**Amplicon analysis**

The pipeline can be divided in 3 steps:

1. QC
2. OTU table generation
3. Data visualisation

N.B. the flags name refer to the analysis pipeline flags, they might be different to the tools flags.

**QC**

The presence of low quality sequencing data and sequencing primers, if not removed from the raw reads, can influence the assembly, clustering and taxonomy assignment, leading to wrong and inconsistent results. The first step of the pipeline is then dedicated to data cleaning and it uses a well-established workflow for Illumina error correction. The combination of Sickle for reads quality trimming, BayesHammer for reads error correction and PANDAseq for overlapping the sequences, has been proved to be the best approach to QC 16S sequencing data (<https://academic.oup.com/nar/article-lookup/doi/10.1093/nar/gku1341>).

**Cutadapt**

Cutadapt is used to remove the sequencing primers. It is an optional step and it is only run when forward and reverse primers sequences are supplied. New sequencers software often removes sequencing primers during the fastq files generation, but, in same occasions, fastq files could still retain them. Sequencing primers then need to be removed or they would interfere with the taxonomy assignment.

Catadapt can also be used as a sequence quality check because, while sequencing primers are removed by the sequencer, PCR primers are often retained. Sequences without PCR primers at the 5’-end can be considered spurious or low quality ones (PCR primer containing many errors). When PCR forward and reverse primers are provided, the reads without the primers will be removed. Because of this, if primers are supplied but the reads do not contain them, they will be considered as spurious, deleted and it will lead to an error.

Cutadapt is used in paired-end mode (-p) and it processes both forward and reverse reads at the same time. It searches for the supplied primers at the 5’-end of the reads ( -g/-G ^Fw\_primer/^Rv\_primer), and the reads containing the primers will be trimmed and written in a different fastq file. Cutadapt allows the usage of IUPAC wildcard characters (such as N). To take into account the high variability in the 16S sequence and MiSeq errors, the maximum level of error tolerance is set to 15% (-e 0.15). Assuming a PCR primer length of 20bp, this would mean a tolerance of up to 3 bases mismatch.

**Sickle**

Most sequencers produce reads with degrading quality score at the 3’-end. Low quality bases would greatly impact the assembly and OTU generation and need to be removed.

Sickle uses a sliding window and the Phred score (scale from 0 to 41 for fastq data) to trim low quality bases at the 3’-end (-x) of paired-end reads ( pe -f forward\_reads -r reverse\_reads ). Sickle supports Illumina, Solexa and Sanger quality values but, for the purpose of this pipeline, the quality value is set to “sanger” (CASAVA >=1.8).

Sickle slides a window of length equal to 10% the length of the read. When the average quality of the window drops below the quality threshold (default -q 20), the algorithm trims both the read and the quality string. If the read length after trimming is shorter than a user defined length (default -l 20), the read along with the corresponding read pair will be discarded.

**FastQC**

It is important to visually analyse the quality of the reads produces by the sequencer. FastQC is used to provide quality information reports for the raw and trimmed fastQ files. It provides per-base sequence quality plots, per-sequence quality scores and sequence length distribution for each sample separately.

**SPAdes**

SPAdes is an assembly toolkit containing many assembly pipelines and tools, and it is the most used short- read assembly tool for metagenomics studies. Within this pipeline, the only tool used is the BayesHammer algorithm for Illumina reads error correction (--only-error-correction). BayesHammer was originally designed for the error correction of single-cell sequencing data but it can greatly improve the quality of multi-cell sequencing data, too (<https://www.ncbi.nlm.nih.gov/pmc/articles/PMC3549815/> ).

**PANDAseq**:

PANDAseq is used to align paired-end reads and reconstruct an overlapping sequence. The algorithm will align the overlap reads (default minimum overlap length -O 10). In case of a mismatch in the overlapping region, the base with the better quality score is chosen. If no overlap is found or in the presence of any uncalled base (N) in either the forward or reverse read, the read pair is discarded. If after overlapping the reads, the sequence length is shorter than a user defined value, the sequence is discarded (default -L 380.)

At the end of this stringent QC, the data is ready for the OTU generation and a file containing reads count for each QC step is available (Reads\_count.txt).

**OTU table generation**

For the OTU generation, the pipeline allows the user to choose between UPARSE, VSERCH and QIIME. For technical reasons, the Galaxy instance of the pipeline currently only uses VSEARCH.

**VSEARCH**

N.B. To generate an OTU table, many steps are needed. For a detailed explanation of every step please refer to GitHub web page <https://github.com/torognes/vsearch>

Vsearch is an alternative, open source tool to USEARCH (UPARSE). USEARCH is considered to be the best tool for chimera removal and *De Novo* OTU generation currently available, but the 32-bit free version is capped to 4GB of memory. This makes the free version not suitable for the analysis of big datasets.

