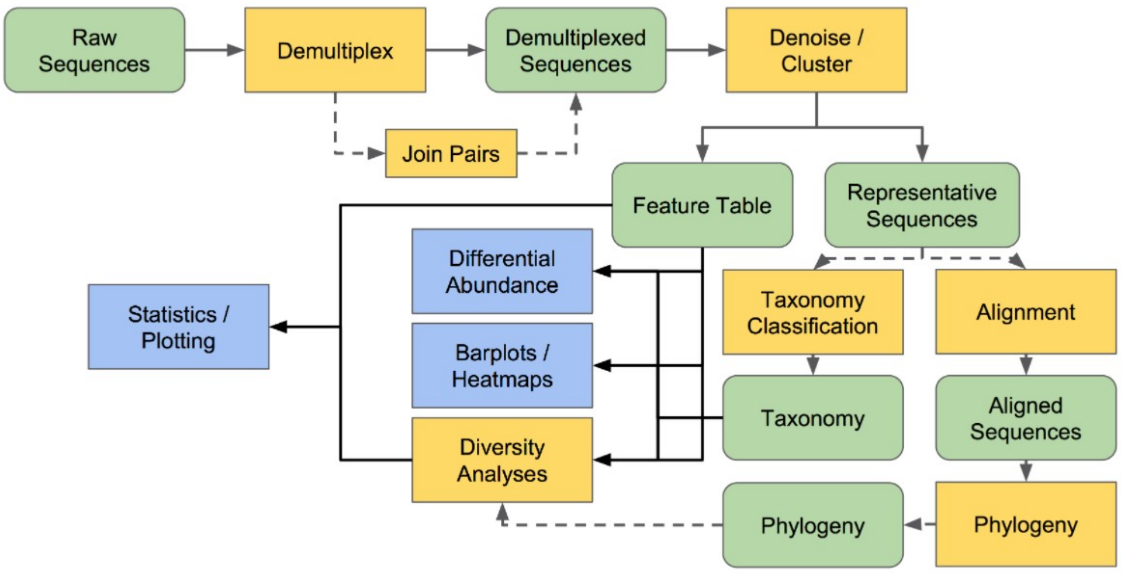
Qiime2官方教程

介绍：

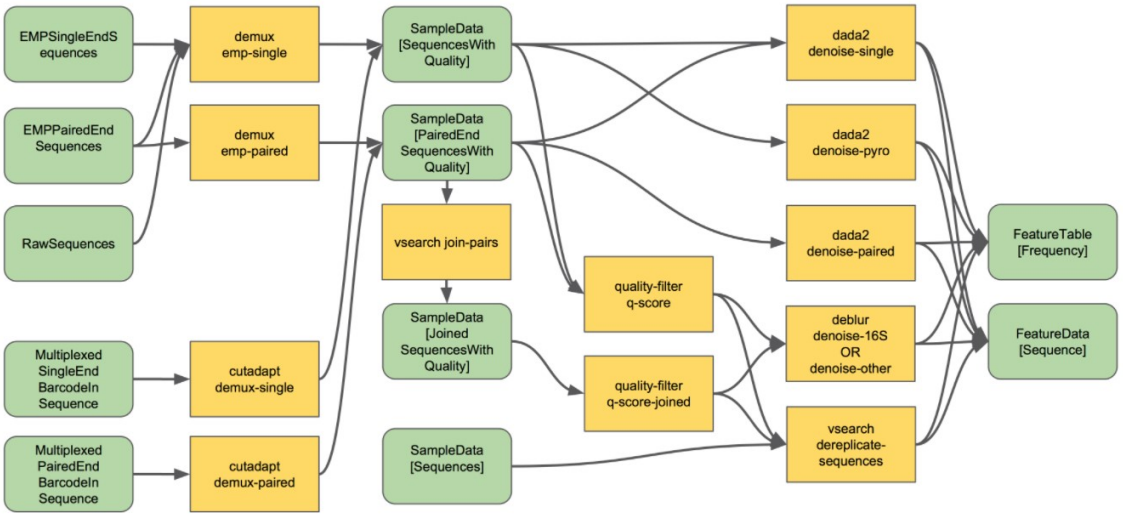
> Remember, many paths lead from the foot of the mountain, but at the peak we all gaze at the same moon.

