

Deblur使用序列的错误情况来关联具有真正生物序列的错误序列reads, 生成高质量序列变异数据(resulting in high quality sequence variant data). 第一步, 基于指定的质量值完成初始质量过滤过程.

```
qiime quality-filter q-score \ --i-demux demux.qza \ --o-filtered-sequences demux-filtered.qza \ --o-filter-stats demux-filter-stats.qza
```

In the Deblur paper, the authors used different quality-filtering parameters than what [they currently recommend after additonal analysis][https://qiita.ucsd.edu/static/doc/html/deblur_quality.html]. The paramters used here are based on those more recent recommendations

下一步, Deblur采用`qiime deblur Denise-16S`方式, 该方式需要质量过滤阶段的一个参数`--p-trim-length n`. 一般, Deblur开发者推荐设置该值为中值开始下降到非常低的位置. 在该类型meta-analysis, 在比较前所有sequencing runs的read length需要一样长, 从而避免引入study-specific bias. 根据上面分析, 这里传递参数`--p-trim-length 120`:

```
qiime deblur denoise-16S \ --i-demultiplexed-seqs demux-filtered.qza \ --p-trim-length 120 \ --o-representative-sequences rep-seqs-deblur.qza \ --o-table table-deblur.qza \ --p-sample-stats \ --o-stats deblur-stats.qza
```

可视化查看过滤结果`qiime metadata tabulate`和`qiime deblur visualize-stats`

```
qiime metadata tabulate \ --m-input-file demux-filter-stats.qza \ --o-visualization demux-filter-stats.qzv qiime deblur visualize-stats \ --i-deblur-stats deblur-stats.qza \ --o-visualization deblur-stats.qzv
```

4. FeatureTable and FeatureData summaries

经过质量过滤后, 若想要继续探索得到的数据. 通过`feature-table summarize`命令获得多少个序列和每个样本, 每个特征(features)相关联, 其分布情况的直方图, 以及相关summary statistics; `feature-table tabulate-seqs`命令提供特征(features)IDs到序列的对应关系, 同时提供用于在NCBI nt数据库blast每个序列的链接.

```
qiime feature-table summarize \ --i-table table-deblur.qza \ --o-visualization table.qzv \ --m-sample-metadata-file sample-metadata.tsv qiime feature-table tabulate-seqs \ --i-data rep-seqs-deblur.qza \ --o-visualization rep-seqs.qzv
```

5. Generate a tree for phylogenetic diversity analyses

QIIME支持多个系统多样性metrics, 包含Faith's Phylogenetic Diversity 和 weighted 和 unweighted UniFrac. 除了计算每个样本的特征(features)的counts(i.e. `featureTable`中的数据), 这些metrics需要一个rooted系统发育树来关联各个特征(features). 该信息将会保存在`Phylogeny[Rooted]`中. 这里使用`q2-phylogeny`插件中的`align-to-tree-mafft-fasttree`流程来生成系统发育树.

首先, 该流程使用`mafft`程序执行多重序列比对(序列保存于`FeatureData[Sequence]`), 来构建`FeatureData[AlignedSequence]`. 下一步, 该流程masks(or filters)比对序列来去除高变异位置. 这些位置一般认为会给最终的系统发育树增加噪音. 随后, 该流程通过FastTree使用masked的比对生成系统发育树. 由于FastTree程序得到的事unrooted树状结构, 因此最终步骤, midpoint rooting is applied to place the root of the tree at the midpoint of the longest tip-to-tip distance in the unrooted tree.

```
qiime phylogeny align-to-tree-mafft-fasttree \ --i-sequences rep-seqs-deblur.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
```

6. Alpha and beta diversity analysis

QIIME2的多样性分析可通过`q2-diversity`插件完成, 该插件支持alpha和beta多样性metrics, 通过相关的统计学检测, 生成交互可视结果. 这里首先使用`core-metrics-phylogenetic`方法, 该方法首先rarefies `FeatureTable[Frequency]`到用户指定的深度, 计算多个alpha和beta diversity metrics, 针对beta diversity metrics使用Emperor生成principle coordinates analysis(PCoA)图. 该metrics默认计算:

- Alpha diversity
    - Shannon's diversity index(a quantitative measure of community richness)
    - Observed OTUs(a qulitative measure of community richness)
    - Faith's Phylogenetic Diversity(a qualitative measure of community richness that incorportates phylogenetic relationships between the features)
    - Evenness (or Pielou's Evenness; a measure of community evenness)
- Beta diversity

- Jaccard distance(a qulitative mearuse of community dissimilarity)
- Bray-Curtis distance(a qualitative measure of community dissimilarity)
- unweighted UniFrac distance (a qualitative measure of community dissimilarity that incorporates phylogenetic relationship between the features)
- weighted UniFac distance(a qualitative measure of community dissimilarity that incorporates phylogenetic relationships between the features)

其中一个重要的参数需要提供`--p-sampling-depth`, 该值为平均的抽样深度(i.e. rarefaction). 因为大多数diversity metrics在不同的样本中抽样时对抽样深度敏感, 该步骤会根据提供的参数值随机从每个样本中抽样. **例如, `--p-sampling-depth 500`, 该步骤将使用不放回取样的方式从每个样本中抽取500个count. 若任何样本的count数目小于该值, 将舍弃该样本. 该值的选择很巧妙, 推荐查看上文`table.qzv`中的信息, 尽可能的选择高的值, 同时舍弃尽可能少的样本(根据Interactive Sample Detail tab).**

```
qiime diversity core-metrics-phylogenetic \ --i-phylogeny rooted-tree.qza \ --i-table table-deblur.qza \ --p-sampling-depth 1103 \ --m-metadata-file sample-metadata.tsv \ --output-dir core-metrics-results
```

Output visualizations:

- `core-metrics-results/unweighted_unifrac_emperor.qzv` : view | download
- `core-metrics-results/jaccard_emperor.qzv` : view | download
- `core-metrics-results/bray_curtis_emperor.qzv` : view | download
- `core-metrics-results/weighted_unifrac_emperor.qzv` : view | download

计算完diversity metrics, 我们开始探索sample metadata文件中的样本的微生物组成. 该信息存在于`sample metadata`文件中.

首先, 计算categorical metadata columns 和 alpha diversity data之间的相关性. 然后分析Faith Phylogenetic Diversity(a measure of community richness)和evenness metrics.

```
qiime diversity alpha-group-significance \ --i-alpha-diversity core-metrics-results/faith_pd_vector.qza \ --m-metadata-file sample-metadata.tsv \ --o-visualization core-metrics-results/faith-pd-group-significance.qzv
```

```
qiime diversity alpha-group-significance \ --i-alpha-diversity core-metrics-results/evenness_vector.qza \ --m-metadata-file sample-metadata.tsv \ --o-visualization core-metrics-results/evenness-group-significance.qzv
```

Output visualizations:

- `core-metrics-results/faith-pd-group-significance.qzv` : view | download
- `core-metrics-results/evenness-group-significance.qzv` : view | download

该数据集中, 因为不存在和alpha diversi相关的连续取样信息列(i.e. Days-since-experiment-start), 因此我们这里不进一步分析. 若对这些分析感兴趣(for this data set, or for others), 可以使用`qiime diversity alpha-correlation`命令实现.

