PipeCraft2

Release 1.0.0

Sten Anslan

Mar 21, 2024

CONTENTS

your pipeline composer

github

01.09.2023 - v1.0.0, get it here

download links for the new release updated, but gradually updating the rest of the user guide. Check *what have changed in the Releases sheet*

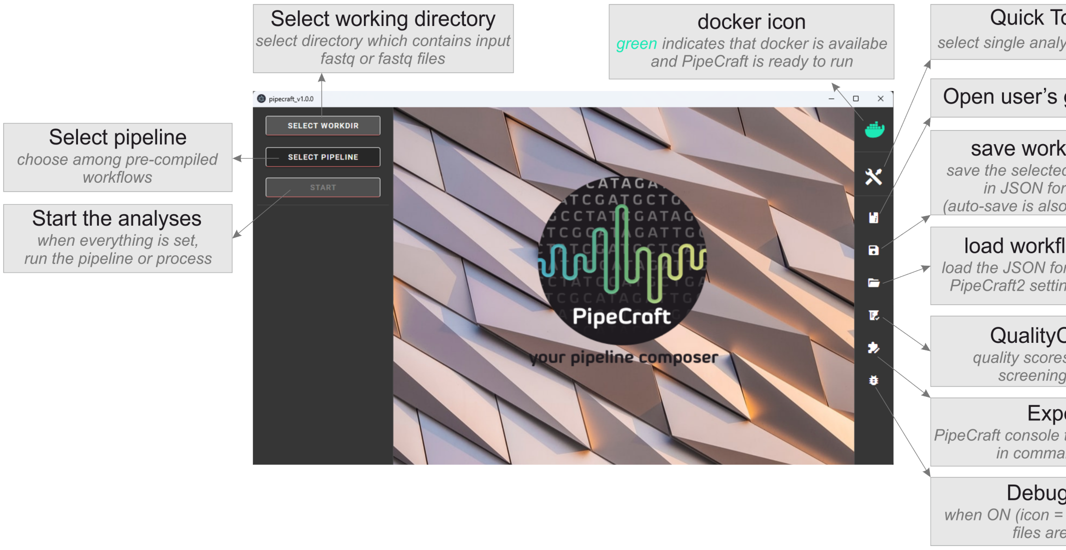
CHAPTER

ONE

MANUAL FOR PIPECRAFT2

PipeCraft2 is a Graphical User Interface (GUI) software that implements *various popular tools* for **metabarcoding** data analyses.

Implements various ready-to-run (pre-defined) pipelines as well as an option to run a variety of individual steps outside of a full-pipeline.



(click on the image for enlargement)

Software settings for pipeline processes contain key options for metabarcoding 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 executing the full pipeline (user specifies the input, and output will be e.g. OTU/ASV table with taxonomic annotations of the generated features), but supports also single-step mode (Quick Tools panel) 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)*.

1.1 Glossary

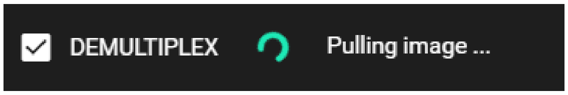
List of terms that you may encounter in this user guide.

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. Note that PipeCraft expects that read 1 file contains the string R1 and read 2 contains R2 (not e.g. my_sample_L001_1.fastq / my_sample_L001_2.fastq)
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 Docker images

All the processes are run through docker, where the PipeCraft's GUI simply 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. Initial PipeCraft2 installation does not contain any software for sequence data processing.

Example: when running DEMULTIPLEXING for the first time



☑ DEMULTIPLEX Pulling image ...

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.

1.3 Save workflow

Once the workflow settings are selected, save the workflow by pressin `SAVE WORKFLOW` button on the *right-ribbon*.

Note: starting from version 0.1.4, PipeCraft2 will automatically save the settings into selected WORKDIR prior starting the analyses (file name = "**pipecraft2_config.json**")

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").*

1.4 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.

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

1.4.1 Contents of this user guide

github

Installation

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

Herein **'PipeCraft' == 'PipeCraft2'**. Using those interchangeably.

Contents

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.

Windows

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

1. Download PipeCraft2 for Windows: v1.0.0
2. Install PipeCraft2 via the setup executable
3. Download Docker for windows - ONLY ONCE (no need, when updating PipeCraft)

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). Default = 50% of total memory on Windows or 8GB, whichever is less. 80% of total memory on Windows on builds before 20175 (Win10, from 2020).

Quick guide to increase Docker accessible RAM size in Windows

Instructions from https://learn.microsoft.com/en-us/windows/wsl/wsl-config#wslconfig

1. This is for Windows Build 19041 and later with WSL 2
2. Open 'File Explorer' and type **%USERPROFILE%** to the address bar to access the %USERPROFILE% directory (generally e.g. "C:Usersmy_user_name").
3. Make new text (txt) document into %USERPROFILE% directory.
4. Paste the following text to that new txt document:

Listing 1: make .wslconfig file

```
# Settings apply across all Linux distros running on WSL 2
[wsl2]

# Limits VM memory to use no more than X GB, this can be set as whole numbers using GB or MB
memory=30GB

# Sets the VM to use X virtual processors
processors=8
```

5. Edit "memory=30GB" and "processors=8" according to your needs
6. Save the file and rename this as .wslconfig

7. Restart Docker.

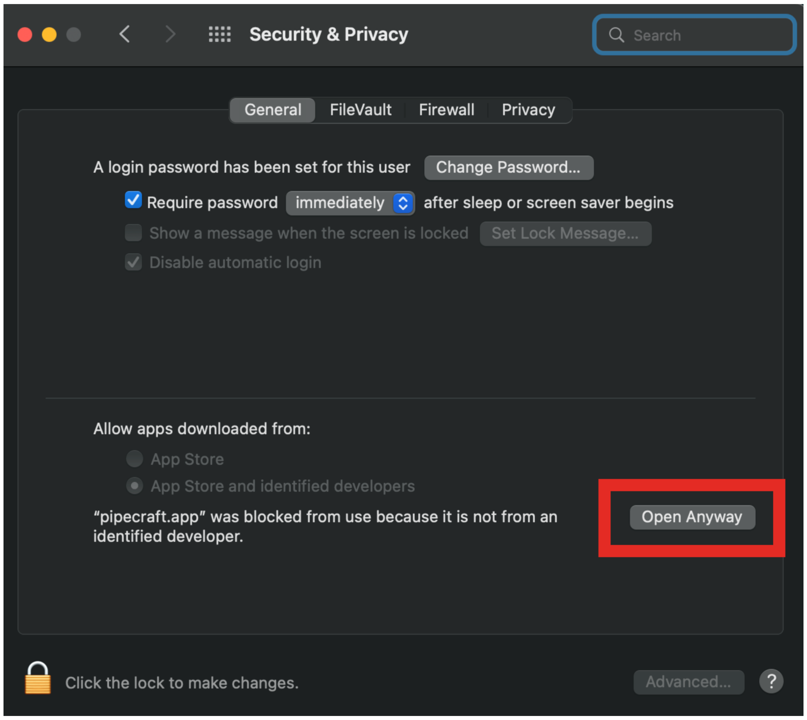
MacOS

PipeCraft2 is supported on macOS 10.15+. Older OS versions might not support PipeCraft GUI workflow through Docker.

Note: If your MacOS has M1/M2 chips, please let us know if you encounter something weird while trying to run some analyses (*contact* or post an issue on the github page).

1. Download PipeCraft2 for Mac: v1.0.0
2. Install PipeCraft2 via **pkg** file
3. Currently macOS will flag pipecraft as an 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. Installing on latest versions of macOS (13.0+) will require you to disable gatekeeper entirely, to do so paste the following command into a terminal.
4. Check your Mac chip (Apple or Intel) and download Docker for Mac - ONLY ONCE (no need, when updating PipeCraft)

```
sudo spctl --master-disable
```



A login password has been set for this user

☑ Require password immediately after sleep or screen saver begins

□ Show a message when the screen is locked

☑ Disable automatic login

Allow apps downloaded from:

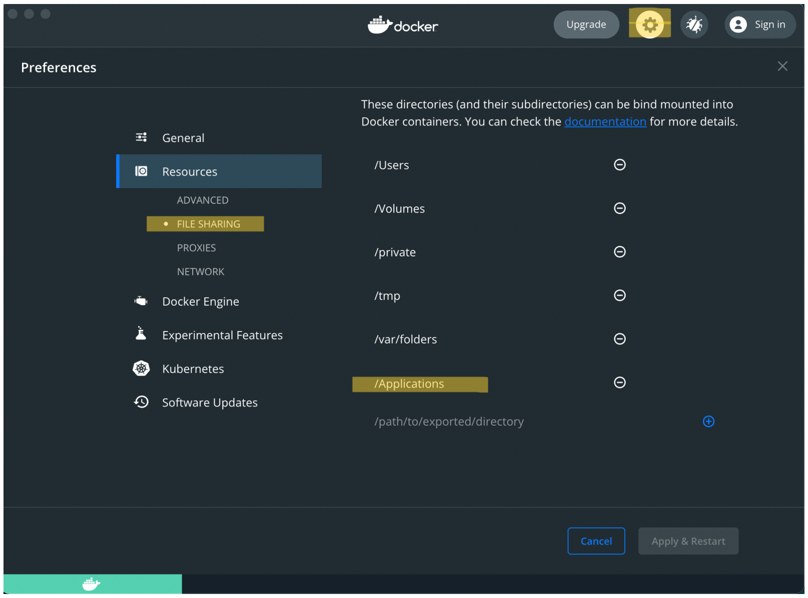
○ App Store

● App Store and identified developers

"pipecraft.app" was blocked from use because it is not from an identified developer.

Click the lock to make changes.

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

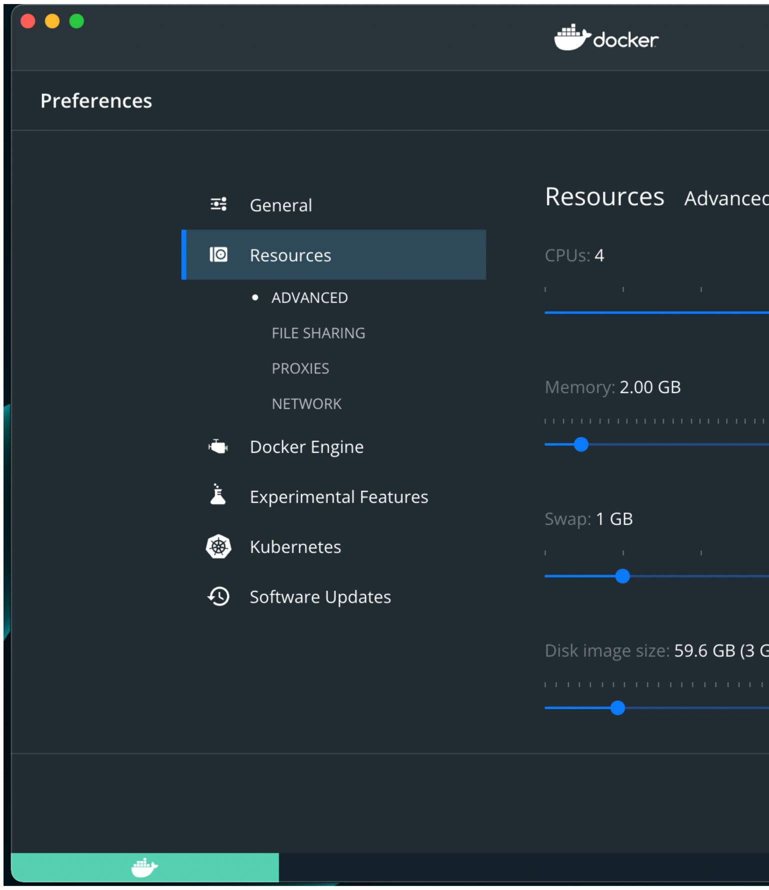


These directories (and their subdirectories) can be bind mounted into Docker containers. You can check the documentation for more details.

