
PipeCraft 2

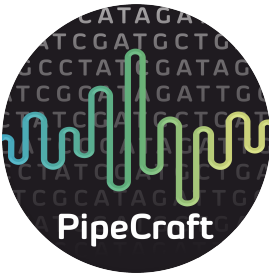
Release 0.1.2

Sten Anslan

Sep 02, 2023

CONTENTS

| | | |
|----------|---------------------------------------|-----------|
| 1 | Contents | 3 |
| 1.1 | Installation | 3 |
| 1.2 | User guide | 9 |
| 1.3 | Walkthrough | 62 |
| 1.4 | Post-processing tools | 82 |
| 1.5 | Troubleshooting | 86 |
| 1.6 | Licence | 88 |
| 1.7 | Contact | 89 |
| 1.8 | How to cite | 89 |
| 1.9 | Releases | 90 |
| 1.10 | Docker images | 93 |
| 2 | Currently implemented software | 95 |

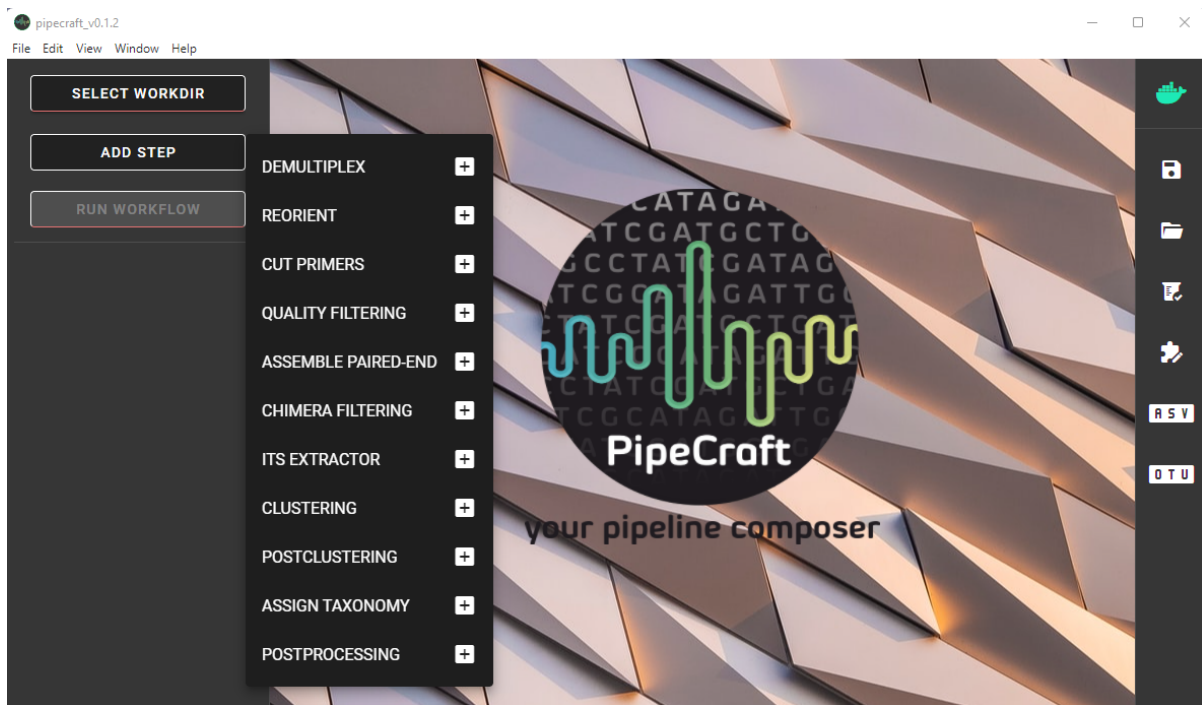


your pipeline composer

[github](#)

PipeCraft is a Graphical User Interface software that implements *various popular tools* for **metabarcoding** data analyses that are linked together to generate a custom bioinformatics pipeline/workflow.

Pre-defined full pipelines for generating *OTUs* or *ASVs* are also implemented.



Panels for pipeline processes contain key options for sequence data analyses, but all options of any implemented program may be accessed via *PipeCraft console (command line)*.

Default settings in the panels represent commonly used options for amplicon sequence data analyses, which may be tailored according to user experience or needs. Custom-designed pipeline settings can be saved, and thus the exact same pipeline may be easily re-run on other sequencing data (and for reproducibility, may be used as a supplement material in the manuscript). PipeCraft enables generation of the full pipeline (user specifies the input data and output will be e.g. OTU/ASV table with taxonomic annotations of the OTUs/ASVs), but supports also single-step mode where analyses may be performed in a step-by-step manner (*e.g. perform quality filtering, then examine the output and decide whether to adjust the quality filtering options of to proceed with next step, e.g. with chimera filtering step*).

CONTENTS



1.1 Installation

Current *versions* do not work on High Performance Computing (**HPC**) clusters **yet**.

Contents

- *Installation*
 - *Prerequisites*
 - *Windows*
 - *MacOS*
 - *Linux*
 - *Updating PipeCraft2*
 - *Uninstalling*
 - *Removing Docker images*

1.1.1 Prerequisites

The only prerequisite is [Docker](#).

See OS-specific (Windows, Mac, Linux) docker installation guidelines below.

Note: Modules of PipeCraft2 are distributed through Docker containers, which will liberate the users from the struggle to install/compile various software for metabarcoding data analyses. **Thus, all processes are run in Docker containers.** Relevant Docker container will be automatically downloaded prior the analysis.

Warning: Your OS might warn that PipeCraft2 is dangerous software! Please ignore the warning in this case.

1.1.2 Windows

PipeCraft2 was tested on **Windows 10** and **Windows 11**. Older Windows versions do not support PipeCraft GUI workflow through Docker.

1. Download [Docker for windows](#)
2. Download PipeCraft for [Windows](#): v0.1.4
3. Install PipeCraft via the setup executable

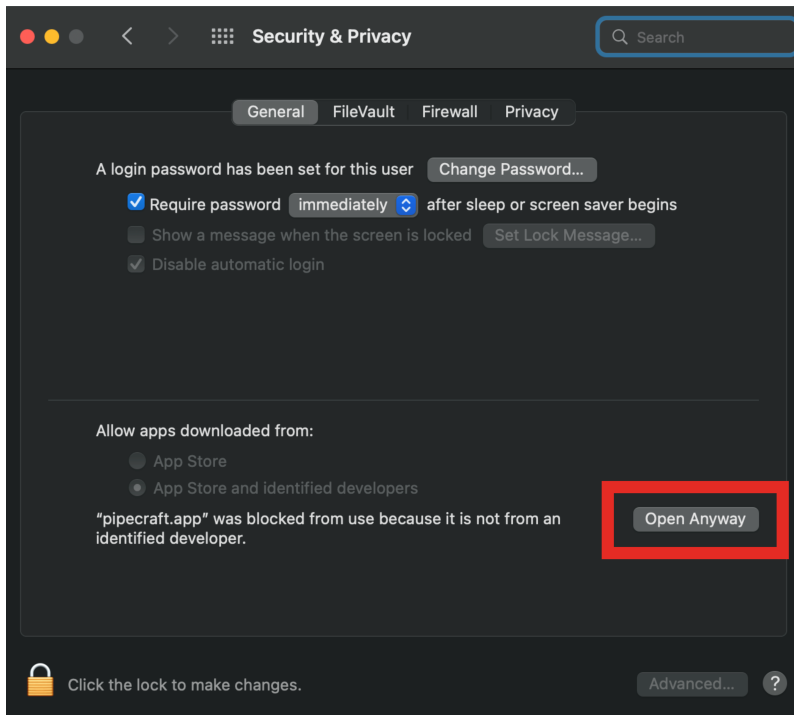
Warning: In Windows, please keep you working directory path as short as possible. Maximum path length in Windows is 260 characters. PipeCraft may not be able to work with files, that are buried “deep inside” (i.e. the path is too long).

Note: Resource limits for Docker are managed by Windows; but you can configure limits in a **.wslconfig** file (see **Settings** -> **Resources** on your Docker desktop app)

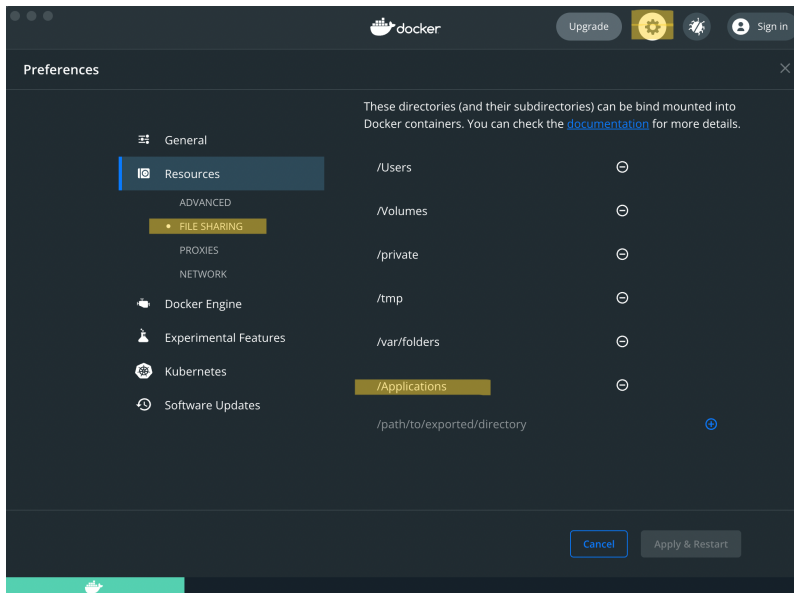
1.1.3 MacOS

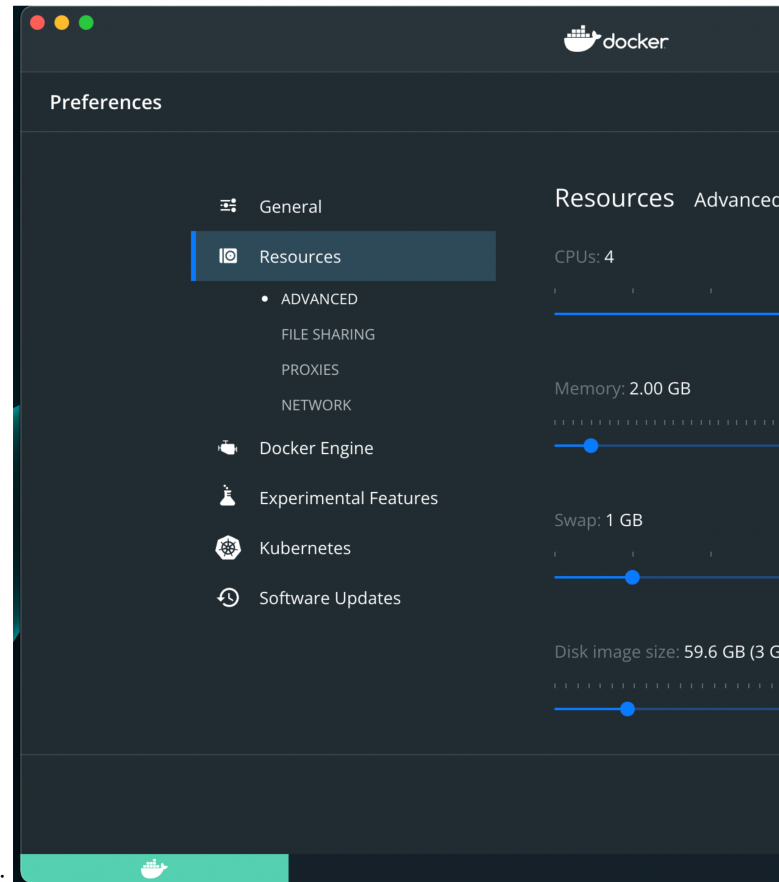
PipeCraft is supported on macOS 10.15+. Older OS versions might not support PipeCraft GUI workflow through Docker.

1. Check your Mac chip (Apple or Intel) and download [Docker for Mac](#)
2. Download PipeCraft for [Mac](#): v0.1.4
3. Install PipeCraft via **pkg** file
4. Currently, this app might be identified as app from an unidentified developer. Grant an exception for a blocked app by clicking the “**Open Anyway**” button in the General panel of **Security & Privacy** preferences



5. Open **Docker dashboard**: Settings -> Resources -> File Sharing; and add the directory where **pipecraft.app** was installed (it is usually /Applications)





Note: Manage Docker resource limits in the Docker dashboard:

1.1.4 Linux

PipeCraft was tested with **Ubuntu 20.04** and **Mint 20.1**. Older OS versions might not support PipeCraft GUI workflow through Docker.

1. Install Docker; [follow the guidelines under appropriate Linux distribution](#)
2. If you are a non-root user complete these [post-install steps](#)
3. Download PipeCraft for [Linux: v0.1.4](#)
4. Right click on the `pipecraft_*.deb` file and “Open With GDebi Package Installer” (Install Package) or `sudo dpkg -i path_to_deb_file`

Note: When you encounter ERROR during installation, then uninstall the previous version of PipeCraft `sudo dpkg --remove pipecraft-v0.1.3`

5. Run PipeCraft. If PipeCraft shortcut does not appear on the Desktop, then search the app and generate shortcut manually (installed in `/opt/pipecraft` directory)

Note: On Linux, Docker can use all available host resources.

1.1.5 Updating PipeCraft2

To avoid any potential software conflicts from PipeCraft2 **v0.1.1 to v0.1.4**, all Docker images of older PipeCraft2 version should be removed. *Starting from v0.1.5 -> if docker container is updated, it will get a new tag for new PipeCraft2 version*

See *removing docker images* section.

Note:

Currently available versions [HERE](#)

1.1.6 Uninstalling

Windows: uninstall PipeCraft via control panel

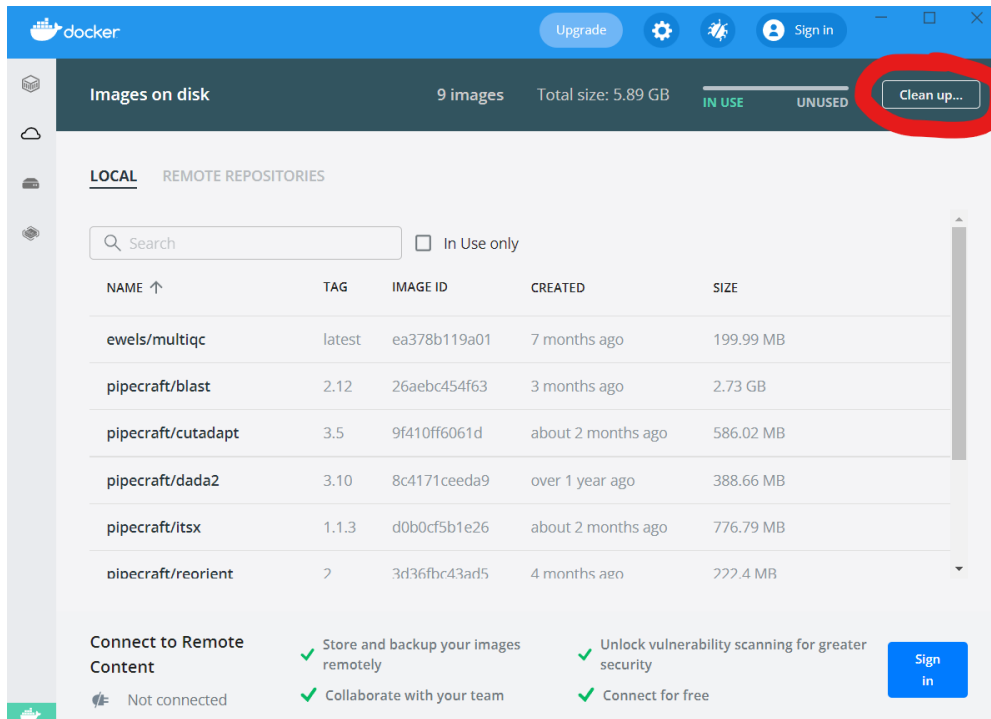
MacOS: Move pipecraft.app to Bin

Linux: remove pipecraft via Software Manager/Software Centre or via terminal `sudo dpkg --remove pipecraft`

1.1.7 Removing Docker images

On **MacOS** and **Windows:** Docker images and container can be easily managed from the Docker dashboard. For more info visit <https://docs.docker.com/desktop/dashboard/>

See **command-line** based way below.