接下来, 通过`beta-group-significance`命令使用PERMANOVA分析categorical metadata内容中的样本组成. 接下来的命令将会检测组内样本之间的距离相比于组间的样本间距离, 是否更加相似(例如来自肠的样本距离, 相比来自舌头, 左手掌, 右手掌的样本距离). 若同时采用了`--p-pairwise`参数, 同时将会进行成对比较, 进而判断哪对组之间存在差异比其他组之间差异大(which specific pairs of groups diff from one another).

```
qiime diversity beta-group-significance \ --i-distance-matrix core-metrics-results/unweighted_unifrac_distance_matrix.qza \ --m-metadata-file sample-metadata.tsv \ --m-metadata-column body-site \ --o-visualization core-metrics-results/unweighted-unifrac-body-site-significance.qzv \ --p-pairwise
```

```
qiime diversity beta-group-significance \ --i-distance-matrix core-metrics-results/unweighted_unifrac_distance_matrix.qza \ --m-metadata-file sample-metadata.tsv \ --m-metadata-column subject \ --o-visualization core-metrics-results/unweighted-unifrac-subject-group-significance.qzv \ --p-pairwise
```

Output visualizations:

- `core-metrics-results/unweighted-unifrac-body-site-significance.qzv` : view | download
- `core-metrics-results/unweighted-unifrac-subject-group-significance.qzv` : view | download

同样, 没有和样本组成相关的连续的采样信息, 无法检测它们之间的关系. 若感兴趣, 可使用`qiime metadata distance-matrix`和`qiime diversity mantel`和`qiime diversity bioenv`命令完成.

最后, ordination是一个用于探索样本信息条件下微生物群落组成的流行方法(ordination is a popular approach for exploring microbial community composition in the context of sample metadata). 这里使用Emperor软件来探究样本信息条件下的principal coordinates plots(PCoA). 之前采用的`core-metrics-phylogenetic`命令已经生成了一些Emperor plots, 这里想传递一些可选参数, `--p-custom-axes`, 对于探索时间系列数据非常有用(which is very useful for exploring time series data). 这里将针对unweighted UniFrac 和 Bray-Curtis生成Emperor plots, 使得最终的到的plot包含坐标, principal coordinate1, principal coordinate2, 和从实验开始的天数. 将使用最后的axis来探索这些样本随着时间的改变.

```
qiime emperor plot \ --i-pcoa core-metrics-results/unweighted_unifrac_pcoa_results.qza \ --m-metadata-file sample-metadata.tsv \ --p-custom-axes days-since-experiment-start \ --o-visualization core-metrics-results/unweighted-unifrac-emperor-days-since-experiment-start.qzv
```

```
qiime emperor plot \ --i-pcoa core-metrics-results/bray_curtis_pcoa_results.qza \ --m-metadata-file sample-metadata.tsv \ --p-custom-axes days-since-experiment-start \ --o-visualization core-metrics-results/bray-curtis-emperor-days-since-experiment-start.qzv
```

Output visualizations:

- `core-metrics-results/bray-curtis-emperor-days-since-experiment-start.qzv` : view | download
- `core-metrics-results/unweighted-unifrac-emperor-days-since-experiment-start.qzv` : view | download

7. Alpha rarefaction plotiing

使用`qiime diversity alpha-rarefaction` visualizer查看抽样深度带来的alpah diversity情况. visualizer在不同的抽样深度计算一个或多个alpha diversity, 其深度变化为1(optional controlled with `--p-min-depth`)到提供值之间(`--p-max-depth`). 在每个抽样深度步骤, 生成10 个rarefied tables, 并且计算该tables中的所有样本的diversity metrics. 迭代次数(在每个抽样深度下rarefied tables计算次数)可通过`--p-iterations`控制. 绘制每个抽样深度下的每个样本的平均diversity values plots, 若提供`--m-metadata-file`参数, 结果中的样本将会根据该参数信息进行分组.

```
qiime diversity alpha-rarefaction \ --i-table table-deblur.qza \ --i-phylogeny rooted-tree.qza \ --p-max-depth 4000 \ --m-metadata-file sample-metadata.tsv \ --o-visualization alpha-rarefaction.qzv
```

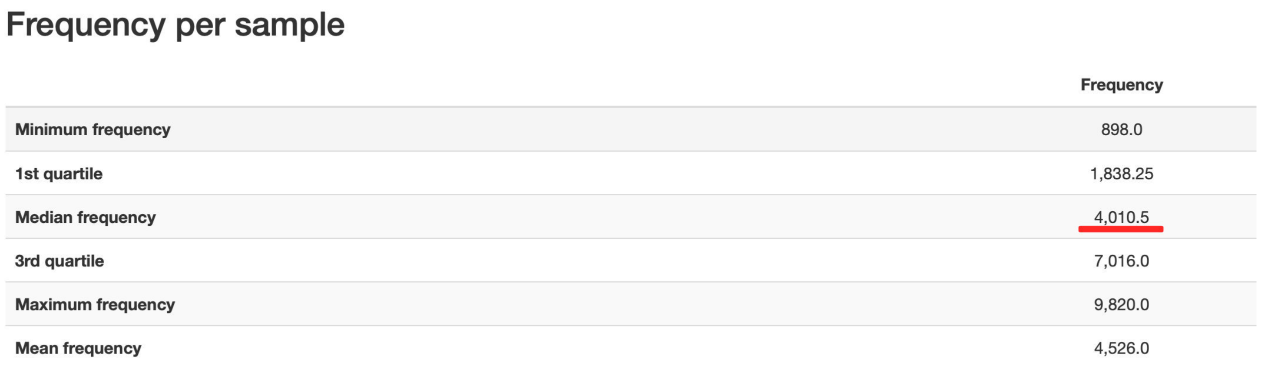
Output visualizations:

- `alpha-rarefaction.qzv` : view | download

该可视化结果包含2个图. 上面的图为alpha rarefaction plot, 主要用于判断样本的richness是否被完全测得(fully observed or sequenced). 若在取样深度下, 图中的线条的斜率接近0(appear to 'level out'), 这表明在抽样深度条件下再获得额外的测序数据不会增加观察到的特征(features); 假如该线条的斜率不是接近0, 这可能是因为样本的richness没有完全被当前得到的测序数据展示出来(because too few sequences were collected), 或者表明在数据中存在许多测序错误(which is mistaken for novel diversity).

下图展示了根据metadata信息的样本聚类图(grouping samples by metadata). 该图展示了在特征表(feature table)根据抽样深度进行rarefaction时, 每个group中包含的样本数目. 假如一个给定的抽样深度d大于一个样本s总的频率(i.e. 针对样本s获得的序列数目), 这将不可能在抽样深度d时计算样本s的diversity metric(**因为抽样过程中, 若任何样本的count数目小于该值, 将舍弃该样本**). If many of the samples in a group have lower total frequencies than d, the average diversity presented for that group at d in the top plot will be unreliable because it will have been computed on relatively few samples. When grouping samples by metadata, it is therefore essential to look at the bottom plot to ensure that the data presented in the top plot is reliable.

注意, `--p-max-depth`值应该根据上面`table.qzv`文件中的'Frequency per sample'来选择. 一般而言, 在频率中值附近选择值都将可行; 假如上图线条并没出现接近0斜率, 此时可能想要增加该值(sampling depth), or decrease that value if you seem to be losing many of your samples due to low total frequencies closer to the minimum sampling depth than the maximum sampling depth(抽样深度大于样本counts/frequency/频率, 将舍弃该样本).



Frequency per sample

	Frequency
Minimum frequency	898.0
1st quartile	1,838.25
Median frequency	4,010.5
3rd quartile	7,016.0
Maximum frequency	9,820.0
Mean frequency	4,526.0

8. Taxonomic analysis

探索每个样本的分类学组成(taxonomic composition), 同时关联到样本信息(sample metadata). 该过程的第一步是将`FeatureData`中的序列给予taxonomy名称. 这里使用pre-trained Naive Bayes classifier和`q2-feature-classifier`插件. 该classifier使用Greengenes 13_8 99% OTUs, 同时这里的序列根据本次分析的16S区域长度, 已经修剪为250bp(V4, 由515F/806R引物对扩增). 将该classifier应用到测序序列, 同时生成可视化比对情况.