- 总体概念

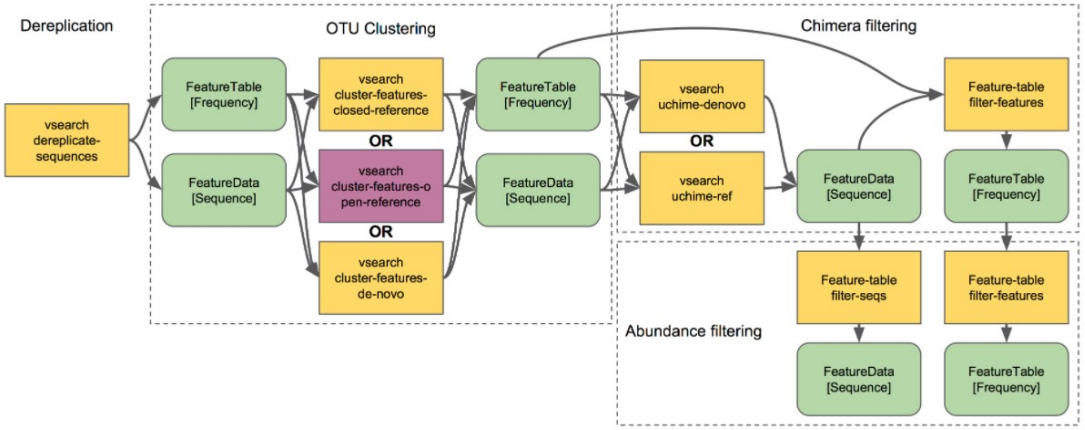


- Demultiplexing：分用
  测序仪的一个lane可以运行几百上千的samples，如何知道每个reads的sample什

么？通过barcode(index)。我们知道每个样的barcode(这个信息再metadata中)，然后将reads上的barcode比对，就可以知道每个reads来自的样。这个过程称为demultiplexing。

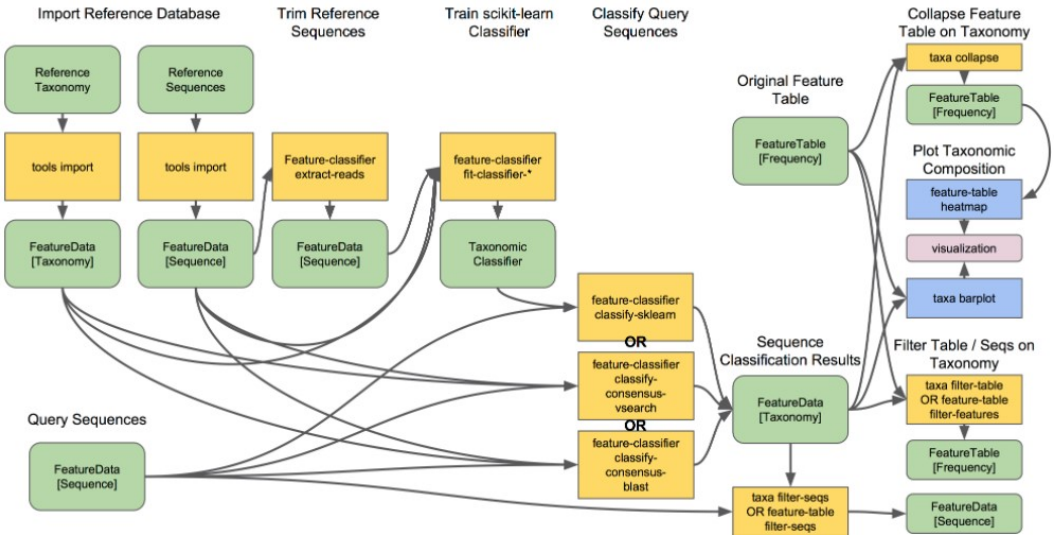


- Denoising and clustering：降噪和聚类
  - 降噪：消除序列的噪音以删除更正嘈杂的reads，qiime会保留重复序列
    - DADA2：
    - Deblur：
  - 聚类：将相似度达到97%重复reads转成一个reads，也叫OTU picking



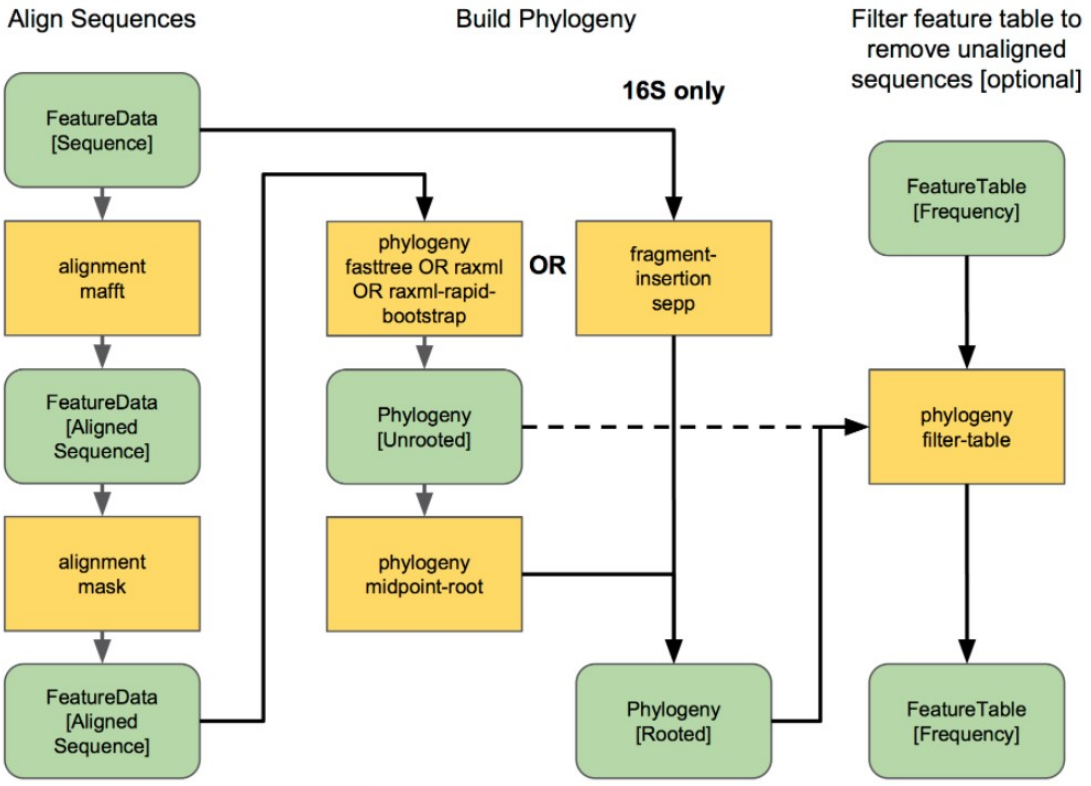
- The feature table：特征表（Qiime2分析的核心）：
  - 降噪和聚类最终会得到 => **FeatureTable[Frequency]**和**FeatureData[Sequence]**，这两张表对下游分析非常重要
- Taxononmy classification and taxonomic analysis：分类学分类和分类学分析
  - 从样品中鉴定到那些生物，属于那些属或者种，他们是不是人类病原体。
  - 将序列比对到参考数据库，仅仅找最接近的比对还不够，因为**有些序列相近的reads可能有不同的分类学分类**
  - q2-feature-classifier包括三种不同的分类方法：