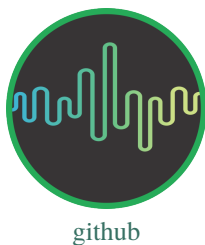
On **Linux** machines: containers and images are managed via the Docker cli commands (<https://docs.docker.com/engine/reference/commandline/rmi/>):

`sudo docker images` -> to see which docker images exist

`sudo docker rmi IMAGE_ID_here` -> to delete selected image

or

`sudo docker system prune -a` -> to delete all unused containers, networks, images



1.2 User guide

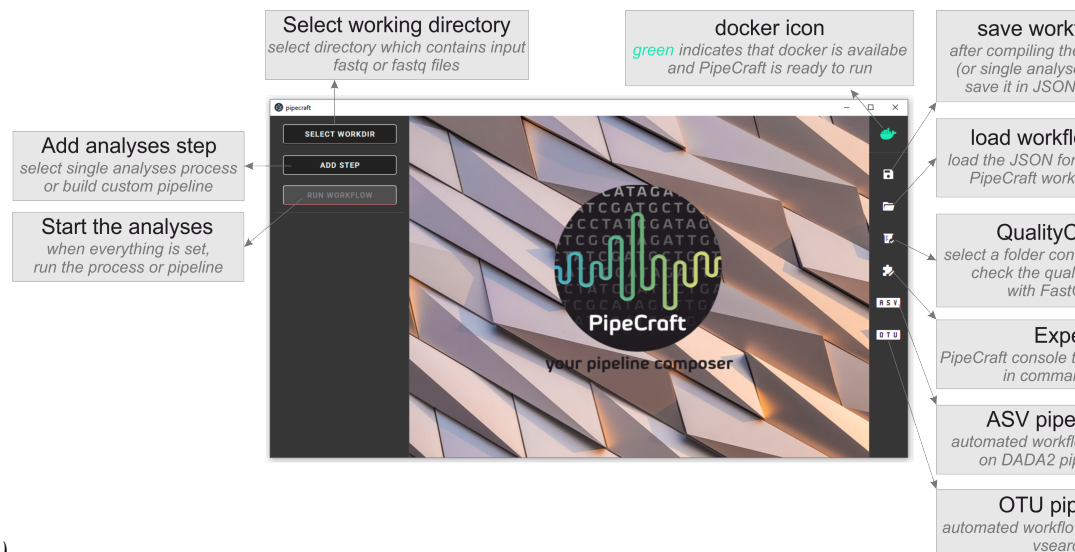
Contents

- *User guide*
 - *The interface*
 - *Glossary*
 - *Docker images*
 - *Save workflow*
 - *Load workflow*
 - *Quality and basic statistics screening of the data*
 - *Select workdir and run analyses*
 - *FULL PIPELINE PANELS*
 - * *ASVs workflow panel (with DADA2)*
 - * *OTUs workflow panel*
 - *ANALYSES PANELS*
 - *DEMULTIPLEX*
 - * *Indexes file example (fasta formatted)*
 - *REORIENT*
 - *CUT PRIMERS*
 - *QUALITY FILTERING*
 - * *vsearch*
 - * *trimmomatic*
 - * *fastp*
 - * *DADA2 ('filterAndTrim' function)*
 - *ASSEMBLE PAIRED-END reads*
 - * *vsearch*
 - * *DADA2*
 - *CHIMERA FILTERING*
 - * *uchime_denovo*
 - * *uchime3_denovo*
 - *ITS Extractor*
 - *CLUSTERING*
 - * *vsearch*
 - * *UNOISE3, with vsearch*
 - *POSTCLUSTERING*

- * *LULU*
- * *DADA2 collapse ASVs*
- *ASSIGN TAXONOMY*
 - * *BLAST (Camacho et al. 2009)*
 - * *DADA2 classifier*
 - * *Sequence databases*
- *POSTPROCESSING*
- *Expert-mode (PipeCraft2 console)*

1.2.1 The interface

The startup panel:



(click on the image for enlargement)

1.2.2 Glossary

List of terms that you may encounter in this user guide.

| | |
|---------------------------|---|
| working directory | the directory (folder) that contains the files for the analyses. The outputs will be written into this directory |
| paired-end data | obtained by sequencing two ends of the same DNA fragment, which results in read 1 (R1) and read 2 (R2) files per library or per sample |
| single-end data | only one sequencing file per library or per sample. Herein, may mean also assembled paired-end data. |
| demultiplexed data | sequences are sorted into separate files, representing individual samples |
| multiplexed data | file(s) that represent a pool of sequences from different samples |
| read/sequence | DNA sequence; herein, reads and sequences are used interchangeably |

1.2.3 Docker images

Initial PipeCraft2 installation does not contain any software for sequence data processing. All the processes are run through [docker](#), where the PipeCraft's simply GUI mediates the information exchange. Therefore, whenever a process is initiated for the **first time**, a relevant Docker image (contains required software for the analyses step) will be pulled from [Docker Hub](#).



Example: running DEMULTIPLEXING for the first time

Thus working **Internet connection** is initially required. Once the Docker images are pulled, PipeCraft2 can work without an Internet connection.

Docker images vary in size, and the speed of the first process is extended by the docker image download time.

1.2.4 Save workflow

Note: starting from version 0.1.4, PipeCraft2 will automatically save the settings into selected WORKDIR prior starting the analyses

Once the workflow settings are selected, save the workflow by pressing SAVE WORKFLOW button on the *right-ribbon*. For saving, working directory (SELECT WORKDIR) does not have to be selected.

Important: When **saiving workflow** settings in **Linux**, specify the file extension as **JSON** (e.g. my_16S_ASVs_pipe.JSON). When trying to load the workflow, only .JSON files will be permitted as input. *Windows and Mac OS automatically extend files as JSON (so you may just save “my_16S_ASVs_pipe”).*

1.2.5 Load workflow

Note: Prior loading the workflow, make sure that the saved workflow configuration has a .JSON extension. Note also that **workflows saved in older PipeCraft2 version** might not run in newer version, but anyhow the selected options will be visible for reproducibility.

Press the LOAD WORKFLOW button on the *right-ribbon* and select appropriate JSON file. The configuration will be loaded; SELECT WORKDIR and run analyses.

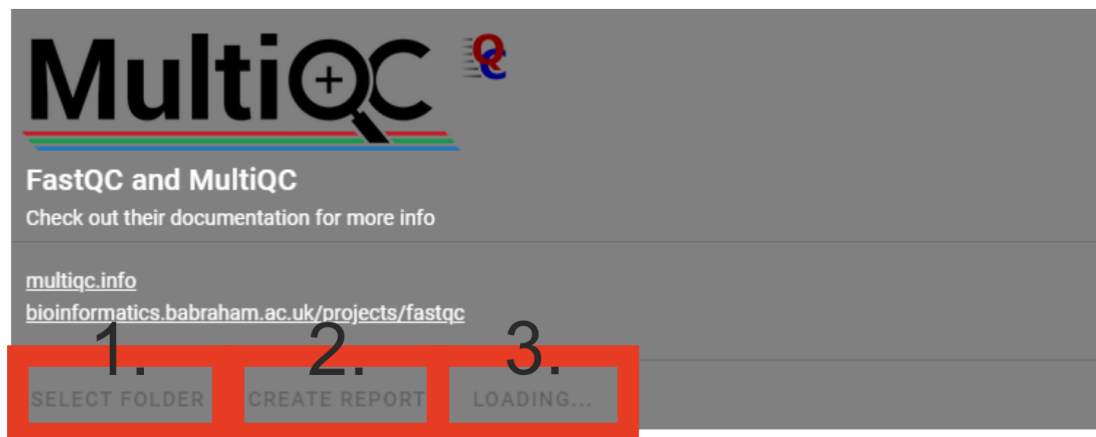
1.2.6 Quality and basic statistics screening of the data

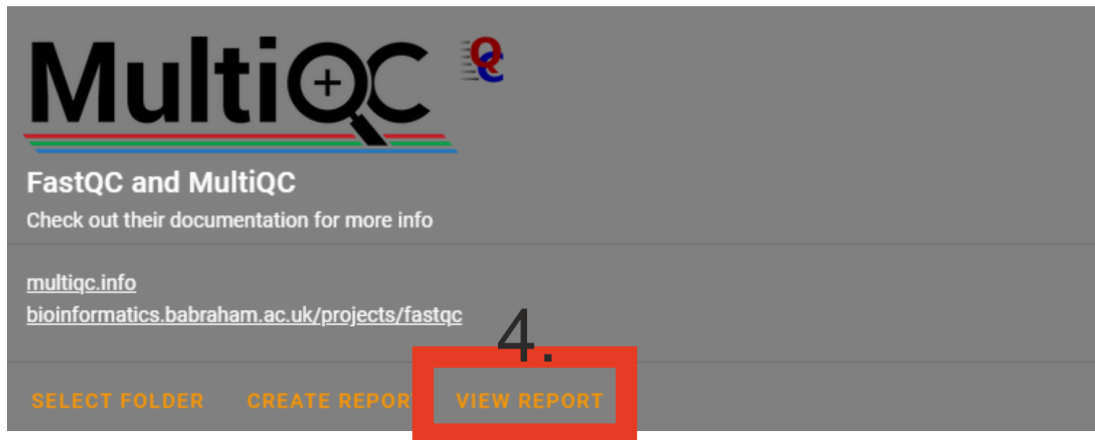
Quality and basic statistics screening of the data can be done via QualityCheck panel. QualityCheck panel implements FastQC and MultiQC to screen the input **fastq** files.



To start:

1. **Select folder** (a working directory) which contains only **fastq** (fastq/fq) files that you aim to inspect.
2. Press **CREATE REPORT** to start MultiQC
3. “LOADING ...” will be displayed while the report is being generated





4. Click VIEW REPORT. A html file (multiqc_report.html) will open in your default web browser.

*If the summary does not open, check your working folder for the presence of **multiqc_report.html** and try to open with some other web browser. Something went wrong if the file **multiqc_report.html** **does not exist** (may fail when maximum number of fastq files in the folder is extremely large, >10 000).*

5. Check out “using MultiQC reports” in MultiQC web page.

Note: Note that ‘_fastqc.zip’ and ‘_fastqc.html’ are generated for each fastq file in the ‘**quality_check**’ directory. These are summarized in **multiqc_report.html**, so you may delete all individual ‘_fastqc.zip’ and ‘_fastqc.html’ files.

1.2.7 Select workdir and run analyses

1. Open your working directory by pressing the SELECT WORKDIR button. E.g., if working with **FASTQ** files, then be sure that the working directory contains **only relevant FASTQ files** because the selected process will be applied to all FASTQ files in the working directory!

Note: When using Windows OS, the selection window might not display the files while browsing through the directories.

After selecting a working directory, PipeCraft needs you to specify if the working directory consists of

- multiplexed or demultiplexed data
- the data is paired-end or single-end
- and the extension of the data (fastq or fasta)

multiplexed → only one file (or a pair of files, R1 and R2) per sequencing data (library)

demultiplexed → multiple per-sample sequencing files per library

paired-end data -> such as data from Illumina or MGI-Tech platforms (R1 and R2 files). Be sure to have ****R1**** and ****R2**** strings in the paired-end files (not simply _1 and _2)

single-end data -> such as data from PacBio, or assembled paired-end data (single file per library or per sample)

2. Select **ASV** or **OTU** workflow panel or press ADD STEP button to select relevant **step** [or **load the PipeCraft settings file**]; edit settings if needed (**SAVE the settings for later use**) and **start running the analyses** by pressing the RUN WORKFLOW button.

Note: Step-by-step analyses: after RUN WORKFLOW is finished, then press SELECT WORKDIR to specify inputs for the next process

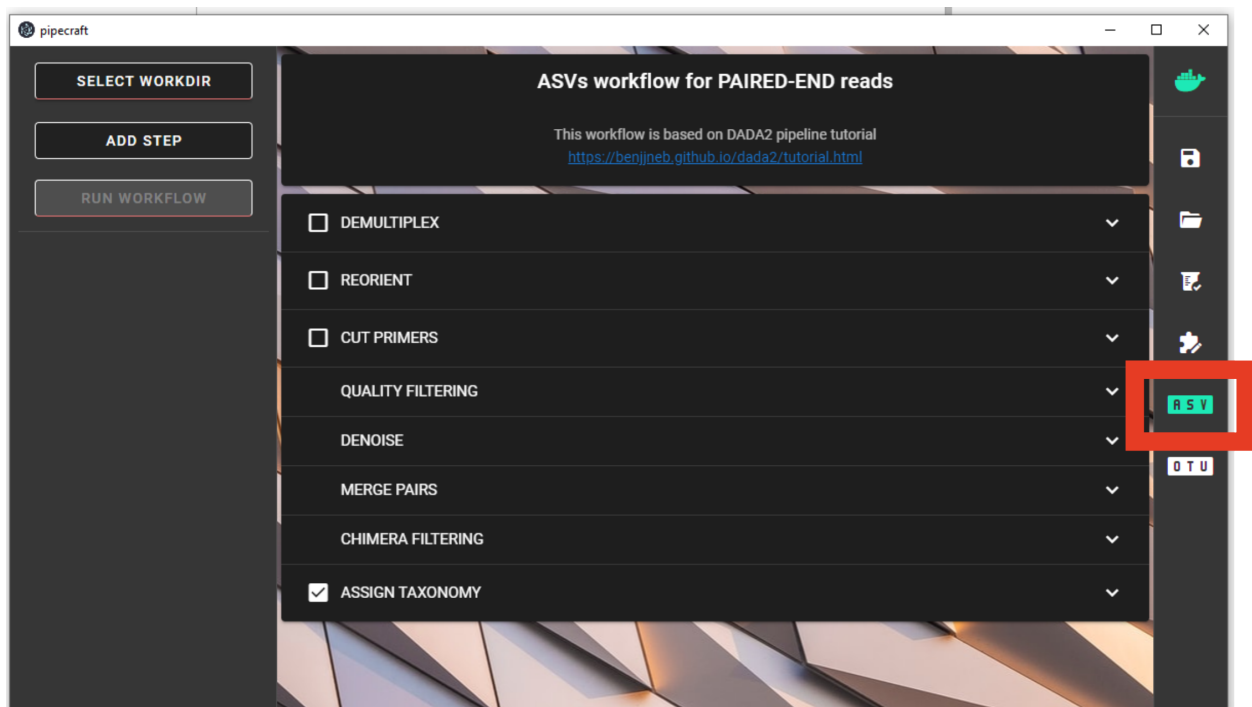
Note: The **output files will be overwritten** if running the same analysis step **multiple times in the same working directory**

3. Each process creates a separate output directory with the processed files inside the selected working directory. **README** file about the process and **sequence count summary** statistics are included in the output directory.

1.2.8 FULL PIPELINE PANELS

ASVs workflow panel (with DADA2)

Note: Current ASVs workflow supports only **PAIRED-END** reads! Working directory must contain paired-end reads for at **least 2 samples**.



ASV workflow is active (green icon)



; ASV workflow is off



This automated workflow is based on the [DADA2 tutorial](#)

Note that demultiplexing, reorienting, and primer removal steps are optional and do not represent parts from the DADA2 tutorial. Nevertheless, it is advisable to *reorient* your reads (to 5'-3') and *remove primers* before proceeding with ASV generation with DADA2.

The official DADA2 manual is available [here](#)

Default options:

| Analyses step | Default setting |
|------------------------------------|---|
| <i>DEMULTIPLEX</i> (optional) | – |
| <i>REORIENT</i> (optional) | – |
| <i>REMOVE PRIMERS</i> (optional) | – |
| <i>QUALITY FILTERING</i> | <pre> read_R1 = \.R1 read_R2 = \.R2 samp_ID = \. maxEE = 2 maxN = 0 minLen = 20 truncQ = 2 truncLen = 0 maxLen = 9999 minQ = 2 matchIDs = TRUE </pre> |
| <i>DENOISE</i> | <pre> pool = FALSE selfConsist = FALSE qualityType = Auto </pre> |
| <i>MERGE PAIRED-END READS</i> | <pre> minOverlap = 12 maxMismatch = 0 trimOverhang = FALSE justConcatenate = FALSE </pre> |
| <i>CHIMERA FILTERING</i> | <pre> method = consensus </pre> |
| <i>Filter ASV table</i> (optional) | <pre> collapseNoMismatch = TRUE by_length = 250 minOverlap = 20 vec = TRUE </pre> |
| <i>ASSIGN TAXONOMY</i> (optional) | <pre> minBoot = 50 tryRC = FALSE dada2 database = select a database </pre> |

QUALITY FILTERING [ASVs workflow]

DADA2 [filterAndTrim](#) function performs quality filtering on input FASTQ files based on user-selected criteria. Outputs include filtered FASTQ files located in the `qualFiltered_out` directory.

Quality profiles may be examined using the *QualityCheck module*.

| Setting | Tooltip |
|-----------------|--|
| read_R1 | <p>applies only for paired-end data.</p> <p>Identifyer string that is common for all R1 reads (e.g. when all R1 files have ‘.R1’ string, then enter ‘\R1’.</p> <p>Note that backslash is only needed to escape dot regex; e.g. when all R1 files have ‘_R1’ string, then enter ‘_R1’).</p> |
| read_R2 | <p>applies only for paired-end data.</p> <p>Identifyer string that is common for all R2 reads (e.g. when all R2 files have ‘.R2’ string, then enter ‘\R2’.</p> <p>Note that backslash is only needed to escape dot regex; e.g. when all R2 files have ‘_R1’ string, then enter ‘_R2’).</p> |
| samp_ID | <p>applies only for paired-end data.</p> <p>Identifyer string that separates the sample name from redundant characters (e.g. file name = sample1.R1.fastq, then underscore ‘\.’ would be the ‘identifier string’ (sample name = sampl84));</p> <p>note that backslash is only needed to escape dot regex (e.g. when file name = sample1_R1.fastq then specify as ‘_’)</p> |
| maxEE | discard sequences with more than the specified number of expected errors |
| maxN | discard sequences with more than the specified number of N’s (ambiguous bases) |
| minLen | <p>remove reads with length less than minLen. minLen is enforced</p> <p>after all other trimming and truncation</p> |
| truncQ | truncate reads at the first instance of a quality score less than or equal to truncQ |
| truncLen | |
| 1.2. User guide | <p>truncate reads after truncLen bases</p> <p>(applies to R1 reads when working with paired-end data).</p> <p>Reads shorter than this are discarded.</p> |

see *default settings*

DENOISING [ASVs workflow]

DADA2 `dada` function to remove sequencing errors. Outputs filtered fasta files into `denoised_assembled.dada2` directory.

| Setting | Tooltip |
|--------------------------|--|
| <code>pool</code> | if TRUE, the algorithm will pool together all samples prior to sample inference. Pooling improves the detection of rare variants, but is computationally more expensive. If <code>pool = 'pseudo'</code> , the algorithm will perform pseudo-pooling between individually processed samples. |
| <code>selfConsist</code> | if TRUE, the algorithm will alternate between sample inference and error rate estimation until convergence |
| <code>qualityType</code> | 'Auto' means to attempt to auto-detect the fastq quality encoding. This may fail for PacBio files with uniformly high quality scores, in which case use 'FastqQuality' |

see *default settings*

MERGE PAIRS [ASVs workflow]

DADA2 `mergePairs` function to merge paired-end reads. Outputs merged fasta files into `denoised_assembled.dada2` directory.

| Setting | Tooltip |
|------------------------------|--|
| <code>minOverlap</code> | the minimum length of the overlap required for merging the forward and reverse reads |
| <code>maxMismatch</code> | the maximum mismatches allowed in the overlap region |
| <code>trimOverhang</code> | if TRUE, overhangs in the alignment between the forwards and reverse read are trimmed off. Overhangs are when the reverse read extends past the start of the forward read, and vice-versa, as can happen when reads are longer than the amplicon and read into the other-direction primer region |
| <code>justConcatenate</code> | if TRUE, the forward and reverse-complemented reverse read are concatenated rather than merged, with a NNNNNNNNNN (10 Ns) spacer inserted between them |

see *default settings*

CHIMERA FILTERING [ASVs workflow]

DADA2 `removeBimeraDenovo` function to remove chimeras. Outputs filtered fasta files into `chimeraFiltered_out.dada2` and final ASVs to `ASVs_out.dada2` directory.

| Setting | Tooltip |
|---------------------|--|
| <code>method</code> | <p>‘consensus’ - the samples are independently checked for chimeras, and a consensus decision on each sequence variant is made.</p> <p>If ‘pooled’, the samples are all pooled together for chimera identification.</p> <p>If ‘per-sample’, the samples are independently checked for chimeras</p> |

see *default settings*

filter ASV table [ASVs workflow]

DADA2 `collapseNoMismatch` function to collapse identical ASVs; and ASVs filtering based on minimum accepted sequence length (custom R functions). Outputs filtered ASV table and fasta files into `ASVs_out.dada2/filtered` directory.