注意: 针对Taxonomic classifier, 需要根据样本制备和测序的情况来训练数据, 这其中包含用于扩增的引物和测序reads读长. 因此, 一般要根据[Trainning feature classifiers with q2-feature-classifier][https://docs.qiime2.org/2019.7/tutorials/feature-classifier/]来训练自己的taxonomic classifiers. 这里已经提供了一些通用的classifiers[data resources page][https://docs.qiime2.org/2019.7/data-resources/], 包括Siva-based 16S classifiers.

gg-13-8-99-515-806-nb-classifier.qza: https://data.qiime2.org/2019.7/common/gg-13-8-99-515-806-nb-classifier.qza

```
qiime feature-classifier classify-sklearn \ --i-classifier gg-13-8-99-515-806-nb-classifier.qza \ --i-reads rep-seqs-deblur.qza \ --o-classification taxonomy.qza
```

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

Output artifacts:

- `taxonomy.qza` : view | download
- `gg-13-8-99-515-806-nb-classifier.qza` : view | download

Output visualizations:

- `taxonomy.qzv` : view | download

使用交互式条形图查看taxonomic composition

```
qiime taxa barplot \ --i-table table-deblur.qza \ --i-taxonomy taxonomy.qza \ --m-metadata-file sample-metadata.tsv \ --o-visualization taxa-bar-plots.qzv
```

Output visualizations:

- `taxa-bar-plots.qzv` : view | download

9. Differential abundance testing with ANCOM

ANCOM可在样本分组中识别差异丰度的特征(features)(i.e. present in different abundances). 和任何生物信息学方法一样, 在使用ANCOM前需要了解其假设和局限性. 推荐在使用前阅读其文章[ANCOM paper][https://www.ncbi.nlm.nih.gov/pubmed/26028277]

注意: microbiome analysis中的差异丰度检测是一个研究热点. QIIME2包括2个方法: `q2-gneiss`和`q2-composition`. 这里使用的是`q2-composition`, 另一个教程使用的是[gneiss][https://docs.qiime2.org/2019.7/tutorials/gneiss/]

在QIIME2中, 通过插件`q2-composition`进行ANCOM分析. ANCOM假设不同的groups间存在少量的特征(features)改变(less than about 25%). 若期待更多的特征出现改变, 那就不要使用ANCOM, 将带来更多错误可能(an increasing in both Type I and Type II errors is possible).

由于我们期待不同的身体不会会出现许多特征(features)发生改变, 因此这里根据two subjects, 仅针对gut smaples做丰度差异分析.

首先构建仅包含gut sample的特征表格(feature table)

```
qiime feature-table filter-samples \ --i-table table-deblur.qza \ --m-metadata-file sample-metadata.tsv \ --p-where "[body-site]='gut'" \ --o-filtered-table gut-table.qza
```

Output artifacts:

- `gut-table.qza` : view | download

ANCOM针对`FeatureTable[Composition]`进行分析, 该软件基于特征的频率(which is based on frequencies of features on a per-sample basis), 但是不能包含频率为0的数据. 因此, 需先对`FeatureTable[Frequency]`做处理

```
qiime composition add-pseudocount \ --i-table gut-table.qza \ --o-composition-table comp-gut-table.qza
```

Output artifacts:

- `comp-gut-table.qza` : view | download

运行`subject`列来判断差异丰度feature

```
qiime composition ancom \ --i-table comp-gut-table.qza \ --m-metadata-file sample-metadata.tsv \ --m-metadata-column subject \ --o-visualization ancom-subject.qzv
```

Output visualizations:

- `ancom-subject.qzv` : view | download

同时, 我们也想在特殊的taxonomic level上执行差异丰度检测. To do this, we can collapse the features in our `FeatureTable[Frequency]` at the taxonomic level of interest, and then re-run the above steps. 这里在genus水平查看feature情况

```
qiime taxa collapse \ --i-table gut-table.qza \ --i-taxonomy taxonomy.qza \ --p-level 6 \ --o-collapsed-table gut-table-l6.qza
```

```
qiime composition add-pseudocount \ --i-table gut-table-l6.qza \ --o-composition-table comp-gut-table-l6.qza
```

```
qiime composition ancom \ --i-table comp-gut-table-l6.qza \ --m-metadata-file sample-metadata.tsv \ --m-metadata-column subject \ --o-visualization l6-ancom-subject.qzv
```

Output artifacts:

- `gut-table-l6.qza` : view | download
- `comp-gut-table-l6.qza` : view | download

Output visualizations:

- `l6-ancom-subject.qzv` : view | download

["Atacama soli microbiome" tutorial][https://docs.qiime2.org/2019.7/tutorials/atacama-soils/#atacama-demux]

sample-metadata.tsv: https://data.qiime2.org/2019.7/tutorials/atacama-soils/sample_metadata.tsv emp-paired-end-sequences/forward.fastq.gz: https://data.qiime2.org/2019.7/tutorials/atacama-soils/10p/forward.fastq.gz

emp-paired-end-sequences/reverse.fastq.gz: https://data.qiime2.org/2019.7/tutorials/atacama-soils/10p/reverse.fastq.gz

emp-paired-end-sequences/barcodes.fastq.gz: https://data.qiime2.org/2019.7/tutorials/atacama-soils/10p/barcodes.fastq.gz

1. Paired-end read analysis commands

```
qiime tools import \ --type EMPPairedEndSequences \ --input-path emp-paired-end-sequences \ --output-path emp-paired-end-sequences.qza
```

拆分序列reads. 需要sample metadata文件, 同时指定包含样本barcodes的列. 该例子中, barcode reads是反向互补地包含在sample metadata文件中, 使用参数`--p-rev-comp-mapping-barcodes`

```
qiime demux emp-paired \ --m-barcodes-file sample-metadata.tsv \ --m-barcodes-column barcode-sequence \ --p-rev-comp-mapping-barcodes \ --i-seqs emp-paired-end-sequences.qza \ --o-per-sample-sequences demux.qza \ --o-error-correction-details demux-details.qza
```

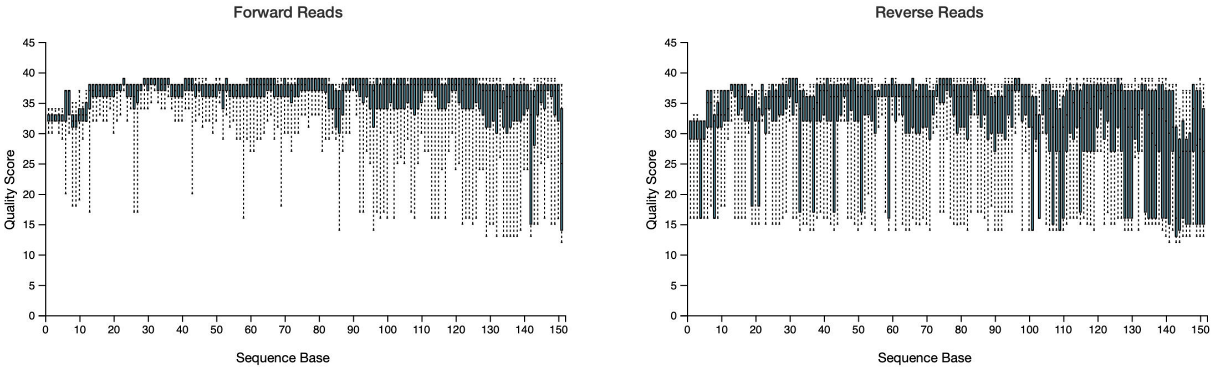
```
qiime demux summarize \ --i-data demux.qza \ --o-visualization demux.qzv
```

Output artifacts:

- `demux-details.qza` : **view** | **download**
- `demux.qza` : **view** | **download**

Output visualizations:

- `demux.qzv` : **view** | **download**



根据forward/reverse reads的质控图. 因为需要read足够长来满足双端read的重叠, 根据图示去除foward/reverse reads的前13bp, 但不修剪reads的末端序列, 避免减少read长度太多:

```
qiime dada2 denoise-paired \ --i-demultiplexed-seqs demux.qza \ --p-trim-left-f 13 \ --p-trim-left-r 13 \ --p-trunc-len-f 150 \ --p-trunc-len-r 150 \ --o-table table.qza \ --o-representative-sequences rep-seqs.qza \ --o-denoising-stats denoising-stats.qza
```