/Users
/Volumes
/private
/tmp
/var/folders
/Applications
/path/to/exported/directory

Se

Note: Manage Docker resource limits in the Docker dashboard:



CPUs: 4

Memory: 2.00 GB

Swap: 1 GB

Disk image size: 59.6 GB (3 G

Linux

PipeCraft2 was tested with **Ubuntu 20.04** and **Mint 20.1**. Older OS versions might not support PipeCraft GUI workflow through Docker.

1. Download PipeCraft2 for Linux: v1.0.0
2. Right click on the pipecraft_*.deb file and "Open With GDebi Package Installer" (Install Package) or `sudo dpkg -i path_to_deb_file`
3. Install Docker - ONLY ONCE (no need, when updating PipeCraft); follow the guidelines under appropriate Linux distribution

> **Warning:**
>
> When installing Docker Engine, make sure you have not Docker Desktop already installed!
> Installing both might have interfering consequences

4. If you are a non-root user complete these post-install steps

Note: When you encounter ERROR during PipeCraft2 installation, then uninstall the previous version of PipeCraft2 `sudo dpkg --remove pipecraft-v0.1.3`

5. Run PipeCraft2. 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.

Updating PipeCraft2

Auto-updates will be available (hopefully) soon [at least for Windows]!

See *PipeCraft2 releases here*.
See *removing docker images* section.

> **Warning:**
>
> To avaoid 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 v1.0.0**, if docker container is updated for the new PipeCraft2 version, then it will get a new tag; so, no need to purge all previous docker containers *(but to save disk space, see which containers you have not used for a while and perhaps delete those)*

Uninstalling PipeCraft2

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`

Purging 'old' Docker installations

Listing 2: To uninstall **docker engine** and all its packages:

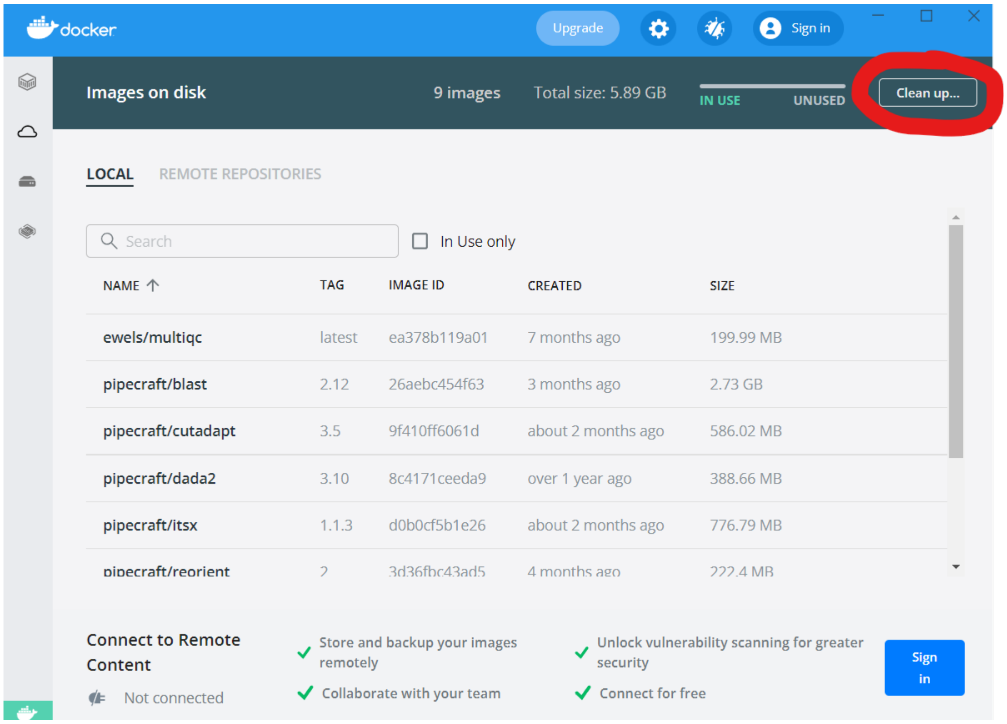
```
sudo apt-get purge docker-ce docker-ce-cli containerd.io docker-buildx-plugin docker-
↪compose-plugin docker-ce-rootless-extras
```

Listing 3: To uninstall **docker desktop** and clean configurations:

```
    rm -r $HOME/.docker/desktop
    sudo rm /usr/local/bin/com.docker.cli
    sudo apt purge docker-desktop
```

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.



NAME	TAG	IMAGE ID	CREATED	SIZE
ewels/multiqc	latest	ea378b119a01	7 months ago	199.99 MB
pipecraft/blast	2.12	26aebc454f63	3 months ago	2.73 GB
pipecraft/cutadapt	3.5	9f410ff6061d	about 2 months ago	586.02 MB
pipecraft/dada2	3.10	8c4171ceeda9	over 1 year ago	388.66 MB
pipecraft/itsx	1.1.3	d0b0cf5b1e26	about 2 months ago	776.79 MB
pipecraft/reorient	2	3d36fbc43ad5	4 months ago	222.4 MB

Connect to Remote Content

✓ Store and backup your images remotely

✓ Collaborate with your team

✓ Unlock vulnerability scanning for greater security

✓ Connect for free

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` –> to delete selected image

or

`sudo docker system prune -a` –> to delete all unused containers, networks, images
`sudo docker images` –> check if images were removed

github

QuickStart

Required data formats

There are few specifc requirements for the input data for PipeCraft.

- **Paired-end data**
  - must contain **R1** and **R2** strings in the paired-end files,
  - **sample names maynot contain 'R1/R2' strings**
    - **OK file names**: `my_sample_01_R1_L001.fastq` and `my_sample_01_R2_L001.fastq`
    - **NOT-OK** file names: `my_R1sample_01_R1_L001.fastq` and `my_R1sample_01_R2_L001.fastq` or `my_sample_01_1.fastq` and `my_sample_01_2.fastq`
- **index/barcodes file for demultiplexing:**
  - *see formatting requirements here*
- **specific directory structure of NextITS pipeline**
  - *see NextITS page here*

How to START

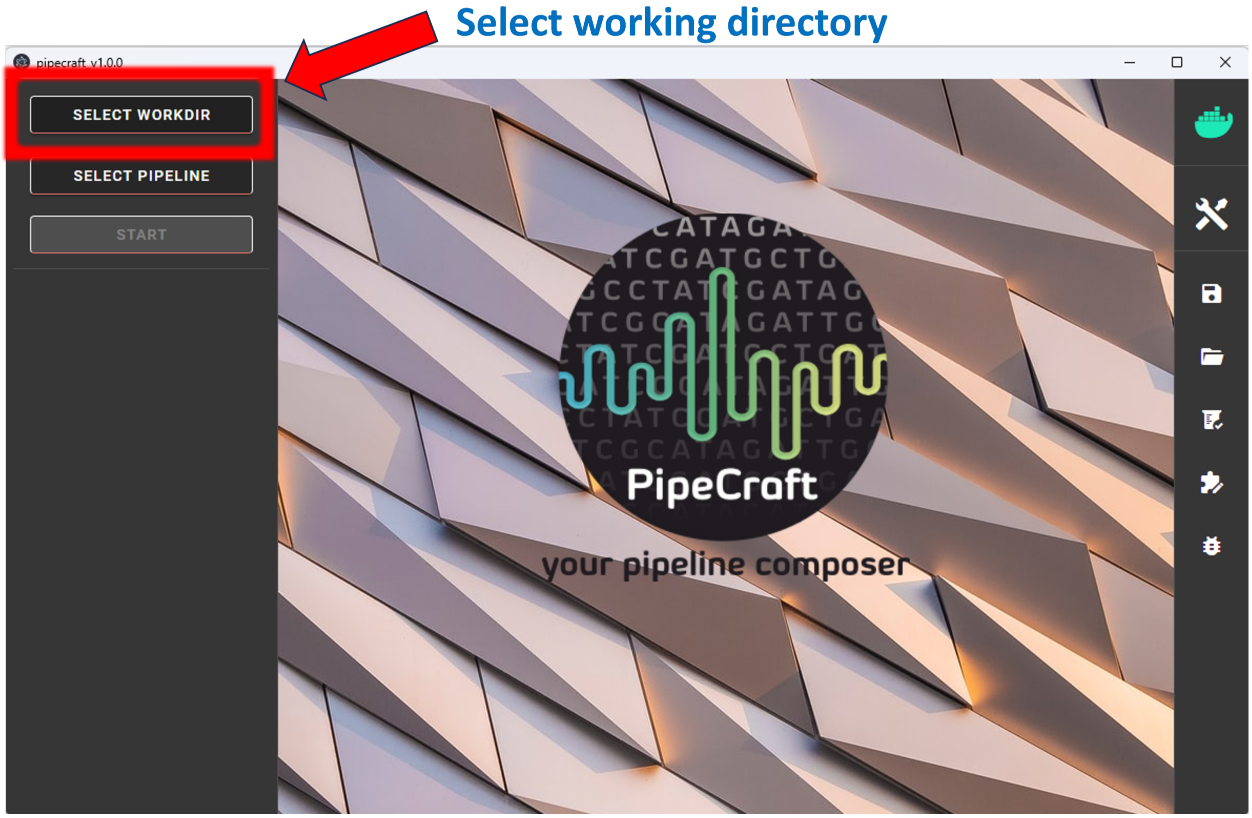
1. To `START` any analyses, you must specify the working directory (WORKDIR) by pressing the `SELECT WORKDIR` button. E.g., if working with **fastq** files, then be sure that the working directory contains **only relevant fastq files** because the selected process will be applied to all fastq files in the working directory!

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

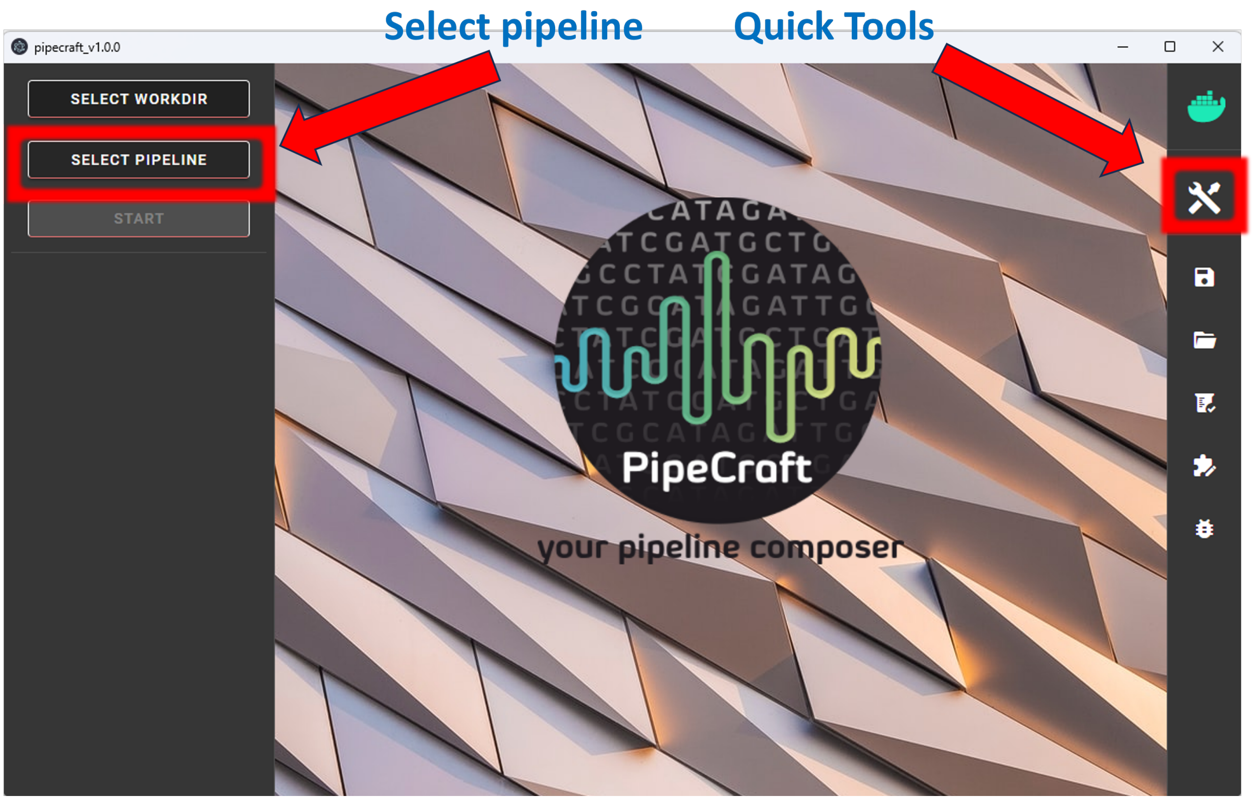
After selecting a working directory, PipeCraft needs you to specify if

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

`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; and sample names maynot contain R1/R2 strings)
`single-end data` –> such as data from PacBio, or assembled paired-end data (single file per library or per sample)



2. SELECT PIPELINE or press Quick Tools button to select relevant *step*; edit settings if needed and **start running the analyses** by pressing the START button.



