Metabarcoding Workshop

Anna Heintz-Buschart

June 2022

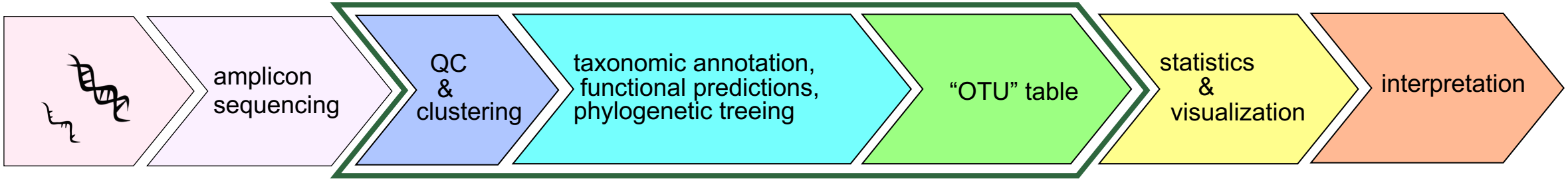
University of Amsterdam
Life Sciences

Overview of today

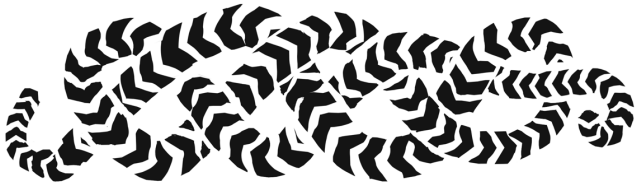
- A look at the aims
- Overview of the method
- Limitations – from sample to sequencing data
- How do we try to deal with these limitations?
- Which problems persist?

- dadasnake – aims and realization
- dadasnake: options in detail
- Q&A
- what to do if it doesn't work

dadasnake pipeline



dadasnake



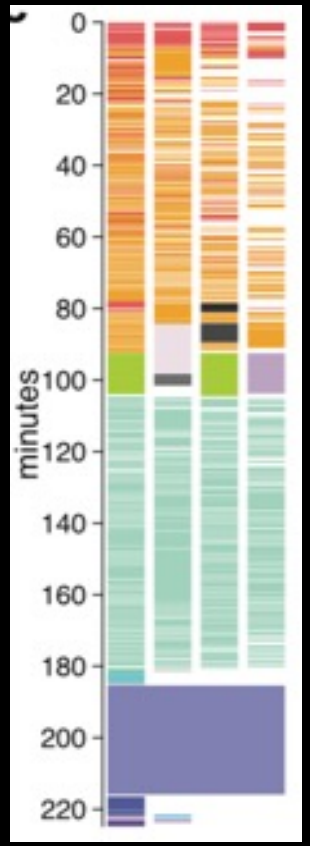
- https://github.com/a-h-b/dadasnake

dadasnake pipeline – aim & ambition

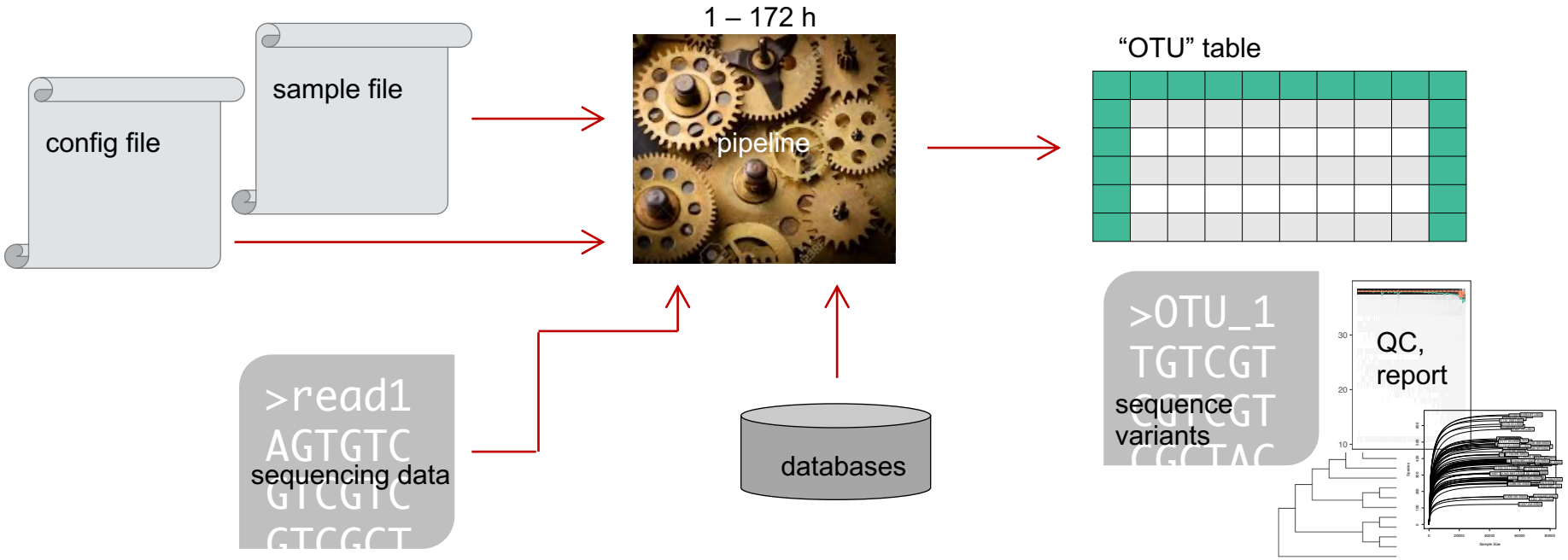
- wrap DADA2 + pre-/post-processing
- be more configurable than qiime2
- be able to use high-performance compute clusters
  = parallelisation, module-based, use big-mem
- be reproducible
- be low-maintenance for the developer

- be really easy to use

pipeline



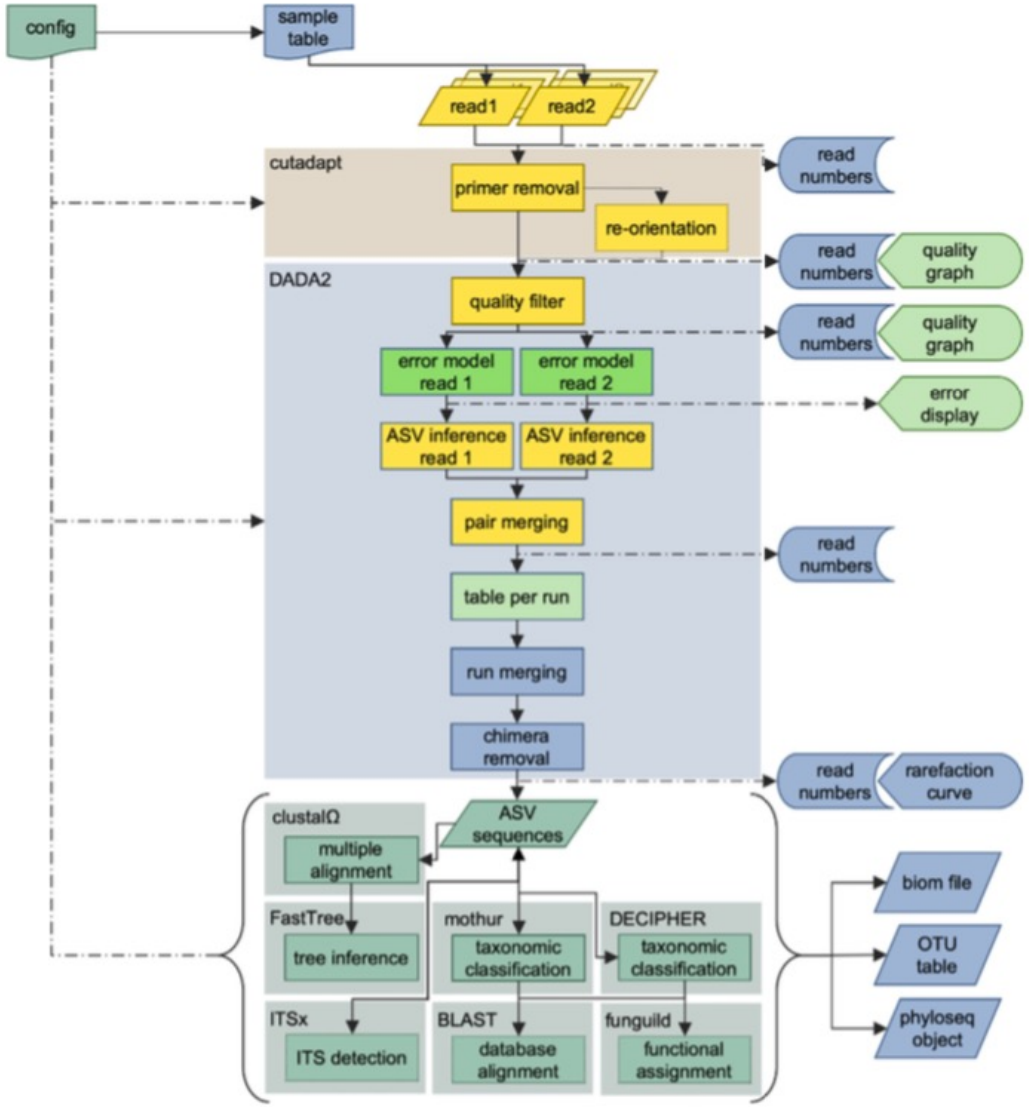
dadasnake pipeline



dadasnake

What does dadasnake do?