根据对应的sample-metadata信息, 获得对应统计信息:

```
qiime feature-table summarize \ --i-table table.qza \ --o-visualization table.qzv \ --m-sample-metadata-file sample-metadata.tsv
qiime feature-table tabulate-seqs \ --i-data rep-seqs.qza \ --o-visualization rep-seqs.qzv
```

同时查看去噪音统计:

```
qiime metadata tabulate \ --m-input-file denoising-stats.qza \ --o-visualization denoising-stats.qzv
```

至此, 剩下的步骤和单端reads数据一样, 可移步[the moving pictures tutorial][https://docs.qiime2.org/2019.7/tutorials/moving-pictures/]

Miscellaneous

1. Filtering data

过滤feature tables, sequences, distance matrices...

```
mkdir qiime2-filtering-tutorial
```

```
cd qiime2-filtering-tutorial
```

下载流程数据

Sample-metadata.tsv: https://data.qiime2.org/2019.7/tutorials/moving-pictures/sample_metadata.tsv

table.qza: https://data.qiime2.org/2019.7/tutorials/filtering/table.qza

distance-matrix.qza: https://data.qiime2.org/2019.7/tutorials/filtering/distance-matrix.qza

taxonomy.qza: https://data.qiime2.org/2019.7/tutorials/filtering/taxonomy.qza

sequences.qza: https://data.qiime2.org/2019.7/tutorials/filtering/sequences.qza

Filtering feature tables

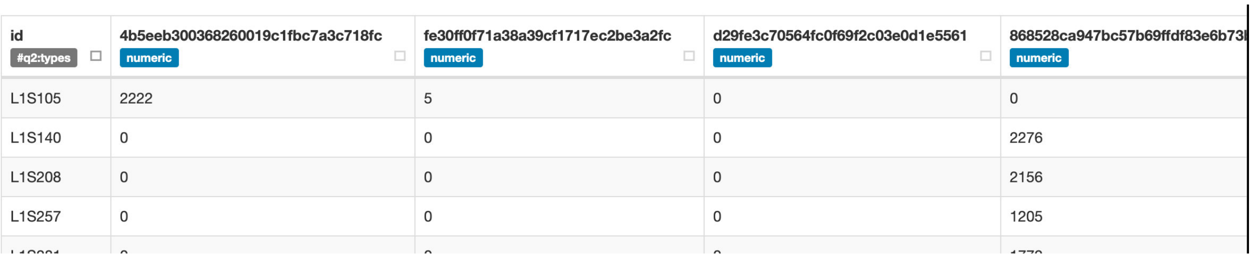
从feature table中取出samples和features. Feature tables拥有两个坐标: sample axis和feature axis. 针对这两个坐标使用的方法为`filter-samples`和`filter-features`. 这两种方法都用在`q2-feature-table`插件中. 同时根据taxonomy信息可以用来从feature table过滤feature, 使用`q2-taxa`的`filter-table`方法.

Total-frequency-based filtering

Total-frequency的过滤可以根据频率过滤feature table中samples或features.

例如, 过滤sample frequencies中的outlier samples. 在许多16S 分析中, 针对一些samples, 仅获得少数序列(perhaps 10s), 可能由于低的样本biomass导致低的DNA提取量. 在这些情形中, 用户可能想根据最小的总的frequencies来去除一些样本(i.e. total number of sequences obtained for the sample). 这里选择1500, 低于1500的总frequency的samples将被过滤掉:

```
qiime feature-table filter-samples \ --i-table table.qza \ --p-min-frequency 1500 \ --o-filtered-table sample-frequency-filtered-table.qza
```



id	4b5eeb300368260019c1fbc7a3c718fc	fe30ff0f71a38a39cf1717ec2be3a2fc	d29fe3c70564fc0f69f2c03e0d1e5561	868528ca947bc57b69ffdf83e6b73
#q2:types	numeric	numeric	numeric	numeric
L1S105	2222	5	0	0
L1S140	0	0	0	2276
L1S208	0	0	0	2156
L1S257	0	0	0	1205

同时也可根据feature来过滤, 过滤掉低丰度低features

```
qiime feature-table filter-features \ --i-table table.qza \ --p-min-frequency 10 \ --o-filtered-table feature-frequency-filtered-table.qza
```

以上过滤过程也可根据最大的total frequency过滤, `--p-max-frequency`, `--p-min-frequency`, `--p-max-frequency`, 且可合并使用过滤

Contingency-based filtering(可能性过滤)

Contingency-based过滤可以根据samples所包含的features, 或根据features所存在的samples来过滤.

该过滤一般用于过滤仅存在一个或多个样本中的features, 该过滤依据是该features可能不是真正存在于biological diversity中, 而是由于PCR或测序错误导致的(例如, PCR chimeras). 例如, 过滤那些仅存在与1个样本的features:

```
qiime feature-table filter-features \ --i-table table.qza \ --p-min-samples 2 \ --o-filtered-table sample-contingency-filtered-table.qza
```

同样, 仅包含少数features的samples也可以过滤:

```
qiime feature-table filter-samples \ --i-table table.qza \ --p-min-features 10 \ --o-filtered-table feature-contingency-filtered-table.qza
```

以上两种contingency过滤也可联合使用, 或过滤那些过大的features/samples, `--p-max-features`, `--p-max-samples`

Identifier-based filtering

该过滤根据指定的samples/features IDs列表来保留对应的数据. 为根据IDs过滤, 应提供metadata file, `--m-metadata-file`参数. 过滤时仅保留第一列中的IDs信息, 所有其他列的信息都不会读取:

首先根据IDs过滤要求, 构建metadata文件, 该文件包含表头信息:

```
echo SampleID > samples-to-keep.tsv echo L1S8 >> samples-to-keep.tsv echo L1S105 >> samples-to-keep.tsv
```

运行`filter-samples`

```
qiime feature-table filter-samples \ --i-table table.qza \ --m-metadata-file samples-to-keep.tsv \ --o-filtered-table id-filtered-table.qza
```

Metadata-based filtering

Metadata-based过滤类似于identifier-based过滤, 不同的是不是根据提供的IDs来保留, 而是根据搜索标准来保留. 用户提供样本描述信息`--p-where`, 保留`--m-metadata-file`中的匹配samples(语法信息, SQLite [WHERE-clause][https://en.wikipedia.org/wiki/Where_(SQL)] syntax).

```
qiime feature-table filter-samples \ --i-table table.qza \ --m-metadata-file sample-metadata.tsv \ --p-where "[subject]='subject-1'" \ --o-filtered-table subject-1-filtered-table.qza
```

若从单个metadata列中选择保留多个值, 使用`IN`来实现:

```
qiime feature-table filter-samples \ --i-table table.qza \ --m-metadata-file sample-metadata.tsv \ --p-where "[body-site] IN ('left palm', 'right palm')" \ --o-filtered-table skin-filtered-table.qza
```

同时`--p-where`表达式可以使用`AND`和`OR`来结合多个列过滤, `AND` 表共同满足:

```
qiime feature-table filter-samples \ --i-table table.qza \ --m-metadata-file sample-metadata.tsv \ --p-where "[subject]='subject-1' AND [body-site]='gut'" \ --o-filtered-table subject-1-gut-filtered-table.qza
```

`OR`表示满足一个即可:

```
qiime feature-table filter-samples \ --i-table table.qza \ --m-metadata-file sample-metadata.tsv \ --p-where "[body-site]='gut' OR [reported-antibiotic-usage]='Yes'" \ --o-filtered-table gut-abx-positive-filtered-table.qza
```

还满足`AND NOT`描述:

```
qiime feature-table filter-samples \ --i-table table.qza \ --m-metadata-file sample-metadata.tsv \ --p-where "[subject]='subject-1' AND NOT [body-site]='gut'" \ --o-filtered-table subject-1-non-gut-filtered-table.qza
```

Taxonomy-based filtering of tables and sequences

基于分类学的过滤是非常常见的feature-metadata-based filtering, `filter-table`可用于简化该步骤. 过滤可以通过`--p-include`来保留指定的taxa或通过`--p-exclude`来排除指定的taxa.