| Setting | Tooltip |
|---------------------------------|---|
| <code>collapseNoMismatch</code> | collapses ASVs that are identical up to shifts or length variation, i.e. that have no mismatches or internal indels |
| <code>by_length</code> | discard ASVs from the ASV table that are shorter than specified value (in base pairs). Value 0 means OFF, no filtering by length |
| <code>minOverlap</code> | <code>collapseNoMismatch</code> setting. Default = 20. The minimum overlap of base pairs between ASV sequences required to collapse them together |
| <code>vec</code> | <code>collapseNoMismatch</code> setting. Default = TRUE. Use the vectorized aligner. Should be turned off if sequences exceed 2kb in length |

see *default settings*

ASSIGN TAXONOMY [ASVs workflow]

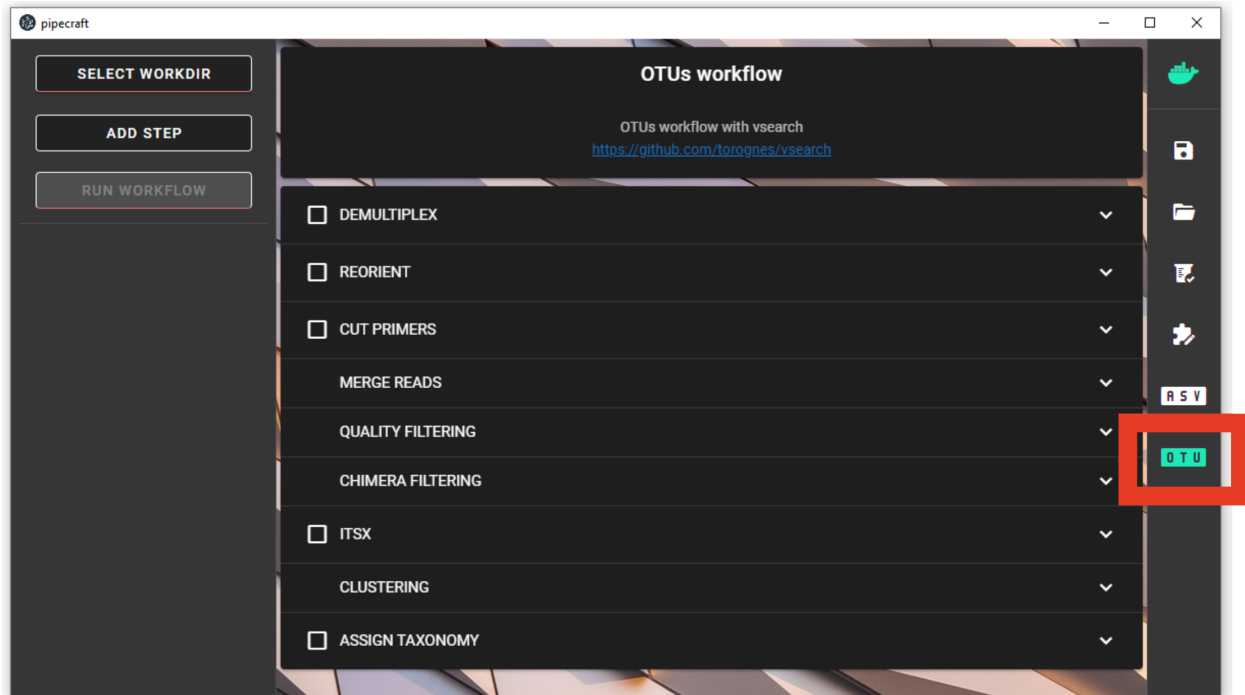
DADA2 `assignTaxonomy` function to classify ASVs. Outputs classified fasta files into `taxonomy_out.dada2` directory.

| Setting | Tooltip |
|-----------------------------|--|
| <code>minBoot</code> | the minimum bootstrap confidence for assigning a taxonomic level |
| <code>tryRC</code> | the reverse-complement of each sequences will be used for classification if it is a better match to the reference sequences than the forward sequence |
| <code>dada2 database</code> | select a reference database fasta file for taxonomy annotation Download DADA2-formatted reference databases here |

see *default settings*

OTUs workflow panel

Note: This OTU workflow works with paired-end (e.g. Illumina, MGI-Tech) as well as single-end reads (e.g. PacBio, assembled Illumina reads)



OTU workflow is active (green icon)  ; OTU workflow is off 

This automated workflow is mostly based on **vsearch** (Rognes et. al 2016) [manual]

Note that demultiplexing, reorient and remove primers steps are optional. Nevertheless, it is advisable to *reorient* your reads (to 5'-3') and *remove primers* before proceeding.

Default options:

click on analyses step for more info

| Analyses step | Default setting |
|---|--|
| <i>DEMULTIPLEX</i> (optional) | – |
| <i>REORIENT</i> (optional) | – |
| <i>REMOVE PRIMERS</i> (optional) | – |
| <i>MERGE READS</i> | <pre> read_R1 = \.R1 min_overlap = 12 min_length = 32 allow_merge_stagger = TRUE include_only_R1 = FALSE max_diffs = 20 max_Ns = 0 max_len = 600 keep_disjoined = FALSE fastq_qmax = 41 </pre> |
| <i>QUALITY FILTERING with vsearch</i> | <pre> maxEE = 1 maxN = 0 minLen = 32 max_length = undefined qmax = 41 qmin = 0 maxee_rate = undefined </pre> |
| <i>CHIMERA FILTERING with uchime_denovo</i> | <pre> pre_cluster = 0.98 min_unique_size = 1 denovo = TRUE reference_based = undefined abundance_skew = 2 min_h = 0.28 </pre> |
| <i>ITS Extractor</i> (optional) | <pre> organisms = all regions = all partial = 50 region_for_clustering = ITS2 cluster_full_and_partial = TRUE e_value = 1e-2 scores = 0 domains = 2 complement = TRUE only_full = FALSE truncate = TRUE </pre> |
| <i>CLUSTERING with vsearch</i> | |
| 1.2. User guide | <pre> OTU_type = centroid similarity_threshold = 0.97 strands = both remove_singletons = false </pre> |

1.2.9 ANALYSES PANELS

1.2.10 DEMULTIPLEX

If data is **multiplexed**, the first step would be **demultiplexing** (using [cutadapt \(Martin 2011\)](#)). This is done based on the user specified *indexes file*, which includes molecular identifier sequences (so called indexes/tags/barcodes) per sample. Note that reverse complementary matches will also be searched.

Fastq/fastq formatted paired-end and single-end data are supported.

Outputs are fastq/fastq files per sample in `demultiplexed_out` directory. Indexes are **truncated** from the sequences.

Paired-end samples get `.R1` and `.R2` read identifiers.

unknown.fastq file(s) contain sequences where specified index combinations were not found.

Note: If found, sequences with any index combination will be outputted **when using paired indexes**. That means, if, for example, your `sample_1` is indexed with `indexFwd_1-indexRev_1` and `sample_2` with `indexFwd_2-indexRev_2`, then files with `indexFwd_1-indexRev_2` and `indexFwd_2-indexRev_1` are also written (although latter index combinations were not used in the lab to index any sample [i.e. represent tag-switches]). Simply remove those files if not needed or use to estimate tag-switching error if relevant.

| Setting | Tooltip |
|----------------|---|
| index file | select your fasta formatted indexes file for demultiplexing (<i>see guide here</i>), where fasta headers are sample names, and sequences are sample specific index or index combination |
| index mismatch | allowed mismatches during the index search |
| overlap | number of overlap bases with the index Recommended overlap is the maximum length of the index for confident sequence assignments to samples |
| min seq length | minimum length of the output sequence |
| no indels | do not allow insertions or deletions in primer search. Mismatches are the only type of errors accounted in the error rate parameter |

Note: Heterogeneity spacers or any redundant base pairs attached to index sequences do not affect demultiplexing. Indexes are trimmed from the best matching position.

Indexes file example (fasta formatted)

Note: Only **IUPAC codes** are allowed in the sequences. Avoid using '.' in the sample names (e.g. instead of sample.1, use sample_1)

1. Demultiplexing using single indexes:

```
>sample1
AGCTGCACCTAA
>sample2
AGCTGTCAAGCT
>sample3
AGCTTCGACAGT
>sample4
AGGCTCCATGTA
```

```
>sample5
AGGCTTACGTGT
>sample6
AGGTACGCAATT
```

2. Demultiplexing using dual (paired) indexes:

Note: IMPORTANT! reverse indexes will be automatically oriented to 5'-3' (for the search); so you can simply copy-paste the indexes from your lab protocol.

```
>sample1
AGCTGCACCTAA...AGCTGCACCTAA
>sample2
AGCTGTCAAGCT...AGCTGTCAAGCT
>sample3
AGCTTCGACAGT...AGCTTCGACAGT
>sample4
AGGCTCCATGTA...AGGCTCCATGTA
>sample5
AGGCTTACGTGT...AGGCTTACGTGT
>sample6
AGGTACGCAATT...AGGTACGCAATT
```

Note: Anchored indexes (<https://cutadapt.readthedocs.io/en/stable/guide.html#anchored-5adapters>) with ^ symbol are **not supported** in PipeCraft demultiplex GUI panel.

DO NOT USE, e.g.

```
>sample1
^AGCTGCACCTAA
```

```
>sample1
^AGCTGCACCTAA...AGCTGCACCTAA
```

How to compose indexes.fasta

In Excel (or any alternative program); first column represents sample names, second (and third) column represent indexes (or index combinations) per sample:

Exaples:

| | |
|---------|--------------|
| sample1 | AGCTGCACCTAA |
| sample2 | AGCTGTCAAGCT |
| sample3 | AGCTTCGACAGT |
| sample4 | AGGCTCCATGTA |
| sample5 | AGGCTTACGTGT |
| sample6 | AGGTACGCAATT |

or

| | | |
|---------|--------------|--------------|
| sample1 | AGCTGCACCTAA | AGCTGCACCTAA |
| sample2 | AGCTGTCAAGCT | AGCTGTCAAGCT |
| sample3 | AGCTTCGACAGT | AGCTTCGACAGT |
| sample4 | AGGCTCCATGTA | AGGCTCCATGTA |
| sample5 | AGGCTTACGTGT | AGGCTTACGTGT |
| sample6 | AGGTACGCAATT | AGGTACGCAATT |

Copy those two (or three) columns to text editor that support regular expressions, such as NotePad++ or Sublime Text. If using **PAIRED** indexes (three columns), proceed to bullet no. 5

- single-end indexes:
 1. Open 'find & replace' Find ^ (which denotes the beginning of each line). Replace with > (and DELETE THE LAST > in the beginning of empty row).
 2. Find \t (which denotes tab). Replace with \n (which denotes the new line).

FASTA FORMATTED (single-end indexes) indexes.fasta file is ready; SAVE the file.
- Only for paired-indexes:
 1. Open 'find & replace': Find ^ (denotes the beginning of each line); replace with > (and DELETE THE LAST > in the beginning of empty row).
 2. Find .*K\t (which captures the second tab); replace with ... (to mark the linked paired-indexes).
 3. Find \t (denotes the tab); replace with \n (denotes the new line).

FASTA FORMATTED (paired indexes) indexes.fasta file is ready; SAVE the file.

1.2.11 REORIENT

Sequences are often (if not always) in both, 5'-3' and 3'-5', orientations in the raw sequencing data sets. If the data still contains PCR primers that were used to generate amplicons, then by specifying these PCR primers, this panel will perform sequence reorientation of all sequences.

For reorienting, first the forward primer will be searched (using [fqgrep](#)) and if detected then the read is considered as forward complementary (5'-3'). Then the reverse primer will be searched (using [fqgrep](#)) from the same input data and if detected, then the read is considered to be in reverse complementary orientation (3'-5'). Latter reads will be transformed to 5'-3' orientation and merged with other 5'-3' reads. Note that for paired-end data, R1 files will be reoriented to 5'-3' but R2 reads will be reoriented to 3'-5' in order to merge paired-end reads.

At least one of the PCR primers must be found in the sequence. For example, read will be recorded if forward primer was found even though reverse primer was not found (and vice versa). **Sequence is discarded if none of the PCR primers are found.**

Sequences that contain **multiple forward or reverse primers (multi-primer artefacts)** are discarded as it is highly likely that these are chimeric sequences. Reorienting sequences **will not remove** primer strings from the sequences.

Note: For single-end data, sequences will be reoriented also during the ‘cut primers’ process (see below); therefore this step may be skipped when working with single-end data (such as data from PacBio machines OR already assembled paired-end data).

Reorienting reads may be relevant for generating ASVs with DADA2 as reverse complement sequences will represent separate ASVs. In the clustering step of an OTU pipeline, both strands of the sequences can be compared prior forming OTUs; thus this step may be skipped in the OTU pipeline.

Supported file formats for paired-end input data are only **fastq**, but also **fasta** for single-end data. **Outputs** are fastq/fasta files in **reoriented_out** directory. Primers are **not truncated** from the sequences; this can be done using *CUT PRIMER panel*

| Setting | Tooltip |
|-----------------|--|
| mismatches | allowed mismatches in the primer search |
| forward_primers | specify forward primer (5’-3’); IUPAC codes allowed; add up to 13 primers |
| reverse_primers | specify reverse primer (3’-5’); IUPAC codes allowed; add up to 13 primers |

1.2.12 CUT PRIMERS

If the input data contains PCR primers (or e.g. adapters), these can be removed in the CUT PRIMERS panel. CUT PRIMERS processes mostly relies on *cutadapt* (Martin 2011).

For generating OTUs or ASVs, it is recommended to truncate the primers from the reads (**unless ITS Extractor is used** later to remove flanking primer binding regions from ITS1/ITS2/full ITS; in that case keep the primers better detection of the 18S, 5.8S and/or 28S regions). Sequences where PCR primer strings were not detected are discarded by default (but stored in ‘untrimmed’ directory). Reverse complementary search of the primers in the sequences is also performed. Thus, primers are clipped from both 5’-3’ and 3’-5’ oriented reads. However, note that **paired-end reads will not be reoriented** to 5’-3’ during this process, but **single-end reads will be reoriented** to 5’-3’ (thus no extra reorient step needed for single-end data).

Note: For paired-end data, the **seqs_to_keep** option should be left as default (**‘keep_all’**). This will output sequences where at least one primer has been clipped. ‘keep_only_linked’ option outputs only sequences where both the forward

and reverse primers are found (i.e. 5'-forward...reverse-3'). 'keep_only_linked' may be used for single-end data to keep only **full-length amplicons**.

Fastq/fastq formatted paired-end and single-end data are supported.

Outputs are fastq/fastq files in `primersCut_out` directory. Primers are **truncated** from the sequences.

| Setting | Tooltip |
|-----------------|--|
| forward primers | specify forward primer (5'-3'); IUPAC codes allowed; add up to 13 primers |
| reverse primers | specify reverse primer (3'-5'); IUPAC codes allowed; add up to 13 primers |
| mismatches | allowed mismatches in the primer search |
| min overlap | number of overlap bases with the primer sequence. Partial matches are allowed, but short matches may occur by chance, leading to erroneously clipped bases. Specifying higher overlap than the length of primer sequence will still clip the primer (e.g. primer length is 22 bp, but overlap is specified as 25 - this does not affect the identification and clipping of the primer as long as the match is in the specified mismatch error range) |
| seqs to keep | keep sequences where at least one primer was found (fwd or rev); recommended when cutting primers from paired-end data (unassembled), when individual R1 or R2 read lengths are shorter than the expected amplicon length. 'keep_only_linked' = keep sequences if primers are found in both ends (fwd...rev); discards the read if both primers were not found in this read |
| pair filter | applies only for paired-end data. 'both', means that a read is discarded only if both, corresponding R1 and R2, reads do not contain primer strings (i.e. a read is kept if R1 contains primer string, but no primer string found in R2 read). Option 'any' discards the read if primers are not found in both, R1 and R2 reads |
| min seq length | Chapter 1. Contents minimum length of the output sequence |
| no indels | |

1.2.13 QUALITY FILTERING

Quality filter and trim sequences.

Fastq formatted paired-end and single-end data are supported.

Outputs are fastq files in `qualFiltered_out` directory.

vsearch

| vsearch setting | Tooltip |
|-----------------|--|
| maxEE | maximum number of expected errors per sequence (see here). Sequences with higher error rates will be discarded |
| maxN | discard sequences with more than the specified number of Ns |
| minLen | minimum length of the filtered output sequence |
| max_length | discard sequences with more than the specified number of bases. Note that if 'trunc length' setting is specified, then 'max length' SHOULD NOT be lower than 'trunc length' (otherwise all reads are discarded) [empty field = no action taken] Note that if 'trunc length' setting is specified, then 'min length' SHOULD BE lower than 'trunc length' (otherwise all reads are discarded) |
| qmax | specify the maximum quality score accepted when reading FASTQ files. The default is 41, which is usual for recent Sanger/Illumina 1.8+ files. For PacBio data use 93 |
| trunc_length | truncate sequences to the specified length. Shorter sequences are discarded; thus if specified, check that 'min length' setting is lower than 'trunc length' ('min length' therefore has basically no effect) [empty field = no action taken] |
| qmin | the minimum quality score accepted for FASTQ files. The default is 0, which is usual for recent Sanger/Illumina 1.8+ files. Older formats may use scores between -5 and 2 |
| 34 maxee_rate | Chapter 1. Contents discard sequences with more than the specified number of expected errors per base |

trimmomatic

| trimmomatic setting | Tooltip |
|----------------------------|--|
| window_size | the number of bases to average base qualities Starts scanning at the 5'-end of a sequence and trims the read once the average required quality (required_qual) within the window size falls below the threshold |
| required_quality | the average quality required for selected window size |
| min_length | minimum length of the filtered output sequence |
| leading_qual_threshold | quality score threshold to remove low quality bases from the beginning of the read. As long as a base has a value below this threshold the base is removed and the next base will be investigated |
| trailing_qual_threshold | quality score threshold to remove low quality bases from the end of the read. As long as a base has a value below this threshold the base is removed and the next base will be investigated |
| phred | phred quality scored encoding. Use phred64 if working with data from older Illumina (Solexa) machines |

fastp

| fastp setting | Tooltip |
|------------------------|---|
| window_size | the window size for calculating mean quality |
| required_qual | the mean quality requirement per sliding window (window_size) |
| min_qual | the quality value that a base is qualified. Default 15 means phred quality $\geq Q15$ is qualified |
| min_qual_thresh | how many percents of bases are allowed to be unqualified (0-100) |
| maxNs | discard sequences with more than the specified number of Ns |
| min_length | minimum length of the filtered output sequence. Shorter sequences are discarded |
| max_length | reads longer than 'max length' will be discarded, default 0 means no limitation |
| trunc_length | truncate sequences to specified length. Shorter sequences are discarded; thus check that 'min length' setting is lower than 'trunc length' |
| aver_qual | if one read's average quality score $<$ 'aver_qual', then this read/pair is discarded. Default 0 means no requirement |
| low_complexity_filter | enables low complexity filter and specify the threshold for low complexity filter. The complexity is defined as the percentage of base that is different from its next base ($\text{base}[i] \neq \text{base}[i+1]$). |
| 1.2. User guide | E.g. vaule 30 means then 30% complexity is required. 37 Not specified = filter not applied |
| cores | |

DADA2 ('filterAndTrim' function)

| DADA2 setting | Tooltip |
|----------------------|---|
| read_R1 | <p>applies only for paired-end data.</p> <p>Identifier string that is common for all R1 reads (e.g. when all R1 files have '.R1' string, then enter '\.R1'.</p> <p>Note that backslash is only needed to escape dot regex; e.g. when all R1 files have '_R1' string, then enter '_R1').</p> |
| read_R2 | <p>applies only for paired-end data.</p> <p>Identifier string that is common for all R2 reads (e.g. when all R2 files have '.R2' string, then enter '\.R2'.</p> <p>Note that backslash is only needed to escape dot regex; e.g. when all R2 files have '_R2' string, then enter '_R2').</p> |
| samp_ID | <p>applies only for paired-end data.</p> <p>Identifier string that separates the sample name from redundant characters (e.g. file name = sample1.R1.fastq, then underscore '\' would be the 'identifier string' (sample name = sample184));</p> <p>note that backslash is only needed to escape dot regex (e.g. when file name = sample1_R1.fastq then specify as '_')</p> |
| maxEE | discard sequences with more than the specified number of expected errors |
| maxN | discard sequences with more than the specified number of N's (ambiguous bases) |
| minLen | <p>remove reads with length less than minLen. minLen is enforced</p> <p>after all other trimming and truncation</p> |
| truncQ | truncate reads at the first instance of a quality score less than or equal to truncQ |

1.2.14 ASSEMBLE PAIRED-END reads

Assemble paired-end sequences (such as those from Illumina or MGI-Tech platforms).