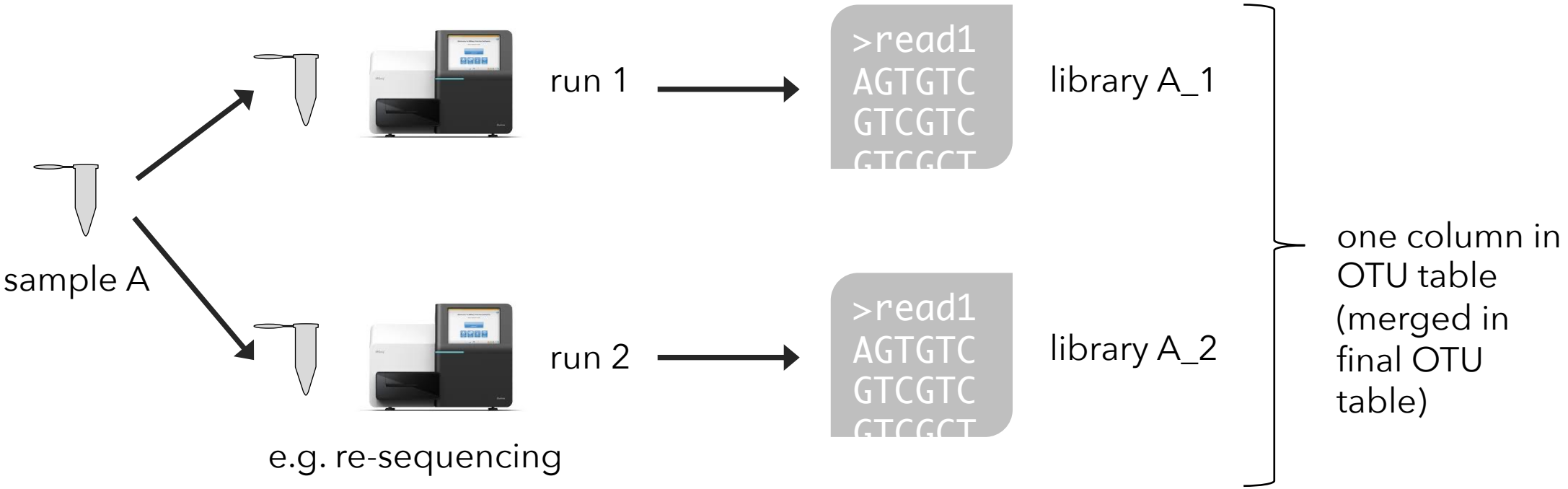
- optional primer removal
- quality filtering and trimming
- optional down-sampling
- error estimation & denoising
- optional paired-ends assembly
- ASV table generation
- optional chimera removal
- taxonomic classification (& ITS detection)
- optional length check, taxonomic filtering
- optional functional annotation/prediction, treeing…
- reporting of stats and quality measures



Output

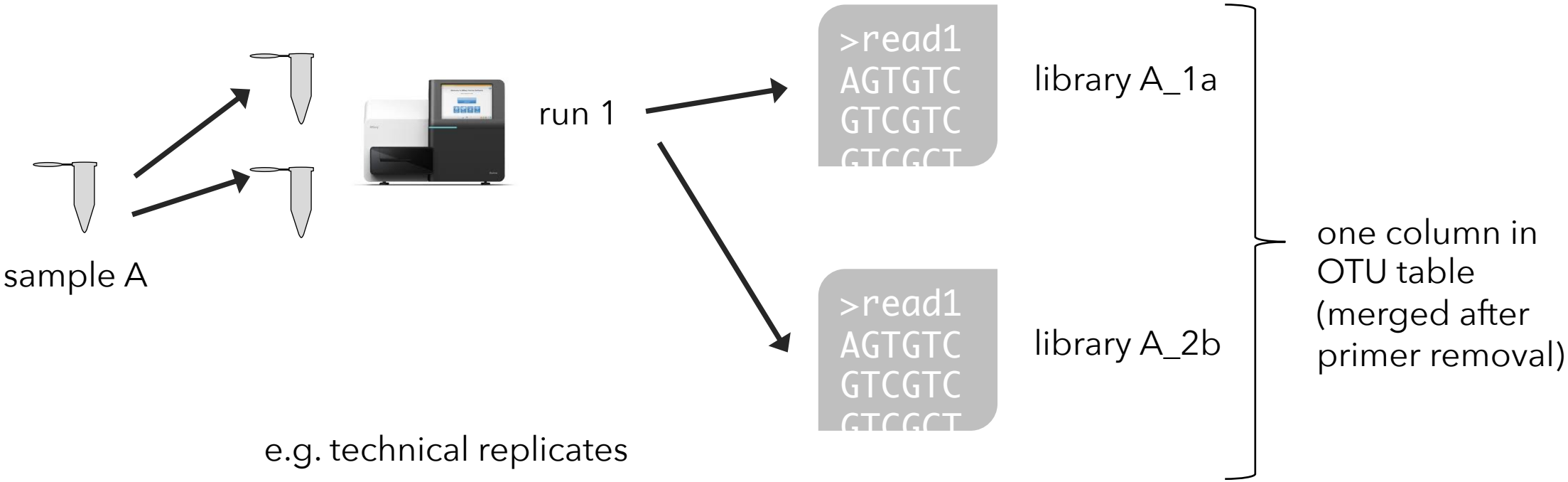
- OTU table with taxonomy and comments
  - .tsv
  - .RDS (optional phyloseq object)
  - optional .biom
- sequences
  - .fasta
- optional phylogenetic tree (.newick)
- optional functional annotation data
- stats (reads at every step, visualization: QC, errors, rarefaction curve)
- configuration, report

Raw data options



sample	library	run	r1_file	r2_file
A	A_1	1	myExp.A_R1.fastq.gz	myExp.A_R2.fastq.gz
A	A_2	2	myExp.A.reseq_R1.fastq.gz	myExp.A.reseq_R2.fastq.gz

Raw data options



sample	library	run	r1_file	r2_file
A	A_1a	1	myExp.A1_R1.fastq.gz	myExp.A1_R2.fastq.gz
A	A_2b	1	myExp.A2_R1.fastq.gz	myExp.A2_R2.fastq.gz

The samples file

- contains all the information on your samples
- must be tab-separated

- should not contain DOS-style end-of-line

- you can change the encoding by opening the config file using vi, then type

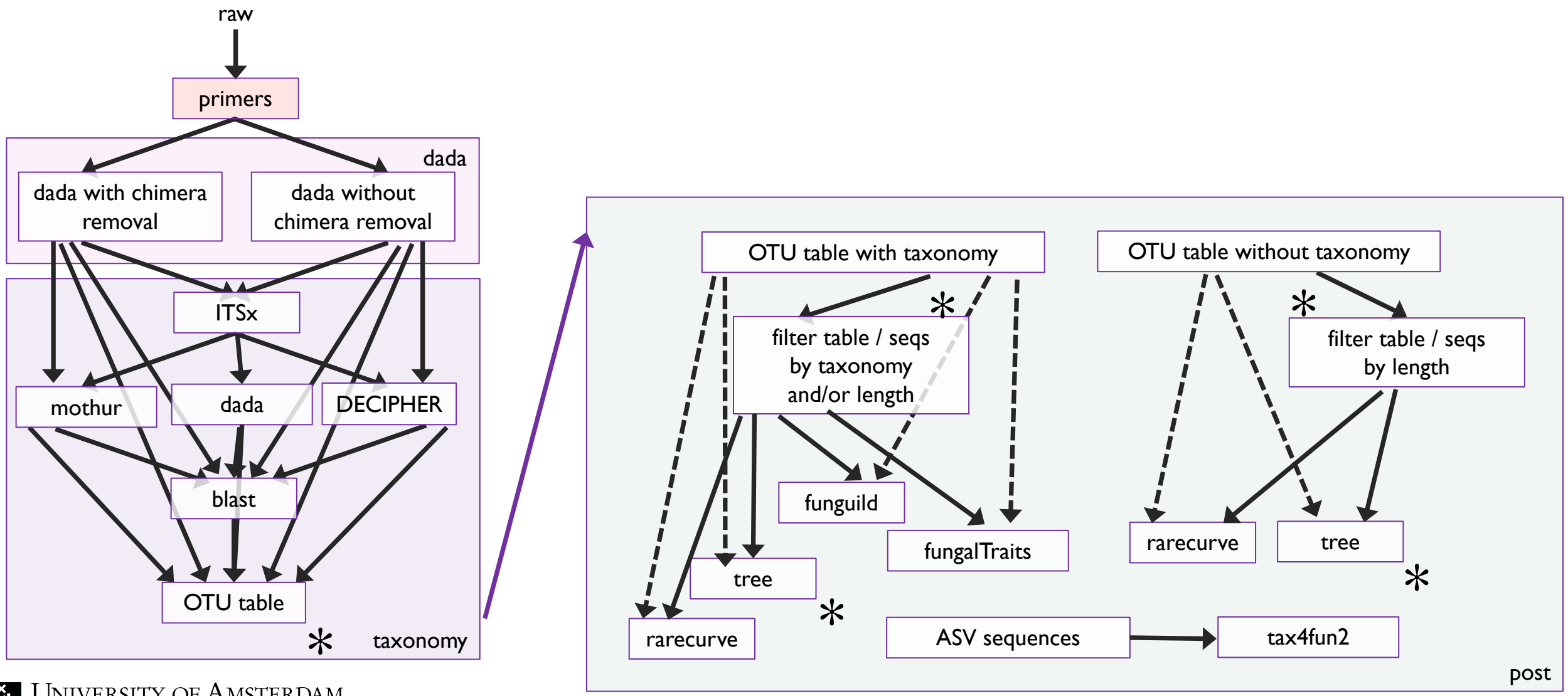
```
:set ff=unix
:wq
```

- must contain named columns: library and r1_file
- can contain named columns: r2_file, sample, run