例如, 排除所有注释为`mitochondria`的所有featues, 当`--p-mode contains`(默认), 该搜索是大小写不敏感的, 因为`Mitochondria`同样搜索到:

```
qiime taxa filter-table \ --i-table table.qza \ --i-taxonomy taxonomy.qza \ --p-exclude mitochondria \ --o-filtered-table table-no-mitochondria.qza
```

根据taxonomic annotation, 删除多个搜索项目的features, 通过逗号分隔搜索条目, 例如去除包含`mitochondira`或`chloroplast`的features:

```
qiime taxa filter-table \ --i-table table.qza \ --i-taxonomy taxonomy.qza \ --p-exclude mitochondria,chloroplast \ --o-filtered-table table-no-mitochondria-no-chloroplast.qza
```

还可以通过`--p-include`来保留最低注释水平的features, 使用`p__`来表示注释到了phylum-level:

```
qiime taxa filter-table \ --i-table table.qza \ --i-taxonomy taxonomy.qza \ --p-include p__ \ --o-filtered-table table-with-phyla.qza
```

同时使用`--p-include`和`--p-exclude`参数来过滤:

```
qiime taxa filter-table \ --i-table table.qza \ --i-taxonomy taxonomy.qza \ --p-include p__ \ --p-exclude mitochondria,chloroplast \ --o-filtered-table table-with-phyla-no-mitochondria-no-chloroplast.qza
```

默认条件下, 根据`--p-include`和`--p-exclude`, 只要匹配的内容包含在taxonomic annotaion中就可实现过滤. 若希望搜索内容完全匹配, 则需要提供提供参数`--p-mode exact`(来表明搜索需要完全匹配方可), 同时该参数表明搜索内容是大小写敏感的.

例如, 确切匹配到该搜索内容:

```
qiime taxa filter-table \ --i-table table.qza \ --i-taxonomy taxonomy.qza \ --p-mode exact \ --p-exclude "k__Bacteria; p__Proteobacteria; c__Alphaproteobacteria; o__Rickettsiales; f__mitochondria" \ --o-filtered-table table-no-mitochondria-exact.qza
```

Taxonomy-based过滤也可以通过`qiime feature-table filter-features`搭配参数`--p-where`来过滤.

Filtering sequences

`filtering-seqs`可以根据feature的taxonomic annotation来过滤`FeatureData[Sequence]`. 该过滤非常类似于`qiime taxa filter-table`. 例如, 保留所有包含phylum-levels annotation, 同时删除`mitochondira`或`chloroplast`的features:

```
qiime taxa filter-seqs \ --i-sequences sequences.qza \ --i-taxonomy taxonomy.qza \ --p-include p__ \ --p-exclude mitochondria,chloroplast \ --o-filtered-sequences sequences-with-phyla-no-mitochondria-no-chloroplast.qza
```

`q2-feature-table`可拥有`filter-seqs`方法, 可用于根据多种标准过滤序列, 其features包含在feature table中; **`q2-quality-control`中的`exclude-seqs`可用于过滤匹配参考序列或引物的序列**

Filtering distance matrices

通过`q2-diversity`的`filter-distance-matrix`根据distance matrix过滤样本. 该过滤同根据identifier或sample metadata过滤 feature table.

例如, 根据identifiers来过滤distance matrix:

```
qiime diversity filter-distance-matrix \ --i-distance-matrix distance-matrix.qza \ --m-metadata-file samples-to-keep.tsv \ --o-filtered-distance-matrix identifier-filtered-distance-matrix.qza
```

根据sample metadata过滤:

```
qiime diversity filter-distance-matrix \ --i-distance-matrix distance-matrix.qza \ --m-metadata-file sample-metadata.tsv \ --p-where "[subject]='subject-2'" \ --o-filtered-distance-matrix subject-2-filtered-distance-matrix.qza
```

2. Training feature classifiers with q2-feature-classifier

这里使用[Greengenes][http://qiime.org/home_static/dataFiles.html]参考序列训练[Naive Bayes classifier][https://scikit-learn.org/stable/modules/naive_bayes.html#multinomial-naive-bayes]; 同时QIIME2 [data resources][https://docs.qiime2.org/2019.7/data-resources/]包含多个训练好的classifiers.

Obtaining and importing reference data sets

训练the classifier需要两个输入: 参考序列(the reference sequences)和相应的分类数据(the corresponding taxonomic classification).

为减少计算时间, 这里将使用小的[Greengenes][http://qiime.org/home_static/dataFiles.html] 13_8 85% OTU data set. 注意, 不要使用这里使用的85% OTU数据集用于真实的实验数据.

推荐使用信息更丰富的序列数据, 根据99%的序列相似度聚类的参考序列用于实际数据的分类. 完整的QIIME兼容的参考数据集: [data resources page][https://docs.qiime2.org/2019.10/data-resources/]

注意: 所有参考序列中的序列IDs必须存在于reference taxonomy. 若使用的参考序列集已经聚到了OTUs内, 需确保使用对应的reference taxonomy. 例如, 使用了Greengenes 99% OTU sequences, 就需要使用99% OTU taxonomy(确保IDs一致).

85_otus.fasta: https://data.qiime2.org/2019.7/tutorials/training-feature-classifiers/85_otus.fasta

85_otu_taxonomy.txt: https://data.qiime2.org/2019.7/tutorials/training-feature-classifiers/85_otu_taxonomy.txt

rep-seqs.qza: https://data.qiime2.org/2019.7/tutorials/training-feature-classifiers/rep-seqs.qza

首先将以上下载数据导入到QIIME2 Artifacts. 因为Greengenes referecne taxonomy file(`85_otu_taxonomy.txt`)是tab分隔的, 不含表头的TSV文件, 这里需要指定`HeaderlessTSVTaxonomyFormat`, 因为默认的格式包含表头行.

```
qiime tools import \ --type 'FeatureData[Sequence]' \ --input-path 85_otus.fasta \ --output-path 85_otus.qza
```

```
qiime tools import \ --type 'FeatureData[Taxonomy]' \ --input-format HeaderlessTSVTaxonomyFormat \ --input-path 85_otu_taxonomy.txt \ --output-path ref-taxonomy.qza
```

Output artifacts:

- `85_otus.qza` : **view** | **download**
- `rep-seqs.qza` : **view** | **download**
- `ref-taxonomy.qza` : **view** | **download**

Extract reference reads

It has been shown that taxonomic classification accuracy of 16S rRNA gene sequences improves when a Naive Bayes classifier is trained on only the region of the target sequences that was sequenced([Werner et al., 2012][https://www.ncbi.nlm.nih.gov/pubmed/21716311]).

已知`Moving Pictures`教程中的测序reads为120bp单端测序, 扩增引物为515F/806R 引物对. 根据匹配的引物对信息和reads长度来从参考数据中提取对应信息:

```
qiime feature-classifier extract-reads \ --i-sequences 85_otus.qza \ --p-f-primer GTGCCAGCMGCCGCGGTAA \ --p-r-primer GGACTACHVGGGTWTCTAAT \ --p-trunc-len 120 \ --p-min-length 100 \ --p-max-length 400 \ --o-reads ref-seqs.qza
```

注意: `--p-trunc-len`仅在query序列被修剪到相同长度或更短时, 用于修建参考序列. 成功合并了的双端测序序列一般在长度上会有不同, 单端测序若不截短到指定长度也不呈现不同长度. 针对双端reads和没有修建的单端reads, 推荐根据适当引物提取的的序列来训练classifier, 但是不进行修剪.