        - based on alignment methods : `classify-consensus-blast`，`classify-consensus-vsearch` 直接对比到参考数据库
        - based on Machine-learning： `classify-sklearn` 需要对分类器进行训练，学习那种feature最能区分每个分类组，分类器是**reference database**和**marker gene specific**，只需要训练一次，Qiime2有预先训练好的。
    - 分类后可以做的事情：对sequences分类后可以做的
        - 将具有所有features中具有相同taxonomic assignment合并成一个feature => taxonomic assignment就成了新的feature表的新ID，这张表可以和之前的表做同样的事情。。。。。后面看不懂
        - 可视化一下分类的组成，看看每个样品中各种分类的丰度
        - 过滤feature table和representative sequences中某些分类组，去除已知的污染物或者宿主的DNA包括(线粒体，叶绿体等等)，这对分析特定组的深入分析很有效。



- sequence alignment and phylogeny building：序列比对和系统发育树构建
    - 特征之间的很多多样性分析需要依赖系统发育树，如果有系统发育的marker(16s etc)，就可以将这些序列进行比对，以评估每个feature间的系统发

生关系，系统发育树可以用于下游的Unifrac distance analysis



- diversity analysis
  - questions:
    - 每个sample中有多少个物种/OTUs/ASVs
    - 每个sample中的系统发育树的多样性?
    - 单个samples间的差异，组之间的差异
    - 那些因素(PH，elevation海拔，血压，身体部位，宿主)与微生物组成和生物多样性差异有关
  - 这些问题可以用Alpha- and beta-diversity analysis
    - Alpha analysis：衡量的单个sample内的多样性，alpha多样性是用来测量群落内生物种类数量以及生物种类间相对多度的一种测量，反映了群落内物种间通过竞争资源或利用同种生境而产生的共存结果。如单个样品中包含的物种种类，群落内每个物种的分布和丰度情况等。常用的alpha多样性指数：*Chao1丰富度估计量(Chao1 richness estimator)*，*香农多样性指数(Shannon diversity index)*，*辛普森多样性指数(Simpson diversity index)*
      - **物种丰富度指数**：Margalef丰富度指数，Menhnick丰富度指数
      - **物种均匀度指数**：Pielou均匀度指数，Sheldon均匀度指数，Hill均匀度指数，Heip均匀度指数，Alatalo均匀度指数
      - **物种多样性指数**：Shannon-Wienner多样性指数，Simpson多样性指数，Hill多样性指数以及中间相遇概率PIE

- Beta analysis：衡量样本间的多样性：生态系之间的物种多样性，包括分类单位的比较，即衡量群落之间的差别，beta多样性不仅描述生境内生物种类的数量，同时也考虑到这些种类的相同性及其彼此之间的位置。说白了是不同样品之间比较或同一样品不同条件的比较。常用指有**Whittaker指数**，**Cody指数**，**Wilson指数**，**Shmida指数**
  - 指示生境物种隔离的程度
  - beta多样性和alpha多样性的测定值可以用来比较不同地段的生境多样性
  - beta多样性和alpha多样性一起构成了总体多样性或一定地段的生物异质性
  - 分析包括：PCoa分析（主坐标，加入了物种进化关系），PCA分析（out丰富度的不同），NMDS分析等
- statistically test whether apha diversity is different between groups of samples

关于metadata

moving picture tutorial(for single-end data)

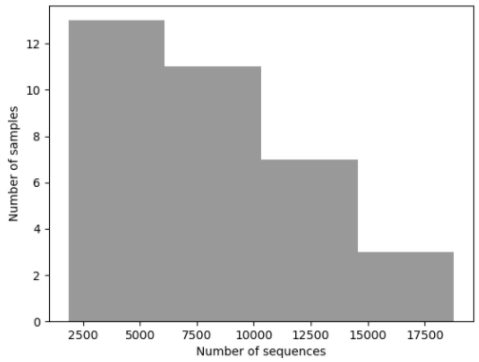
- 做了个啥：两个人，四个身体部位（肠子，左右掌，舌头），五个时间点（差不多隔一个月），第一个人用抗生素处理。
- 单端测序，有barcode