- libraries and samples should not have the same name, if there are libraries that have different names

Workflows

dadasnake

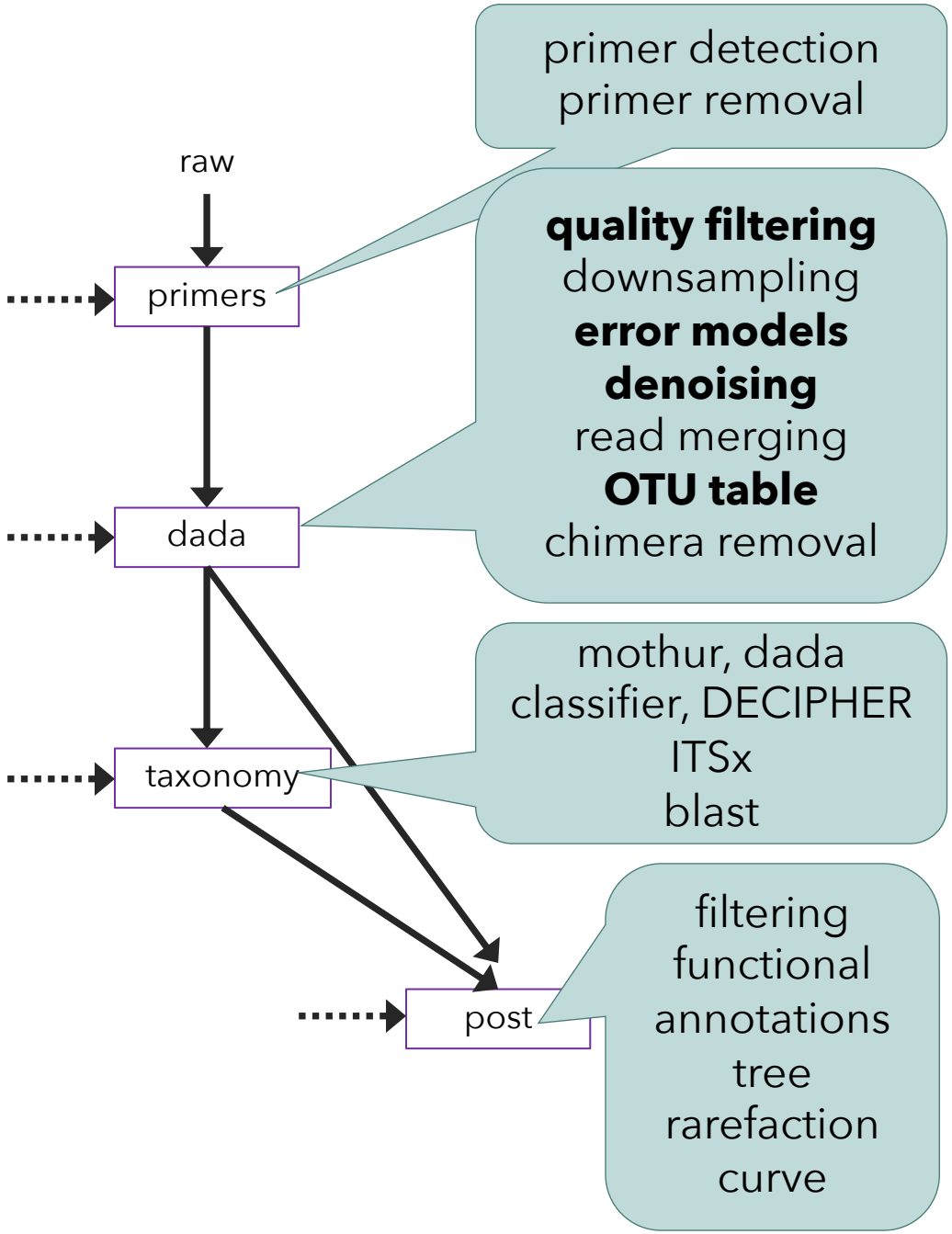


Steps

- by default, all steps are done

➢ `do_step: true`

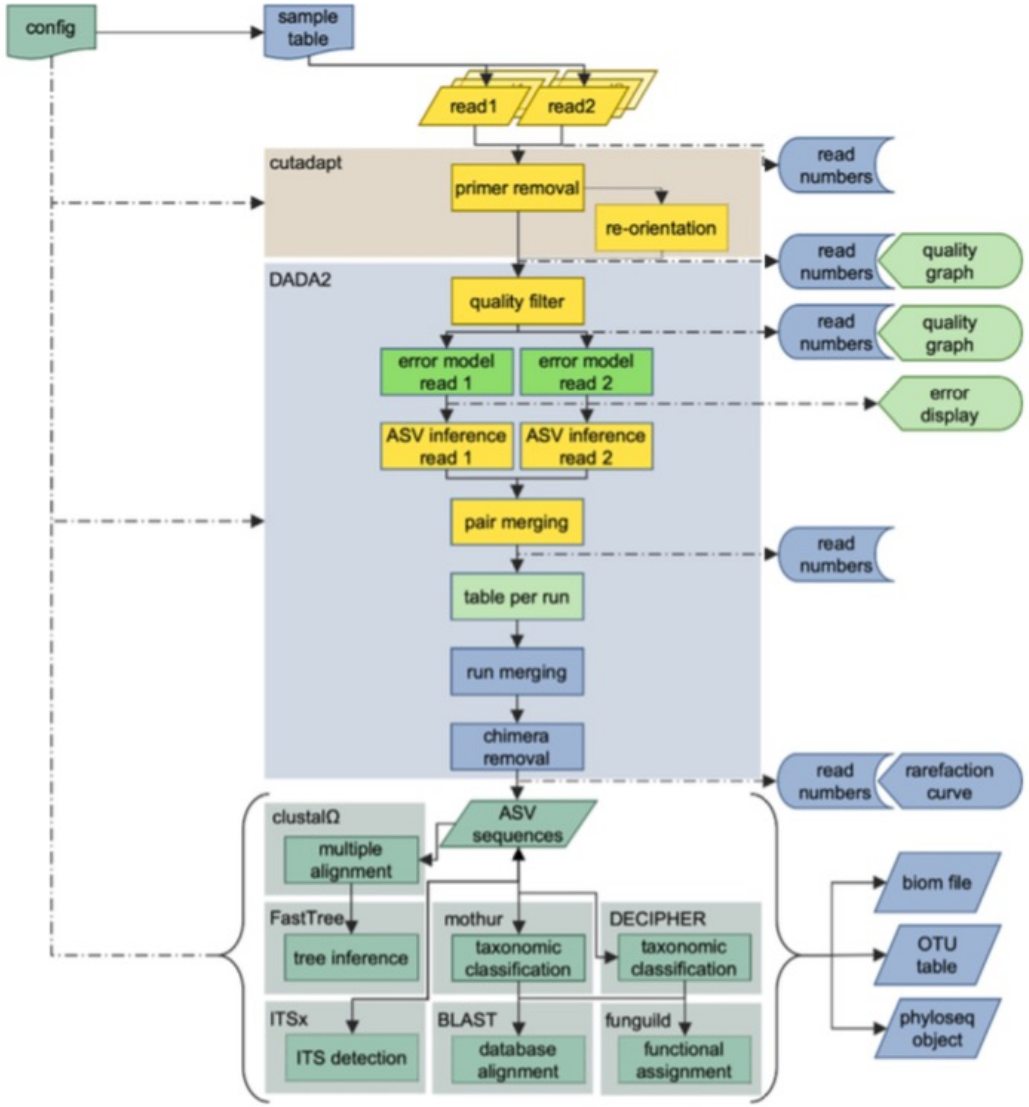
```
do_primers: true
do_dada: true
do_taxonomy: true
do_postprocessing: true
```