注意: `min-lenght`和`max-length`参数用于排除过远偏离当前引物下的扩增产物分布. 这些过分偏离的扩增产物很可能是non-target hits应该被舍弃. 实际应用时, `min-length`参数在`trim-left`和`trunc-len`之后运行, `max-length`参数是在`trim-left`和`trunc-len`之前作用.

Train the classifier

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

Test the classifier

```
qiime feature-classifier classify-sklearn \ --i-classifier classifier.qza \ --i-reads rep-seqs.qza \ --o-classification taxonomy.qza
```

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

Output artifacts:

- `taxonomy.qza` : view | download

Output visualizations:

- `taxonomy.qzv` : view | download

Classification of fungal ITS sequences

推荐使用full reference sequences来训练UNIT classifiers. 根据[UNITE reference database][https://unite.ut.ee/repository.php], 而不是根据引物位置提取或修剪的reads, 来训练fungal ITS classifiers. Furthermore, we recommend the “developer” sequences (located within the QIIME-compatible release download) because the standard versions of the sequences have already been trimmed to the ITS region (excluding portions of flanking rRNA genes that may be present in amplicons generated with standard ITS primers).

---

3. Exporting data

需要导出的数据必须为QIIME2 artifacts(i.e. `.qza`).

Exporting a feature table

`FeatureTable[Frequency]` artifacts将会被导出为[BIO v2.1.0 formated file][http://biom-format.org/documentation/format_versions/biom-2.1.html].

feature-table.qza: https://data.qiime2.org/2019.7/tutorials/exporting/feature-table.qza `qiime tools export \ --input-path feature-table.qza \ --output-path exported-feature-table`

Exporting a phylogenetic tree

`Phylogeny[Unrooted]` artifacts将会被导出为[newick formated file][http://scikit-bio.org/docs/latest/generated/skbio.io.format.newick.html].

unrooted-tree.qza: https://data.qiime2.org/2019.7/tutorials/exporting/unrooted-tree.qza

`qiime tools export \ --input-path unrooted-tree.qza \ --output-path exported-tree`

Exporting versus extracting

`qiime tools extract`, extracting an artifact 不同于 exporting an artifact. 在exporting an artifact时, 只有数据文件被导出到输出目录, extracting会提取出额外关于artifact的QIIME2' metadata, 例如, 包括artifact的出处. 同时, 该用于提取的目录必须已经存在.

`mkdir extracted-feature-table qiime tools extract \ --input-path feature-table.qza \ --output-path extracted-feature-table`

---

4. Improting data

导入数据类型:

- Sequence data with sequence quality information (i.e. FASTQ)
- Sequence without quality information (i.e. FASTA)
- Per-feature unaligned sequence data (i.e. representative FASTA sequences)
- Per-feature aligned sequence data (i.e. aligned representative FASTA sequences)
- Feature table data
- Phylogenetic trees
- Other data types

为使用QIIME2, 输入数据必须为QIIME2 artifacts(i.e. `.qza`)格式.

Sequence data with sequence quality information (i.e. FASTQ)

不同类型FASTQ数据:

- FASTQ data with the EMP Protocol format
- FASTQ data with the Casava 1.8 dumultiplexed format
- Any other kind of FASTQ data

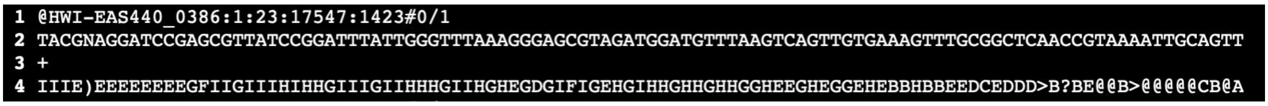
1. "EMP protocol" multiplexed single-end fastq

单端'Earth Microbiom Project(EMP) protocol'格式reads应包含两个`fastq.gz`文件:

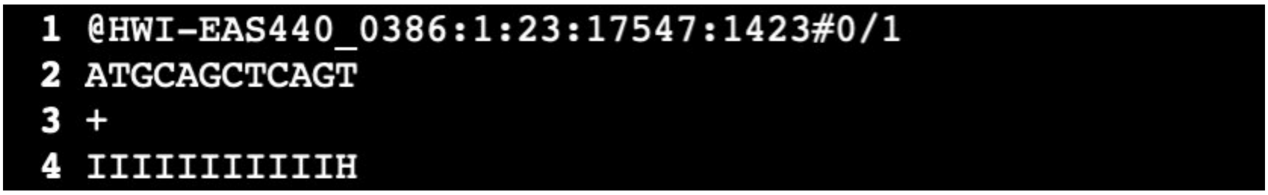
- `fastq.gz`文件包含单端reads
- `barcodes.fastq.gz`文件包含相关的barcode reads

这两个`fastq.gz`中的内容顺序是对应的序列read和其barcode read(第一个barcode read对应第一个序列read, 第二个barcode read对应第二个序列read, 依次类推)

```
mkdir emp-single-end-sequences
```



```
@HWI-EAS440_0386:1:23:17547:1423#0/1
TACGNAGGATCCGAGCGTTATCCGGATTTATTGGGTTTAAAGGGAGCGTAGATGGATGTTTAAGTCAGTTGTGAAAGTTTGCGGCTCAACCGTAAAATTGCAGTT
+
IIIE)EEEEEEEEGFIIGIIIHIHHGIIIGIIHHHGIIHGHEGDGIFIGEHGIHHGHHGHHGGHEEGHEGGEHEBBHBBEEDCEDDD>B?BE@@B>@@@@@CB@A
```



```
@HWI-EAS440_0386:1:23:17547:1423#0/1
ATGCAGCTCAGT
+
IIIIIIIIIIIH
```

```
qiime tools import \ --type EMPSingleEndSequences \ --input-path emp-single-end-sequences \ --output-path emp-single-end-sequences.qza
```

2. "EMP protocol" multiplexed paired-end fastq

拥有3个`fastq.gz`文件: 正向/反向/barcode reads

方法同上"EMP protocol" single-end fastq

3. Casava 1.8 single-end demultiplexed fastq

[Casava 1.8 demultiplexed][http://illumina.bioinfo.ucr.edu/ht/documentation/data-analysis-docs/CASAVA-FASTQ.pdf/view] (single-end) 格式, 针对每个样本各一个`fastq.gz`文件. 文件名称包含样本identifier, `L2S357_15_L001_R1_001.fastq.gz`, 其下划线分隔的文件名称为:

- 样本identifier
- barcode序列或barcode identifier
- lane数目
- read方向(i.e. 只有R1, 因为单端reads)
- the set number

casava-18-single-end-demultiplexed.zip: https://data.qiime2.org/2019.7/tutorials/importing/casava-18-single-end-demultiplexed.zip

```
unzip -q casava-18-single-end-demultiplexed.zip
```

```
qiime tools import \ --type 'SampleData[SequencesWithQuality]' \ --input-path casava-18-single-end-demultiplexed \ --input-format CasavaOneEightSingleLanePerSampleDirFmt \ --output-path demux-single-end.qza
```

4. Casava 1.8 paired-end demultiplexed fastq

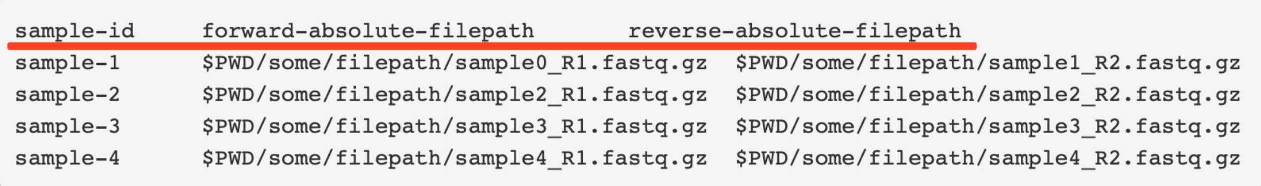
格式和方法同"Casava 1.8 single-end demultiplexd fastq"

5. "Fastq manifest" formats

如果不是EMP或Casava 格式, 那么要导入QIIME2, 首先就要构建'manifest file', 然后使用`qiime tools import`命令导入.