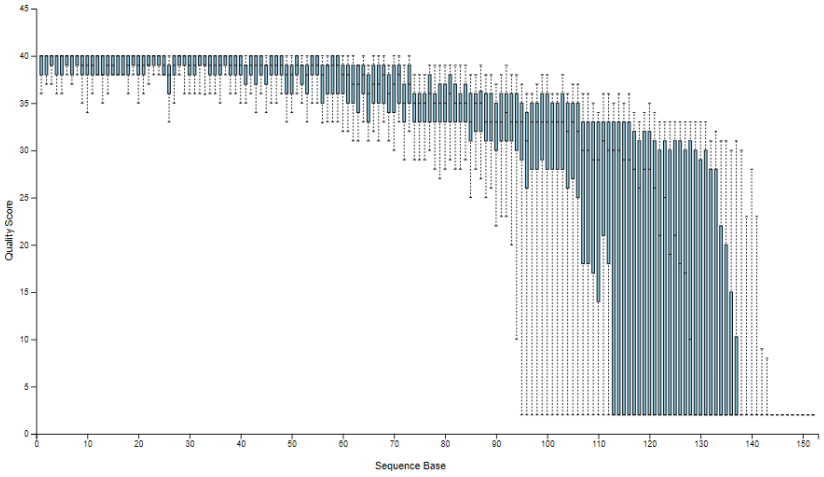
01.导入Qiime2创建qza对象

- 注意输入的是文件夹，将barcode和fastq序列放在同一个文件夹下本测试放在`emp-single-end-sequences`，还有一点这里的所有fastq文件最好是压缩的，不然会报错，这种写死的方式就这点不好，如果给manifest.tsv如果不是压缩文件也不要紧。
- 输出是`demux.qza`

```
$ head barcodes.fastq
@HWI-EAS440_0386:1:23:17547:1423#0/1
ATGCAGCTCAGT
+
IIIIIIIIIIIH
@HWI-EAS440_0386:1:23:14818:1533#0/1
CCCCTCAGCGGC
+
```

```
DDD@D?@B<<+/

$ head sample-metadata.tsv
sample-id barcode-sequence body-site year month day
#q2:types categorical categorical numeric numeric numeric
L1S8 AGCTGACTAGTC gut 2008 10 28
L1S57 ACACACTATGGC gut 2009 1 20
L1S76 ACTACGTGTGGT gut 2009 2 17

$ head sequences.fastq
@HWI-EAS440_0386:1:23:17547:1423#0/1
TACGNAGGATCCGAGCGTTATCCGGATTTATTGGGTTTAAAGGGAGCGTAGATGGATGTTTAAGTCAG
TTGTGAAAGTTTGCGGCTCAACCGTAAAATTGCAGTTGATACTGGATATCTTGAGTGCAGTTGAGGCA
GGGGGGGATTGGTGTG
+
IIIE)EEEEEEEEGFIIGIIIHIHHGIIIGIIHHHGIIHGHEGDGIFIGEHGIHHGH

$ qiime tools import \
      --type EMPSingleEndSequences \ #指定类型，单端
      --input-path  emp-single-end-sequences \ # 测序数据路径(barcode和测序数据
      --output-path emp-single-end-sequences.qza #输出的是Qiime的对象

$ qiime demux emp-single \ #分解
      --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

$ qiime demux summarize \
      --i-data demux.qza \
      --o-visualization demux.qzv #可视化看一下质量

# 深入理解一下这个demux.qza对象，压缩后看到其实就是把barcode和测序数据整合在一起放
# 估计是方便软件调用符合软件的固用格式可能压缩了，因为整合了这么多文件大小反而减小，
# 顺路看了一下如果是双端的话，里面包含一个manifest的表和我们给他的有一点不一致，
# 每个文件后面多了个001不知是为何
# 分解是根据metadata样本的barcode进行分解，结果导致qza的data里从之前一个测序按找样
```





02.降噪和构建特征矩阵

- DADA2和Deblur两种方法都可以，最终会得到两张重要的表，作者建议两个都做选择最好的一个，本次使用是的DADA2
  - FeatureTable[Frequency]（唯一seq在每个样本中的counts数）
  - FeatureData[Sequence]（唯一seq的序列，对应FeatureTable的标识符）

```
#===> DADA2降噪:
# DADA2检验和矫正illumina扩增子序列，会过滤掉phiXreads(illumina经常产生的错误gene
# 同时会过滤chimeric(嵌合序列)
$ qiime dada2 denoise-single \ # 指明单端
      --i-demultiplexed-seqs demux.qza \
      --p-trim-left 0 \ #修建
      --p-trunc-len 120 \ #截取
      --o-representative-sequences rep-seqs-dada2.qza \ #FeatureData[Sequence
      --o-table table-dada2.qza \ #FeatureTable[Frequency]
      --o-denoising-stats stats-dada2.qza #统计表

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

#===> Deblur
```

```
# 有点不太懂原理
# 第一步：按质量过滤序列
$ qiime quality-filter q-score \
     --i-demux demux.qza \
     --o-filtered-sequences demux-filtered.qza \ #过滤后的demux.qza
     --o-filter-stats demux-filter-stats.qza #统计
# 第二步：获得代表序列，需要选择一个trim-length开发者建议截取中等质量的开始下降的那
# 大概115-130之间按照这个套数据来讲
$ qiime deblur denoise-16S \
    --i-demultiplexed-seqs demux-filtered.qza \ #过滤后的demux.qza
    --p-trim-length 120 \ #截取
    --o-representative-sequences rep-seqs-deblur.qza \ #特征序列
    --o-table table-deblur.qza \ #特征表
    --p-sample-stats \ #?
    --o-stats deblur-stats.qza # 统计

$ qiime metadata tabulate \  #降噪后的统计
    --m-input-file demux-filter-stats.qza \ #过滤后的demux.qza
    --o-visualization demux-filter-stats.qzv
$ qiime deblur visualize-stats \ # 特征表可视化
    --i-deblur-stats deblur-stats.qza \ #统计
    --o-visualization deblur-stats.qzv
```