dadasnake

Options

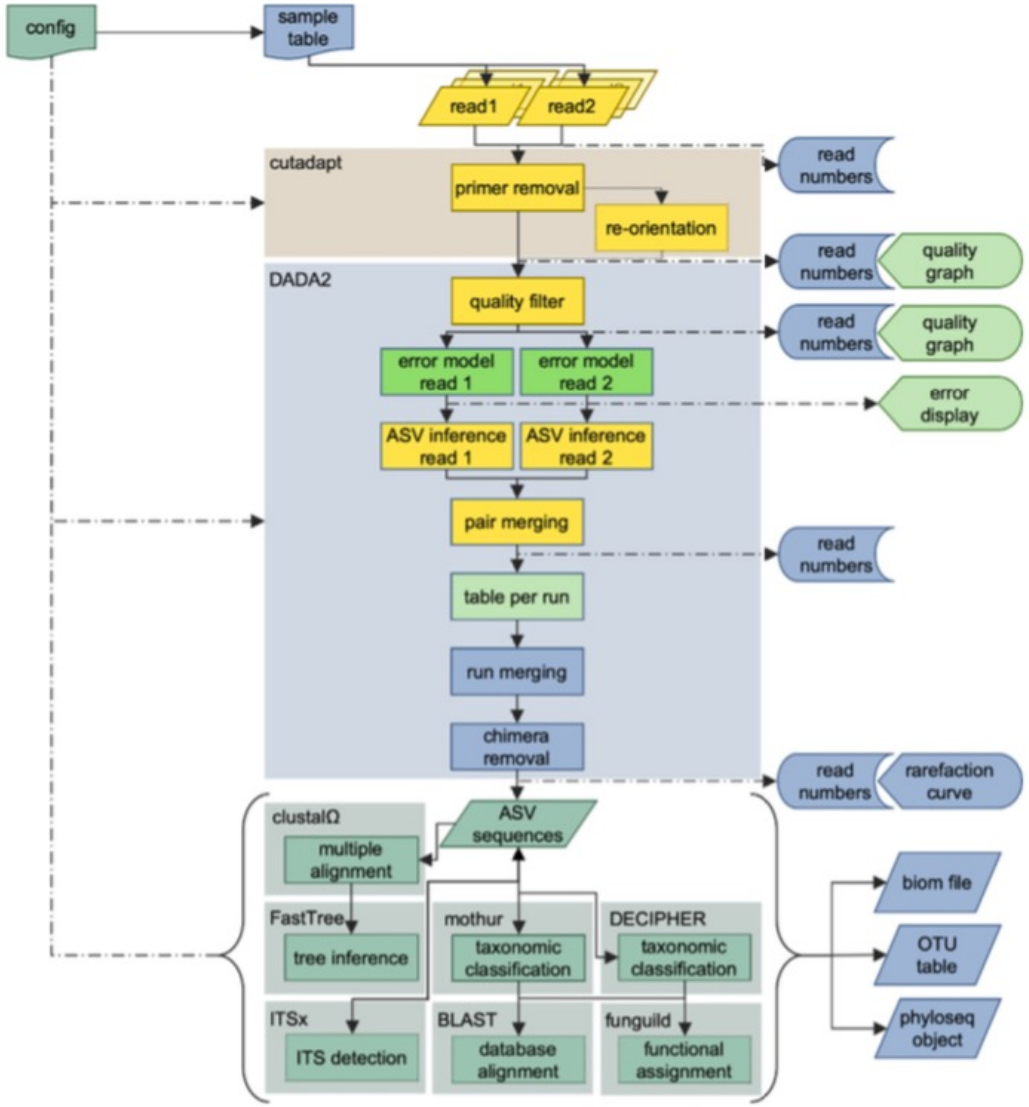
- dadasnake defaults are for 16S rRNA V4 amplicons (515-806)
- it was also extensively benchmarked for fungal ITS2
- suggestions for other targets: AMF, archaea, nematodes, trnL



dadasnake

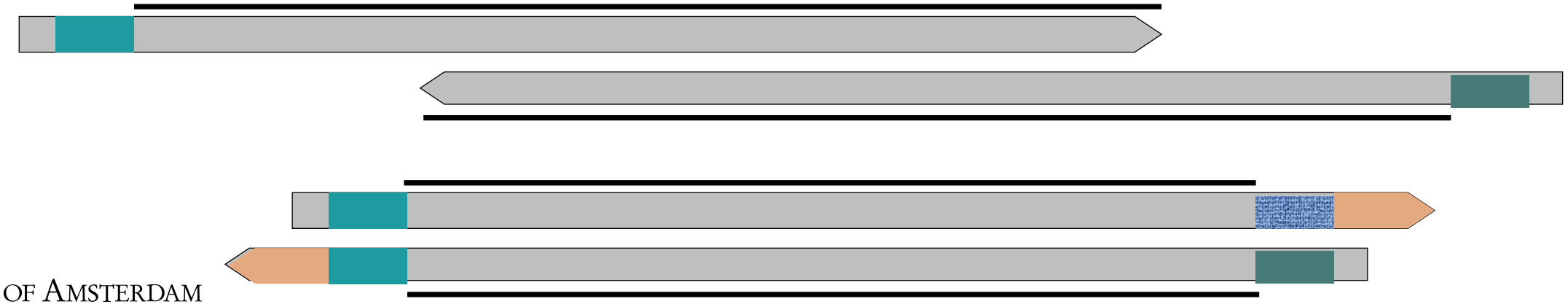
Steps in detail

- please interrupt me at any point in time to ask questions/comment on the steps and options



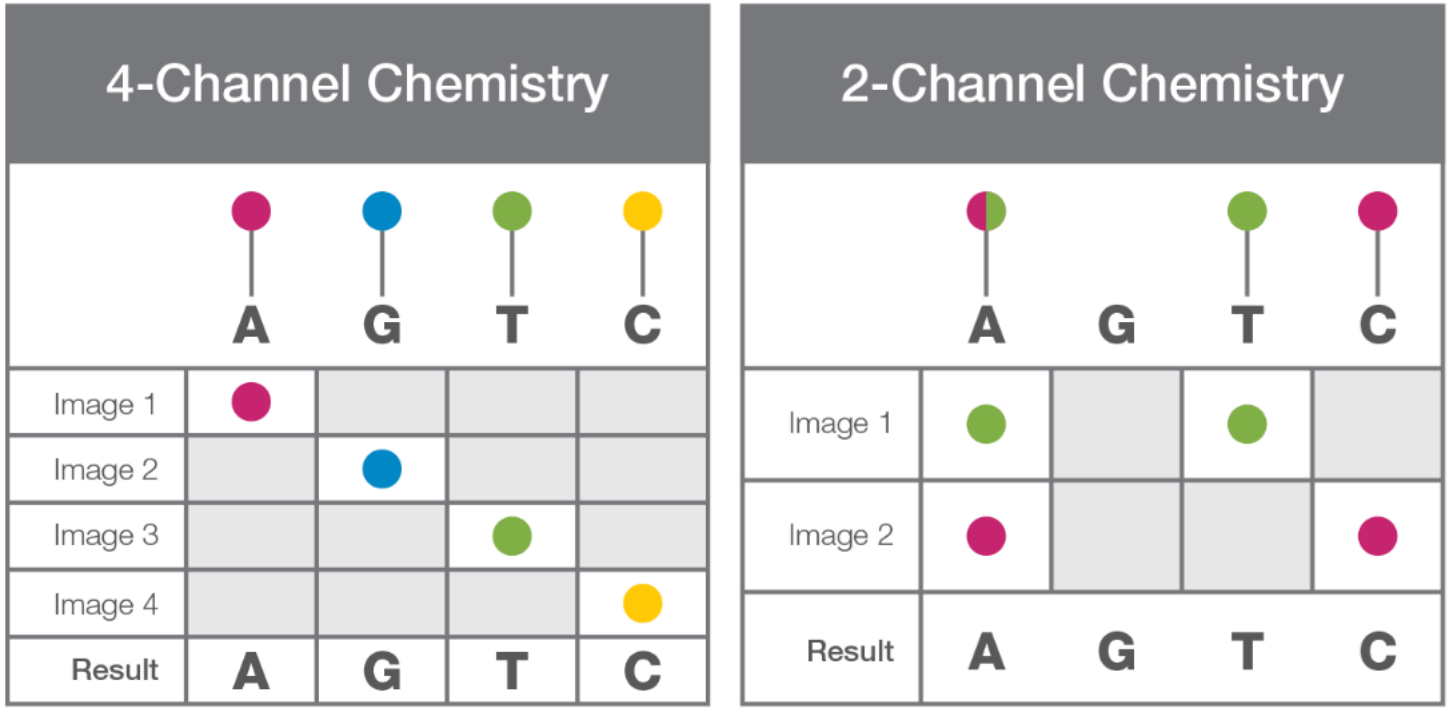
Primer removal

- using cutadapt
- flexible minimal overlap (default 10)
- flexible mismatches (default 20%)
- flexible AND/OR matching (default “any”, i.e. both reads need primers)
- flexible sequencing direction, or automatic detection
- removal of reverse-complement second primer