`include_only_R1` represents additional in-built module. If `TRUE`, unassembled R1 reads will be included to the set of assembled reads per sample. This may be relevant when working with e.g. ITS2 sequences, because the ITS2 region in some taxa is too long for paired-end assembly using current short-read sequencing technology. Therefore longer ITS2 amplicon sequences are discarded completely after the assembly process. Thus, including also unassembled R1 reads (`include_only_R1 = TRUE`), partial ITS2 sequences for these taxa will be represented in the final output. But when using *ITSx*, keep `only_full = FALSE` and `include_partial = 50`.

Fastq formatted paired-end data is supported. **Outputs** are fastq files in `assembled_out` directory.

vsearch

| Setting | Tooltip |
|------------------------|---|
| read_R1 | applies only for paired-end data. Identifier string that is common for all R1 reads (e.g. when all R1 files have ‘.R1’ string, then enter ‘\R1’. Note that backslash is only needed to escape dot regex; e.g. when all R1 files have ‘_R1’ string, then enter ‘_R1’) |
| min_overlap | minimum overlap between the merged reads |
| min_length | minimum length of the merged sequence |
| allow_merge_stagger | allow to merge staggered read pairs. Staggered pairs are pairs where the 3’ end of the reverse read has an overhang to the left of the 5’ end of the forward read. This situation can occur when a very short fragment is sequenced |
| include_only_R1 | include unassembled R1 reads to the set of assembled reads per sample |
| max_diffs | the maximum number of non-matching nucleotides allowed in the overlap region |
| max_Ns | discard sequences with more than the specified number of Ns |
| max_len | maximum length of the merged sequence |
| keep_disjoined | output reads that were not merged into separate FASTQ files |
| fastq_qmax | |
| 1.2. User guide | maximum quality score accepted when reading FASTQ files. The default is 41, which is usual for recent Sanger/Illumina 1.8+ files |

DADA2

Important: Here, dada2 will perform also denoising (function ‘dada’) before assembling paired-end data. Because of that, input sequences (in **fastq** format) must consist of only A/T/C/Gs.

| Setting | Tooltip |
|------------------------|---|
| read_R1 | <p>identifyer string that is common for all R1 reads (e.g. when all R1 files have '.R1' string, then enter '\.R1'.</p> <p>Note that backslash is only needed to escape dot regex; e.g. when all R1 files have '_R1' string, then enter '_R1'.)</p> |
| read_R2 | <p>identifyer string that is common for all R2 reads (e.g. when all R2 files have '.R2' string, then enter '\.R2'.</p> <p>Note that backslash is only needed to escape dot regex; e.g. when all R2 files have '_R1' string, then enter '_R2'.)</p> |
| samp_ID | <p>identifyer string that separates the sample name from redundant characters (e.g. file name = sample1.R1.fastq, then underscore '\' would be the 'identifier string' (sample name = sampl84));</p> <p>note that backslash is only needed to escape dot regex (e.g. when file name = sample1_R1.fastq then specify as '_')</p> |
| minOverlap | the minimum length of the overlap required for merging the forward and reverse reads |
| maxMismatch | the maximum mismatches allowed in the overlap region |
| trimOverhang | <p>if TRUE, overhangs in the alignment between the forwards and reverse read are trimmed off. Overhangs are when the reverse read extends past the start of the forward read, and vice-versa, as can happen when reads are longer than the amplicon and read into the other-direction primer region</p> |
| justConcatenate | <p>if TRUE, the forward and reverse-complemented reverse read are concatenated rather than merged, with a NNNNNNNNNNN (10 Ns) spacer inserted between them</p> |
| 1.2. User guide | 43 |
| pool | denoising setting. If TRUE, the algorithm will pool |

1.2.15 CHIMERA FILTERING

Perform de-novo and reference database based chimera filtering.

Chimera filtering is performed by **sample-wise approach** (i.e. each sample (input file) is treated separately).

Fastq/fastq formatted single-end data is supported [fastq inputs will be converted to fasta].

Outputs are fasta files in `chimera_Filtered_out` directory.

uchime_denovo

Perform chimera filtering with **uchime_denovo** and **uchime_ref** algorithms in [vsearch](#)

| Setting | Tooltip |
|------------------------------|--|
| <code>pre_cluster</code> | identity percentage when performing ‘pre-clustering’ with <code>-cluster_size</code> for denovo chimera filtering with <code>-uchime_denovo</code> |
| <code>min_unique_size</code> | minimum amount of a unique sequences in a fasta file. If value = 1, then no sequences are discarded after dereplication; if value = 2, then sequences, which are represented only once in a given file are discarded; and so on |
| <code>denovo</code> | if TRUE, then perform denovo chimera filtering with <code>-uchime_denovo</code> |
| <code>reference_based</code> | perform reference database based chimera filtering with <code>-uchime_ref</code> . Select fasta formatted reference database (e.g. UNITE for ITS reads). If <code>denovo = TRUE</code> , then reference based chimera filtering will be performed after denovo. |
| <code>abundance_skew</code> | the abundance skew is used to distinguish in a threeway alignment which sequence is the chimera and which are the parents. The assumption is that chimeras appear later in the PCR amplification process and are therefore less abundant than their parents. The default value is 2.0, which means that the parents should be at least 2 times more abundant than their chimera. Any positive value equal or greater than 1.0 can be used |
| <code>min_h</code> | minimum score (h). Increasing this value tends to reduce the number of false positives and to decrease sensitivity. Values ranging from 0.0 to 1.0 included are accepted |

uchime3_denovo

Perform chimera filtering with **uchime3_denovo** algorithm in [vsearch](#)

Designed for denoised amplicons.

uchime3_denovo can be applied also in [UNOISE3 clustering](#)

| Setting | Tooltip |
|------------------------------|--|
| <code>pre_cluster</code> | identity percentage when performing ‘pre-clustering’ with <code>-cluster_size</code> for denovo chimera filtering with <code>-uchime_denovo</code> |
| <code>min_unique_size</code> | minimum amount of a unique sequences in a fasta file. If value = 1, then no sequences are discarded after dereplication; if value = 2, then sequences, which are represented only once in a given file are discarded; and so on |
| <code>denovo</code> | if TRUE, then perform denovo chimera filtering with <code>-uchime_denovo</code> |
| <code>reference_based</code> | perform reference database based chimera filtering with <code>-uchime_ref</code> . Select fasta formatted reference database (e.g. UNITE for ITS reads). If <code>denovo = TRUE</code> , then reference based chimera filtering will be performed after denovo. |
| <code>abundance_skew</code> | the abundance skew is used to distinguish in a threeway alignment which sequence is the chimera and which are the parents. The assumption is that chimeras appear later in the PCR amplification process and are therefore less abundant than their parents. The default value is 2.0, which means that the parents should be at least 2 times more abundant than their chimera. Any positive value equal or greater than 1.0 can be used |
| <code>min_h</code> | minimum score (h). Increasing this value tends to reduce the number of false positives and to decrease sensitivity. Values ranging from 0.0 to 1.0 included are accepted |

1.2.16 ITS Extractor

When working with ITS amplicons, then extract ITS regions with [ITS Extractor](#) (Bengtsson-Palme et al. 2013)

Note: Note that for better detection of the 18S, 5.8S and/or 28S regions, keep the primers (i.e. do not use ‘CUT PRIMERS’)

Fastq/fastq formatted single-end data is supported [fastq inputs will be converted to fasta].

Outputs are fasta files in ITSx_out directory.

Note: To **START**, specify working directory under **SELECT WORKDIR** and the **sequence files extension**, but the read types (single-end or paired-end) and data format (demultiplexed or multiplexed) does not matter here (just click ‘Next’).

| Setting | Tooltip |
|------------|--|
| organisms | set of profiles to use for the search. Can be used to restrict the search to only a few organism groups types to save time, if one or more of the origins are not relevant to the dataset under study |
| regions | ITS regions to output (note that 'all' will output also full ITS region [ITS1-5.8S-ITS2]) |
| partial | if larger than 0, ITSx will save additional FASTA-files for full and partial ITS sequences longer than the specified cutoff value. If his setting is left to 0 (zero), it means OFF |
| e-value | domain e-value cutoff a sequence must obtain in the HMMER-based step to be included in the output |
| scores | domain score cutoff that a sequence must obtain in the HMMER-based step to be included in the output |
| domains | the minimum number of domains (different HMM gene profiles) that must match a sequence for it to be included in the output (detected as an ITS sequence). Setting the value lower than two will increase the number of false positives, while increasing it above two will decrease ITSx detection abilities on fragmentary data |
| complement | if TRUE, ITSx checks both DNA strands for matches to HMM-profiles |
| only full | If TRUE, the output is limited to full-length ITS1 and ITS2 regions only |
| truncate | removes ends of ITS sequences if they are outside of the ITS region. If FALSE, the whole input sequence is saved |

1.2.17 CLUSTERING

Cluster sequences, generate OTUs or zOTUs (with UNOISE3)

Supported file format for the input data is **fasta**.

Outputs are **OTUs.fasta**, **OTU_table.txt** and **OTUs.uc** files in **clustering_out** directory.

Note: output OTU table is tab delimited text file.

vsearch

| Setting | Tooltip |
|------------------------|--|
| OTU_type | centroid” = output centroid sequences; “consensus” = output consensus sequences |
| similarity_threshold | define OTUs based on the sequence similarity threshold; 0.97 = 97% similarity threshold |
| strands | when comparing sequences with the cluster seed, check both strands (forward and reverse complementary) or the plus strand only |
| remove_singletons | if TRUE, then singleton OTUs will be discarded (OTUs with only one sequence) |
| similarity_type | pairwise sequence identity definition <code>-iddef</code> |
| sequence_sorting | size = sort the sequences by decreasing abundance; “length” = sort the sequences by decreasing length (<code>-cluster_fast</code>); “no” = do not sort sequences (<code>-cluster_smallmem</code> <code>-usersort</code>) |
| centroid_type | “similarity” = assign representative sequence to the closest (most similar) centroid (distance-based greedy clustering); “abundance” = assign representative sequence to the most abundant centroid (abundance-based greedy clustering; <code>-sizeorder</code>), <code>max_hits</code> should be > 1 |
| max_hits | maximum number of hits to accept before stopping the search (should be > 1 for abundance-based selection of centroids [centroid type]) |
| mask | |
| 1.2. User guide | mask regions in sequences using the “dust” method, or do not mask (“none”) 51 |
| dbmask | |

UNOISE3, with vsearch

| Setting | Tooltip |
|----------------------|---|
| zOTUs_thresh | sequence similarity threshold for zOTU table creation; 1 = 100% similarity threshold for zOTUs |
| similarity_threshold | optionally cluster zOTUs to OTUs based on the sequence similarity threshold; if id = 1, no OTU clustering will be performed |
| similarity_type | pairwise sequence identity definition for OTU clustering <code>-iddef</code> |
| maxaccepts | maximum number of hits to accept before stopping the search |
| maxrejects | maximum number of non-matching target sequences to consider before stopping the search |
| mask | mask regions in sequences using the “dust” method, or do not mask (“none”) |
| strands | when comparing sequences with the cluster seed, check both strands (forward and reverse complementary) or the plus strand only |
| minsize | minimum abundance of sequences for denoising |
| unoise_alpha | alpha parameter to the vsearch <code>-cluster_unoise</code> command. default = 2.0. |
| denoise_level | at which level to perform denoising; global = by pooling samples, individual = independently for each sample (if samples are denoised individually, reducing minsize to 4 may be more reasonable for higher sensitivity) |
| remove_chimeras | |

1.2.18 POSTCLUSTERING

Perform OTU post-clustering. Merge co-occurring ‘daughter’ OTUs.

LULU

LULU description from the [LULU repository](#): the purpose of LULU is to reduce the number of erroneous OTUs in OTU tables to achieve more realistic biodiversity metrics. By evaluating the co-occurrence patterns of OTUs among samples LULU identifies OTUs that consistently satisfy some user selected criteria for being errors of more abundant OTUs and merges these. It has been shown that curation with LULU consistently result in more realistic diversity metrics.

Additional information:

- [LULU repository](#)
- [LULU paper](#)

Input data is tab delimited **OTU table** (table) and **OTU sequences** (rep_seqs) in fasta format (see input examples below).

[EXAMPLE table here](#) (from *LULU repository*)

[EXAMPLE fasta here](#) (from *LULU repository*)

Note: To **START**, specify working directory under SELECT WORKDIR, but the file formats do not matter here (just click ‘Next’).

Output files in lulu_out directory:

lulu_out_table.txt = curated table in tab delimited txt format

lulu_out_RepSeqs.fasta = fasta file for the molecular units (OTUs or ASVs) in the curated table

match_list.lulu = match list file that was used by LULU to merge ‘daughter’ molecular units

discarded_units.lulu = molecular units (OTUs or ASVs) that were merged with other units based on specified thresholds)

| Setting | Tooltip |
|-------------------------|---|
| table | select OTU/ASV table. If no file is selected, then PipeCraft will look OTU_table.txt or ASV_table.txt in the working directory. EXAMPLE table here |
| rep_seqs | select fasta formatted sequence file containing your OTU/ASV reads. EXAMPLE file here |
| min_ratio_type | sets whether a potential error must have lower abundance than the parent in all samples 'min' (default), or if an error just needs to have lower abundance on average 'avg' |
| min_ratio | set the minimum abundance ratio between a potential error and a potential parent to be identified as an error |
| min_match | specify minimum threshold of sequence similarity for considering any OTU as an error of another |
| min_rel_cooccurrence | minimum co-occurrence rate. Default = 0.95 (meaning that 1 in 20 samples are allowed to have no parent presence) |
| match_list_soft | use either 'blastn' or 'vsearch' to generate match list for LULU. Default is 'vsearch' (much faster) |
| vsearch_similarity_type | applies only when 'vsearch' is used as 'match_list_soft'. Pairwise sequence identity definition (-iddef) |
| perc_identity | percent identity cutoff for match list. Excluding pairwise comparisons with lower sequence identity percentage than specified threshold |
| 1.2. User guide | 55 |
| coverage_perc | percent query coverage per hit. Excluding pairwise |

DADA2 collapse ASVs

DADA2 `collapseNoMismatch` function to collapse identical ASVs; and ASVs filtering based on minimum accepted sequence length (custom R functions).

To **START**, specify working directory under `SELECT WORKDIR`, but the file formats do not matter here (just click 'Next').

Output files in `filtered_table` directory:

`ASVs_table_collapsed.txt` = ASV table after collapsing identical ASVs

`ASVs_collapsed.fasta` = ASV sequences after collapsing identical ASVs

`ASV_table_collapsed.rds` = ASV table in RDS format after collapsing identical ASVs.

If length filtering was applied (if 'by length' setting > 0) [performed after collapsing identical ASVs]:

`ASV_table_lenFilt.txt` = ASV table after filtering out ASVs with shorter than specified sequence length

`ASVs_lenFilt.fasta` = ASV sequences after filtering out ASVs with shorter than specified sequence length

| Setting | Tooltip |
|--------------------|---|
| DADA2 table | select the RDS file (ASV table), output from DADA2 workflow; usually in ASVs_out.dada2/ASVs_table.denoised-merged.rds |
| collapseNoMismatch | collapses ASVs that are identical up to shifts or length variation, i.e. that have no mismatches or internal indels |
| by_length | discard ASVs from the ASV table that are shorter than specified value (in base pairs). Value 0 means OFF, no filtering by length |
| minOverlap | collapseNoMismatch setting. Default = 20. The minimum overlap of base pairs between ASV sequences required to collapse them together |
| vec | collapseNoMismatch setting. Default = TRUE. Use the vectorized aligner. Should be turned off if sequences exceed 2kb in length |

1.2.19 ASSIGN TAXONOMY

Implemented tools for taxonomy annotation:

BLAST (Camacho et al. 2009)

BLAST search sequences against selected *database*.

Important: BLAST database needs to be an unzipped fasta file in a separate folder (fasta will be automatically converted to BLAST database files). If converted BLAST database files (.ndb, .nhr, .nin, .not, .nsq, .ntf, .nto) already exist, then just SELECT **one** of those files as BLAST database in 'ASSIGN TAXONOMY' panel.

Supported file format for the input data is **fasta**.

Output files in ``taxonomy_out`` directory:

BLAST_1st_best_hit.txt = BLAST results for the 1st best hit in the used database.

BLAST_10_best_hits.txt = BLAST results for the 10 best hits in the used database.

Note: To **START**, specify working directory under SELECT WORKDIR and the **sequence files extension** (to look for input OTUs/ASVs fasta file), but the read types (single-end or paired-end) and data format (demultiplexed or multiplexed) does not matter here (just click 'Next').