- stats-dada2.qzv

sample-id #q2:types	input numeric	filtered numeric	percentage of input passed filter numeric	denoised numeric	non-chimeric numeric	percentage of input non-chimeric numeric
L1S105	11340	8571	75.58	8476	7788	68.68
L1S140	9738	7677	78.84	7605	7163	73.56
L1S208	11337	9261	81.69	9156	8162	71.99
L1S257	8216	6705	81.61	6627	6405	77.96
L1S281	8907	7067	79.34	6983	6630	74.44
L1S57	11752	9299	79.13	9260	8716	74.17
L1S76	10101	8395	83.11	8339	7871	77.92
L1S8	12388	7663	61.86	7628	7037	56.8
L2S155	9263	4112	44.39	3932	3932	42.45
L2S175	10692	4546	42.52	4386	4386	41.02
L2S204	7299	3379	46.29	3199	3161	43.31

- demux-filter-stats.qzv

sample-id #q2:types	total-input-reads numeric	total-retained-reads numeric	reads-truncated numeric	reads-too-short-after-truncation numeric	reads-exceeding-maximum-ambiguous-bases numeric
L1S105	11340	9232	10782	2066	42
L1S140	9738	8585	9459	1113	40
L1S208	11337	10149	10668	1161	27
L1S257	8216	7302	7672	876	38
L1S281	8907	7764	8346	1118	25
L1S57	11752	10001	11002	1717	34
L1S76	10101	8985	9679	1092	24
L1S8	12388	8434	12037	3916	38
L2S155	9263	5066	8934	4169	28
L2S175	10692	5575	10217	5092	25

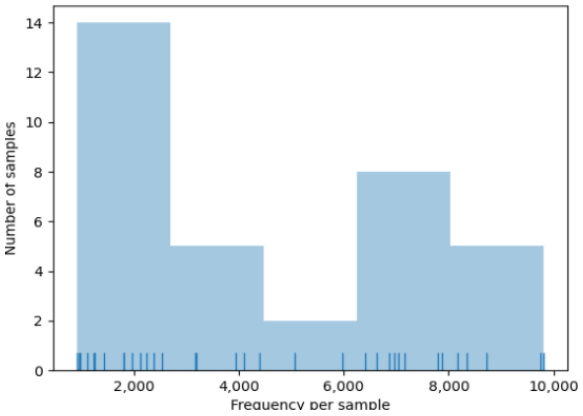
- deblur-stats.qzv

	sample-id	reads-raw	fraction-artifact-with-minsize	fraction-artifact	fraction-missed-reference	unique-reads-derep	reads-derep	unique-reads-deblur	reads-deblur	unique-reads-hit-artifact	reads-hit-artifact	unique-reads-chimeric	reads-chimeric	unique-reads-hit-reference	reads-hit-reference	unique-reads-missed-reference	reads-missed-reference
0	L3S360	1343	0.517498	0.0	0.011745	119	648	112	602	0	0	1	6	74	514	3	7
1	L2S222	4459	0.498094	0.0	0.019076	327	2238	288	2000	0	0	4	8	147	1603	3	38
2	L2S309	1905	0.457743	0.0	0.003185	124	1033	99	942	0	0	0	0	76	895	1	3
3	L3S341	1293	0.438515	0.0	0.000000	95	726	86	675	0	0	0	0	78	653	0	0
4	L2S204	4350	0.428966	0.0	0.012845	236	2484	152	2103	0	0	1	1	106	1969	2	27
5	L2S357	3100	0.419032	0.0	0.000000	149	1801	87	1554	0	0	0	0	75	1533	0	0
6	L3S294	1523	0.401182	0.0	0.002398	82	912	65	836	0	0	1	2	52	800	1	2
7	L3S313	1340	0.391045	0.0	0.000000	85	816	69	747	0	0	1	1	66	741	0	0
8	L2S240	7110	0.390717	0.0	0.000000	253	4332	78	3578	0	0	9	17	59	3535	0	0

- 特征表和特征序列的可视化(DADA的结果)

```
$ 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
```

- table.qzv：特征表



Sampling Depth

0

(zero implies no even sampling)

Retained 153,884 (100.00%) features in 34 (100.00%) samples at the specifed sampling depth.