Quality filtering / trimming

- removal of trailing Gs (dark-cycle) for novaseq/nextseq



https://www.illumina.com/content/dam/illumina-marketing/documents/products/techspotlights/cmos-tech-note-770-2013-054.pdf

Quality filtering / trimming

- removal of trailing Gs (dark-cycle) for novaseq/nextseq
- rest is part of DADA2 pipeline:
- visualization of quality before and after – including fastQC/multiQC
- options:
  - minimum length
  - maximum length
  - truncation at specific length (too short kicked out)
  - truncation before first position with low quality (cut-off user-defined)
  - maximum overall error (based on quality)
  - trim positions from the left

Down-sampling

- quality-filtered/trimmed data can be down-sampled (rarefied) to a specified or minimum number of reads
- if reads of one sample are split into several libraries, the number of reads is adjusted to that

Error profile & denoising

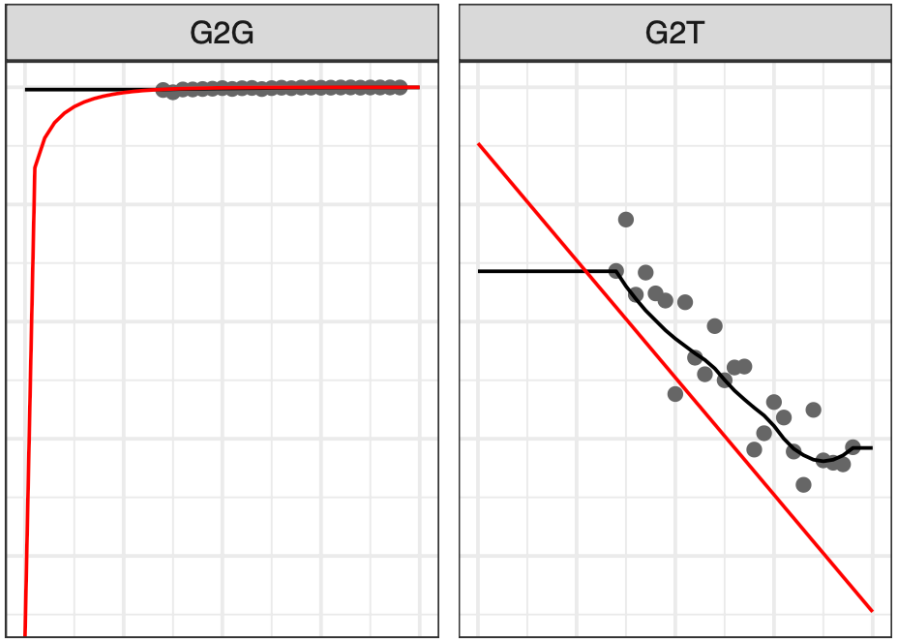
- part of DADA2 pipeline
- build ASVs per sample, per run, or for the whole study
- visualization
- experimental error-models for novaseq data
- settings can be adjusted for non-Illumina data

```
s: ATTAACGAGATTATAACCAGAGTACGAATA...
     |                |
r: ATCAACGAGATTATAACAAGAGTACGAATA...
```

$$p(r|s) = \prod_{i=1}^{L} p(r(i)|s(i), q_r(i), Z)$$

Reminder: error models

- model substitutions for every run




```
s: ATTAACGAGATTATAACCAGAGTACGAATA...
     |                 |
r: ATCAACGAGATTATAACAAGAGTACGAATA...
```

$$p(r|s) = \prod_{i=1}^{L} p(r(i)|s(i), q_r(i), Z)$$

Error rates depend on....

- Substitution (eg. A->C)
- Quality score (eg. Q=30)
- Batch effect (eg. run)

Paired-ends assembly

- part of DADA2 pipeline
- options:
  - minimum overlap (can be 0)
  - number of mismatches

- single-end data can also be used

Chimera removal

- part of DADA2 pipeline
- is done after the “OTU table” is made
- options:
  - consensus
  - pool

- chimera removal is optional

Taxonomic annotation/classification

- choices:
  - DECIPHER algorithm
  - works better than DADA2-native algorithm
  - annotation to genus level
  - but doesn't scale (don't use for large datasets)
- and/or Bayesian classifier from mothur or from dada2 (slower than mothur)
- optional BLAST for unclassified sequences or all sequences, best hit and LCA can be added to ASV table, thanks to BASTA
- options:
  - databases
  - direction
  - before or after optional ITSx

Database choices

- dadasnake does not provide databases
- go get them from the people who make them

- dadasnake comes with a script to prune databases for the mothur classifier
  - select taxa (e.g. Fungi, Bacteria etc.)
  - select based on primer sequences
  - cut to region of interest

Functional annotation/prediction

- dadasnake does not provide databases
- go get them from the people who make them

- funguild
- fungalTraits

- tax4fun2

Other functional information

- bacterial traits DB

- https://github.com/bacteria-archaea-traits/bacteria-archaea-traits
- https://www.nature.com/articles/s41597-020-0497-4

scientific **data**

Data Descriptor | Open Access | Published: 05 June 2020

A synthesis of bacterial and archaeal phenotypic trait data

Joshua S. Madin, Daniel A. Nielsen, [...]Mark Westoby

Scientific Data **7**, Article number: 170 (2020) | Cite this article

Abstract

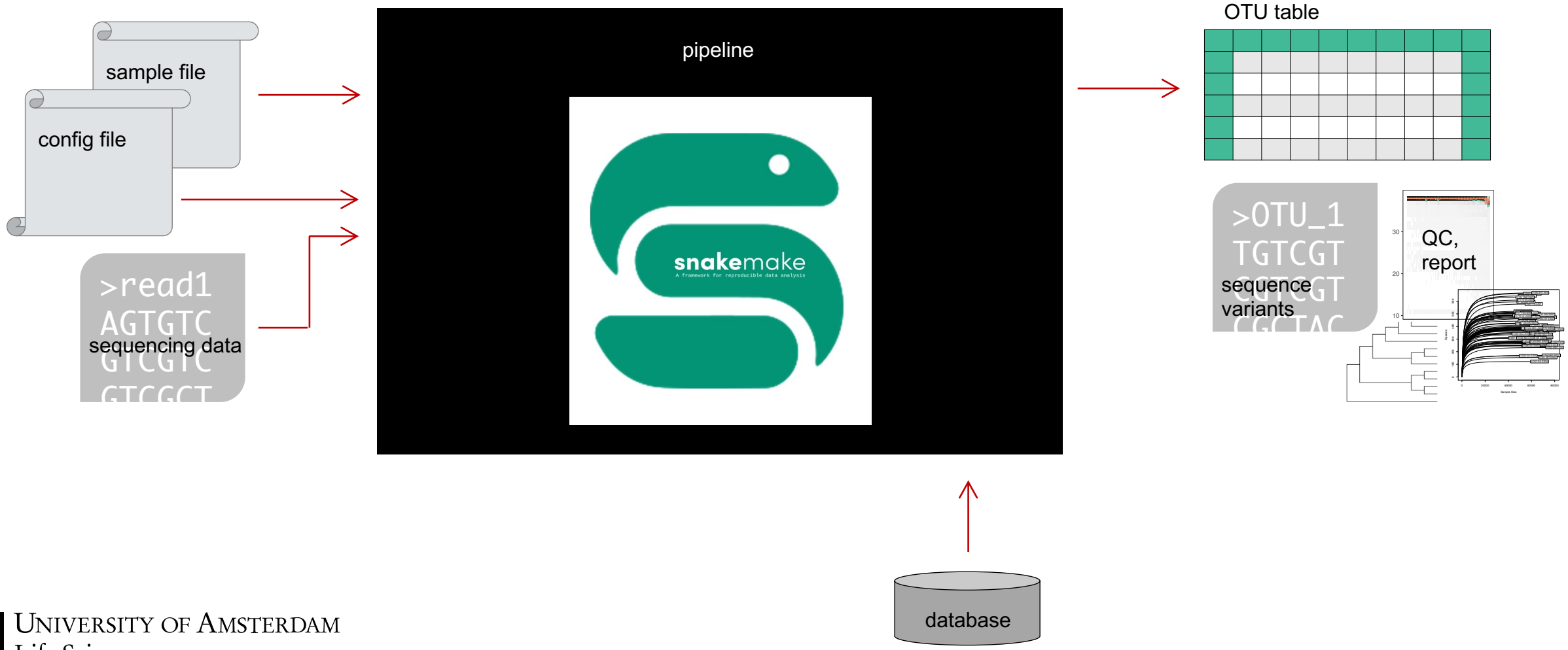
A synthesis of phenotypic and quantitative genomic traits is provided for bacteria and archaea, in the form of a scripted, reproducible workflow that standardizes and merges 26 sources. The resulting unified dataset covers 14 phenotypic traits, 5 quantitative genomic traits, and 4 environmental characteristics for approximately 170,000 strain-level and 15,000 species-aggregated records. It spans all habitats including soils, marine and fresh waters and sediments, host-associated and thermal. Trait data can find use in clarifying major dimensions of ecological strategy variation across species. They can also be used in conjunction with species and abundance sampling to characterize trait mixtures in communities and responses of traits along environmental gradients.