Note: BLAST values filed separator is '+'. When pasting the taxonomy results to e.g. Excel, then first denote '+' as as filed separator to align the columns.

| Setting | Tooltip |
|---------------|---|
| database_file | select a database file in fasta format. Fasta format will be automatically converted to BLAST database |
| task | BLAST search settings according to blastn or megablast |
| strands | query strand to search against database. Both = search also reverse complement |
| e_value | a parameter that describes the number of hits one can expect to see by chance when searching a database of a particular size. The lower the e-value the more 'significant' the match is |
| word_size | the size of the initial word that must be matched between the database and the query sequence |
| reward | reward for a match |
| penalty | penalty for a mismatch |
| gap_open | cost to open a gap |
| gap_extend | cost to extend a gap |

DADA2 classifier

Classify sequences with DADA2 RDP naive Bayesian classifier (function `assignTaxonomy`) against selected *database*.

Supported file format for the input data is **fasta**.

Output files in ``taxonomy_out.dada2`` directory:
taxonomy.txt = classifier results with bootstrap values.

Note: To **START**, specify working directory under **SELECT WORKDIR** and the **sequence files extension** (to look for input OTUs/ASVs fasta file), but the read types (single-end or paired-end) and data format (demultiplexed or multiplexed) does not matter here (just click 'Next').

| Setting | Tooltip |
|----------------|--|
| dada2_database | select a reference database fasta file for taxonomy annotation |
| minBoot | the minimum bootstrap confidence for assigning a taxonomic level |
| tryRC | the reverse-complement of each sequences will be used for classification if it is a better match to the reference sequences than the forward sequence |

Sequence databases

A (*noncomprehensive*) list of public databases available for taxonomy annotation

| Database | Version | Description (click to download) |
|---|---------|--|
| UNITE | 8.3 | ITS region, all Eukaryotes |
| SILVA | 138.1 | 16S/18S (SSU), Bacteria, Archaea and Eukarya |
| SILVA 99% | 138.1 | 16S/18S (SSU), Bacteria, Archaea and Eukarya |
| MIDORI | 246 | Eukaryota mitochondrial genes |
| COI Classifier | 4 | Metazoa COI |
| DADA2-formatted reference databases | | DADA2-formatted reference databases |
| DIAT.BARCODE database | | rbcL/18S, diatoms |

1.2.20 POSTPROCESSING

Post-processing tools. *See this page*

1.2.21 Expert-mode (PipeCraft2 console)

Bioinformatic tools used by PipeCraft2 are stored on [Dockerhub](#) as Docker images. These images can be used to launch any tool with the Docker CLI to utilize the compiled tools. Especially useful in Windows OS, where majority of implemented modules are not compatible.

See list of docker images with implemented software here

Show a list of all images in your system (using e.g. **Expert-mode**):

```
docker images
```

Download an image if required (from [Dockerhub](#)):

Listing 1: docker pull pipecraft/IMAGE:TAG

```
docker pull pipecraft/vsearch:2.18
```

Delete an image

Listing 2: docker rmi IMAGE

```
docker rmi pipecraft/vsearch:2.18
```

Run docker container in your working directory to access the files. Outputs will be generated into the specified working directory. Specify the working directory under the -v flag:

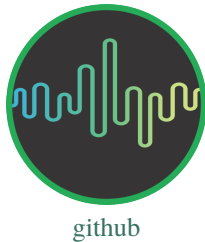
```
docker run -i --tty -v users/Tom/myFiles/:/Files pipecraft/vsearch:2.18
```

Once inside the container, move to /Files directory, which represents your working directory in the container; and run analyses

```
cd Files
vsearch --help
vsearch *--whateversettings*
```

Exit from the container:

```
exit
```



1.3 Walkthrough

Some example analyses pipelines.

Note: When samples of interest are distributed between different sequencing libraries, then first demultiplex (if needed) libraries separately and place samples of interest into separate working directory.

Contents

- *Walkthrough*
 - *Inspect quality profiles*
 - *Paired-end Illumina (or MGI-Tech) data*
 - * *Demultiplexed paired-end data; ASV workflow with DADA2*
 - *Examine the outputs*

- * *Demultiplexed paired-end data; OTU workflow*
 - *Examine the outputs*
- * *Multiplexed library*
- *Single-end (PacBio or assembled paired-end) data*

1.3.1 Inspect quality profiles

Examine the quality profiles and basic statistics of the your data set using **QualityCheck module**. [See here](#).

1.3.2 Paired-end Illumina (or MGI-Tech) data

Example analyses of paired-end data. Starting with raw paired-end fastq files, finishing with ASV/OTU table and taxonomy table.

Demultiplexed paired-end data; ASV workflow with DADA2

Note: This tutorial follows [DADA2 Pipeline Tutorial](#).

Here, we perform example analyses of paired-end data using [mothur MiSeq SOP example data set](#). [Download example data set here](#) (35.1 Mb) and unzip it. This data set represents demultiplexed set (per-sample fastq files) of 16S rRNA gene V4 amplicon sequences where sample indexes and primers have already been removed.

- If working with multiplexed data, [see here](#).
- If you need to reorient reads (based on primer sequences), [see here](#).
- If you need to trim the primers/adapters, [see here](#).

Warning: Be sure that all sequences have **same orientation** (5'-3' or 3'-5') in your input data set(s)! If sequences are in **mixed orientation** (i.e. some sequences are recorded as 5'-3' and some as 3'-5'; as usually in the raw data), then exactly the same ASV may be reported twice, where one is just the reverse complementary ASV: 1) ASV with sequence orientation of 5'-3'; and 2) ASV with sequence orientation of 3'-5'. **Reorient sequences** based on primer sequence using REORIENT panel; [see here](#).

Important: When working with your own data, then please check that the paired-end data file names contain “R1” and “R2” strings (to correctly identify the paired-end reads by PipeCraft).

Example:

F3D0_S188_L001_R1_001.fastq

F3D0_S188_L001_R2_001.fastq


1. Select working directory by pressing the 'SELECT WORKDIR' button.

Specify

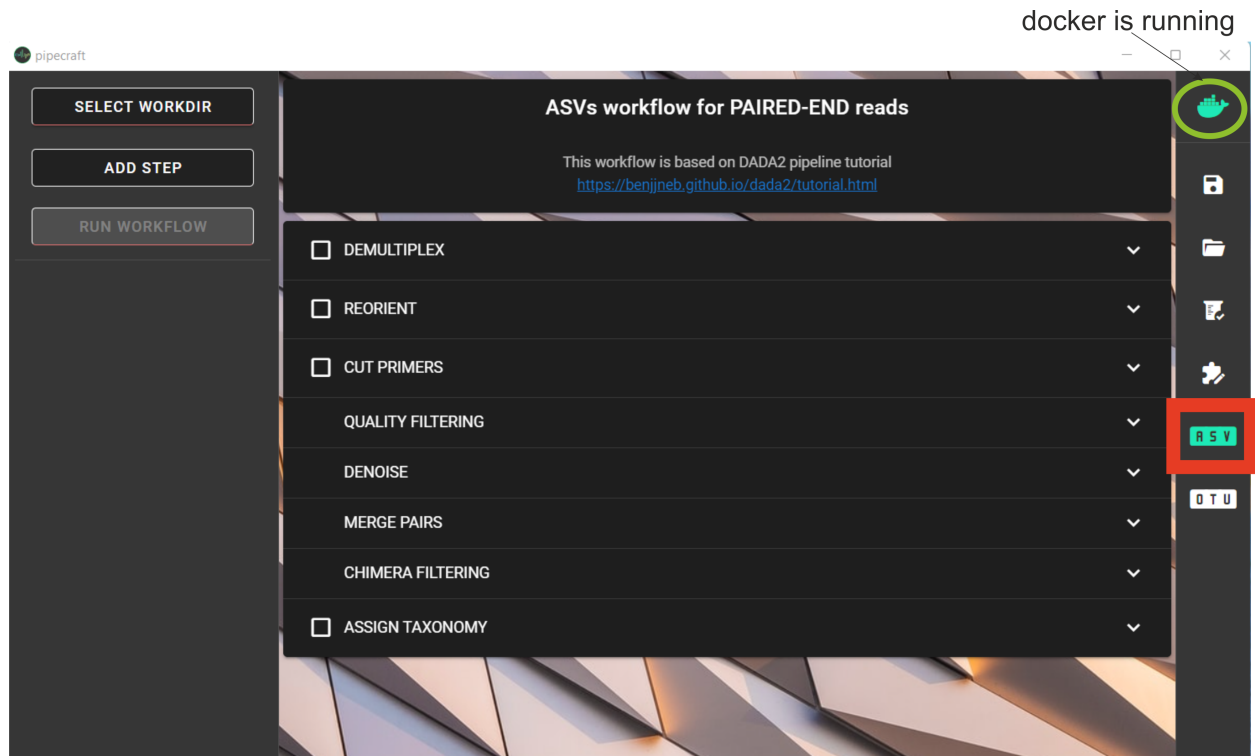
sequencing data format as **demultiplexed**;

sequence files extension as ***.fastq**;

sequencing read types as **paired-end**.

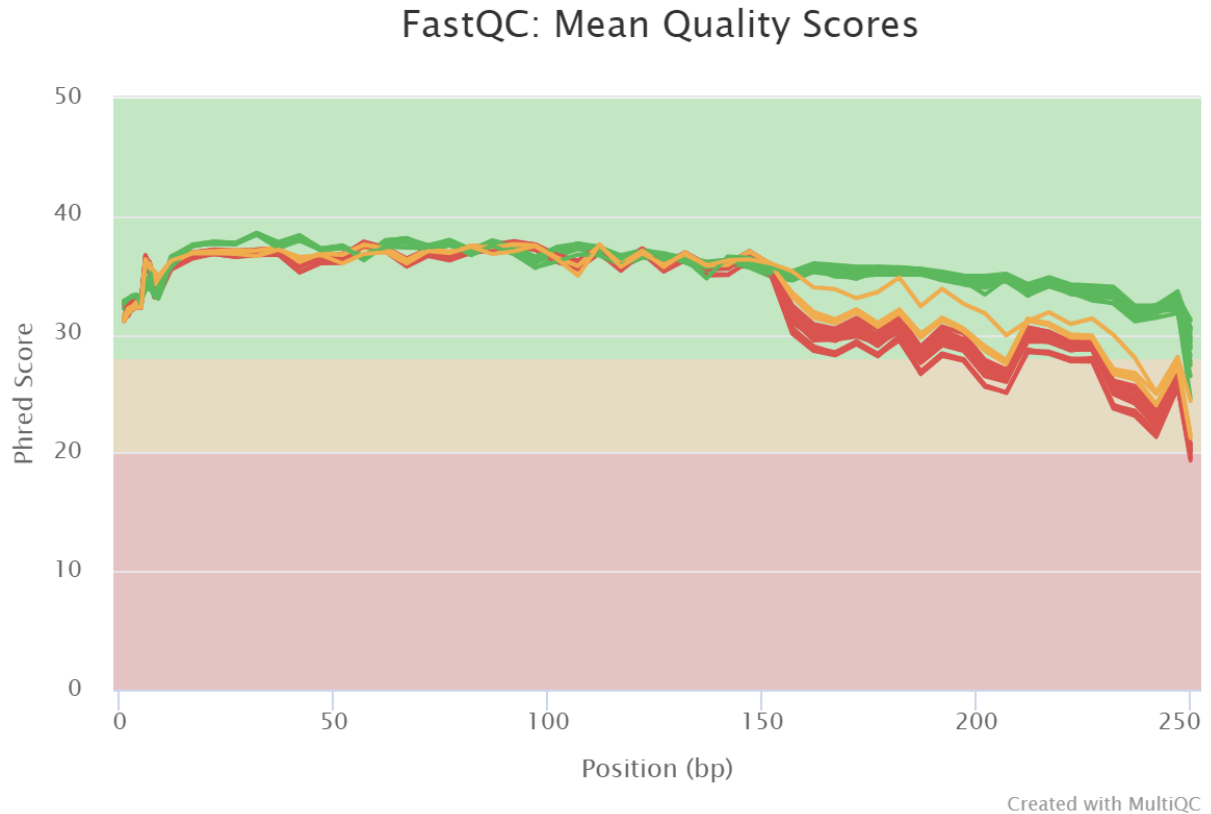
2. Select 'ASVs workflow' panel (right-ribbon) **and** check that docker **is** running (green  icon);

- Here, working with demultiplexed data, where primers have already been removed; so **do not tick** DEMULTIPLEX, REORIENT, CUT PRIMERS ([see here](#) to analyse multiplexed data, and [here](#) if you need to cut primers/adapters).



3. 'QUALITY FILTERING'

Before adjusting quality filtering settings, let's have a look on the **quality profile** of our example data set. Below quality profile plot was generated using QualityCheck panel ([see here](#)).



In this case, all **R1 files are represented with green lines**, indicating good average quality per file. However, all **R2 files are either yellow or red**, indicating a drop in quality scores. Lower qualities of R2 reads are characteristic for Illumina sequencing data, and is not too alarming. DADA2 algorithm is robust to lower quality sequences, but removing the low quality read parts will improve the DADA2 sensitivity to rare sequence variants.

- **Click on QUALITY FILTERING to expand the panel**
- specify identifier strings for read R1 and read R2. Here, fastq file names = F3D0_S188_L001_R1_001.fastq, F3D0_S188_L001_R2_001.fastq etc...; **_R1** and **_R2** are common identifiers for all files.
- specify samp ID (sample identifier). Here _ (underscore), which denotes that sample name is a string before the first _ in the fastq file name.
- trim reads to specified length to remove low quality ends. Set truncLen to 240 for trimming R1 reads and truncLen R2 to 160 to trim R2 reads. Latter positions represent the approximate positions where sequence quality drps notably (quality profile figure above). Be sure to consider the amplicon length before applying truncLen options, so that R1 and R2 reads would still overlap for the MERGE PAIRS process.
- other settings as default.

QUALITY FILTERING

| | | |
|----------------|-----------------|--------------------|
| read R1 _R1 | read R2 _R2 | samp ID - |
| maxEE 2 | maxN 0 | minLen 20 |
| truncQ 2 | truncLen 240 | truncLen R2 160 |

(click on the image for enlargement)

This step performs quality filtering.

Quality filtering settings [here](#)

Output directory = qualFiltered_out:

*_filt.fastq = quality filtered sequences per sample in FASTQ format

seq_count_summary.txt = summary of sequence counts per sample

FASTA/*_filt.fasta = quality filtered sequences per sample in FASTA format

4. Here, we use default 'DENOISE' and 'MERGE PAIRS' settings

This step performs denoising and merging of paired-end sequences.

Denoise settings : [here](#), merge pairs settings [here](#))

Output directory = denoised_assembled.dada2.

*.merged_ASVs.fasta = denoised and assembled ASVs per sample. 'Size' denotes the abundance of the ASV sequence

Error_rates_R1.pdf = plots for estimated R1 error rates

Error_rates_R2.pdf = plots for estimated R2 error rates

seq_count_summary.txt = summary of sequence and ASV counts per sample

5. Default settings for 'CHIMERA FILTERING'

(method = consensus)

This step performs chimera filtering on denoised and merged reads.

ASV table is generated during this step

Chimera filtering settings [here](#)

Output directories ->

chimeraFiltered_out.dada2:

*.chimFilt_ASVs.fasta = chimera filtered ASVs per sample. 'Size' denotes the abundance of the ASV sequence.

seq_count_summary.txt = summary of sequence and ASV counts per sample

*.chimeras.fasta = ASVs per sample that were flagged as chimeras (and thus discarded)

ASVs_out.dada2:

ASVs_table.txt = ASV distribution table per sample (tab delimited file)

ASVs.fasta = FASTA formatted representative ASV sequences (this file is used for taxonomy assignment)

6. 'ASSIGN TAXONOMY'

- Click on 'ASSIGN TAXONOMY' to expand the panel
- press **DOWNLOAD DATABASES** which direct you to the DADA2-formatted reference databases [web page](#).
- download SILVA (silva_nr99_v138.1_wSpecies_train_set.fa.gz) database for assigning taxonomy to our 16S ASVs. [Download link here](#)
- specify the location of your downloaded DADA2 database by pressing **SELECT FASTA**
- since primers were already removed from this data set, we could not *reorient all sequences to uniform orientation as based on primers*. Therefore, **switch ON** tryRC to include reverse-complement search.

This step assigns taxonomy to ASVs using DADA2 [assignTaxonomy](#) function.

Assign taxonomy settings [here](#)

Output directory = taxonomy_out.dada2:
taxonomy.txt = classifier results with bootstrap values

6.1. Save the workflow by pressing ``SAVE WORKFLOW`` button on the right-ribbon.

7. Press** 'RUN WORKFLOW' **to start the analyses.

Note: When running the panel for the first time, a docker image will be pulled first to start the process.

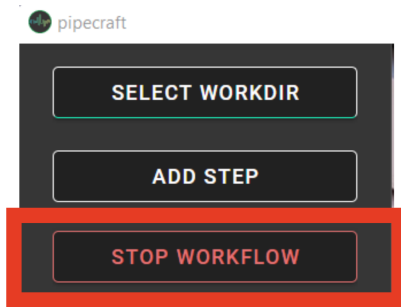
When done, 'workflow finished' window will be displayed.

Workflow finished

OK

Note:

Press **STOP WORKFLOW** to stop.



->

Examine the outputs

Several process-specific output folders are generated:

qualFiltered_out -> quality filtered paired-end **fastq** files per sample
denoised_assembled.dada2 -> denoised and assembled **fasta** files per sample (and error rate plots)
chimeraFiltered_out.dada2 -> chimera filtered **fasta** files per sample
ASVs_out.dada2 -> **ASVs table** (ASVs_table.txt), and ASV sequences (ASVs.fasta) file

taxonomy_out.dada2→ ASVs **taxonomy table** (taxonomy.txt)

Each folder (except ASVs_out.dada2 and taxonomy_out.dada2) contain **summary of the sequence counts** (seq_count_summary.txt). Examine those to track the read counts throughout the pipeline.