Sample ID	Feature Count
L4S137	9820
L4S63	9744
L1S57	8716
L4S112	8340
L1S208	8162
L1S76	7871
L1S105	7788
L1S140	7163
L1S8	7037
L6S93	6953
L6S20	6857
L1S281	6630

- rep-seqs.qzv特征序列

Feature ID	Sequence Length	Sequence
4b5eeb300368260019c1fbc7a3c718fc	120	TACGGAGGATCCGAGCGTTATCCGGATTTATTGGGTTTAAAGGGAGCGTAGATGGATGTTTAAGTCAGTTGTGAAAGTTTGCGGCT
fe30ff0f71a38a39cf1717ec2be3a2fc	120	TACGTAGGGTGCGAGCGTTAATCGGAATTACTGGGCGTAAAGCGAGCGCAGACGGTTACTTAAGCAGGATGTGAAATCCCCGGGCT
d29fe3c70564fc0f69f2c03e0d1e5561	120	TACGTAGGTCCCGAGCGTTGTCCGGATTTATTGGGCGTAAAGCGAGCGCAGGCGGTTAGATAAGTCTGAAGTTAAAGGCTGTGGCT
868528ca947bc57b69ffdf83e6b73bae	120	TACGGAGGATCCGAGCGTTATCCGGATTTATTGGGTTTAAAGGGAGCGTAGATGGATGTTTAAGTCAGTTGTGAAAGTTTGCGGCT
154709e160e8cada6bfb21115acc80f5	120	TACGGAGGATCCGAGCGTTATCCGGATTTATTGGGTTTAAAGGGAGCGTAGGTGGATTGTTAAGTCAGTTGTGAAAGTTTGCGGCT
1d2e5f3444ca750c85302ceee2473331	120	TACGGAGGGTGCGAGCGTTAATCGGAATAACTGGGCGTAAAGGGCACGCAGGCGGTGACTTAAGTGAGGTGTGAAAGCCCCGGGCT
0305a4993ecf2d8ef4149fdfc7592603	120	TACGGAGGATCCGAGCGTTATCCGGATTTATTGGGTTTAAAGGGAGCGTAGGCGGACGCTTAAGTCAGTTGTGAAAGTTTGCGGCT
cb2fe0146e2fbcb101050edb996a0ee2	120	TACGTAGGTGGCAAGCGTTATCCGGAATTATTGGGCGTAAAGCGCGCGTAGGCGGTTTTTTAAGTCTGATGTGAAAGCCCACGGCT

03.构建系统发育树用于多样性分析

- Generate a tree for phylogenetic diversity analyses
  - Qiime2支持：
    - Faith’s Phylogenetic Diversity
    - weighted and unweighted UniFrac
- 首先mafft对特征序列进行多序列比对，创建一个FeatureData[AlignedSeqence]，接着，过滤掉那些高可变位置，这些位置被认为会给系统发育树增加噪音。然后，FastTree通过之前的masked alignment创建一个无根树，因此最后一步将最远的两个根之间的中点作为这个无根树的根。结果可以在mega里面看看。

```
$ qiime phylogeny align-to-tree-mafft-fasttree \
    --i-sequences rep-seqs.qza \ #特征序列
    --o-alignment aligned-rep-seqs.qza \ #mafft的结果
    --o-masked-alignment masked-aligned-rep-seqs.qza \ #masks一下
    --o-tree unrooted-tree.qza \ #fasttree生成无根树
    --o-rooted-tree rooted-tree.qza #生成有根树
```

03.alpha和beta多样性分析(what is diversity metrics?)

- 首先需要对特征表稀疏到用户指定的深度，因此有一个关键的参数`--p-sampling-depth`，用来均匀深度（类似于标准化吧），这种稀疏的方式是随机抽样不放回，小于抽样数的样将被舍弃，所以尽可能写大，根据特征表的.qzv可以看出，L3S313这个样本相对于特征序列总数较小的样本，他是大的。这个样本一下的样本将被舍弃，从一个元数据类别中丢失不成比例的样本数并不理想，但是我们在此处丢去的样本足够少是对总序列分析和总样本分析的最佳折中。下游分析都用的是这个稀疏后的特征表
- alpha diversity
  - Shannon’s diversity index (a quantitative measure of community richness)
  - Observed Features (a qualitative measure of community richness)
  - Faith’s Phylogenetic Diversity (a qualitiative measure of community richness that incorporates phylogenetic relationships between the features)
  - Evenness (or Pielou’s Evenness; a measure of community evenness)

```
# diversity metrics多样性指标
$ qiime diversity core-metrics-phylogenetic \
      --i-phylogeny rooted-tree.qza \ #有根树
```

```
    --i-table table.qza \ #特征表
    --p-sampling-depth 1103 \ #深度肉眼看
    --m-metadata-file sample-metadata.tsv \ #metadata
    --output-dir core-metrics-results #一系列稀疏后的结果，看不太懂，大概都是看
```

- **这一步多样性分析的输出结果是整个实验输出的核心**结果如下：

> 物种丰富度：有多少种不同的物种
>
> 物种均匀度：不同物种的占比

  - rarefied_table：稀疏后的特征表，可以看到丢掉了3个样本，不太明白为什么要丢弃样本
  - distance_matrix：用于beta多样性分析
    - weighted_unifrac_distance_matrix.qza #考虑丰度
    - unweighted_unifrac_distance_matrix.qza #不考虑丰度
    - bray_curtis_distance_matrix #
    - jaccard_distance_matrix #
  - vector：用于alpha多样性分析
    - faith_pd_vector.qza #考虑物种间进化关系指数
    - shannon_vector.qza # 考虑物种和丰度
    - evenness_vector.qza # 考虑物种和丰度
    - observed_feature_vecotr.qza #考虑物种和丰度
  - results:
    - bray_curtis_pcoa_results.qza
    - jaccard_pcoa_results.qza
    - weighted_unifrac_pcoa_results.qza
    - unweighted_unifrac_pcoa_results.qza
  - visualizations：对应四种多样性分析的方法
    - unweighted_unifrac_emperor.qzv
    - jaccard_emperor.qzv
    - bray_curtis_emperor.qzv
    - weighted_unifrac_emperor.qzv
- after computing **diversity metrics**, we can begin to explore the mirobial composition of the sample in the context of sample metadata