Note: **When running 'step-by-step analyses with Quick Tools'**: when one workflow is finished, then press `SELECT WORKDIR` to specify inputs for the next process to ensure the correct workflow piping.

Warning: The **outputs will be overwritten** if running the same analysis step **multiple times in the same working directory**. If needed, edit the default output directory name to prevent that.

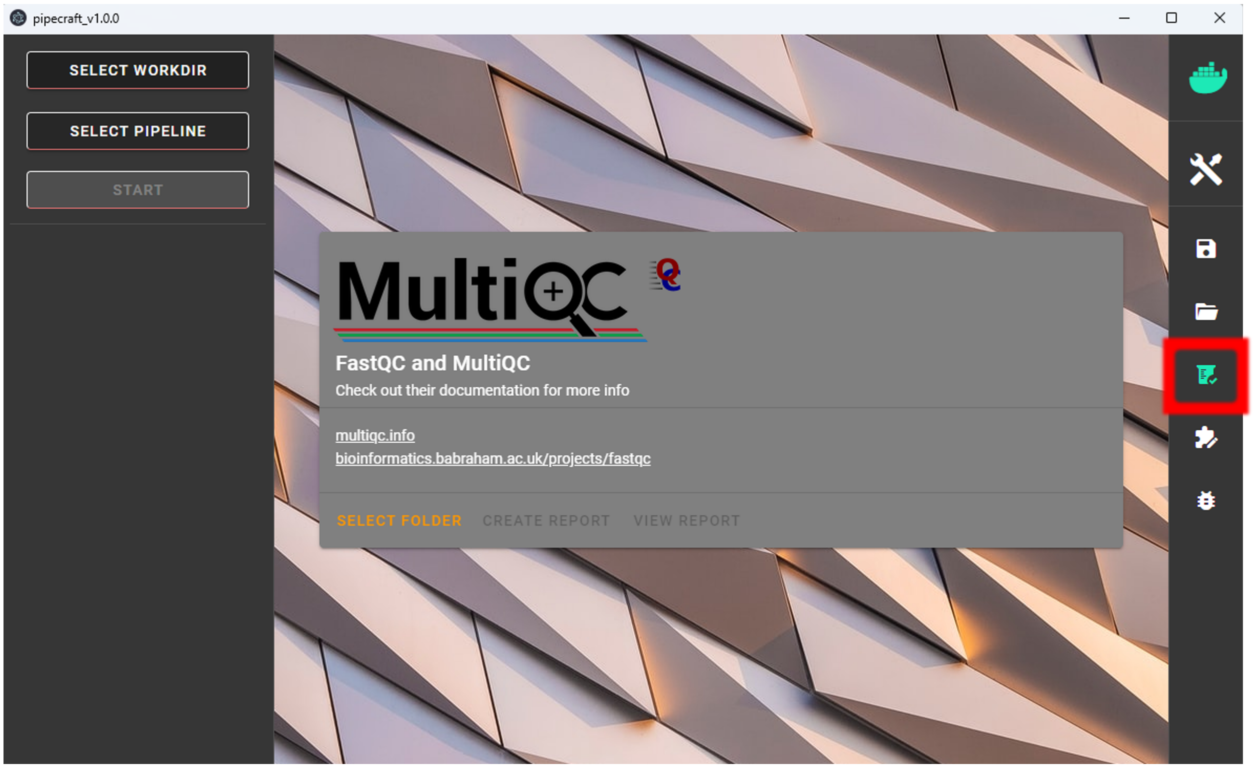
Each process creates a separate output directory with the processed files. The **README** file in the output directory states some of the details about the finished process.

Ready-to-run pre-defined pipelines here

QuickTools page here

Quality scores 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.



FastQC and MultiQC

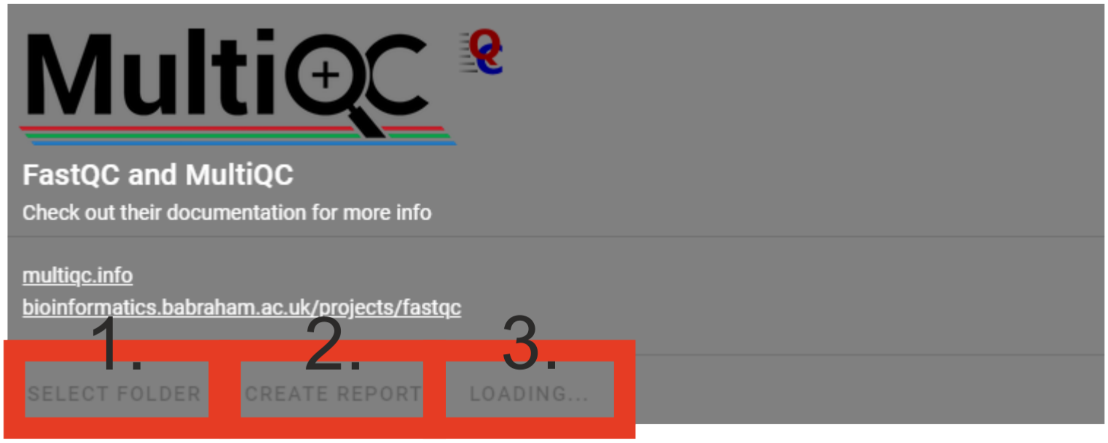
Check out their documentation for more info

multiqc.info

bioinformatics.babraham.ac.uk/projects/fastqc

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

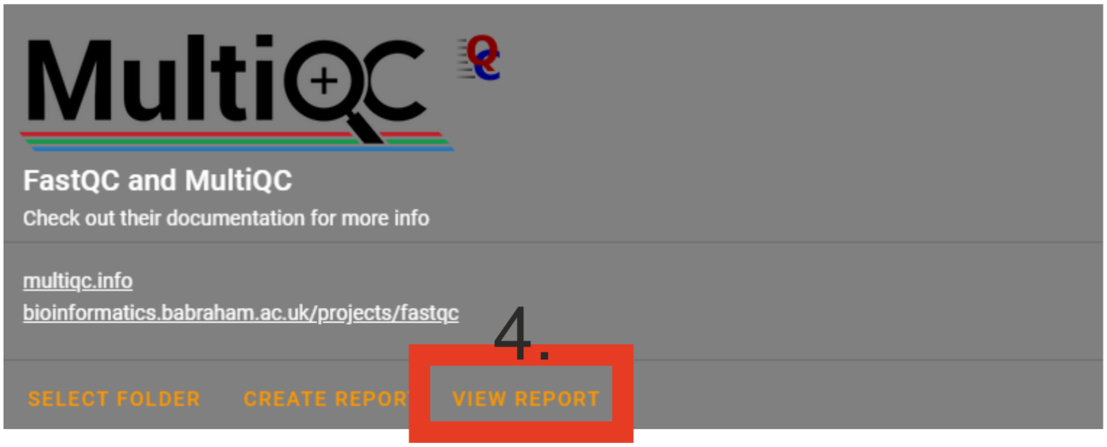


FastQC and MultiQC

Check out their documentation for more info

multiqc.info

bioinformatics.babraham.ac.uk/projects/fastqc



MultiQC

FastQC and MultiQC

Check out their documentation for more info

multiqc.info

bioinformatics.babraham.ac.uk/projects/fastqc

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 floder 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 if those are of no interest.

github

Pre-defined pipelines

vsearch OTUs

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

Default options:
click on analyses step for more info

Analyses step	Default setting
DEMULTIPLEX (optional)	–
REORIENT (optional)	–
REMOVE PRIMERS (optional)	–

MERGE READS	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

QUALITY FILTERING with vsearch	maxEE = 1
	maxN = 0
	minLen = 32
	max_length = undefined
	qmax = 41
	qmin = 0
	maxee_rate = undefined

CHIMERA FILTERING with uchime_denovo	pre_cluster = 0.98
	min_unique_size = 1
	denovo = TRUE
	reference_based = undefined
	abundance_skew = 2
	min_h = 0.28

ITS Extractor (optional)	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

CLUSTERING with vsearch	OTU_type = centroid
	similarity_threshold = 0.97
	strands = both
	remove_singletons = false

DADA2 ASVs

ASVs workflow panel (with DADA2)

Note: Working directory must contain **at least 2 samples** for DADA2 pipeline.

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 *remove primers* before proceeding with ASV generation with DADA2.

The official DADA2 manual is available here

Default options:

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

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`	applies only for **paired-end** data. Identifyer 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’.).
`read_R2`	applies only for **paired-end** data. Identifyer string that is common for all R2 reads (e.g. when all R2 files have ‘.R2’ string, then enter ‘\.R2’. Note that backslash is only needed to escape dot regex; e.g. when all R2 files have ‘_R1’ string, then enter ‘_R2’.).
`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`	remove reads with length less than minLen. minLen is enforced after all other trimming and truncation
`truncQ`	truncate reads at the first instance of a quality score less than or equal to truncQ
`truncLen`	truncate reads after truncLen bases (applies to **R1 reads** when working with **paired-end** data). Reads shorter than this are discarded. Explore quality profiles (with QualityCheck module) and see whether poor quality ends needs to be truncated
`truncLen_R2`	applies only for **paired-end** data. Truncate **R2 reads** after truncLen bases. Reads shorter than this are discarded. Explore quality profiles (with QualityCheck module) and see whether poor quality ends needs to truncated

see *default settings*

DENOISING [ASVs workflow]

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

Setting	Tooltip
`pool`	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 pool = 'pseudo', the algorithm will perform pseudo-pooling between individually processed samples.
`selfConsist`	if TRUE, the algorithm will alternate between sample inference and error rate estimation until convergence
`qualityType`	'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
`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`	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
`justConcatenate`	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
`method`	'consensus' - the samples are independently checked for chimeras, and a consensus decision on each sequence variant is made. If 'pooled', the samples are all pooled together for chimera identification. If 'per-sample', the samples are independently checked for chimeras

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
`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

see *default settings*

ASSIGN TAXONOMY [ASVs workflow]

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

Setting	Tooltip
`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
`dada2 database`	select a reference database fasta file for taxonomy annotation Download DADA2-formatted reference databases here

see *default settings*

UNOISE ASVs

UNOISE3 pipeline for making ASVs (zOTUs). Can optionally do automatic clustering of those ASVs (zOTUs) to OTUs by specifying the similarity threshold < 1. Uses UNOISE3 and clustering algorithms in vsearch.

NextITS

NextITS is an automated pipeline for analysing **full-length ITS** reads obtained via **PacBio** sequencing.

This pipeline implements:
* primer trimming
* quality filtering
* full-length ITS region extraction
* correction of homopolymer errors
* chimera filtering (get database for reference-based chimera filtering here)
* recovery of sequences false-positively annotated as chimeric
* detection of tag-switching artifacts per sequencing run
* multiple options for sequence clustering
* post-clustering with LULU

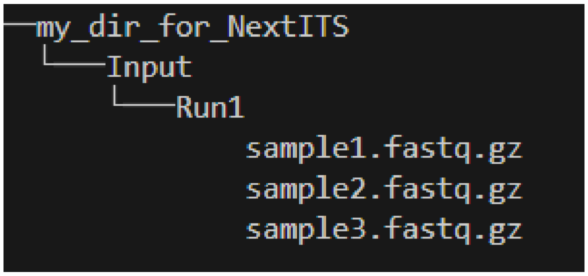
Please see other details here: https://next-its.github.io

Important: NextITS requires your data and folders to be structured in a specific way (see below)! Directory `my_dir_for_NextITS` contains `Input` [hard-coded requirement here] and one or multiple sequencing runs. In the below example, the sequencing runs [`RunID`] are named as Run1, Run2 and Run3 (but naming can be different).