For example, merging the seq_count_summary.txt file in qualFiltered_out with the seq_count_summary.txt file from chimeraFiltered_out.dada2 forms a table for examining sequence counts throughout the pipeline and number of ASVs per sample.

| sample | input | qualFiltered | merged | chimeraFiltered | no.of ASVs |
|--------|-------|--------------|--------|-----------------|------------|
| F3D0 | 7793 | 7113 | 6540 | 6528 | 106 |
| F3D141 | 5958 | 5463 | 4986 | 4863 | 74 |
| F3D142 | 3183 | 2914 | 2595 | 2521 | 48 |
| F3D143 | 3178 | 2941 | 2552 | 2518 | 56 |
| F3D144 | 4827 | 4312 | 3627 | 3488 | 47 |
| F3D145 | 7377 | 6741 | 6079 | 5820 | 72 |
| F3D146 | 5021 | 4560 | 3968 | 3879 | 84 |
| F3D147 | 17070 | 15637 | 14231 | 13006 | 103 |
| F3D148 | 12405 | 11413 | 10529 | 9935 | 97 |
| F3D149 | 13083 | 12017 | 11154 | 10653 | 112 |
| F3D150 | 5509 | 5032 | 4349 | 4240 | 78 |
| F3D1 | 5869 | 5299 | 5028 | 5017 | 100 |
| F3D2 | 19620 | 18075 | 17431 | 16835 | 134 |
| F3D3 | 6758 | 6250 | 5853 | 5491 | 68 |
| F3D5 | 4448 | 4052 | 3716 | 3716 | 86 |
| F3D6 | 7989 | 7369 | 6865 | 6679 | 90 |
| F3D7 | 5129 | 4765 | 4428 | 4217 | 61 |
| F3D8 | 5294 | 4871 | 4576 | 4547 | 99 |
| F3D9 | 7070 | 6504 | 6092 | 6015 | 106 |
| Mock | 4779 | 4314 | 4269 | 4269 | 20 |

ASVs_out.dada2 directory contains **ASVs table** (ASVs_table.txt), where the **1st column** represents ASV identifiers, **2nd column** representative sequences of ASVs, and all following columns represent samples (number of sequences per ASV in a sample). This is tab delimited text file.

ASVs_table.txt; first 4 samples

| ASV | Sequence | F3D0 | F3D141 | F3D142 | F3D143 |
|-------|----------------|------|--------|--------|--------|
| ASV_1 | TACGGAGGATG... | 579 | 444 | 289 | 228 |
| ASV_2 | TACGGAGGATG... | 345 | 362 | 304 | 176 |
| ASV_3 | TACGGAGGATG... | 449 | 345 | 158 | 204 |
| ASV_4 | TACGGAGGATG... | 430 | 502 | 164 | 231 |
| ASV_5 | TACGGAGGATC... | 154 | 189 | 180 | 130 |
| ASV_6 | TACGGAGGATG... | 470 | 331 | 181 | 244 |
| ASV_7 | TACGGAGGATG... | 282 | 243 | 163 | 152 |
| ASV_8 | TACGGAGGATT... | 184 | 321 | 89 | 83 |
| ASV_9 | TACGGAGGATG... | 45 | 167 | 89 | 109 |

The **ASV sequences** are represented also in the fasta file (ASVs.fasta) in ASVs_out.dada2 directory.

Result from the taxonomy annotation process - **taxonomy table** (taxonomy.txt) - is located at the taxonomy_out.dada2 directory. "NA" denotes that the ASV was not assigned to corresponding taxonomic unit. Last columns with integers (for 'Kingdom' to 'Species') represent bootstrap values for the corresponding taxonomic unit.

taxonomy.txt; first 10 ASVs

| ASV | Se- quence | King- dom | Phy- lum | Class | Or- der | Fam- ily | Genus | Species | King- dom | Phy- lum | Class | Or- der | Fam- ily | Genus | Species |
|--------|------------------------------|----------------------|-------------------|--------------------------|-------------------------------|--------------------------|-------|---------|--------------|-------------|-------|------------|-------------|-------|---------|
| ASV_1 | TACG-Bac- GAG..te- ria | Bac- teroidota | Bac- teroidota | Bac- teroidota | Murib- alaceae | NA | NA | 100 | 100 | 100 | 100 | 100 | 100 | 100 | 100 |
| ASV_2 | TACG-Bac- GAG..te- ria | Bac- teroidota | Bac- teroidota | Bac- teroidota | Murib- alaceae | NA | NA | 100 | 100 | 100 | 100 | 100 | 100 | 100 | 100 |
| ASV_3 | TACG-Bac- GAG..te- ria | Bac- teroidota | Bac- teroidota | Bac- teroidota | Murib- alaceae | NA | NA | 100 | 100 | 100 | 100 | 100 | 100 | 100 | 100 |
| ASV_4 | TACG-Bac- GAG..te- ria | Bac- teroidota | Bac- teroidota | Bac- teroidota | Rikenel- laceae | Alistipes | NA | 100 | 100 | 100 | 100 | 100 | 100 | 100 | 100 |
| ASV_5 | TACG-Bac- GAG..te- ria | Bac- teroidota | Bac- teroidota | Bac- teroidota | Murib- alaceae | NA | NA | 100 | 100 | 100 | 100 | 100 | 100 | 100 | 100 |
| ASV_6 | TACG-Bac- GAG..te- ria | Bac- teroidota | Bac- teroidota | Bac- teroidota | Murib- alaceae | NA | NA | 100 | 100 | 100 | 100 | 100 | 95 | 95 | 95 |
| ASV_7 | TACG-Bac- TAG..te- ria | Fir- mi- cutes | Clostri- ales | Lach- nospira- les | Lach- nospira- les | Lach- nospira- les | NA | 100 | 100 | 100 | 100 | 100 | 100 | 100 | 99 |
| ASV_8 | TACG-Bac- GAG..te- ria | Bac- teroidota | Bac- teroidota | Bac- teroidota | Murib- alaceae | NA | NA | 100 | 100 | 100 | 100 | 100 | 100 | 100 | 100 |
| ASV_9 | TACG-Bac- GAG..te- ria | Bac- teroidota | Bac- teroidota | Bac- teroidota | Bac- teroidac- teroides | caec- imuris | 100 | 100 | 100 | 100 | 100 | 100 | 100 | 77 | 77 |
| ASV_10 | TACG-Bac- GAG..te- ria | Bac- teroidota | Bac- teroidota | Bac- teroidota | Murib- alaceae | NA | NA | 100 | 100 | 100 | 100 | 100 | 99 | 99 | 99 |

Demultiplexed paired-end data; OTU workflow

Note: Built-in panel for OTU workflow with (mostly) vsearch.

Here, we perform example analyses of paired-end data using [mothur MiSeq SOP example data set](#). [Download example data set here](#) (35.1 Mb) and unzip it. This data set represents demultiplexed set (per-sample fastq files) of 16S rRNA gene V4 amplicon sequences where sample indexes and primers have already been removed.

- If working with multiplexed data, [see here](#).
- If you need to trim the primers/adapters, [see here](#).

Note: When working with your own data, then consider **reorienting** reads; [see here](#). Although, in the OTU formation step (clustering), both sequence strands will be compared to generate OTUs, the time for BLAST (taxonomy annotation step) can be reduced when there is no need to search reverse complementary matches.

Important: When working with your own data, then please check that the paired-end data file names contain “R1” and “R2” strings (to correctly identify the paired-end reads by PipeCraft).

Example:

F3D0_S188_L001_R1_001.fastq

F3D0_S188_L001_R2_001.fastq


1. Select working directory by pressing the 'SELECT WORKDIR' button.

Specify

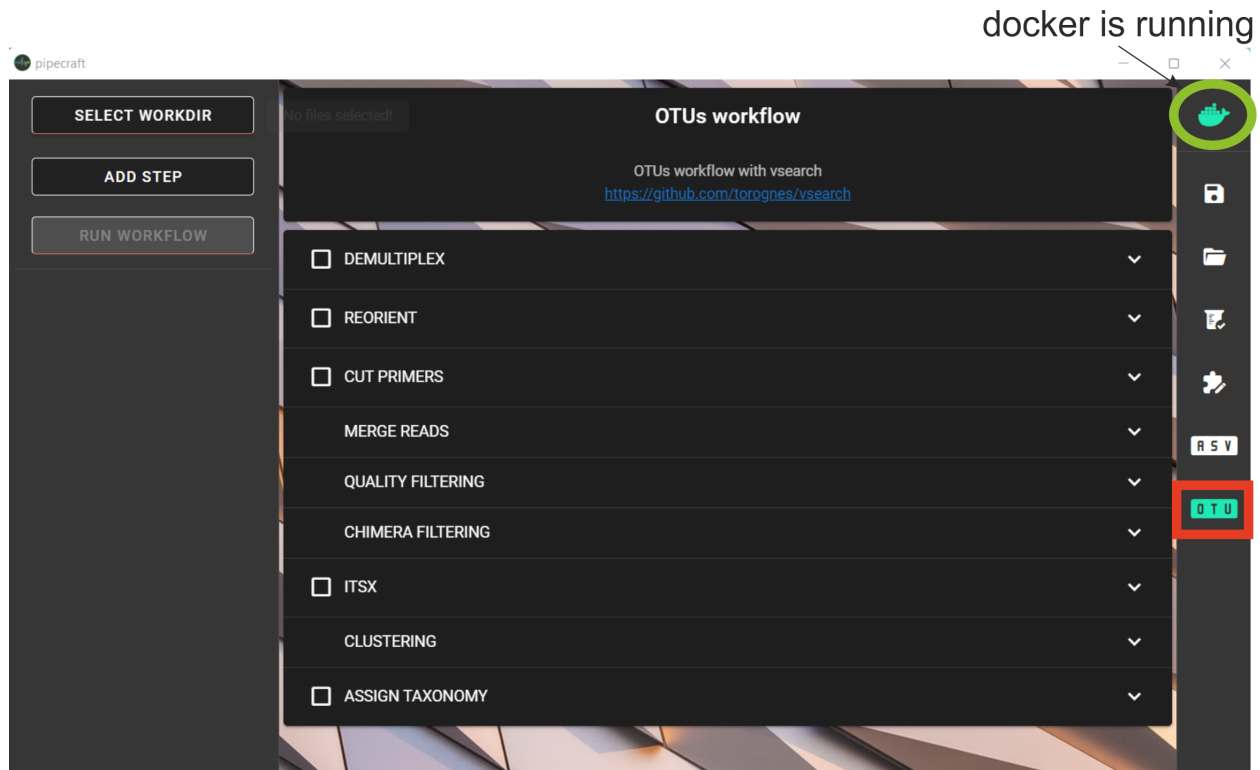
sequencing data format as **demultiplexed**;

sequence files extension as ***.fastq**;

sequencing read types as **paired-end**.

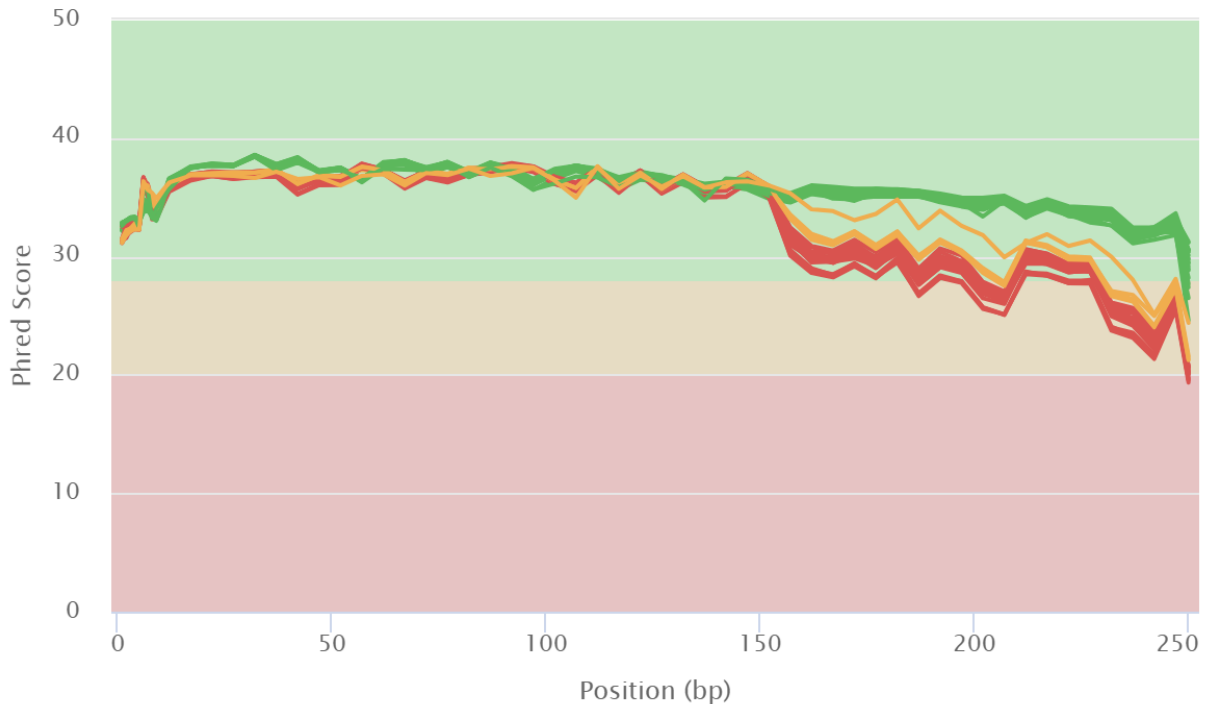
2. Select 'OTU workflow' panel (right-ribbon) **and** check that docker **is** running (green  icon);

- Here, working with demultiplexed data, where primers have already been removed; so **do not tick** DEMULTIPLEX, REORIENT, CUT PRIMERS ([see here](#) to analyse multiplexed data, and [here](#) if you need to cut primers/adapters).



Before proceeding, let's have a look on the **quality profile** of our example data set. Below quality profile plot was generated using QualityCheck panel ([see here](#)).

FastQC: Mean Quality Scores



In this case, all **R1 files are represented with green lines**, indicating good average quality per file. However, all **R2 files are either yellow or red**, indicating a drop in quality scores. Lower qualities of R2 reads are characteristic for Illumina sequencing data, and is not too alarming. Nevertheless, we need to quality filter the data set.

3. 'MERGE PAIRS'

- Here, we use default settings.

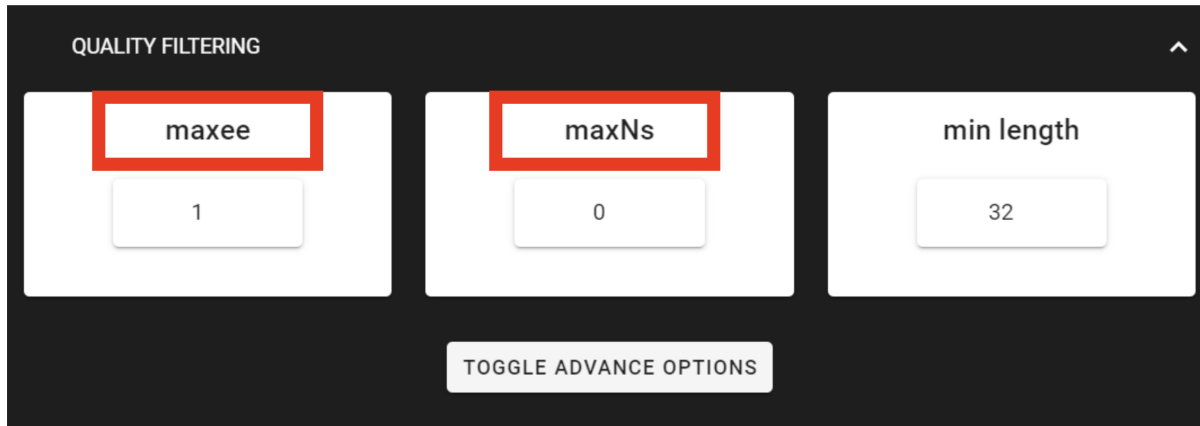
Note: If `include_only_R1` option = TRUE, then unassembled R1 reads will be included to the set of assembled reads per sample. This may be useful when working with e.g. ITS2 sequences, because the ITS2 region in some taxa is too long for paired-end assembly using current short-read sequencing technology. Therefore longer ITS2 amplicon sequences are discarded completely after the assembly process. Thus, including also unassembled R1 reads (`include_only_R1` = TRUE), partial ITS2 sequences for these taxa will be represented in the final output. But when using *ITSx*, `keep_only_full` = FALSE and `include_partial` = 50. | If include only R1 option = TRUE, then other specified options (length, max error rate etc.) have not been applied to R1 reads in the 'assembled' file. Thus, additional quality filtering (if this was done before assembling) should be run on the 'assembled' data. But in this built-in OTU workflow, the quality filtering step is anyway performed after merge pairs step.

*This step performs merging of paired-end sequences using `vsearch -fastq_mergepairs`.
Merge pairs settings [here](#))*

Output directory = `assembled_out`.

4. 'QUALITY FILTERING'

- **Click on QUALITY FILTERING to expand the panel**
- specify maxee (maximum number of expected errors per sequence), here we use 1 ([see here what is maxee](#)).
- specify maxNs (maximum number of Ns in the sequences). Here, we will discard any sequence that contains N (ambiguously recorded nucleotide) by setting the value to 0.
- other settings as default.



QUALITY FILTERING

maxee

1

maxNs

0

min length

32

TOGGLE ADVANCE OPTIONS

*This step performs quality filtering using vsearch.
vsearch quality filtering settings [here](#)*

Output directory = qualFiltered_out

5. 'CHIMERA FILTERING'

- **Click on CHIMERA FILTERING to expand the panel**
- specify pre cluster threshold as 0.97 (that is 97%; when planning to use 97% sequence similarity threshold also for clustering reads into OTUs).
- here, we perform only denovo chimera filtering
- other settings as default.

Note: Tick reference based if there is appropriate database for reference based chimera filtering (such as e.g. [UNITE](#) for ITS reads).

This step performs chimera filtering using vsearch
Chimera filtering settings [here](#)

Output directory = chimeraFiltered_out

6. Consideration when working **with** ITS data

Identify and extract the ITS regions using ITSx; [see here](#)

Note: because ITSx outputs multiple directories for different ITS sub-regions CLUSTERING and ASSIGN TAXONOMY will be disabled after 'ITS EXTRACTOR'. Select appropriate ITSx output folder for CLUSTERING after the process is finished ['ADD STEP' -> CLUSTERING (vsearch)].