- test for associations between **categorical metadata columns** and **alpha diversity data**，这里用了alpha的
  - Faith Phylogenetic Diversity
  - evenness metrics

```
# 注意一个细节，这里的分析是看metadata分来变量内部是否存在显著alpha显著差异，
# 因此这里是忽略了连续变量，如果想看连续变量和alpha之间的关系可以用`qiime diversit
$ 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

# 连续型变量与物种丰富度
$ qiime diversity alpha-correlation \
    --i-alpha-diversity core-metrics-results/faith_pd_vector.qza \
    --m-metadata-file sample-metadata.tsv \
    --o-visualization faith-pd-significance.qzv
```

- next we'll analyze sample composition in the context of catrgorical metadata using PERMANOVA using `beta-group-significance`
  - 同样没有演示连续变量：`metadata distance-matrix`，`qiime diversity mantel`，`qiime diversity bioenv`

```
$ qiime diversity beta-group-significance \
      --i-distance-matrix core-metrics-results/unweighted_unifrac_distance_ma
      --m-metadata-file sample-metadata.tsv \
      --m-metadata-column body-site \ # 指定一列
      --o-visualization core-metrics-results/unweighted-unifrac-body-site-sig
      --p-pairwise

$ qiime diversity beta-group-significance \
      --i-distance-matrix core-metrics-results/unweighted_unifrac_distance_ma
      --m-metadata-file sample-metadata.tsv \
      --m-metadata-column subject \ # 指定一列
      --o-visualization core-metrics-results/unweighted-unifrac-subject-group
```

```
      --p-pairwise
```

04.物种分类 taxonomic composition of the samples

- 对FeatureData[Sequence]进行物种分类
- 命令 `q2-feature-classifier`

```
$ wget https://data.qiime2.org/2020.8/common/gg-13-8-99-515-806-nb-classifier
$ qiime feature-classifier classify-sklearn \
      --i-classifier gg-13-8-99-515-806-nb-classifier.qza \
      --i-reads rep-seqs.qza \
      --o-classification taxonomy.qza

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

- 自己训练：
  - 引物，截了多少

Atacama soils tutorial(for paired-end data)

01.导入Qiime2，并可视化

- 类似于创建Qiime2的对象
- input：
  - 测序fastq数据
  - manifest.tsv
- output:
  - demux_seqs.qza：类似于Qiime2的类吧
- command:

```
$ qiime tools import \
      --type "SampleData[SequencesWithQuality]" \
      --input-format SingleEndFastqManifestPhred33V2 \
      --input-path ./manifest.tsv \
      --output-path ./demux_seqs.qza #对象名字

```

```
$ qiime demux summarize \
      --i-data ./demux_seqs.qza \ #对象
      --o-visualization ./demux_seqs.qzv #.qzv为可视化对象

$ head manifest.tsv #
sample-id       absolute-filepath
recip.220.WT.OB1.D7     $PWD/demultiplexed_seqs/10483.recip.220.WT.OB1.D7_30_
recip.290.ASO.OB2.D1    $PWD/demultiplexed_seqs/10483.recip.290.ASO.OB2.D1_27
recip.389.WT.HC2.D21    $PWD/demultiplexed_seqs/10483.recip.389.WT.HC2.D21_1_
```

02.降噪（DADA2或者Deblur）

- 特征表feature table获得的两种方式**OTU**和**ASV**
- 输入：
  - qiime对象
- 输出：
  - dada2_stats.qza #过程统计
  - data2_table.qza #特征表 FeatureTable[Frequency]
  - data2_rep_set.qza #代表序列 FeatureData[Sequence]
- command

```
$ qiime dada2 denoise-single \ #数据为单端
      --i-demultiplexed-seqs ./demux_seqs.qza \
      --p-trunc-len 150 \
      --o-table ./dada2_table.qza \
      --o-representative-sequences ./dada2_rep_set.qza \
      --o-denoising-stats ./dada2_stats.qza

#===>
# 对于最终的到dada2_table.qza和dada2_rep_set.qza的深入一点理解:
# dada2讲测序数据过滤后，会生成特征表，这个特征表是按照唯一序列进行分组创建的，
# 因为他们等同于100% OTU，通常被称为sequence variants，质量更高，更准确地估计多样
# 每个样品中的生物分类，因此有两张表一张是特征表（包含特征序列的ID及每个样本中含特征
# 另外一张表是特征序列的具体fasta序列，两张表等长，说白了这个就很类似于表达矩阵，把
# 一个物种这个物种是通过DADA2找到的，成为SV，后面肯定需要标准化，需要对序列进行注释
$ unzip dada2_table.qza
$ biom convert -i feature-table.biom -o feature-table.txt --to-tsv && head fe
# Constructed from biom file
#OTU ID recip.220.WT.OB1.D7 recip.290.ASO.OB2.D1 recip.389.WT.HC2.D21 recip.3
```