Vsearch is a free, open source 64-bit tool that has been designed to mimic USEARCH functionalities. It has been proved to be at least as accurate and fast as USEARCH (<https://www.ncbi.nlm.nih.gov/pubmed/27781170> ).

Because Vsearch has the same functionalities as USEARCH, the pipeline will follow USEARCH workflow as found on the USEARCH web site.

**Dereplication**

Before going through chimera removal and OTU clustering, the sequences need to be pooled and dereplicated (--derep\_fulllength). This step will produce a set of unique sequences with abundance annotation added to them. The sequences will be then sorted (--sizeout) by abundance and singletons, sequences not found more the once in the dataset, will be discarded (--minuniquesize 2). Singletons are likely to contain errors and discarding them will greatly improve the specificity of the OTU clustering.

**De novo chimera removal**

Chimeras are DNA sequences that are not actually present in the original samples. They usually form during the amplification step. They are a problem especially when amplifying sequences from a single region. In the 16S, the high sequences similarities cause, partially amplified sequences from one species, to anneal and act as primers for a different species. The result is a sequence that is made by the “fusion” of the 16S region from two different species. Chimeras would lead to overestimated and biased estimates of diversity measures.

It performs de novo chimera detection using an algorithm similar to UCHIME (<https://www.ncbi.nlm.nih.gov/pubmed/21700674> ). Briefly, in the de novo mode, Vsearch uses the assumption that the parents (the original sequences present in the samples) are at least twice as abundant as the chimeric sequences, because they undergo more rounds of amplification. Sequences are considered in the order of decreasing abundance. If a sequence is classified as chimeric, it is discarded; otherwise it is added to the reference database.

**Cluster OTUs**

The set of unique, chimera-free sequences can now be clustered into OTUs. The standard 97% similarity threshold is used (--id 0.97). Once the clusters are formed, the centroids for each cluster are written to a FASTA file.

**Reference chimera removal**

After de novo chimera removal, some chimeric sequences could still be present in the dataset. The OTU centroids are then checked against a user-supplied reference database. The database is assumed to be chimera-free and containing the parent sequences. The database used in this pipeline is RDPClassifier\_16S\_trainsetNo14\_rawtrainingdata/trainset14\_032015.fasta.The FASTA file obtained from this step is available for download (Vsearch\_multiplexed\_linearized\_dereplicated\_mc2\_repset\_nonchimeras\_OTUs.fasta).

**Map reads back to OTUs**

We now have a database of unique and chimera-free OTUs. We can now map the original sequences (including the singletons) back to these OTUs, keeping track of what sample they come from (--usearch\_global). If the sequence is >97% similar to OTU in the database (--id 0.97), it is considered to be part of the cluster, and it is assigned to that OTU. If the sequence is <97% similar to any of the OTU in the database, the sequence is discarded .Because we constructed the database, we know that the sequences are oriented the same way as the database and we can perform the alignment on a single strand (--strand plus), which will save computational time.

**Make OTU table (convert to BIOM)**

Most of the tools for alpha and beta diversity use as input file a BIOM table. We then need to convert our OTU table to BIOM format. To do this, the pipeline firstly converts the OTU table created in the previous step into a txt file using uc2otutab.py, and then converts this file into the BIOM format using biom convert (--table-type="otu table")

**Assign taxonomy**

For the taxonomy assignment the pipeline uses RDP classifier. The tool requires a reference file to be used to train the classifier and a taxonomy file to map the OTUs to a particular taxonomy. The classifier uses a Naïve Bayesian approach and, for each rank, it returns a confidence in the assignment. If the confidence score is higher than a threshold (default minimum confidence score used is 0.50) then the taxonomy is assigned.

The user can choose which database to use for this step, between Greengenes v13.8 and SILVA v119.

The BIOM file with the taxonomy assignment is available for download (Vsearch\_tax\_OTU\_table (biom format)).

**Create tree**

For some diversity measures, a phylogenetic tree is needed. To obtain the phylogenetic tree, the pipeline uses QIIME scripts. It first aligns the OTU sequences to a pre-aligned database using align\_seqs.py using default values. The alignment is then filtered for those positions where only gaps are present (filter\_alignment.py with default values). The filtered aligned file is used to make the phylogenetic tree with make\_phylogeny.py (default values).

The phylogenetic tree is available for download (Vsearch\_OTUs.tree).

**Data visualisation**

The plotting is entirely performed using QIIME scripts.

1. Summary tables: biom summarize-table –qualitative
2. Exclude samples with less than 5000 OTUs. They are not reliable and they would cause an error during alpha diversity calculations.

**Alpha diversity (alpha diversity measures used: observed\_species,shannon,simpson,fisher\_alpha)**

1. Create an interactive heatmap: make\_otu\_heatmap\_html.py
2. Rarefy the samples and compare alpha diversity measures
3. Create bar plots for Phylum, Genus and Species abundances

**Beta diversity**

1. Filter for low abundance OTUs?
2. PCoA plots for weighted and un-weighted UniFrac