This step extracts ITS reads using ITSx
ITSx settings [here](#)

Output directory = ITSx_out

7. 'CLUSTERING'

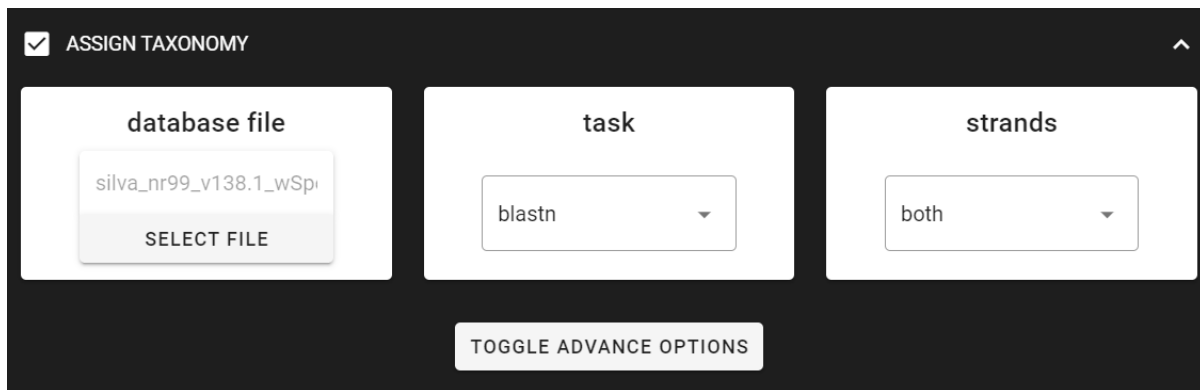
- Here, we use default settings by clustering the reads using 97% similarity threshold

This step performs clustering using vsearch.
vsearch clustering settings [here](#)

Output directory = clustering_out

8. 'ASSIGN TAXONOMY'

- Tick ASSIGN TAXONOMY to perform taxonomy assignment with BLAST
- download SILVA 99% database [here](#) (SILVA_138.1_SSURef_NR99_tax_silva.fasta.gz)
- **unzip** the downloaded database and place this into separate folder (to automatically make blast database from that fasta file)
- specify the location of your downloaded SILVA database by pressing SELECT FILE under 'database file' option
- since primers were already removed from this data set, we could not *reorient all sequences to uniform orientation as based on primers*. Therefore, **keep ON** the strands = both to include reverse-complement search.



This step assigns taxonomy to OTUs using BLAST
Assign taxonomy settings [here](#)

Output directory = taxonomy_out

8.1. Save the workflow by pressing ``SAVE WORKFLOW`` button on the right-ribbon.

1. Press** 'RUN WORKFLOW' **to start the analyses.

Note: When running the panel for the first time, a docker image will be pulled first to start the process.

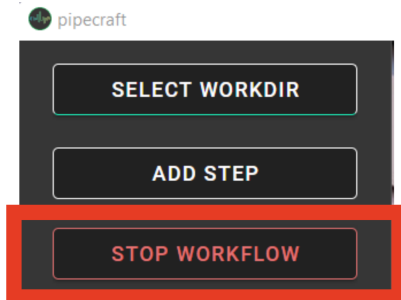
When done, 'workflow finished' window will be displayed.

Workflow finished

OK

Note:

Press **STOP WORKFLOW** to stop.



->

Examine the outputs

Several process-specific output folders are generated:

assembled_out -> assembled **fastq** files per sample

qualFiltered_out -> quality filtered **fastq** files per sample

chimeraFiltered_out -> chimera filtered **fasta** files per sample

clustering_out -> **OTU table** (OTU_table.txt), and OTU sequences (OTUs.fasta) file

taxonomy_out -> BLAST hits for the OTUs (BLAST_1st_best_hit.txt and BLAST_10_best_hits.txt)

Each folder (except clustering_out and taxonomy_out) contain **summary of the sequence counts** (seq_count_summary.txt). Examine those to track the read counts throughout the pipeline ([example here](#))

clustering_out directory contains **OTUs table** (OTUs_table.txt), where the **1st column** represents OTU identifiers, and all following columns represent samples (number of sequences per OTU in a sample). The **OTU sequences** are represented in the fasta file (OTUs.fasta) in clustering_out directory.

OTUs_table.txt; first 4 samples

| OTU_id | F3D0_S188_L00 | F3D1_S189_L00 | F3D2_S190_L00 | F3D3_S191_L00 |
|--|---------------|---------------|---------------|---------------|
| 00fc1569196587dde0462c7d806cc0577410fbfa | 271 | 584 | 20 | |
| 02d84ed0175c2c79e8379a99cffb6dbc7f686bd9 | 44 | 88 | 14 | |
| 0407ee3bd15ca7206a75d02bb41732516a3aaa88 | 4 | 3 | 0 | |
| 042e5f0b5e38dff09f7ad58b6849fb17ec5203b9 | 83 | 131 | 4 | |
| 07411b848fcda497fd29944d351b8a2ec7dc2bd4 | 0 | 2 | 0 | |
| 07e7806a732c67ef090b6b279b74a87feda18e8e | 22 | 83 | 7 | |
| 0836d270877aed22cd247f7e703b9247fb339127 | 1 | 0 | 0 | |
| 0aa6e7da5819c11973f186cb35b3f4f58275fb04 | 4 | 5 | 0 | |
| 0c1c219a4756bb729e5f0ceb7d82d932bb180c5e | 17 | 40 | 7 | |

Results from the taxonomy annotation process (BLAST) are located at the `taxonomy_out` directory (BLAST_1st_best_hit.txt and BLAST_10_best_hits.txt). **Blast values are separated by + and tab** [be sure to specify the delimiter when aligning columns in e.g. LibreOffice or Excel]. “NO_BLAST_HIT” denotes that the OTU sequence did not get any match against the selected database.

| blast values | |
|--------------|---|
| score | blast score |
| e-value | blast e-value |
| query len | query (i.e. OTU/ASV) sequence length |
| query start | start position of match in the query seq |
| query end | end position of match in the query seq |
| target len | target seq length in the database |
| target start | start position of match in the target seq |
| target end | end position of match in the target seq |
| align len | alignment length of query and target |
| identities | number of identical matches |
| gaps | number of gaps in the alignment |
| coverage | query coverage percentage against the target sequence (100 percent is full-length match; low coverage may indicate presence of chimeric sequence/OTU/ASV) |
| id | identity percentage against the target sequence |

Multiplexed library

Working with paired-end raw multiplexed data.

1. Select working directory by pressing the '`SELECT WORKDIR`' button.

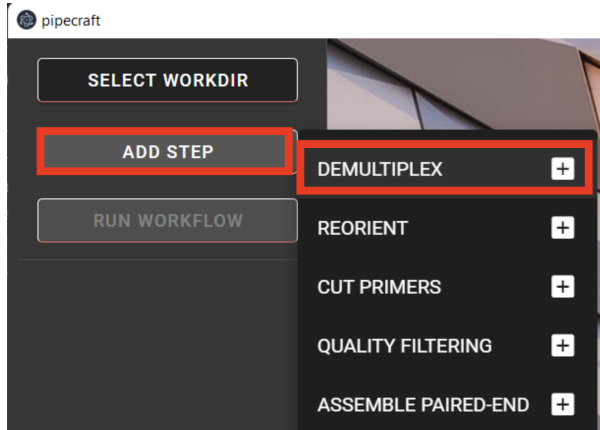
Specify

sequencing data format as **multiplexed**;

sequence files extension as **may be fastq or fasta** formatted files;
sequencing read types as **paired-end**.

2. 'DEMUTIPLEX'

2.1 Press ADD STEP -> DEMULTIPLEX

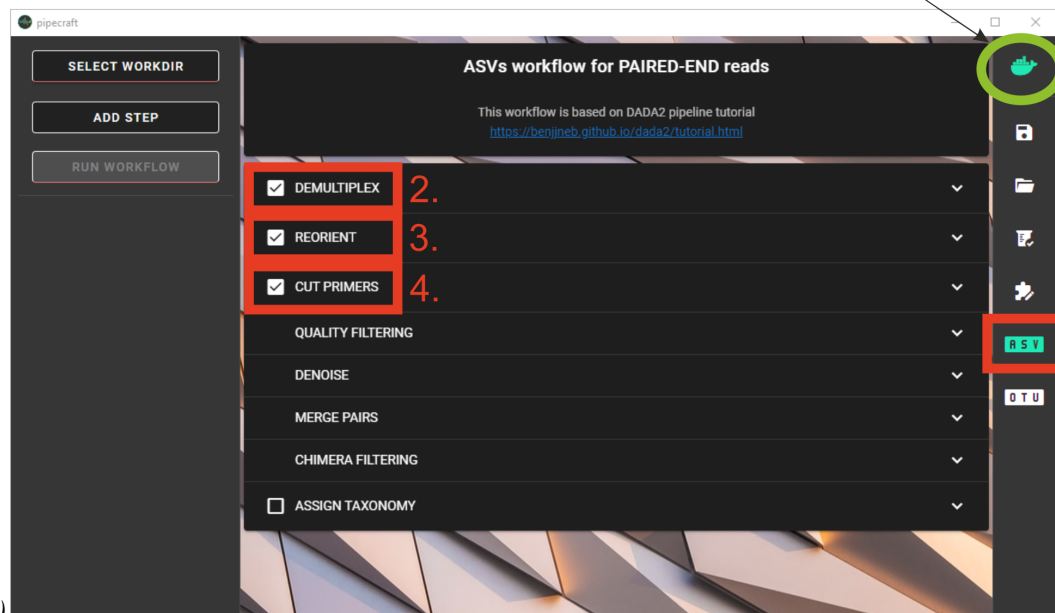


or

2.2. Select ASVs workflow or OTUs workflow panel

- tick DEMULTIPLEX, REORIENT and CUT PRIMERS;
- check that the docker is running (green icon [red = not running])

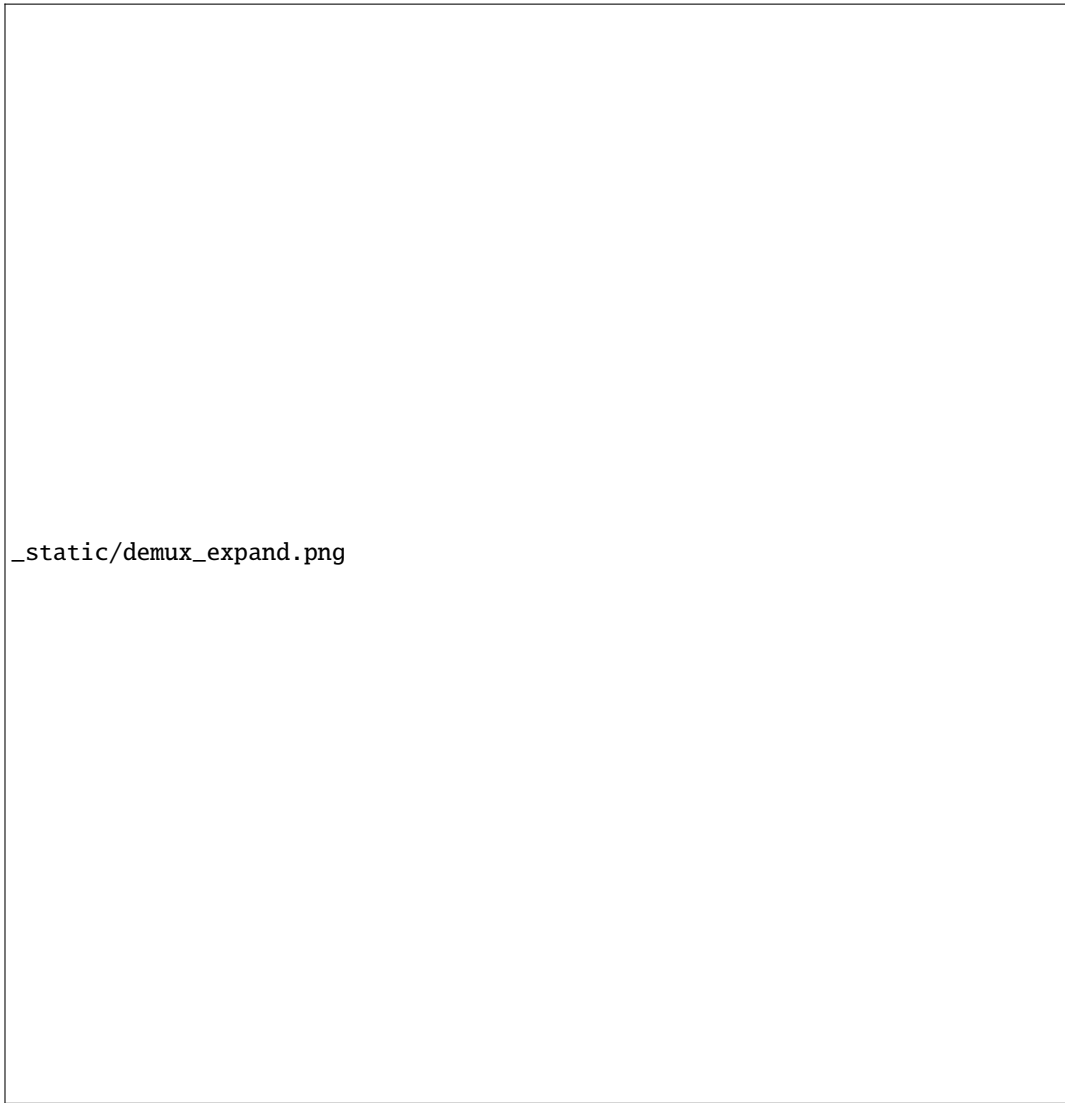
docker is running



(click on the image for enlargement)

3. Click on 'DEMULTIPLEX' to expand the panel

- select your FASTA formatted **index_file.fasta** (*general index file guide here*)
- adjust overlap setting to fully match the length (in base pairs) of the indexes in the index_file.fasta.



_static/demux_expand.png

(click on the image for enlargement)

This step distributes sequences to samples according to the information in the index_file.fasta. See *specifics here*

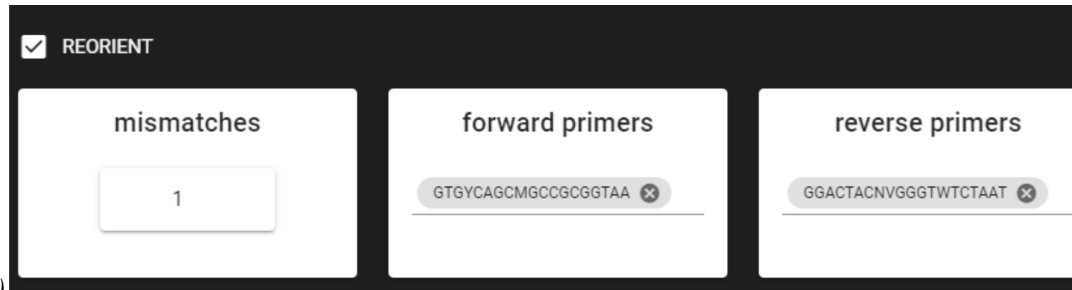
Output directory = demultiplex_out:

* fastq or fasta files per sample (as specified in the *index file*)

* unknown.fastq/fastq files contain sequences where specified index combinations were not found.

1. 'REORIENT'

- specify allowed mismatches during the primer search; >2 not recommended.
- specify forward primer: 5'-GTGYCAGCMGCCGCGGTAA-3' (example)
- specify reverse primer: 3'-GGCCGYCAATTYMTTTRAGTTT-5' (example)



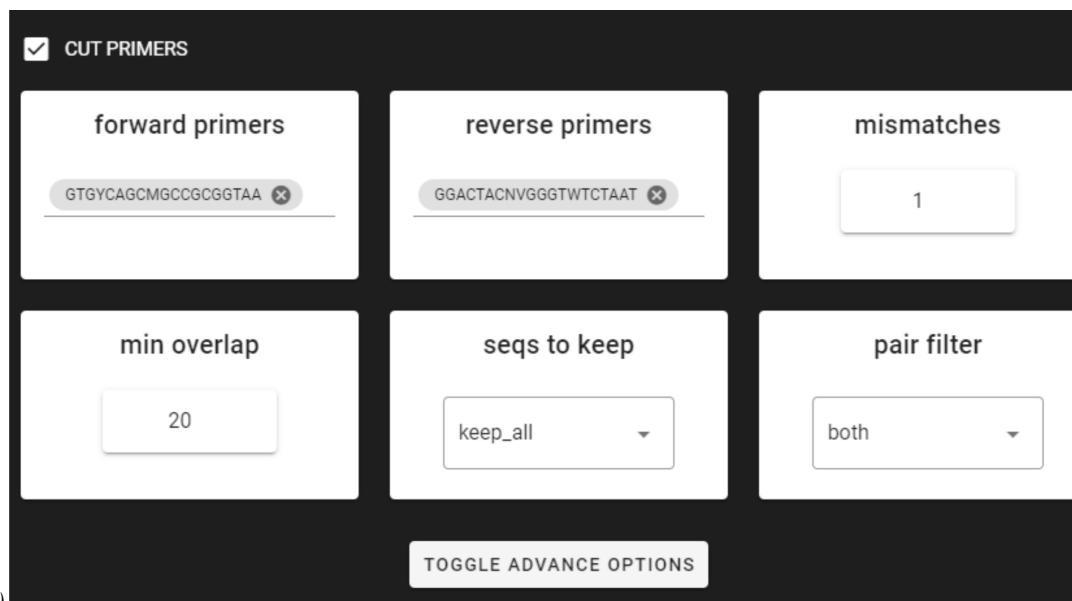
(click on the image for enlargement)

This step reorients sequences to 5'-3' as based on specified forward and reverse primers. See [specifics here](#)

Output directory = reorient_out

5. Click on 'CUT PRIMERS' to expand the panel

- specify forward primer: 5'-GTGYCAGCMGCCGCGGTAA-3' (example)
- specify reverse primer: 3'-GGCCGYCAATTYMTTTRAGTTT-5' (example)
- specify allowed mismatches during the primer search; >2 not recommended
- for paired-end reads keep seqs to keep and pair filter as default (**keep_all** and **both**, respectively)



(click on the image for enlargement)

This step clips specified primer sequences from the reads (if primers are found). See [specifics here](#). Discards the reads where primer sequences are not detected.

Output directory = primersCut_out

6. Follow the rest of the [ASV workflow](#) or [OTU workflow](#)

1.3.3 Single-end (PacBio or assembled paired-end) data

coming soon ...



1.4 Post-processing tools

Note: All post-processing tools accessible under **ADD STEP -> POSTPROCESSING**

Contents

- *Post-processing tools*
 - *LULU*
 - *DEICODE*

1.4.1 LULU

LULU description from the [LULU repository](#): the purpose of LULU is to reduce the number of erroneous OTUs in OTU tables to achieve more realistic biodiversity metrics. By evaluating the co-occurrence patterns of OTUs among samples LULU identifies OTUs that consistently satisfy some user selected criteria for being errors of more abundant OTUs and merges these. It has been shown that curation with LULU consistently result in more realistic diversity metrics.

This is implemented also under POSTCLUSTERING panel, [see here](#)

1.4.2 DEICODE

DEICODE (Martino et al., 2019) is used to perform beta diversity analysis by applying robust Aitchison PCA on the OTU/ASV table. To consider the compositional nature of data, it preprocesses data with rCLR transformation (centered log-ratio on only non-zero values, without adding pseudo count). As a second step, it performs dimensionality reduction of the data using robust PCA (also applied only to the non-zero values of the data), where sparse data are handled through matrix completion.

Additional information:

- [DEICODE tutorial](#)
- [DEICODE repository](#)
- [DEICODE paper](#)

Input data is tab delimited **OTU table** and optionally **subset of OTU ids** to generate results also for the selected subset (see input examples below).