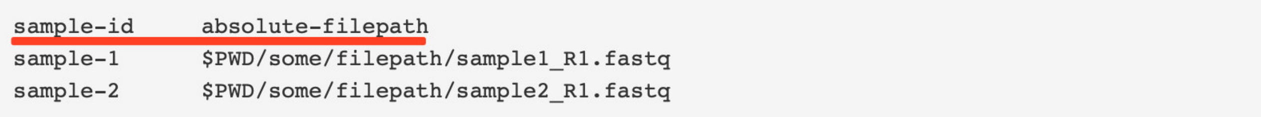
首先构建一个text文件, 'manifest file', 对应样本identifiers到`fastq.gz`或`fastq`绝对文件路径(包含样本的序列和质量数据, fastq文件). 该manifest文件同样指明了`fastq.gz`或`fastq`文件中reads的方向.

manifest文件是tab-分隔的文本文件(i.e. `.tsv`). 第一列定义了样本的ID, 第二列定义了正向reads文件的绝对路径, 第三列定义了对应反向reads文件的绝对路径(该绝对路径可包含环境变量, `$HOME/$PWD`).



```
sample-id     forward-absolute-filepath       reverse-absolute-filepath
sample-1      $PWD/some/filepath/sample0_R1.fastq.gz  $PWD/some/filepath/sample1_R2.fastq.gz
sample-2      $PWD/some/filepath/sample2_R1.fastq.gz  $PWD/some/filepath/sample2_R2.fastq.gz
sample-3      $PWD/some/filepath/sample3_R1.fastq.gz  $PWD/some/filepath/sample3_R2.fastq.gz
sample-4      $PWD/some/filepath/sample4_R1.fastq.gz  $PWD/some/filepath/sample4_R2.fastq.gz
```

单端fastq文件的manifest文件如下:



```
sample-id     absolute-filepath
sample-1      $PWD/some/filepath/sample1_R1.fastq
sample-2      $PWD/some/filepath/sample2_R1.fastq
```

SingleEndFastqManifestPhred33V2

se-33.zip: https://data.qiime2.org/2019.7/tutorials/importing/se-33.zip

se-33-manifest: https://data.qiime2.org/2019.7/tutorials/importing/se-33-manifest

`unzip -q se-33.zip`

```
qiime tools import \ --type 'SampleData[SequencesWithQuality]' \ --input-path se-33-manifest \ --output-path single-end-demux.qza \ --input-format SingleEndFastqManifestPhred33V2
```

PairedEndFastqManifestPhred64V2

pe-64.zip: https://data.qiime2.org/2019.7/tutorials/importing/pe-64.zip

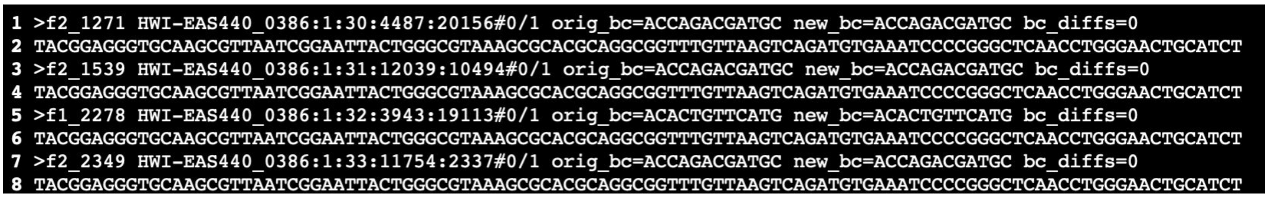
pe-64-manifest: https://data.qiime2.org/2019.7/tutorials/importing/pe-64-manifest

`unzip -q pe-64.zip`

```
qiime tools import \ --type 'SampleData[PairedEndSequencesWithQuality]' \ --input-path pe-64-manifest \ --output-path paired-end-demux.qza \ --input-format PairedEndFastqManifestPhred64V2
```

6. Sequences without quality information (i.e. FASTA)

支持导入QIIME1 `seqs.fna` 文件格式, 每个记录包含2行内容: header和sequence. 每个sequence必须仅且占一行. header中的ID格式为: `<sample-id>_<seq-id>`. `<sample-id>`为样本sequence所属的样本名称(ID), `<seq-id>`为样本内的sequence的名称(ID).



```
>f2_1271 HWI-EAS440_0386:1:30:4487:20156#0/1 orig_bc=ACCAGACGATGC new_bc=ACCAGACGATGC bc_diffs=0
TACGGAGGGTGCAAGCGTTAATCGGAATTACTGGGCGTAAAGCGCACGCAGGCGGTTTGTTAAGTCAGATGTGAAATCCCCGGGCTCAACCTGGGAACTGCATCT
>f2_1539 HWI-EAS440_0386:1:31:12039:10494#0/1 orig_bc=ACCAGACGATGC new_bc=ACCAGACGATGC bc_diffs=0
TACGGAGGGTGCAAGCGTTAATCGGAATTACTGGGCGTAAAGCGCACGCAGGCGGTTTGTTAAGTCAGATGTGAAATCCCCGGGCTCAACCTGGGAACTGCATCT
>f1_2278 HWI-EAS440_0386:1:32:3943:19113#0/1 orig_bc=ACACTGTTCATG new_bc=ACACTGTTCATG bc_diffs=0
TACGGAGGGTGCAAGCGTTAATCGGAATTACTGGGCGTAAAGCGCACGCAGGCGGTTTGTTAAGTCAGATGTGAAATCCCCGGGCTCAACCTGGGAACTGCATCT
>f2_2349 HWI-EAS440_0386:1:33:11754:2337#0/1 orig_bc=ACCAGACGATGC new_bc=ACCAGACGATGC bc_diffs=0
TACGGAGGGTGCAAGCGTTAATCGGAATTACTGGGCGTAAAGCGCACGCAGGCGGTTTGTTAAGTCAGATGTGAAATCCCCGGGCTCAACCTGGGAACTGCATCT
```

7. Per-feature unaligned sequence data (i.e., representative fasta sequences)

为比对的序列数据为FASTA格式文件, 包含为比对的DNA序列(i.e. , 不包含`-`或`.`符号). 该序列可以包含degenerate核酸字符, 例如`N`.

sequences.fna: https://data.qiime2.org/2019.7/tutorials/importing/sequences.fna

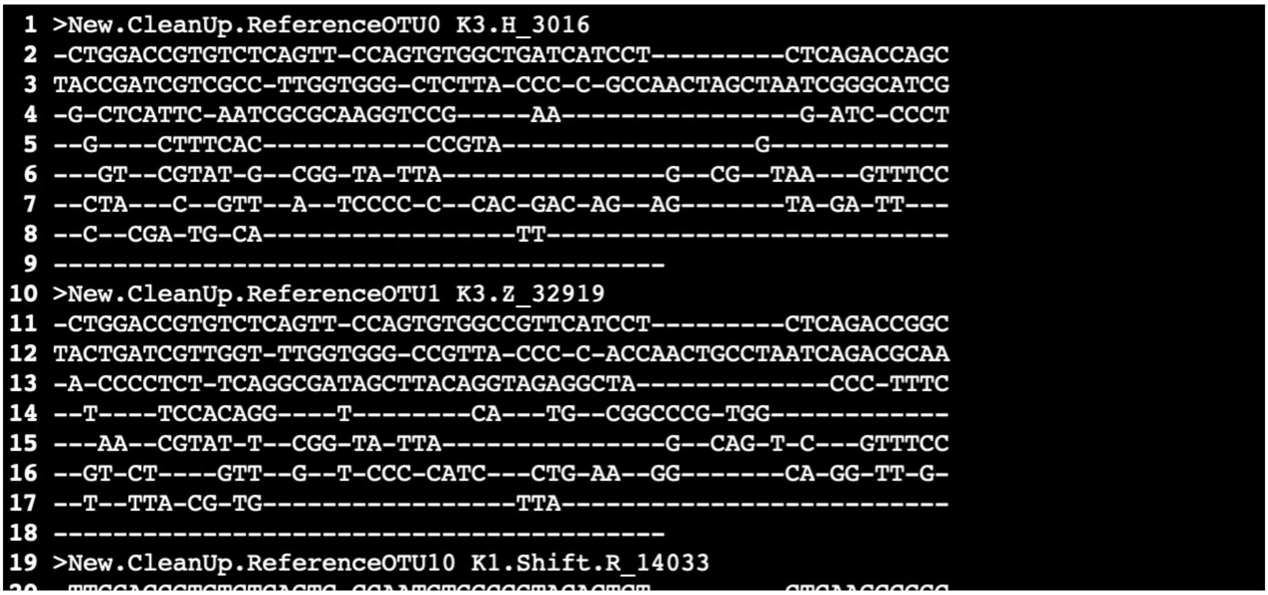
```
qiime tools import \ --input-path sequences.fna \ --output-path sequences.qza \ --type 'FeatureData[Sequence]'
```