In PipeCraft2, following the examples below, select `my_dir_for_NextITS` as a WORKDIR.

Single sequencing run

Select `my_dir_for_NextITS` as a WORKDIR in PipeCraft2.
Directory structure for analysing a single sequencing run:



```
──my_dir_for_NextITS
  └───Input
      └───Run1
              sample1.fastq.gz
              sample2.fastq.gz
              sample3.fastq.gz
```

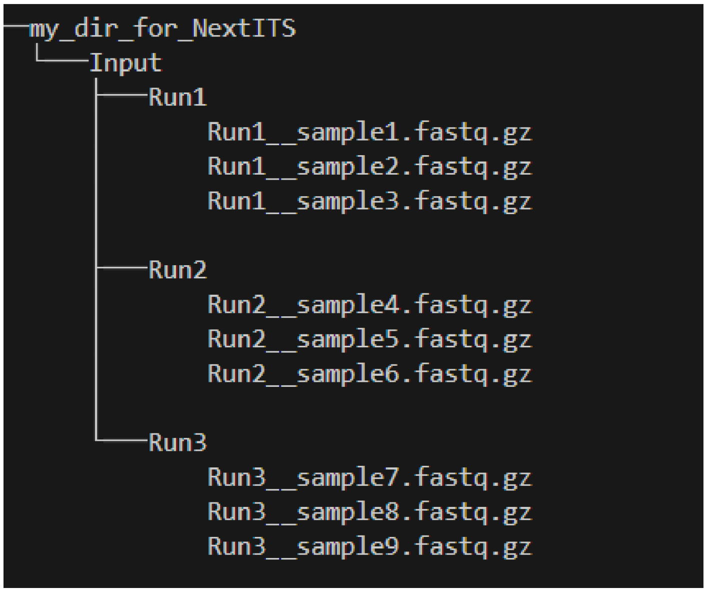
Input data for this pipeline **must be demultiplexed**, if your data is multiplexed use the demultiplexer from **QuickTools** before running the pipeline.

Sample naming

Please avoid non-ASCII symbols in `SampleID`, and do not use the period symbol (.), as it represents the wildcard character in regular expressions. Also, it is preferable not to start the sample name with a number.

Multiple sequencing runs

Select `my_dir_for_NextITS` as a WORKDIR in PipeCraft2.
Directory structure for analysing multiple sequencing runs:



```
─my_dir_for_NextITS
  └───Input
      ├───Run1
      │       Run1__sample1.fastq.gz
      │       Run1__sample2.fastq.gz
      │       Run1__sample3.fastq.gz
      │
      ├───Run2
      │       Run2__sample4.fastq.gz
      │       Run2__sample5.fastq.gz
      │       Run2__sample6.fastq.gz
      │
      └───Run3
              Run3__sample7.fastq.gz
              Run3__sample8.fastq.gz
              Run3__sample9.fastq.gz
```

Input data for this pipeline **must be demultiplexed**, if your data is multiplexed use the demultiplexer from **QuickTools** before running the pipeline.

Sample naming

Please avoid non-ASCII symbols in `RunID` and `SampleID`, and do not use the period symbol (.), as it represents the wildcard character in regular expressions. Also, it is preferable not to start the sample name with a number.

NextITS uses the `SequencingRunID__SampleID` naming convention (please note the double underscore separating `RunID` and `SampleID` parts). This naming scheme allows to easily trace back sequences, especially if the same sample was sequenced several times and is present in multiple sequencing runs. In the later steps, extracting the SampleID part and summarizing read counts for such samples is easy.

Default settings:

Analyses step	Default setting

STEP 1: QUALITY CONTROL, ARTEFACT REMOVAL	primer_mismatch = 2
	its_region = full
	qc_maxhomopolymerlen = 25
	qc_maxn = 4
	ITSx_evalue = 1e-2
	ITSx_partial = 0
	ITSx_tax = all
	chimera_rescue_occurrence = 2
	tj f = 0.01
	tj p = 1
	hp = TRUE

STEP 2: DATA AGGREGATION, CLUSTERING	otu_id = 0.98
	swarm_d = 1
	lulu = TRUE
	unoise = FALSE
	otu_id_def = 2
	otu_qmask = dust
	swarm_fastidious = TRUE
	unoise_alpha = 2
	unoise_minsize = 8
	max_MEEP = 0.5
	max_chimera_score = 0.5
	lulu_match = 95
	lulu_ratio = 1
	lulu_ratiotype = min
	lulu_relcooc = 0.95
	lulu_maxhits = 0

Cut primers

Setting	Tooltip
`primer_forward`	Specify forward primer, IUPAC codes allowed
`primer_reverse`	Specify reverse primer, IUPAC codes allowed
`primer_mismatch`	Specify allowed number of mismatches for primers

Quality filtering

Filter sequences based on expected errors per sequence and per base, compress and correct homopolymers.

Setting	Tooltip
`qc_maxee`	Maximum number of expected errors
`qc_maxeerate`	Maximum number of expected error per base
`qc_maxn`	Discard sequences with more than the specified number of ambiguous nucleotides (N’s)
`qc_maxhomopolymerlen`	Threshold for a homopolymer region lenght in a sequence
`hp`	Enable or disable homopolymer correction

ITS extraction

When performing ITS metabarcoding, it may be beneficial to trim the flanking 18S and 28S rRNA genes; because:

- these conserved regions don't offer species-level differentiation.
- random errors in these areas can disrupt sequence clustering.
- chimeric breakpoints, which are common in these regions, are hard to detect in short fragments ranging from 10 to 70 bases.

NextITS deploys the ITSx software (Bengtsson-Palme et al. 2013) for extracting the ITS sequence.

Setting	Tooltip
`its_region`	ITS part selector (ITS1, ITS2 or full)
`ITSx_tax`	Taxonomy profile for ITSx can be used to restrict the search to only taxon(s) of interest.
`ITSx_evalue`	E-value cutoff threshold for ITSx
`ITSx_partial`	Keep partial ITS sequences (specify a minimum length cutoff)

Chimera filtering

NextITS employs a two-pronged strategy to detect chimeras: de novo and reference-based chimera filtering.
A **reference database** for chimera filtering from full-length ITS data is accessible here. This database is based on EUKARYOME database

Additional step in NextITS is a **"rescue" of sequences** that were initially flagged as chimeric, but are occur at least in 2 samples (which represent independent PCR reactions); thus are likely false-positive chimeric sequences. The chimeric sequence occurrence frequency can be edited using the –chimera_rescue_occurrence parameter.

Setting	Tooltip
`chimera_database` (optional)	Database for reference based chimera removal (UDB)
`chimera_rescue_occurence`	A minimum occurence of initially flagged chimeric sequence required to rescue them

Tag-jump correction

Tag-jumps, sometimes referred to as index-switches or index cross-talk, may represent a significant concern in high-throughput sequencing (HTS) data. They can cause technical cross-contamination between samples, potentially distorting estimates of community composition. Here, tag-jump events are evaluated the UNCROSS2 algorithm (Edgar 2018) are removed.

Setting	Tooltip
`tj_f`	UNCROSS parameter f for tag-jump filtering
`tj_p`	UNCROSS parameter p for tag-jump filtering

UNOISE denoising

The UNOISE algorithm (Edgar 2016) focuses on error-correction (or denoising) of amplicon reads. Essentially, UNOISE operates on the principle that if a sequence with low abundance closely resembles another sequence with high abundance, the former is probably an error. This helps differentiate between true biological variation and sequencing errors. It's important to note that UNOISE was initially designed and optimized for Illumina data. Because of indel errors stemming from inaccuracies in homopolymeric regions, UNOISE might not work well with data that hasn't undergone homopolymer correction.

Setting	Tooltip
`unoise`	Enable or disable denoising with UNOISE algorithm
`unoise_alpha`	Alpha parameter for UNOISE
`unoise_minsize`	Minimum sequence abundance

Clustering

NextITS supports 3 different clustering methods:

- vsearch: this employs greedy clustering using a fixed sequence similarity threshold with VSEARCH (Rognes et al., 2016,);
- swarm: dynamic sequence similarity threshold for clustering with SWARM (Mahé et al., 2021,);
- unoise: creates zero-radius OTUs (zOTUs) based on the UNOISE3 algorithm (Edgar 2016);

Setting	Tooltip
`clustering_method`	Sequence clustering method (choose from: vsearch, swarm, unoise)
`otu_id`	Sequence similarity threshold
`otu_iddef`	Sequence similarity definition (applied to UNOISE as well)
`otu_qmask`	Method to mask low-complexity sequences (applied to UNOISE as well)
`swarm_d`	SWARM clustering resolution (d)
`swarm_fastidious`	Link nearby low-abundance swarms (fastidious option)

Post-clustering with LULU

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-occurence 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 OTUs.

Setting	Tooltip
`lulu`	Enable or disable post-clustering curation with lulu
`lulu_match`	Minimum similarity threshold
`lulu_ratio`	Minimum abundance ratio
`lulu_ratiotype`	Abundance ratio type - “min” or “avg
`lulu_relcooc`	Relative co-occurrence
`lulu_maxhits`	Maximum number of hits (0 = unlimited)

your pipeline composer

github

Individual steps (Quick Tools)

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/fasta formatted paired-end and single-end data are supported.
Outputs are fastq/fasta 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: **When using paired indexes**, then sequences with any index combination will be outputted to 'unnamed_index_combinations' dir. 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 (*see guide here*), 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
`search window`	the index search window size. The default 35 means that the forward index is searched among the first 35 bp and the reverse index among the last 35 bp. This search restriction prevents random index matches in the middle of the sequence
`min seq length`	minimum length of the output sequence
`no indels`	do not allow insertions or deletions is primer search. Mismatches are the only type of errors accounted in the error rate parameter

Note: Heterogenity 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 paired (dual) 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:

Example of **single-end indexes**

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

Example of **paired indexes**

```
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**.

- 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.
- 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.

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).

If 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

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/fasta formatted paired-end and single-end data are supported.

Outputs are fastq/fasta 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 sequnce 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`	minimum length of the output sequence

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 discared) [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 discared)
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
maxee_rate	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 trimms 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 >=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 (base[i] != base[i+1]). E.g. vaule 30 means then 30% complexity is required. Not specified = filter not applied
`cores`	

DADA2 (‘filterAndTrim’ function)

DADA2 setting	Tooltip
`read_R1`	applies only for **paired-end** data. Identifyer 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’.).
`read_R2`	applies only for **paired-end** data. Identifyer string that is common for all R2 reads (e.g. when all R2 files have ‘.R2’ string, then enter ‘\.R2’. Note that backslash is only needed to escape dot regex; e.g. when all R2 files have ‘_R1’ string, then enter ‘_R2’.).
`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`	remove reads with length less than minLen. minLen is enforced after all other trimming and truncation
`truncQ`	truncate reads at the first instance of a quality score less than or equal to truncQ
`truncLen`	truncate reads after truncLen bases (applies to **R1 reads** when working with **paired-end** data). Reads shorter than this are discarded. Explore quality profiles (with QualityCheck module) and see whether poor quality ends needs to be truncated
`truncLen_R2`	applies only for **paired-end** data. Truncate **R2 reads** after truncLen bases. Reads shorter than this are discarded. Explore quality profiles (with QualityCheck module)

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. Identifyer 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`	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`	identifyer 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’.)
`read_R2`	identifyer string that is common for all R2 reads (e.g. when all R2 files have ‘.R2’ string, then enter ‘\.R2’. Note that backslash is only needed to escape dot regex; e.g. when all R2 files have ‘_R1’ string, then enter ‘_R2’.)
`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`	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
`justConcatenate`	if TRUE, the forward and reverse-complemented reverse read are concatenated rather than merged, with a NNNNNNNNNN (10 Ns) spacer inserted between them
`pool`	denoising setting. 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 pool = ‘pseudo’, the algorithm will perform pseudo-pooling between individually processed samples.
`selfConsist`	

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/fasta 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
`pre_cluster`	identity percentage when performing 'pre-clustering' with –cluster_size for denovo chimera filtering with –uchime_denovo
`min_unique_size`	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
`denovo`	if TRUE, then perform denovo chimera filtering with –uchime_denovo
`reference_based`	perform reference database based chimera filtering with –uchime_ref. Select fasta formatted reference database (e.g. UNITE for ITS reads). If denovo = TRUE, then reference based chimera filtering will be performed after denovo.
`abundance_skew`	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
`min_h`	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
`pre_cluster`	identity percentage when performing 'pre-clustering' with –cluster_size for denovo chimera filtering with –uchime_denovo
`min_unique_size`	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
`denovo`	if TRUE, then perform denovo chimera filtering with –uchime_denovo
`reference_based`	perform reference database based chimera filtering with –uchime_ref. Select fasta formatted reference database (e.g. UNITE for ITS reads). If denovo = TRUE, then reference based chimera filtering will be performed after denovo.
`abundance_skew`	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
`min_h`	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

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/fasta 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) 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

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 –iddef
`sequence_sorting`	size = sort the sequences by decreasing abundance; “length” = sort the sequences by decreasing length (–cluster_fast); “no” = do not sort sequences (–cluster_smallmem –usersort)
`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; –sizeorder), `max_hits` 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`	mask regions in sequences using the “dust” method, or do not mask (“none”)

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 –iddef
`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 –cluster_unoise 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`	

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-occurence 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 minimim 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_cooccurence	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
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 shorther than specified sequence length
# ASVs_lenFilt.fasta = ASV sequences after filtering out ASVs with shorther 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

ASSIGN TAXONOMY

Implemented tools for taxonomy annotation:

BLAST (Camacho et al. 2009)

BLAST search sequences againt 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) againt 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
CO1 Classifier	5.1	Metazoa COI
DADA2-formatted reference databases		DADA2-formatted reference databases
DIAT.BARCODE database		rbcL/18S, diatoms

POSTPROCESSING

Post-processing tools. *See this page*

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 4: docker pull pipecraft/IMAGE:TAG

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

Delete an image

Listing 5: 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
```

github

Post-processing tools

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

Contents

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-occurence 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*

### 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 then specified cutoff will be excluded from the analysis
`min_sample_reads`	cutoff for reads per sample. Samples with lower reads then 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()
```

github

Example data analyses

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

Inspect quality profiles

Examine the quality profiles and basic statistics of the your data set using **QualityCheck module**. *See here*.

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 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 sequene 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).

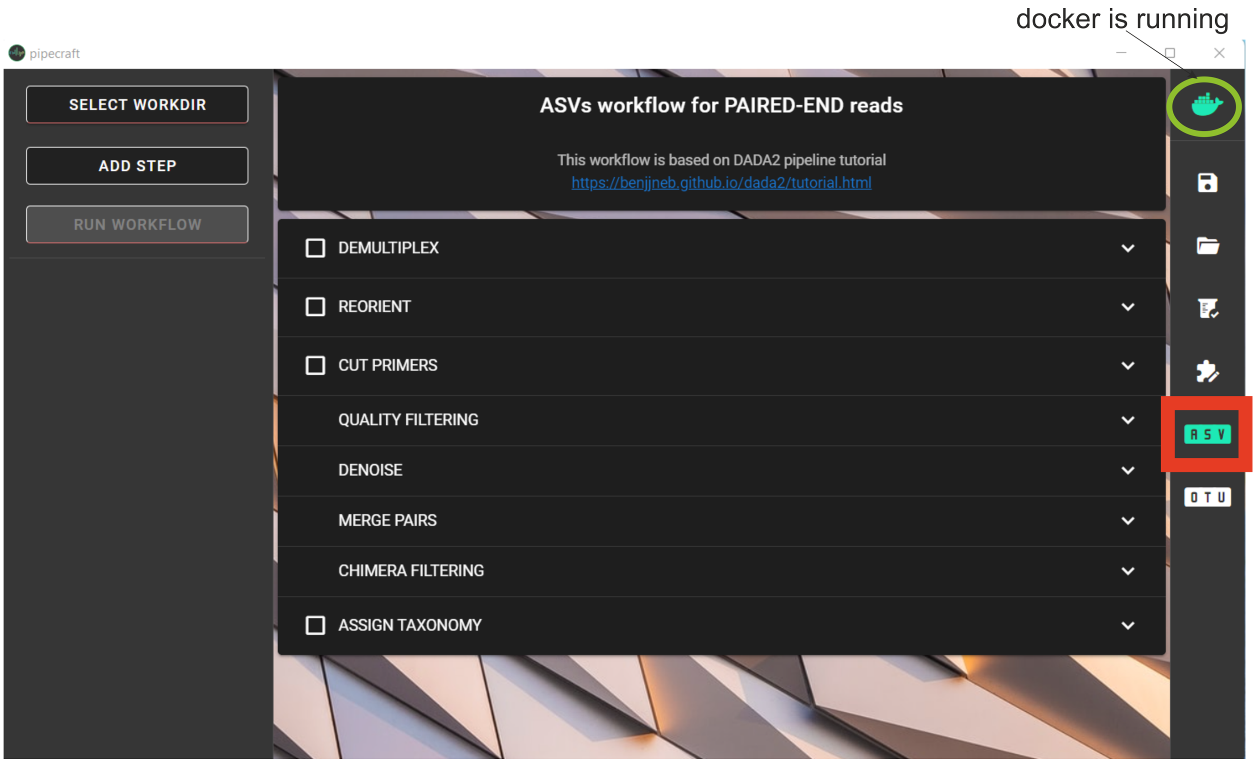
Example:
F3D0_S188_L001_R1_001.fastq
F3D0_S188_L001_R2_001.fastq

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

Secify
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).



ASVs workflow for PAIRED-END reads

This workflow is based on DADA2 pipeline tutorial
https://benjjneb.github.io/dada2/tutorial.html

☐ DEMULTIPLEX

☐ REORIENT

☐ CUT PRIMERS

QUALITY FILTERING

DENOISE

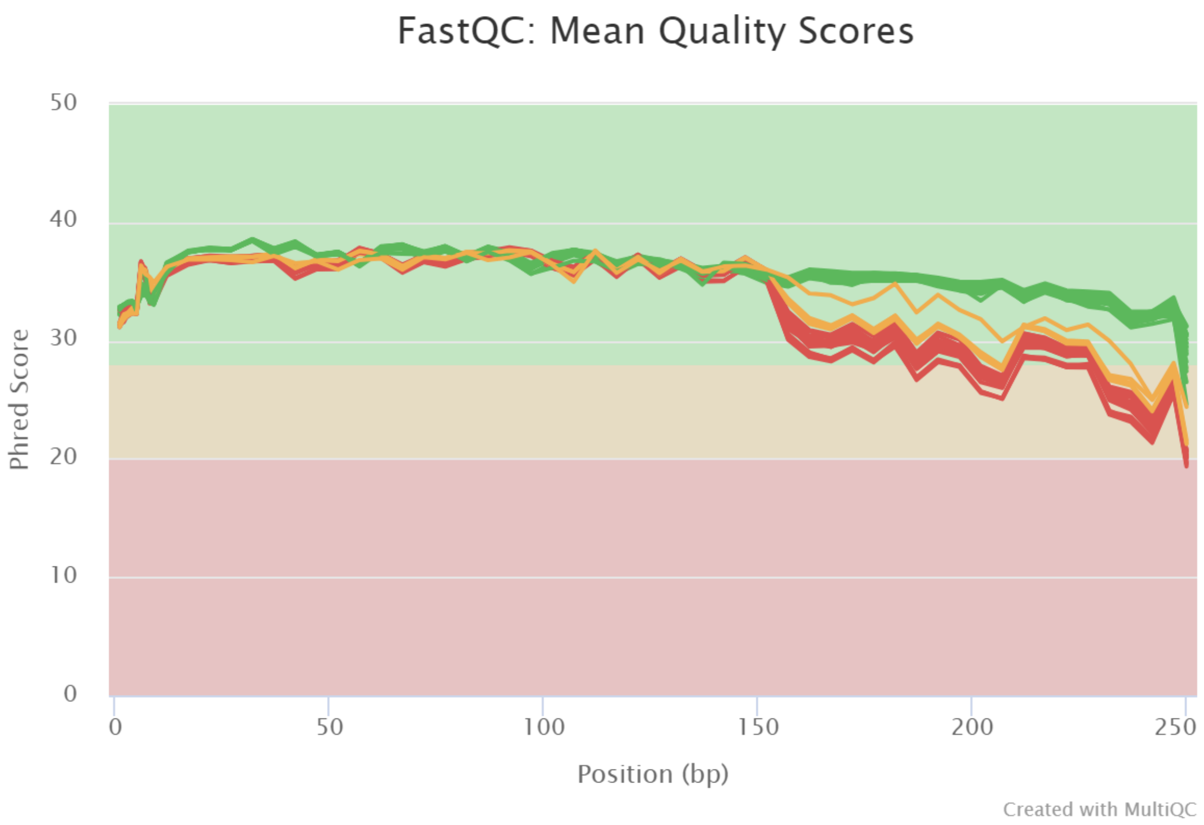
MERGE PAIRS

CHIMERA FILTERING

☐ ASSIGN TAXONOMY

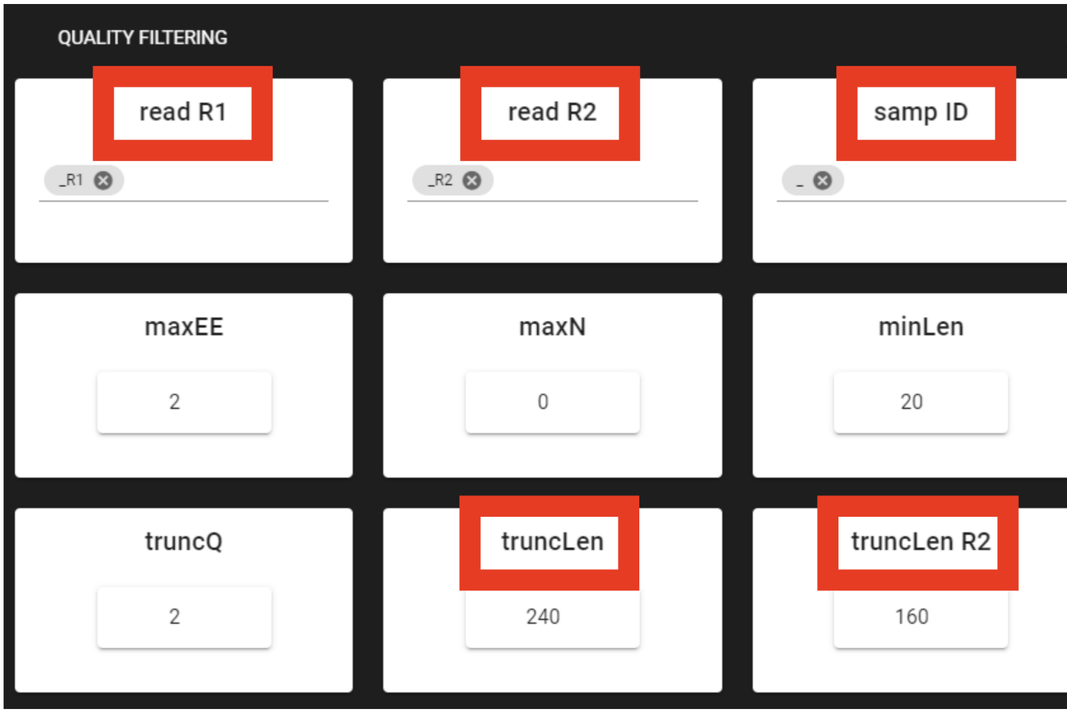
```
3. 'QUALITY FILTERING'
```

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 algoritm 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)

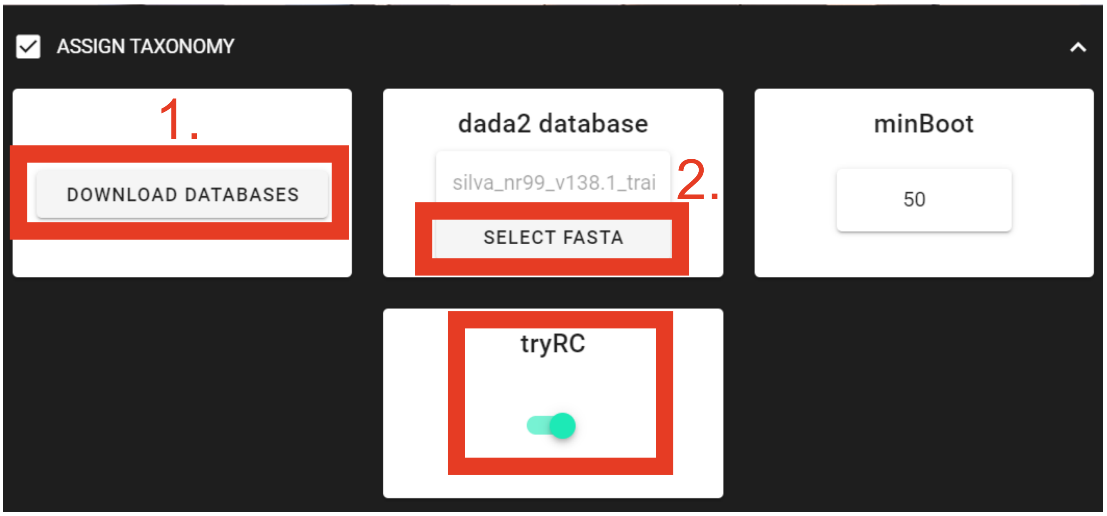
This step performs chimera filtering on denoised and merged reads.
ASV table is generated during this step
Chimera filtering settings here

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, **swithc ON** `tryRC` to include reverse-complement search.



This step assigns taxonomy to ASVs using DADA2 assignTaxonomy function.
Assign taxonomy settings here

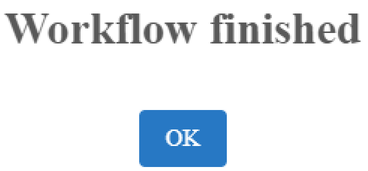
Output directory = `taxonomy_out.dada2`:
taxonomy.txt = classifier results with bootstrap values

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