Note: To **START**, specify working directory under SELECT WORKDIR, but the file formats do not matter here (just click 'Next').

Output files in DEICODE_out directory:

```
# - otutab.biom = full OTU table in BIOM format
# - rclr_subset.tsv = rCLR-transformed subset of OTU table *
# DEICODE_out/full/
# - distance-matrix.tsv = distance matrix between the samples, based on full OTU table
# - ordination.txt = ordination scores for samples and OTUs, based on full OTU table
# - rclr.tsv = rCLR-transformed OTU table
# DEICODE_out/subs/
# - distance-matrix.tsv = distance matrix between the samples, based on a subset of OTU table *
# - ordination.txt = ordination scores for samples and OTUs, based a subset of OTU table *
# *, files are present only if 'subset_IDS' variable was specified
```

| Setting | Tooltip |
|------------------|--|
| table | select OTU/ASV table. If no file is selected, then PipeCraft will look OTU_table.txt or ASV_table.txt in the working directory. See OTU table example below |
| subset_IDs | select list of OTU/ASV IDs for analysing a subset from the full table see subset_IDs file example below |
| min_otu_reads | cutoff for reads per OTU/ASV. OTUs/ASVs with lower reads than specified cutoff will be excluded from the analysis |
| min_sample_reads | cutoff for reads per sample. Samples with lower reads than specified cutoff will be excluded from the analysis |

Example of input table (tab delimited text file):

| OTU_id | sample1 | sample2 | sample3 | sample4 |
|-------------------|---------|---------|---------|---------|
| 00fc1569196587dde | 106 | 271 | 584 | 20 |
| 02d84ed0175c2c79e | 81 | 44 | 88 | 14 |
| 0407ee3bd15ca7206 | 3 | 4 | 3 | 0 |
| 042e5f0b5e38dff09 | 20 | 83 | 131 | 4 |
| 07411b848fcda497f | 1 | 0 | 2 | 0 |
| 07e7806a732c67ef0 | 18 | 22 | 83 | 7 |
| 0836d270877aed22c | 1 | 1 | 0 | 0 |
| 0aa6e7da5819c1197 | 1 | 4 | 5 | 0 |
| 0c1c219a4756bb729 | 18 | 17 | 40 | 7 |

Example of input subset_IDs:

```
07411b848fcda497f
042e5f0b5e38dff09
0836d270877aed22c
0c1c219a4756bb729
...
```

PERMANOVA and PERMDISP example using the robust Aitchison distance

```
library(vegan)

## Load distance matrix
dd <- read.table(file = "distance-matrix.tsv")

## You will also need to load the sample metadata
## However, for this example we will create a dummy data
meta <- data.frame(
  SampleID = rownames(dd),
  TestData = rep(c("A", "B", "C"), each = ceiling(nrow(dd)/3))[1:nrow(dd)]

## NB! Ensure that samples in distance matrix and metadata are in the same order
meta <- meta[ match(x = meta$SampleID, table = rownames(dd)), ]

## Convert distance matrix into 'dist' class
dd <- as.dist(dd)

## Run PERMANOVA
adon <- adonis2(formula = dd ~ TestData, data = meta, permutations = 1000)
adon

## Run PERMDISP
permdisp <- betadisper(dd, meta$TestData)
plot(permdisp)
```

Example of plotting the ordination scores

```
library(ggplot2)

## Load ordination scores
ord <- readLines("ordination.txt")

## Skip PCA summary
ord <- ord[ 8:length(ord) ]

## Break the data into sample and species scores
breaks <- which(! nzchar(ord))
ord <- ord[1:(breaks[2]-1)] # Skip biplot scores
ord_sp <- ord[1:(breaks[1]-1)] # species scores
ord_sm <- ord[(breaks[1]+2):length(ord)] # sample scores

## Convert scores to data.frames
ord_sp <- as.data.frame( do.call(rbind, strsplit(x = ord_sp, split = "\t")) )
colnames(ord_sp) <- c("OTU_ID", paste0("PC", 1:(ncol(ord_sp)-1)))

ord_sm <- as.data.frame( do.call(rbind, strsplit(x = ord_sm, split = "\t")) )
colnames(ord_sm) <- c("Sample_ID", paste0("PC", 1:(ncol(ord_sm)-1)))

## Convert PCA to numbers
ord_sp[colnames(ord_sp)[-1]] <- sapply(ord_sp[colnames(ord_sp)[-1]], as.numeric)
ord_sm[colnames(ord_sm)[-1]] <- sapply(ord_sm[colnames(ord_sm)[-1]], as.numeric)
```

(continues on next page)

(continued from previous page)

```
## At this step, sample and OTU metadata could be added to the data.frame

## Example plot
ggplot(data = ord_sm, aes(x = PC1, y = PC2)) + geom_point()
```



1.5 Troubleshooting

This page is developing based on the user feedback.

1.5.1 General

Error: Conflict. The container name XXX is already in use by container “XXX”. You have to remove (or rename) that container to be able to reuse that name.

Reason: Process stopped unexpectedly and docker container was not closed.

Fix: Remove the docker container (not image!) that is causing the conflict

Error: No files in the output folder, but PipeCraft said “Workflow finished”.

Reason: ?

Fix: Check if there was a README.txt output and read that. Please *report* unexpexted errors.

1.5.2 ASVs workflow

Error:

“Workflow stopped”

Workflow stopped

OK

Possible reason: Computer's memory is full, cannot finish the analyses.

Fix: Analyse fewer number of samples or increase RAM size.

Error: "Error in derepFastq(fls[[i]], qualityType = qualityType) : Not all provided files exist. Calls: learnErrors -> derepFastq. Execution halted"

```

Loading required package:
  Rcpp
Error in derepFastq(fls[[i]],
qualityType = qualityType) :
  Not all provided files exist.
Calls: learnErrors ->
      derepFastq
Execution halted

```

OK

Possible reason: Some samples have completely discarded by quality filtering process.

Fix: Examine `seq_count_summary.txt` file in `qualFiltered_out` folder and discard samples, which had 0 quality filtered sequences (poor quality samples). Or edit the quality filtering settings.

Error: Error in filterAndTrim. Every input file must have a corresponding output file.

```

Loading required package:
  Rcpp
Error in filterAndTrim(fnFs,
  filtFs, fnRs, filtRs, maxN =
maxN, maxEE = c(maxEE, :
Every input file must have a
corresponding output file.
Execution halted

```

OK

Possible reason: wrong read identifiers for read R1 and read R2 in QUALITY FILTERING panel.

Fix: Check the input fastq file names and edit the identifiers. Specify identifier string that is common for all R1 reads (e.g. when all R1 files have `‘.R1’` string, then enter `‘\R1’`. Note that backslash is only needed to escape dot regex; e.g. when all R1 files have `‘_R1’` string, then enter `‘_R1’`). When demultiplexing data in during ASV (DADA2) workflow, then specify as `‘\R1’` _____

Error: “Error rates could not be estimated (this is usually because of very few reads). Error in getErrors(err, enforce = TRUE) : Error matrix is null.”

```
Loading required package:
  Rcpp
Error rates could not be
estimated (this is usually
because of very few reads).
Error in getErrors(err, enforce
= TRUE) : Error matrix is
NULL.
Calls: learnErrors -> dada ->
      getErrors
Execution halted
```

OK

Possible reason: Too small data set; samples contain too few reads for DADA2 denoising.

Fix: use OTU workflow.



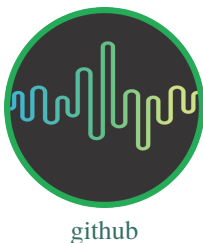
1.6 Licence

Copyright (C) 2022, Sten Anslan, Martin Metsoja

This program is free software: you can redistribute it and/or modify it under the terms of the GNU General Public License as published by the Free Software Foundation, either version 3 of the License, or any later version.

This program is distributed in the hope that it will be useful, but WITHOUT ANY WARRANTY; without even the implied warranty of MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE.

See the GNU General Public License for more details, <https://www.gnu.org/licenses/gpl-3.0.html>.



1.7 Contact

Sten Anslan <sten.anslan[at]ut.ee>
 Martin Metsoja <martin.metsoja[at]gmail.com>
 Vladimir Mikryukov <vladimir.mikryukov[at]ut.ee>
 Ali Hakimzadeh <ali.hakimzadeh[at]ut.ee>

Feel free to propose new pipelines/modules/software to be implemented to PipeCraft.



1.8 How to cite

For now, please cite the first release of PipeCraft:

Anslan, S, Bahram, M, Hiiesalu, I, Tedersoo, L. **PipeCraft: Flexible open-source toolkit for bioinformatics analysis of custom high-throughput amplicon sequencing data.** Mol Ecol Resour. 2017; 17: e234– e240. <https://doi.org/10.1111/1755-0998.12692>

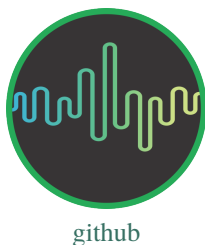
But please also include PipeCraft version 2 manual link: <https://pipecraft2-manual.readthedocs.io/en/stable>

e.g. “... using PipeCraft 2 (Anslan et al 2017; pipecraft2-manual.readthedocs.io/en/stable)”

PipeCraft version 1.0 is [here](#)

1.8.1 Work that has cited PipeCraft:

Records from google scholar https://scholar.google.com/scholar?cites=16831399634985547679&as_sdt=2005&sciodt=0,5&hl=en



1.9 Releases

Contents

- *Releases*
 - *0.1.4 (15.12.2022)*
 - *0.1.3 (28.07.2022)*
 - *0.1.2 (07.06.2022)*
 - *0.1.1 (01.04.2022)*
 - *0.1.0 pre-release (14.12.2021)*

1.9.1 0.1.4 (15.12.2022)

[DOWNLOAD link for v0.1.4](#)

- added 2nd round of cut primers to properly remove fwd and rev primers form the paired-end data set
- added UNOISE3 module to generate zOTUs (under clustering)
- added uchime3 chimera filtering (for denoised amplicons)
- edited sequence count statistics process after the process (using seqkit)
- only fasta (fa, fas) format is accepted for clustering
- edited OTU table making strategy for OTU clustering (was `-usearch_global` before)
- added table filtering options for DADA2 ASV table (collapse mismatch, filter by length)
- added ASV to OTU module (clustering DADA2 ASVs into OTUs)
- select region to cluster after ITSx in OTUs workflow
- automatically saves the PipeCraft workflow settings into loadable JSON file
- outputs log file (in development)
- merged vsearch and dada2 containers (had a lot in common)

Implemented software: (*software version in bold denotes version upgrade*)

| Software | version | Reference |
|---------------|---------------|-----------------------------|
| DADA2 | 1.20 | Callahan et. al 2016 |
| vsearch | 2.22.1 | Rognes et. al 2016 |
| trimmomatic | 0.39 | Bolger et al. 2014 |
| seqkit | 2.3.0 | Shen et al. 2016 |
| cutadapt | 3.5 | Martin 2011 |
| mothur | 1.46.1 | Schloss et al. 2009 |
| ITS Extractor | 1.1.3 | Bengtsson-Palme et al. 2013 |
| fqgrep | 0.4.4 | Indraniel Das 2011 |
| BLAST | 2.11.0+ | Camacho et al. 2009 |
| FastQC | 0.11.9 | Andrews 2019 |
| MultiQC | 1.12 | Ewels et al. 2016 |
| LULU | 0.1.0 | Froslev et al. 2017 |
| fastp | 0.23.2 | Chen et al. 2018 |
| DEICODE | 0.2.4 | Martion et al. 2019 |

1.9.2 0.1.3 (28.07.2022)

DOWNLOAD link for v0.1.3

- updated BLAST 2.11.0+ to BLAST 2.12.0+ and added biopython to BLAST container (fixed the coverage% calculation)
- fixed the megaBLAST, when gapextend=undefined
- quality Check module edit (does not stop when browsing around)
- fixed ASVs workflow error message when using <2 samples
- added lock panels when starting a process
- few cosmetic front-end adds

1.9.3 0.1.2 (07.06.2022)

DOWNLOAD link for v0.1.2

- added LULU post-clustering
- added DEICODE (postprocessing)
- added fastp quality filtering
- added DADA2 quality filtering under 'ADD STEP' -> 'QUALITY FILTERING' panel
- added DADA2 denoise and assemble paired-end data under 'ADD STEP' -> 'ASSEMBLE PAIRED-END' panel
- added DADA2 assignTaxonomy under 'ADD STEP' -> 'ASSIGN TAXONOMY' panel
- added trunc_length option for vsearch quality filtering
- python3 module fix for ITSx for removing empty sequences

Implemented software: *(software in red font denote new additions; ‘version’ in bold denotes version upgrade)*

| Software | version | Reference |
|----------------|-------------|-----------------------------|
| DADA2 | 1.20 | Callahan et. al 2016 |
| vsearch | 2.18.0 | Rognes et. al 2016 |
| trimmomatic | 0.39 | Bolger et al. 2014 |
| seqkit | 2.0.0 | Shen et al. 2016 |
| cutadapt | 3.5 | Martin 2011 |
| mothur | 1.46.1 | Schloss et al. 2009 |
| ITS Extractor | 1.1.3 | Bengtsson-Palme et al. 2013 |
| fggrep | 0.4.4 | Indraniel Das 2011 |
| BLAST | 2.11.0+ | Camacho et al. 2009 |
| FastQC | 0.11.9 | Andrews 2019 |
| MultiQC | 1.12 | Ewels et al. 2016 |
| LULU (link) | 0.1.0 | Froslev et al. 2017 |
| fastp (link) | 0.23.2 | Chen et al. 2018 |
| DEICODE (link) | 0.2.4 | Martion et al. 2019 |

1.9.4 0.1.1 (01.04.2022)

Minor cosmetic changes and bug fixes. [DOWNLOAD link for v0.1.1](#)

- separate output folder for unused index combinations in demultiplexing.
- resolved issues with sample renaming when using dual combinational indexes for paired-end data (DEMUTIPLEX)
- minBoot option fixed in DADA2 taxonomy annotation
- vsearch quality filtering “minsize” not working (option currently removed).

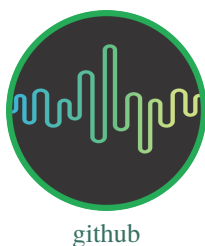
1.9.5 0.1.0 pre-release (14.12.2021)

[DOWNLOAD link for v0.1.0](#)

- ASV workflow with DADA2 for paired-end data.
- vsearch based OTU workflow.
- QualityCheck module with MultiQC and FastQC

Implemented software:

| Software | version | Reference |
|---------------|---------|-----------------------------|
| DADA2 | 1.14 | Callahan et. al 2016 |
| vsearch | 2.18.0 | Rognes et. al 2016 |
| trimmomatic | 0.39 | Bolger et al. 2014 |
| seqkit | 2.0.0 | Shen et al. 2016 |
| cutadapt | 3.5 | Martin 2011 |
| mothur | 1.46.1 | Schloss et al. 2009 |
| ITS Extractor | 1.1.3 | Bengtsson-Palme et al. 2013 |
| fqgrep | 0.4.4 | Indraniel Das 2011 |
| BLAST | 2.11.0+ | Camacho et al. 2009 |
| FastQC | 0.11.9 | Andrews 2019 |
| MultiQC | 1.12 | Ewels et al. 2016 |



1.10 Docker images

Bioinformatic tools used by PipeCraft2 are stored on [Dockerhub](#) as Docker images. These images can be used to launch any tool with the Docker CLI to utilize the compiled tools.

1.10.1 Images in use

| Image | Software |
|----------------------------|---|
| pipecraft/vsearch_dada2:1 | vsearch v2.22.1, dada2 v1.20, seqkit v2.3.0, lulu v0.1.0, R, GNU parallel |
| ewels/multiqc:latest | mutliqc v1.12 |
| staphb/fastqc:0.11.9 | fastqc v0.11.9 |
| pipecraft/cutadapt:3.5 | cutadapt v3.5, seqkit v2.3.0, python3, biopython |
| pipecraft/dada2:1.20 | dada2 v1.20, seqkit v2.3.0, lulu v0.1.0, R |
| pipecraft/reorient:1 | fqgrep v0.4.4, seqkit v2.3.0 |
| pipecraft/trimmomatic:0.39 | trimmomatic 0.39, seqkit v2.3.0 |
| pipecraft/vsearch:2.18 | vsearch v2.18, seqkit v2.3.0, GNU parallel |
| pipecraft/itsx:1.1.3 | ITSx v1.1.3, seqkit v2.3.0, mothur v1.46.1 |
| pipecraft/deicode:0.2.4 | DEICODE v0.2.4, qiime2-2002.2 |
| pipecraft/fastp:0.23.2 | fastp v0.23.2 |
| pipecraft/blast:2.12 | BLAST 2.12.0+, biopython, python3, gawk |

1.10.2 Other images

| Image | Software |
|------------------------|--|
| pipecraft/dada2:1.20 | dada2 v1.20, seqkit v2.3.0, lulu v0.1.0, R |
| pipecraft/vsearch:2.18 | vsearch v2.18, seqkit v2.3.0, GNU parallel |

Manual may contain some typos! Fixing those on the way.

CURRENTLY IMPLEMENTED SOFTWARE

See software version on the ‘Releases’ page

| Software | Reference | Task |
|---------------|---|---|
| docker | https://www.docker.com | building, sharing and running applications |
| DADA2 | Callahan et. al 2016 | ASVs workflow (from raw reads to ASV table) |
| vsearch | Rognes et. al 2016 | quality filtering, assemble paired-end reads, chimera filtering, clustering |
| trimmomatic | Bolger et al. 2014 | quality filtering |
| fastp | Chen et al. 2018 | quality filtering |
| seqkit | Shen et al. 2016 | multiple sequence manipulation operations |
| cutadapt | Martin 2011 | demultiplexing, cut primers |
| biopython | Cock et al. 2009 | multiple sequence manipulation operations |
| GNU Parallel | Tangle 2021 | executing jobs in parallel |
| mothur | Schloss et al. 2009 | submodule in ITSx to make unique and deunique seqs |
| ITS Extractor | Bengtsson-Palme et al. 2013 | extract ITS regions |
| fggrep | Indrani Das 2011 | core for reorient reads |
| BLAST | Camacho et al. 2009 | assign taxonomy |
| FastQC | Andrews 2019 | QualityCheck module |
| MultiQC | Ewels et al. 2016 | QualityCheck module |
| LULU | Frøslev et al. 2017 | post-clustering curation |
| DEICODE | Martino et al. 2019 | dissimilarity analysis |

Let us know if you would like to have a specific software implemented to PipeCraft ([contacts](#)) or create an issue in the main repository.