8. Per-feature aligned sequence data (i.e. , aligned representative FASTA sequences)

比对了的序列数据是通过包含了DNA序列比对到其他序列的FASTA格式文件导入而来的. 所有比对的序列必须一样长, 该序列可以包含`N`.

aligned-sequences.fna: https://data.qiime2.org/2019.7/tutorials/importing/aligned-sequences.fna

```
qiime tools import \ --input-path aligned-sequences.fna \ --output-path aligned-sequences.qza \ --type 'FeatureData[AlignedSequence]'
```



```
>New.CleanUp.ReferenceOTU0 K3.H_3016
-CTGGACCGTGTCTCAGTT-CCAGTGTGGCTGATCATCCT---------CTCAGACCAGC
TACCGATCGTCGCC-TTGGTGGG-CTCTTA-CCC-C-GCCAACTAGCTAATCGGGCATCG
-G-CTCATTC-AATCGCGCAAGGTCCG-----AA----------------G-ATC-CCCT
--G----CTTTCAC-----------CCGTA-----------------G------------
---GT--CGTAT-G--CGG-TA-TTA---------------G--CG--TAA---GTTTCC
--CTA---C--GTT--A--TCCCC-C--CAC-GAC-AG--AG-------TA-GA-TT---
--C--CGA-TG-CA-----------------TT---------------------------
-----------------------------------------
>New.CleanUp.ReferenceOTU1 K3.Z_32919
-CTGGACCGTGTCTCAGTT-CCAGTGTGGCCGTTCATCCT---------CTCAGACCGGC
TACTGATCGTTGGT-TTGGTGGG-CCGTTA-CCC-C-ACCAACTGCCTAATCAGACGCAA
-A-CCCCTCT-TCAGGCGATAGCTTACAGGTAGAGGCTA-------------CCC-TTTC
--T----TCCACAGG----T--------CA---TG--CGGCCCG-TGG------------
---AA--CGTAT-T--CGG-TA-TTA---------------G--CAG-T-C---GTTTCC
--GT-CT----GTT--G--T-CCC-CATC---CTG-AA--GG-------CA-GG-TT-G-
--T--TTA-CG-TG-----------------TTA--------------------------
-----------------------------------------
>New.CleanUp.ReferenceOTU10 K1.Shift.R_14033
```

9. Feature table data

导入pre-processed feature table

[BIO v1.0.0][http://biom-format.org/documentation/format_versions/biom-1.0.html]

feature-table-v100.biom: https://data.qiime2.org/2019.7/tutorials/importing/feature-table-v100.biom

```
qiime tools import \ --input-path feature-table-v100.biom \ --type 'FeatureTable[Frequency]' \ --input-format BIOMV100Format \ --output-path feature-table-1.qza
```

[BIOM v2.1.0][http://biom-format.org/documentation/format_versions/biom-2.1.html]

feature-table-v210.biom: https://data.qiime2.org/2019.7/tutorials/importing/feature-table-v210.biom

10. Phylogenetic trees

Phylogenetic trees 是从newick 格式文件导入而来的.

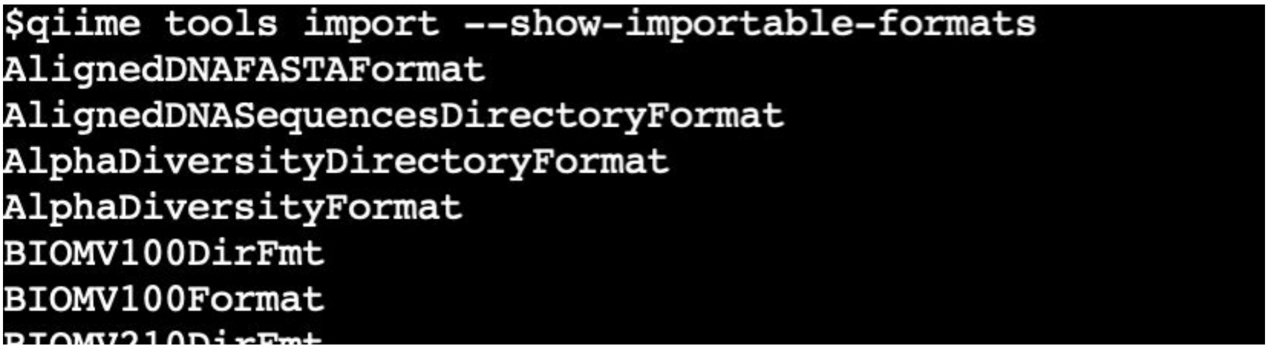
unrooted-tree.tre: https://data.qiime2.org/2019.7/tutorials/importing/unrooted-tree.tre

```
qiime tools import \ --input-path unrooted-tree.tre \ --output-path unrooted-tree.qza \ --type 'Phylogeny[Unrooted]'
```

11. Other data types

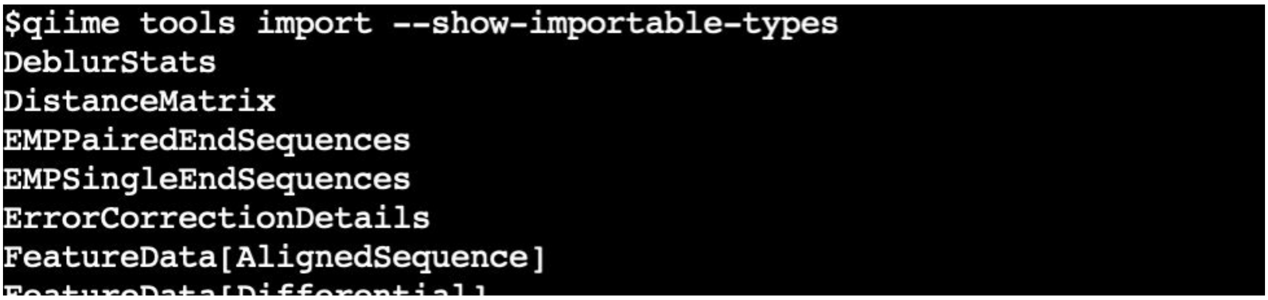
查看帮助:

```
qiime tools import \ --show-importable-formats
```



```
$qiime tools import --show-importable-formats
AlignedDNAFASTAFormat
AlignedDNASequencesDirectoryFormat
AlphaDiversityDirectoryFormat
AlphaDiversityFormat
BIOMV100DirFmt
BIOMV100Format
BIOMV210DirFmt
```

```
qiime tools import \ --show-importable-types
```



```
$qiime tools import --show-importable-types
DeblurStats
DistanceMatrix
EMPPairedEndSequences
EMPSingleEndSequences
ErrorCorrectionDetails
FeatureData[AlignedSequence]
FeatureData[Differential]
```