Where do I get more information?

- primer removal: cutadapt - http://gensoft.pasteur.fr/docs/cutadapt/1.18/guide.html
- DADA2 – steps: http://benjjneb.github.io/dada2/index.html
  - quality filtering and trimming, error estimation & denoising, paired-ends assembly, OTU table generation, chimera removal, taxonomic annotation
- taxonomic classification (& ITS detection):
  - DECIPHER: http://www2.decipher.codes/Bioinformatics.html
  - mothur classification: https://www.mothur.org/wiki/Classify.seqs
  - ITSx: https://microbiology.se/software/itsx/
  - BASTA: https://github.com/timkahlke/BASTA/wiki
- functional annotation, treeing…
  - funguild: https://github.com/UMNFuN/FUNGuild
  - fungalTraits: https://github.com/traitecoevo/fungaltraits
  - tax4fun2: https://github.com/bwemheu/Tax4Fun2
  - GTDB: https://gtdb.ecogenomic.org/
  - treeing: http://www.microbesonline.org/fasttree/ http://www.clustal.org/omega/

Questions/comments on steps/options?

How does dadasnake work?



How does snakemake work?

Job execution

A job is executed if and only if

- output file is target and does not exist
- output file needed by another executed job and does not exist
- input file newer than output file
- input file will be updated by other job

How does snakemake make dadasnake work?

Start here

Use these settings

you:

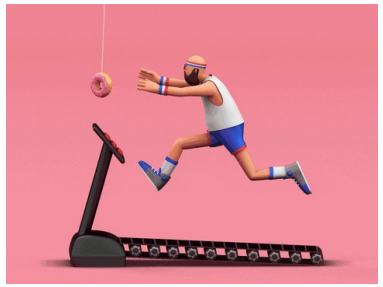
This needs to be done

These are the rules

You're welcome

keeps track of software

submits jobs
and monitors execution



cluster compute nodes

We would like this output

Questions/comments on snakemake?

How to run dadasnake? – installation

- connect to cluster

- install/set up conda
- run `conda config --set auto_activate_base false`
- set up mamba:

  `conda install -n base -c conda-forge mamba`
- set up snakemake

How to run dadasnake? – installation

- clone dadasnake and prepare run script

```
git clone https://github.com/a-h-b/dadasnake.git
cd dadasnake
cp auxiliary_files/dadasnake_tmux dadasnake
chmod 755 dadasnake
```

- adjust VARIABLE_CONFIG to your cluster

How to run dadasnake? – installation

- initialize dadasnake

```
./dadasnake -i config/config.init.yaml
sed -i "s/R CMD javareconf/#R CMD javareconf/” \
  conda/*/etc/conda/activate.d/activate-r-base.sh
```

- test dadasnake

```
./dadasnake -l -n "TESTRUN" -r config/config.test.yaml
```

- download (and prepare) databases

How to run dadasnake? – set up your files

- your reads:
  - all of your reads need to be in the same directory. Alternatively, you can set links to all of your reads into one directory. Reads can be gzipped or not (fastq.gz or fastq)
- config file:
  - you can copy one of the files in dadasnake/config and adjust the settings
- sample file*:
  - you can quickly generate a sample table like this:

```
paste <(ls *_R1_*fastq.gz | sed "s#_R.*##g") <(ls *_R1_*fastq.gz) \
 <(ls *_R1_*fastq.gz | sed "s#_R1_#_R2_#g")  >> samples.new.tsv
```

  - then, open in vi and introduce a header, containing:

library, r1_file, r2_file, (run) - separated by tabs

  - fix sample names, if you wish

*for multiple runs in the sample file, you can do this for the first run from the first run's directory:

```
paste <(ls *_R1_*fastq.gz | \
sed "s#_R.*##g") <(ls
*_R1_*fastq.gz) \
<(ls *_R1_*fastq.gz | \
sed "s#_R1_#_R2_#g") | \
sed 's#$#\trun1#' >>
../samples.2run.tsv
```

and then from the second run's directory:

```
paste <(ls *_R1_*fastq.gz | \
sed "s#_R.*##g") <(ls
*_R1_*fastq.gz) \
<(ls *_R1_*fastq.gz | \
sed "s#_R1_#_R2_#g") | \
sed 's#$#\trun2#' >>
../samples.2run.tsv
```

then fix header in vi

How to run dadasnake? – run

- connect to your server, navigate to your config file

```
/path/2/dadasnake/dadasnake -d /path/to/your/configuration/file
```

- check output
- then start dadasnake, e.g.:

```
/path/2/dadasnake/dadasnake -c -r \
-n ANYNAME /path/to/configuration/file
```

- wait, check status in output folder
- download results

How can I re-start the pipeline?

Job execution

A job is executed if and only if

- output file is target and does not exist
- output file needed by another executed job and does not exist
- input file newer than output file
- input file will be updated by other job

How can I re-start the pipeline?

- if the pipeline failed:
  - you can usually just repeat the start command, once the error is fixed

- if you want to re-do something: you have to delete all the file that you want to have redone. Then you can restart.

Current developments

- better options for re-starting
- keeping steps' results to test multiple options

- more example config files
- some bug-fixes

ideas, suggestions?

How to get help

- read the manuals ("RTFM")
  - https://github.com/a-h-b/dadasnake/
- think before you run
- ask other users
- issue issue

Github issue tracker

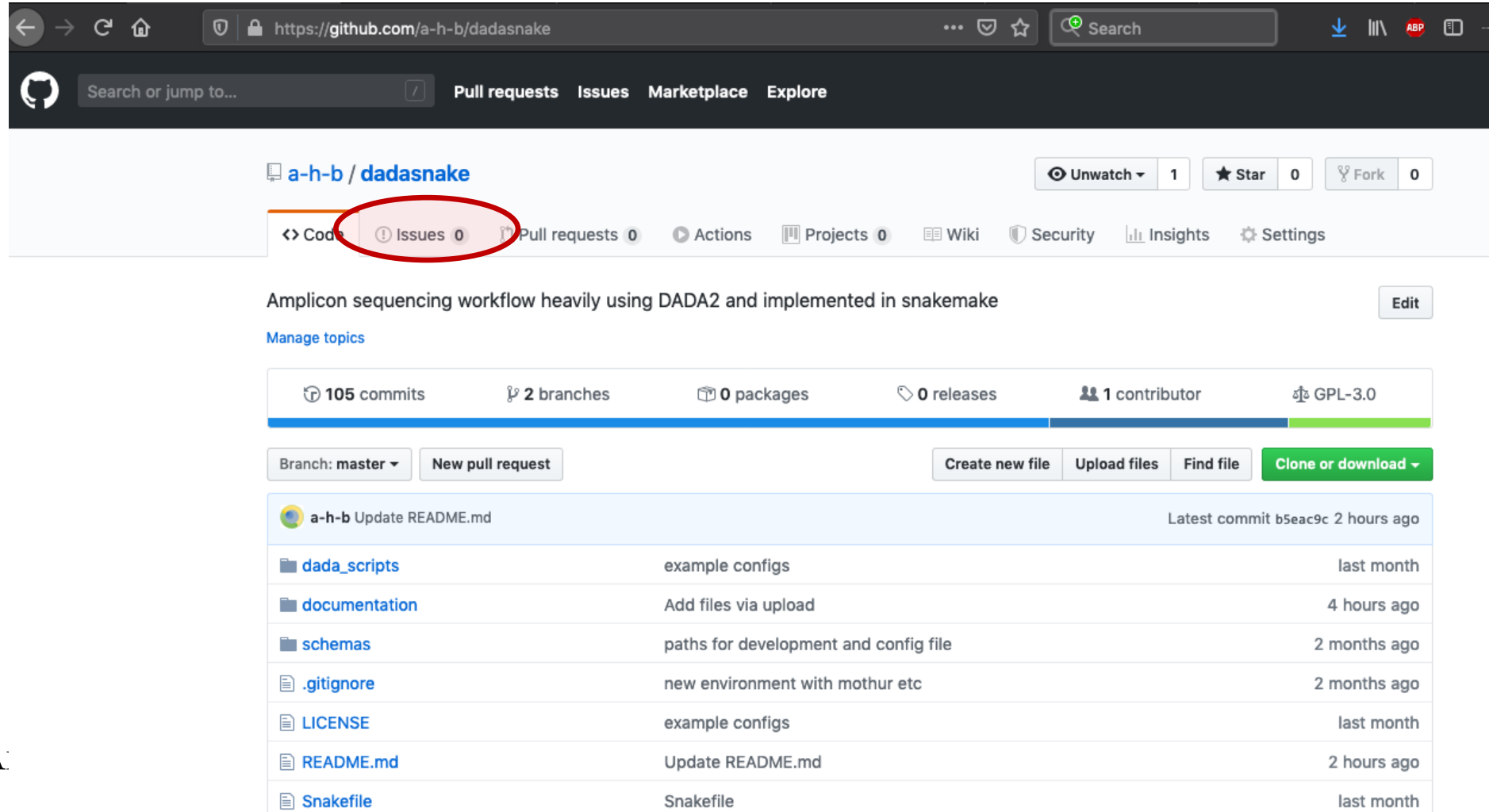
- public
- permanent
- searchable

- you can attach files (logs, screenshots)
- I can reply, you can reply

- fixes can be linked directly to versioning

Github issue tracker

https://github.com/a-h-b/dadasnake/issues/new



How to find out what's wrong

- typos, file formatting, wrong directory
  - missing permissions
  - software malfunctioning
    - `squeue`
    - `sacct`
    - check the logs for error messages

Common errors

- typos and formatting: config file, sample file, command
- you are not where you think you are: different directory, module not loaded, trying to write to directories without permission
- your environment isn't properly set up
- asking the impossible
- too strict filtering -> no data left
- time out
- errors in dadasnake's rules

How to find the error?