```
7.  Press** 'START' **to start the analyses.
```

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

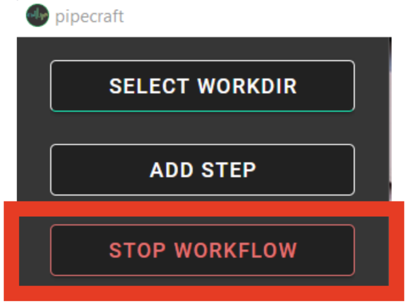
```
When done, 'workflow finished' window will be displayed.
```



Workflow finished

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 representad 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 correspoinding taxonomic unit.

taxonomy.txt; first 10 ASVs

ASV	Sequen	Kingdom	Phylum	Clas	Order	Family	Genus	Spec	Kingdom	Phylum	Clas	Order	Family	Genu	Species
ASV_	TACC GAG.	Bacteria	Bacteroic	Bacteroic	Bacteroid	Muribaculaceae	NA	NA	100	100	100	100	100	100	100
ASV_	TACC GAG.	Bacteria	Bacteroic	Bacteroic	Bacteroid	Muribaculaceae	NA	NA	100	100	100	100	100	100	100
ASV_	TACC GAG.	Bacteria	Bacteroic	Bacteroic	Bacteroid	Muribaculaceae	NA	NA	100	100	100	100	100	100	100
ASV_	TACC GAG.	Bacteria	Bacteroic	Bacteroic	Bacteroid	Rikenelaceae	Alistipes	NA	100	100	100	100	100	100	100
ASV_	TACC GAG.	Bacteria	Bacteroic	Bacteroic	Bacteroid	Muribaculaceae	NA	NA	100	100	100	100	100	100	100
ASV_	TACC GAG.	Bacteria	Bacteroic	Bacteroic	Bacteroid	Muribaculaceae	NA	NA	100	100	100	100	100	95	95
ASV_	TACC TAG.	Bacteria	Firmicutes	Clost	Lachnospirales	Lachnospir	Lachnospiraceae NK4A136 group	NA	100	100	100	100	100	100	99
ASV_	TACC GAG.	Bacteria	Bacteroic	Bacteroic	Bacteroid	Muribaculaceae	NA	NA	100	100	100	100	100	100	100
ASV_	TACC GAG.	Bacteria	Bacteroic	Bacteroic	Bacteroid	Bacteroida	Bacteroides	caecimuri	100	100	100	100	100	100	77
ASV_	TACC GAG.	Bacteria	Bacteroic	Bacteroic	Bacteroid	Muribaculaceae	NA	NA	100	100	100	100	100	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).

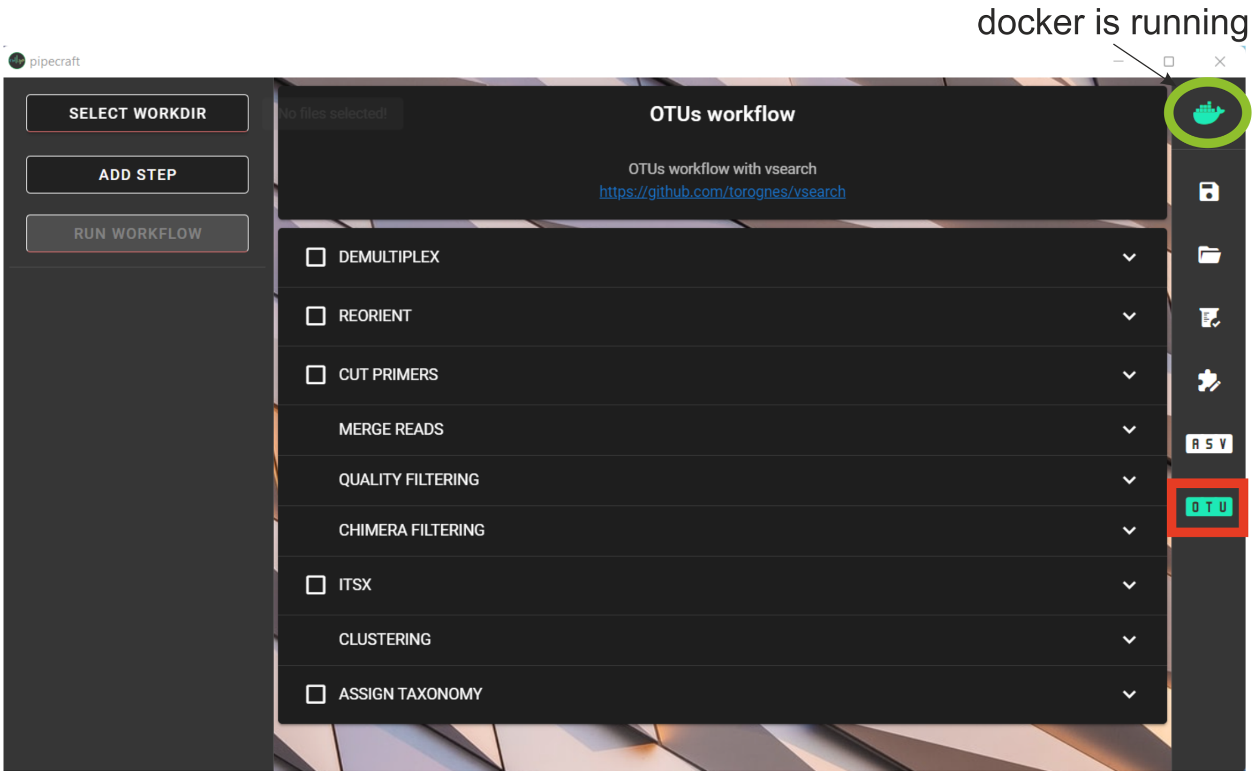
Example:
F3D0_S188_L001_R1_001.fastq
F3D0_S188_L001_R2_001.fastq

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

Secify
`sequencing data format` as **demultiplexed**;
`sequence files extension` as ***.fastq**;
`sequencing read types` as **paired-end**.

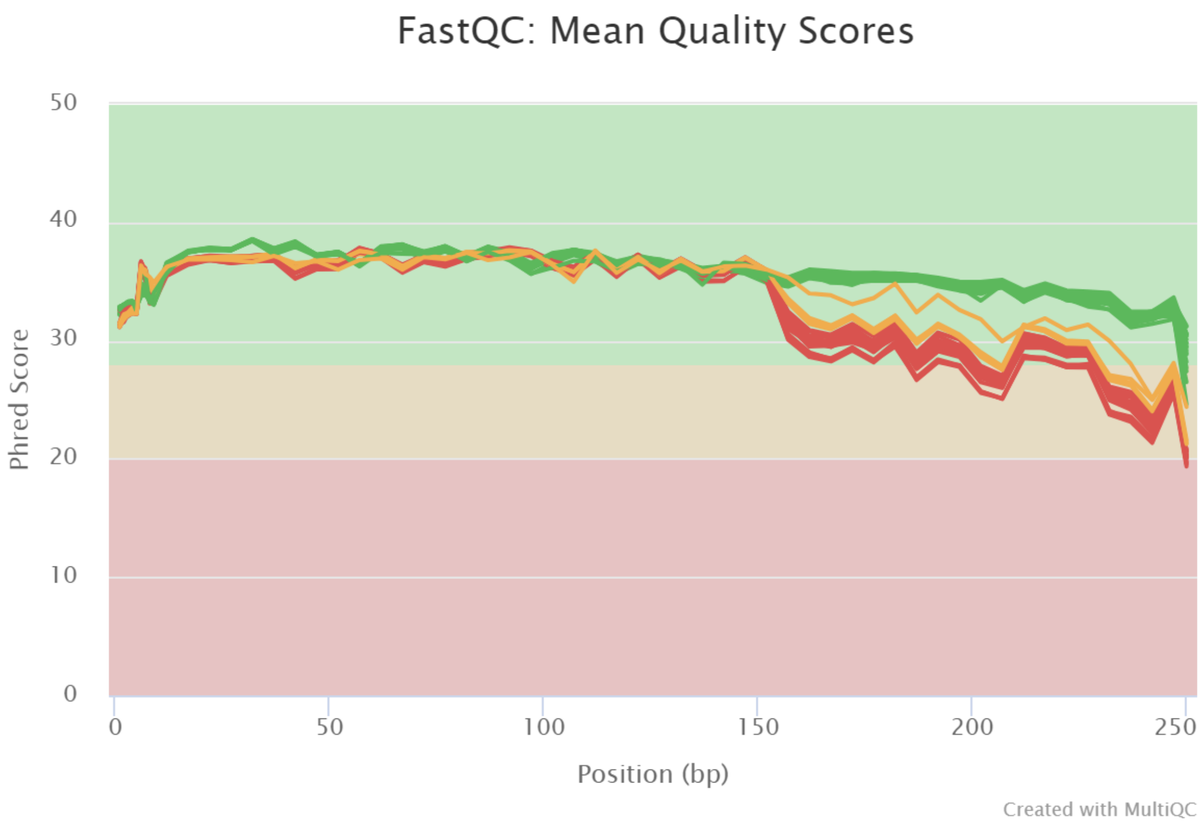
```
2. Select 'OTU workflow' panel (right-ribbon) and check that docker is running (green␣
↪icon);
```

- 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).



docker is running

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*).



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.

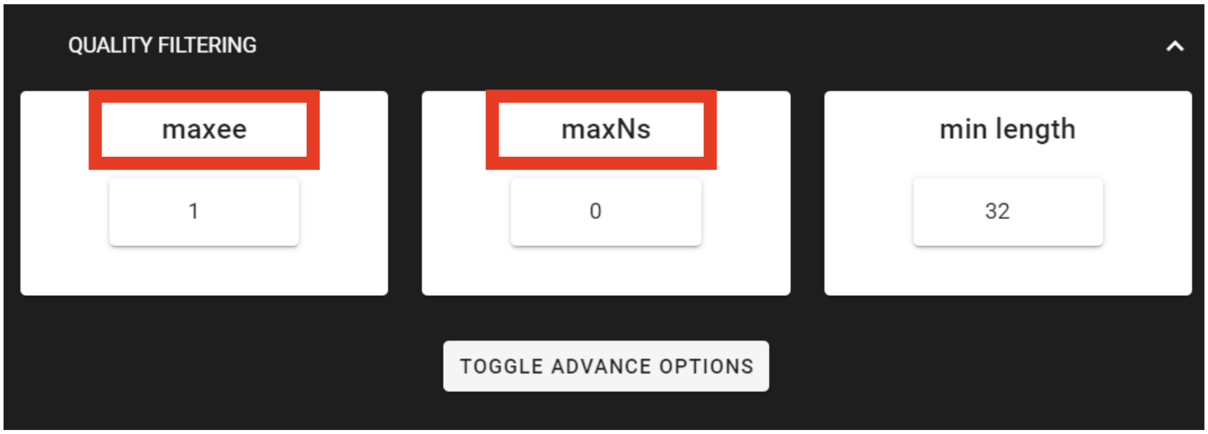
Note: If `include_only_R1` option = TRUE, then unassembled R1 reads will be included to the set of assembled reads per sample. This may be useful when working with e.g. ITS2 sequences, because the ITS2 region in some taxa is too long for paired-end assembly using current short-read sequencing technology. 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 (lenght, 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

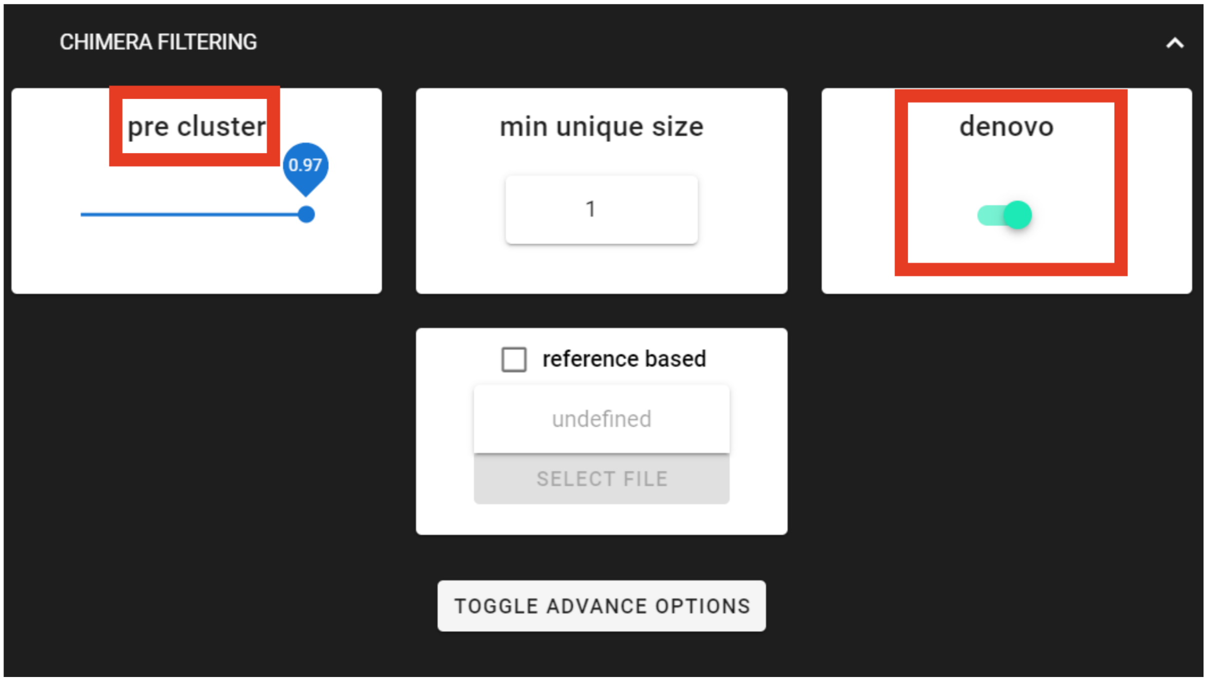
This step performs quality filtering using vsearch.
vsearch quality filtering settings here

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).



CHIMERA FILTERING

pre cluster 0.97

min unique size 1

denovo ●

☐ reference based

undefined

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'
```