```
04c8be5a3a6ba2d70446812e99318905 64.0 116.0 323.0 215.0 326.0
ea2b0e4a93c24c6c3661cbe347f93b74 287.0 346.0 551.0 402.0 370.0
1ad289cd8f44e109fd95de0382c5b252 57.0 351.0 173.0 304.0 235.0
3d9838f12f6ff5591dbadeb427a855f1 302.0 718.0 839.0 418.0 760.0
ac5402de1ddf427ab8d2b0a8a0a44f19 848.0 501.0 524.0 430.0 464.0
e5a43b018c81cedd19e9a5354d32d469 0.0 0.0 0.0 0.0 0.0
c4f9ef34bd2919511069f409c25de6f1 0.0 0.0 356.0 390.0 420.0
74ec9fe6ffab4ecff6d5def74298a825 11.0 27.0 76.0 244.0 100.0

$ unzip dada2_rep_set.qza && head dna-sequences.fasta
>04c8be5a3a6ba2d70446812e99318905
TACGGAGGATCCGAGCGTTATCCGGATTTATTGGGTTTAAAGGGAG
CGTAGGTGGACAGTTAAGTCAGTTGTGAAAGTTTGCGGCTCAACC
>ea2b0e4a93c24c6c3661cbe347f93b74
TACGGAGGATCCGAGCGTTATCCGGATTTATTGGGTTTAAAGGGAG
CGTAGGCGGACGCTTAAGTCAGTTGTGAAAGTTTGCGGCTCAACC

$ qiime metadata tabulate \ #可视化去噪后的信息
      --m-input-file ./dada2_stats.qza  \
      --o-visualization ./dada2_stats.qzv
$ qiime feature-table summarize \ # 特征表可视化
      --i-table table.qza \
      --o-visualization table.qzv \
      --m-sample-metadata-file sample-metadata.tsv #因为需要知道每个sample的ID
$ qiime feature-table tabulate-seqs \ #特征序列可视化
      --i-data rep-seqs.qza \
      --o-visualization rep-seqs.qzv
```

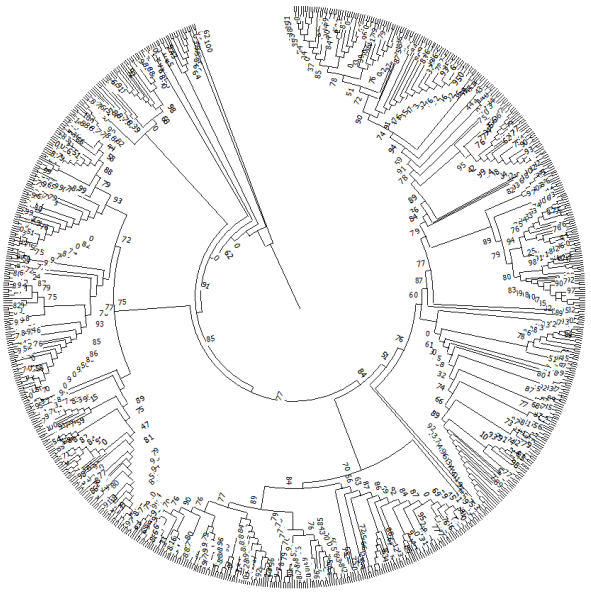
03.生成多样性的系统发育进化树

- 系统发育树为数据提供固有的结构，使我们考虑生物间的进化关系
- 多样性指标：
  - Faith’s Phylogenetic Diversity
  - UniFrac distance
  - feature-based
- input：
  - dada2_rep_set.qza #代表序列
  - sepp-refs-gg-13-8.qza #参考数据库
- output：

- tree.qza
- tree_placements.qza
- command

```
$ qiime fragment-insertion sepp \
      --i-representative-sequences ./dada2_rep_set.qza \ #
      --i-reference-database sepp-refs-gg-13-8.qza \ #greengenes数据库
      --o-tree ./tree.qza \
      --o-placements ./tree_placements.qza \
      --p-threads 1  # update to a higher number if you can
```

- 去掉了标签的树，这个数应该是所有样本中所有的物种的这里用的数据不是本教程的是第一个教程的用本教程的建树方法用mega不可以可视化不晓得为什么



04.Alpha Rarefaction和Selecting a Rarefaction Depth：标准化

- 各种微生物多样性分析
  - 每个样品的ASVs特征表
  - 代表这些ASVs的系统树
- 步骤：
  - 数据标准化：解决样品测序深度不均匀的问题
  - rarefaction：不需要替换的二次采样标准化（抽样不放回?）
    - 1.从特征表中过滤掉低于稀疏深度样
    - 2.剩余样品抽样不放回以达到指定的测序深度（难点确定稀疏深度）
- commond

```
# 生成的文件都可以解压后在data里面看到nwk树文件，可以用mega可视化一下，
# 不过应该是没有物种信息的，大概率是特征序列的id
$ qiime diversity alpha-rarefaction \
      --i-table ./dada2_table.qza \
      --m-metadata-file ./metadata.tsv \
      --o-visualization ./alpha_rarefaction_curves.qzv \
      --p-min-depth 10 \
      --p-max-depth 4250
```