- start from outside to inside, from back to beginning

How to say that something went wrong

- meaningful summary
- what did you do?
- what did you expect to happen?
- what happened?

☹️ "it doesn't work"
"error when output of step X is empty"

☹️ "I ran the pipeline"
add your config file

☹️ "there's nothing there"
I am looking for the output of step Y

☹️ "I don't know what happened"
add the error messages and logs

Thanks to:

DADA2 Amplicon Sequencing. Exactly. Version 1.12

UFZ
Christina Weißbecker
Bea Schnabel
Julia Moll
Kezia Goldmann

iDiv
Christian Krause

CONDA

git

R

Bioconductor
OPEN SOURCE SOFTWARE FOR BIOINFORMATICS

DFG Deutsche Forschungsgemeinschaft

iDiv

a.u.s.heintzbuschart@uva.nl

SP C2.205

github.com/a-h-b

twitter.com/_a_h_b_

University of Amsterdam
Life Sciences

Thanks for your attention!

a.u.s.heintzbuschart@uva.nl

SP C2.205

github.com/a-h-b

twitter.com/_a_h_b_

The config file: raw files and sample list

- where you keep the raw files

➢ all raw files must be in the same directory

- you will usually have to set this
  - you can use absolute paths or paths relative to where you are when you start dadasnake

```
raw_directory: "testdata"
sample_table: "testdata/samples.small.tsv"
```

The samples files

- contains all the information on your samples
- must be tab-separated

- should not contain DOS-style end-of-line

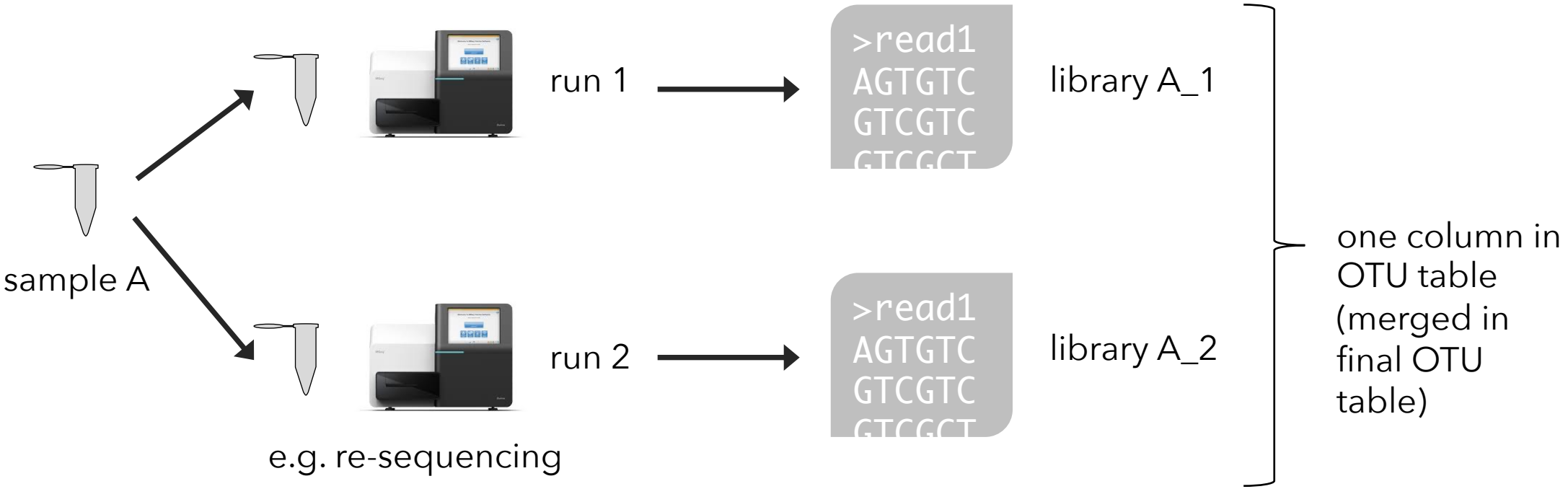
- you can change the encoding by opening the config file on Eve using vi, then type

```
:set ff=unix
:wq
```

- must contain named columns: library and r1_file
- can contain named columns: r2_file, sample, run

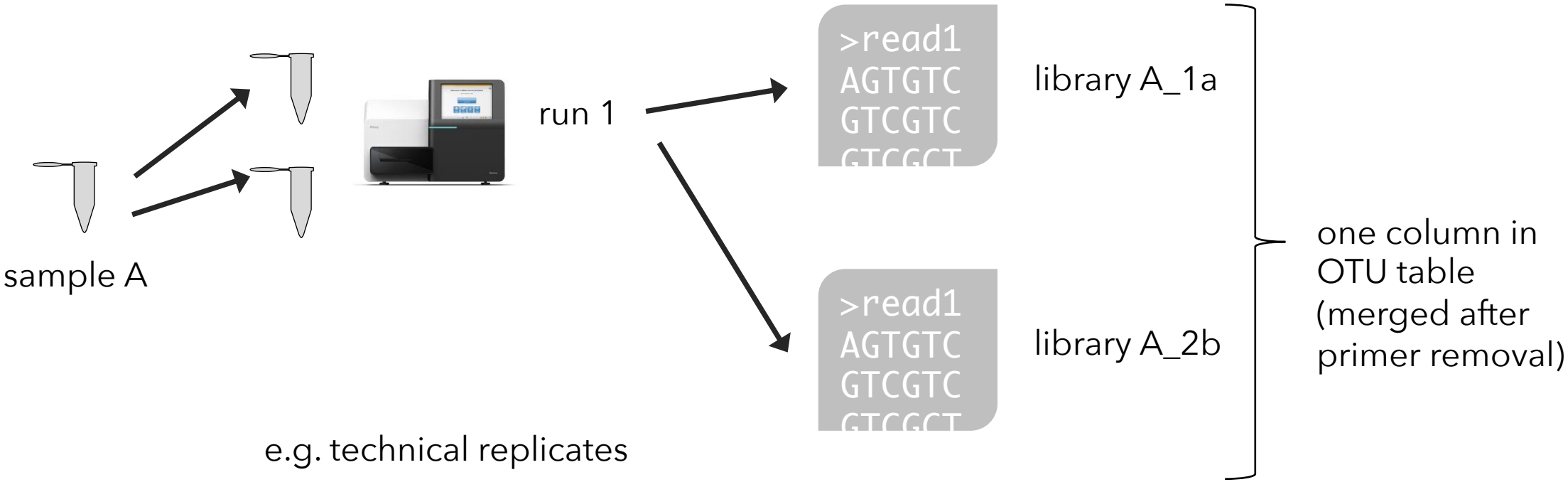
- libraries and samples should not have the same name, if there are libraries that have different names

The samples files



sample	library	run	r1_file	r2_file
A	A_1	1	myExp.A_R1.fastq.gz	myExp.A_R2.fastq.gz
A	A_2	2	myExp.A.reseq_R1.fastq.gz	myExp.A.reseq_R2.fastq.gz