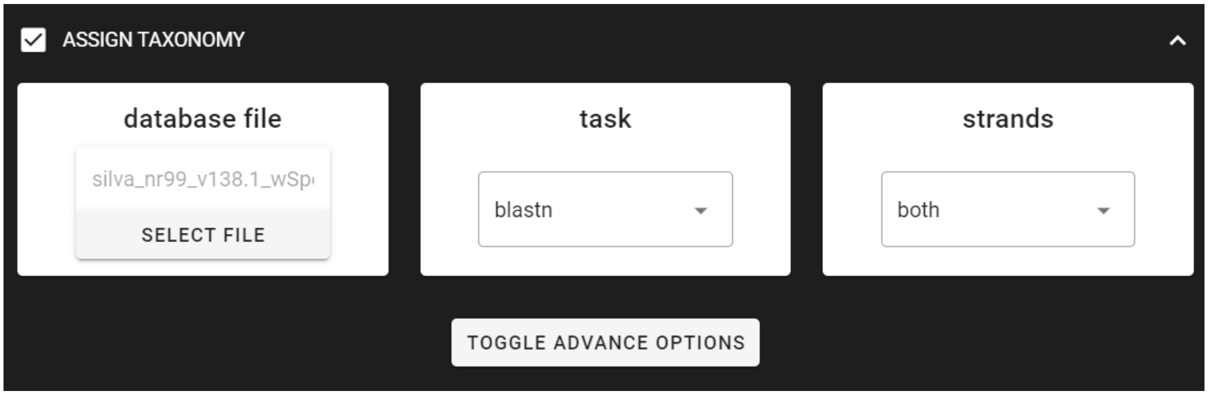
- Here, we use default settings by clustering the reads using 97% similarity threshold

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 separete 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.



☑ ASSIGN TAXONOMY

database file: silva_nr99_v138.1_wSp

SELECT FILE

task: blastn

strands: both

TOGGLE ADVANCE OPTIONS

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.
```

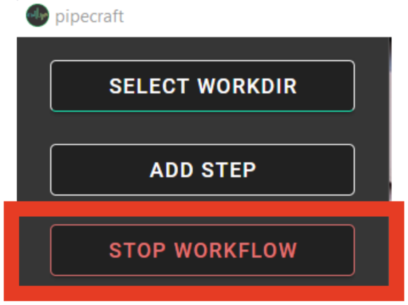
```
1.  Press** 'START' **to start the analyses.
```

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.
```

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 representad in the fasta file (OTUs.fasta) in `clustering_out` directory.

OTUs_table.txt; first 4 samples

OTU_id	F3D0_S188_L0	F3D1_S189_L0	F3D2_S190_L0	F3D3_S191_L001
00fc1569196587dde0462c7d806cc0577	106	271	584	20
02d84ed0175c2c79e8379a99cffb6dbc7f	81	44	88	14
0407ee3bd15ca7206a75d02bb4173251	3	4	3	0
042e5f0b5e38dff09f7ad58b6849fb17ec	20	83	131	4
07411b848fcda497fd29944d351b8a2ec	1	0	2	0
07e7806a732c67ef090b6b279b74a87fe	18	22	83	7
0836d270877aed22cd247f7e703b9247f	1	1	0	0
0aa6e7da5819c11973f186cb35b3f4f58	1	4	5	0
0c1c219a4756bb729e5f0ceb7d82d932b	18	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 againt 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

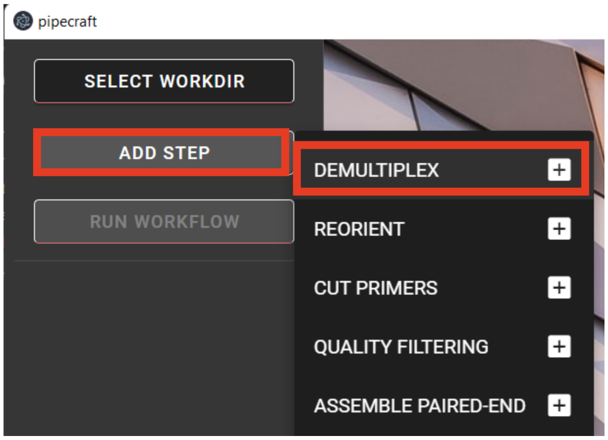
Working with paired-end raw multiplexed data.

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

Secify
`sequencing data format` as **multiplexed**;
`sequence files extension` as **may be fastq or fasta** formatted files;
`sequencing read types` as **paired-end**.

```
2. 'DEMULTIPLEX'
```

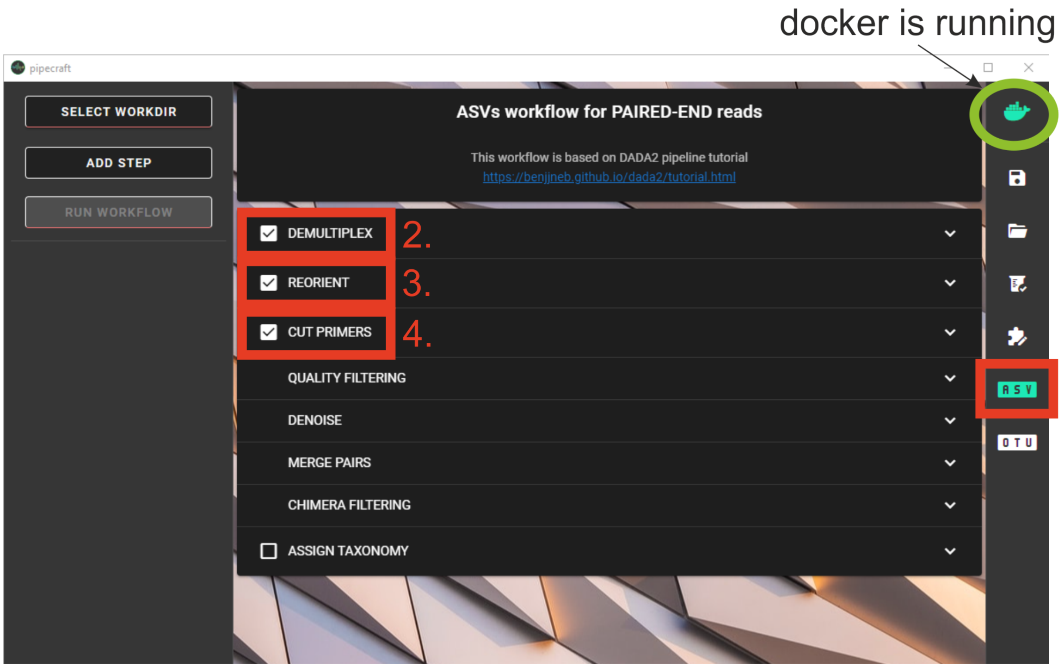
2.1 Press `ADD STEP` -> `DEMULTIPLEX`



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])



ASVs workflow for PAIRED-END reads

This workflow is based on DADA2 pipeline tutorial
https://benjjneb.github.io/dada2/tutorial.html

(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

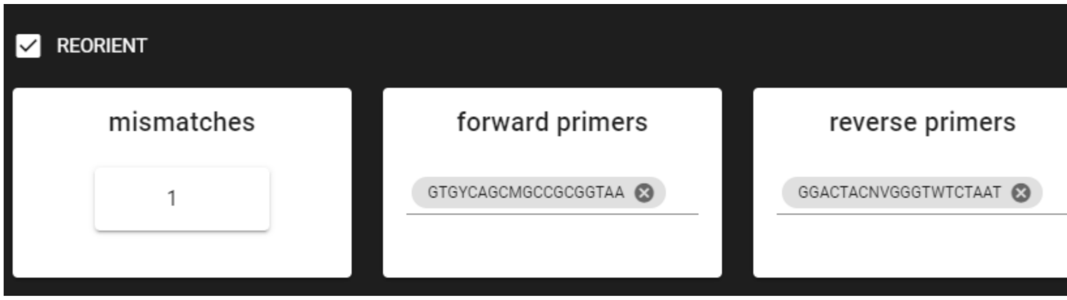
(click on the image for enlargement)

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/fasta 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)



☑ REORIENT

mismatches: 1

forward primers: GTGYCAGCMGCCGCGGTAA

reverse primers: GGACTACNVGGGTWTCTAAT

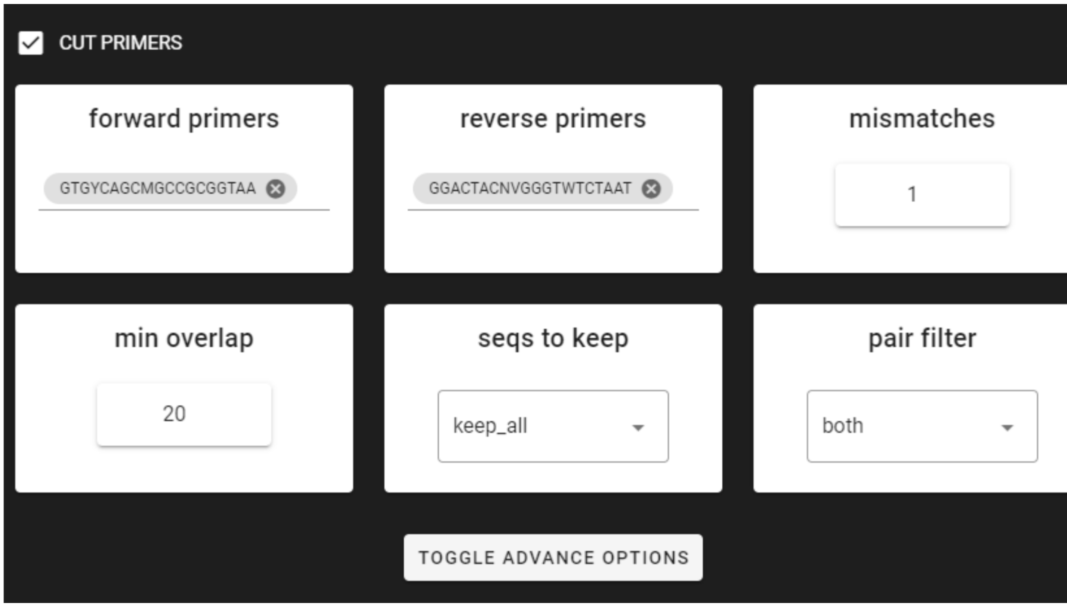
(click on the image for enlargement)

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)



☑ CUT PRIMERS

forward primers: GTGYCAGCMGCCGCGGTAA

reverse primers: GGACTACNVGGGTWTCTAAT

mismatches: 1

min overlap: 20

seqs to keep: keep_all

pair filter: both

(click on the image for enlargement)

This step clipps 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*

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

coming soon . . .

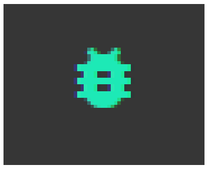
github

Troubleshooting

This page is developing based on the user feedback.

Debugging mode

Turn on '**debugging mode**' (bottom-right button) to keep temporary (log) files for identifying the cause of the error



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".

Possible reason: Computer's memory (RAM) is full, and process was killed. Cannot finish the analyses with those local resources.

Possible fix: In Windows, try to increase the RAM size accessible to Docker (see *here*).

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

Error: No OTU_table.txt with version v0.1.4

Reason: known bug.

Fix: Fixed the bug. Reinstall PipeCraft v0.1.4 (or higher)

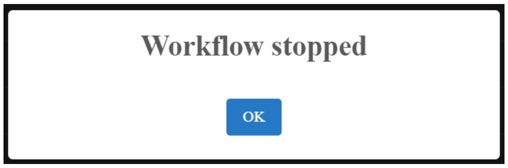
Error: "ERROR]: cannot find files with specified extension"

Reason: wrongly specified working directory or extension; OR issues with external hard drives in Windows.

Fix: Double-check the specified directory and extention; OR restart Windows.

Error:

"Workflow stopped"

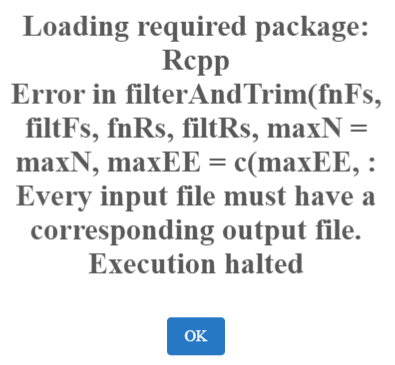


Workflow stopped

Possible reason: Computer's memory (RAM) is full, and process was killed. Cannot finish the analyses with those local resources.

Possible fix: In Windows, try to increase the RAM size accessible to Docker (see *here*).

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

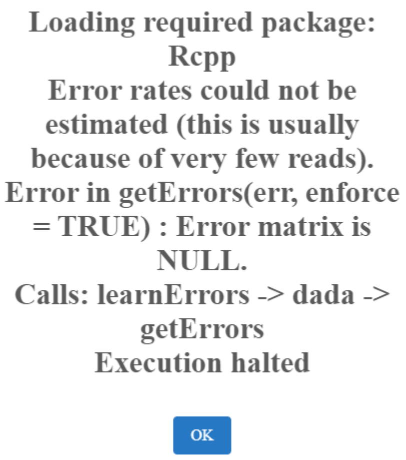


Loading required package:
Rcpp
Error in filterAndTrim(fnFs, filtFs, fnRs, filtRs, maxN = maxN, maxEE = c(maxEE, :
Every input file must have a corresponding output file.
Execution halted

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 identifyer 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'.).

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

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

Fix: use OTU workflow.

github

Licence

github

Contact and Acknowledgements

```
Sten Anslan <sten.s.anslan[at]jyu.fi>
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.

.

Acknowledgements

PipeCraft2 development has been supported by:

- European Regional Development Fund and the programme Mobilitas Pluss, MOBTP198.
- Horizon Europe under the Biodiversity, Circular Economy and Environment (REA.B.3); co-funded by the Swiss State Secretariat for Education, Research and Innovation (SERI) under contract number 22.00173; and by the UK Research and Innovation (UKRI) under the Department for Business, Energy and Industrial Strategy's Horizon Europe Guarantee Scheme.

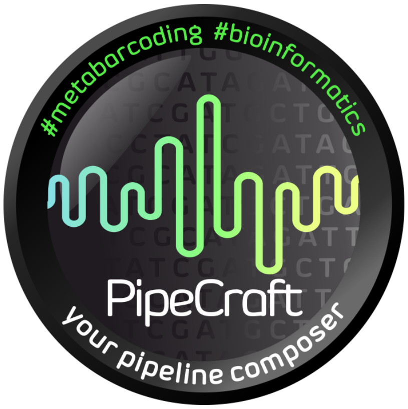
Mobilitas+

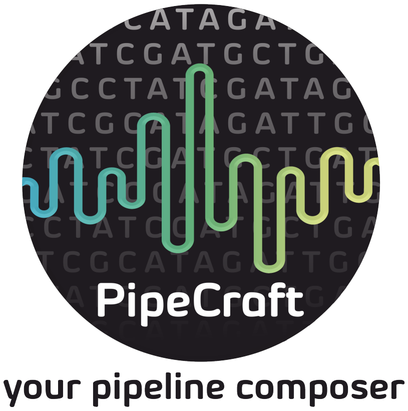
European Union
European Regional
Development Fund

Investing
in your future

PipeCraft2 logos/icons

design by © **Olesia Dulia**





github

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: pipecraft2-manual.rtfd.io/

e.g. "… using PipeCraft 2 (Anslan et al 2017; pipecraft2-manual.rtfd.io/)"

PipeCraft version 1.0 is here

Work that has cited PipeCraft:

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

github

Releases

Contents

1.0.0 (01.09.2023)

DOWNLOAD link for v1.0.0

- major updates in the front-end; individual tools on the right, pipelines on left.
- added debugging mode and improved log info
- added NextITS pipeline for PacBio ITS sequences (not available for MacOS release)
- added ORFfinder + HMM bsed pseudogene/off-targets filtering for protein coding genes
- added RDP classifier
- added DADA2 pipeline for PacBio data
- added DADA2 pipeline for paired-end mixed oriented amplicons (fwd_orient and rev_orient are denoised separately and then merged)
- implemented DADA2 denoising sensitivity editing
- all features will get sha1 ID
- added ASVs to OTUs module (cluster ASVs into OTUs with vsearch)
- added tag-jumps filtering module (UNCROSS2)

- fixed the vsearch_dada2 container issues for MacOS

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

Software	version
NextITS pipeline (link)	**0.5.0**
ORFfinder (link)	**v0.4.3**
RDP classifier	**v2.13**
DADA2	**1.27**
vsearch	**2.23**
trimmomatic	0.39
seqkit	2.3.0
cutadapt	**4.4**
mothur	1.46.1
ITS Extractor	1.1.3
fqgrep	0.4.4
BLAST	**2.14.0+**
FastQC	0.11.9
MultiQC	1.12
LULU	0.1.0
fastp	0.23.2
DEICODE	0.2.4

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

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

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 sequeces

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
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 (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

0.1.1 (01.04.2022)

Minor cosmetic changes and bug fixes. DOWNLOAD link for v0.1.1

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

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

github

Docker images

Docker images (with the bioinformatic tools) used in PipeCraft2 are stored on Dockerhub. These images can be used to launch any tool with the Docker CLI to utilize the compiled tools.

Images in use

Image	Software
pipecraft/vsearch_dada2:2	vsearch v2.23, dada2 v1.27, seqkit v2.3.0, lulu v0.1.0, R, GNU parallel etc
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/reorient:1	fqgrep v0.4.4, seqkit v2.3.0
pipecraft/trimmomatic:0.39	trimmomatic 0.39, seqkit v2.3.0
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

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
pipecraft/vsearch_dada2:	has issues with MacOS. vsearch v2.22.1, dada2 v1.20, seqkit v2.3.0, lulu v0.1.0, R, GNU parallel

github

For Developers

Prerequisites

Docker Desktop
https://www.docker.com/products/docker-desktop/

NodeJS 14
https://nodejs.org/download/release/latest-v14.x/

Yarn

```
npm install --global yarn
```

Git
(https://git-scm.com/downloads)

Setting up the environment

Clone the repository:

```
git clone https://github.com/pipecraft2/pipecraft
```

Install and run:

```
cd pipecraft
yarn run install_pipe
yarn electron:serve
```

Developer tools

The apps front-end desing, data storage and navigation are built using the Vue framework and its plugins. To build new or to modify exiting components will reiqure a some proficiency in JavaScript and the Vue framework.
Other important tools include electron which is mainly used for interacting with the file system and dockerode which is used for controlling docker.

Electron 13:
https://www.electronjs.org/
Vue 2:
https://v2.vuejs.org/v2/guide/
Vuetify 2:
https://v2.vuetifyjs.com/en/introduction/why-vuetify/#what-is-vuetify3f
Vue Router 3:
https://v3.router.vuejs.org/guide/
Vuex 3:
https://v3.vuex.vuejs.org/#what-is-vuex
dockerode:
https://github.com/apocas/dockerode

Note: All of these tools are automatically installed during the install_pipe command.

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

1.4.2 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	full pipeline operations
vsearch	Rognes et. al 2016	quality filtering, assemble paired-end reads, chimera filtering, clustering
NextITS	Mikryukov et. al	pipeline for fungal full-ITS (PacBio); not available in Mac version of PipeCraft2
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
fqgrep	Indraniel Das 2011	core for reorient reads
BLAST	Camacho et al. 2009	assign taxonomy
RDP classifier	Wang et al. 2007	assign taxonomy
ORFfinder	NCBI Tool	finding open reading frames of protein coding genes (filtering pseudogenes/off-targets)
HMMER	Web site	HMM based filtering if the sequences (filtering pseudogenes/off-targets)
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 implemeted to PipeCraft (*contacts*) or create an issue in the main repository.