The samples files



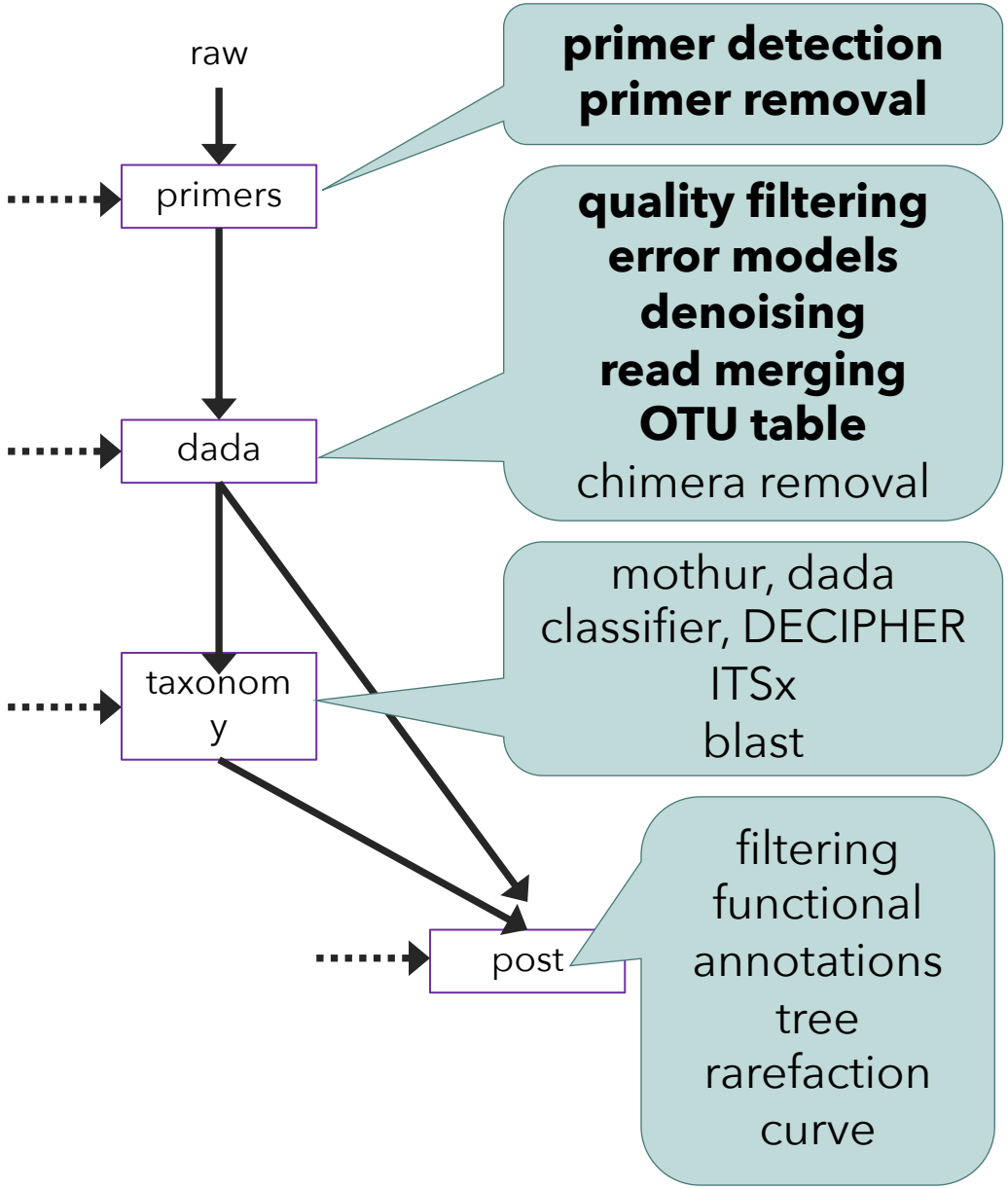
sample	library	run	r1_file	r2_file
A	A_1a	1	myExp.A1_R1.fastq.gz	myExp.A1_R2.fastq.gz
A	A_2b	1	myExp.A2_R1.fastq.gz	myExp.A2_R2.fastq.gz

The config file: step selection

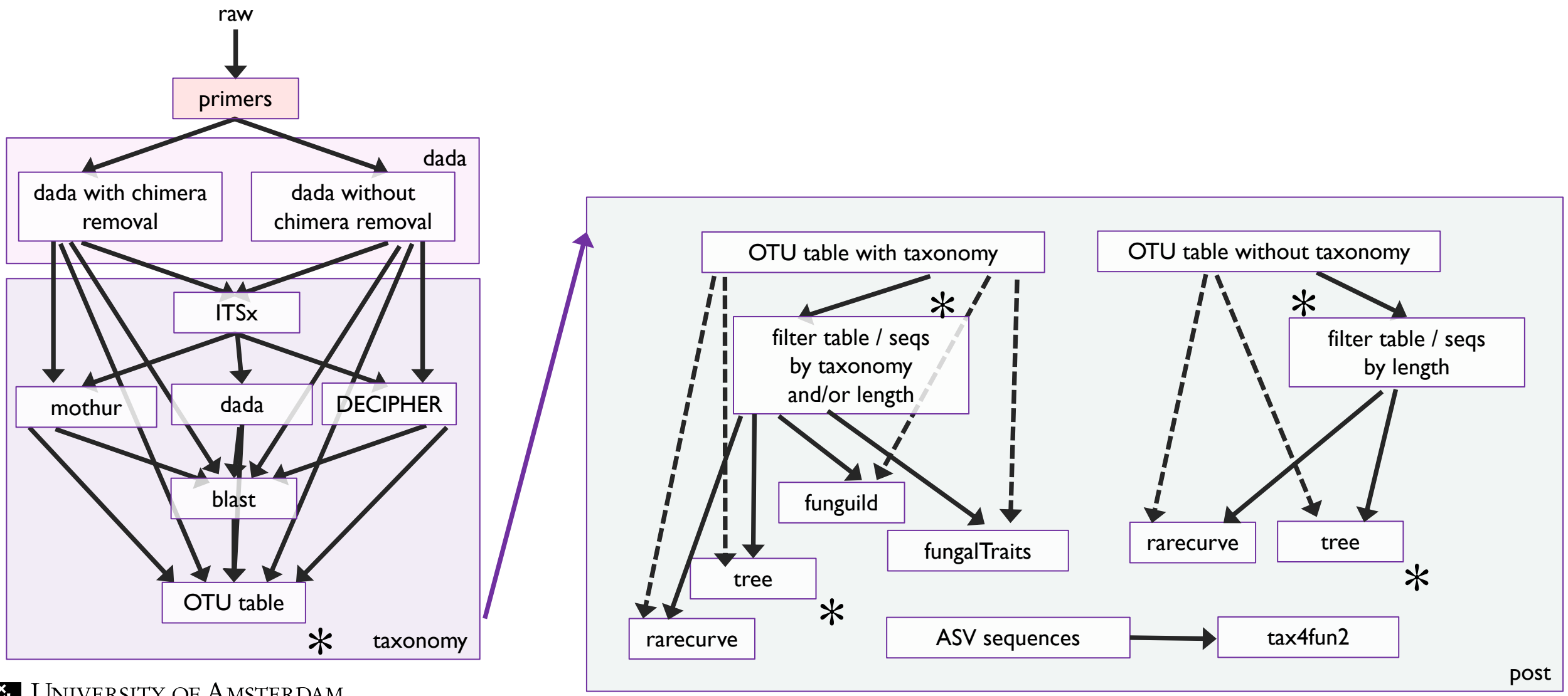
- by default, all steps are done
- ➢ `do_step: true`

```
do_primers: true
do_dada: true
do_taxonomy: true
do_postprocessing: true
```

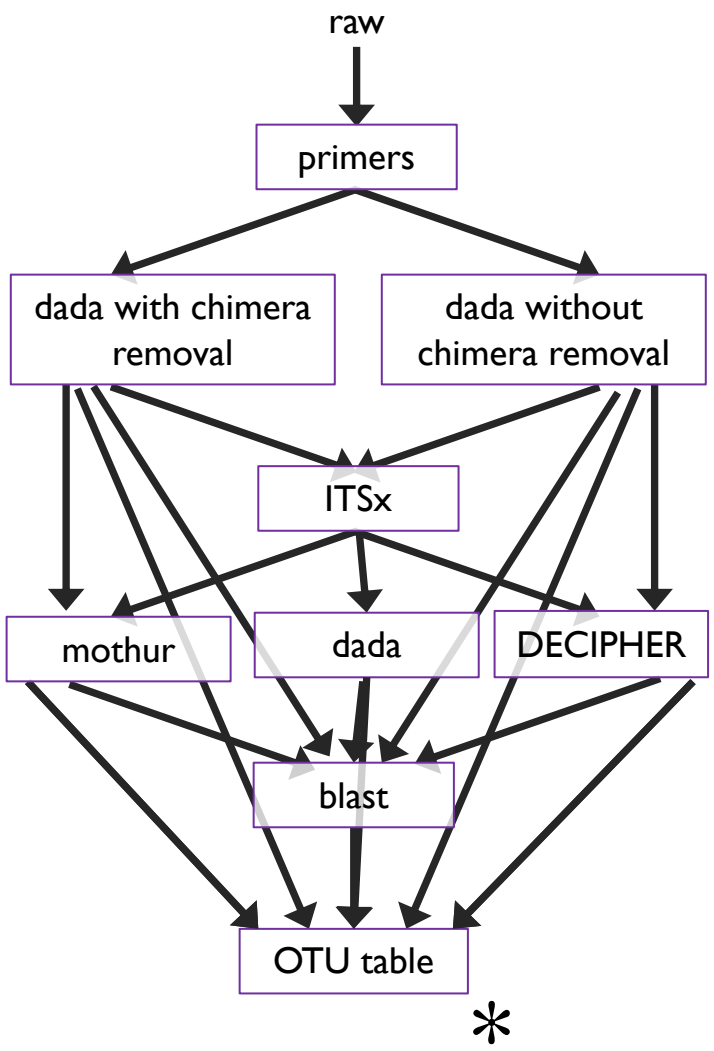
- set to false, if you don't want to run the whole workflow
- ➢ if you disable the first steps, you need to provide the input to the later steps



Workflows

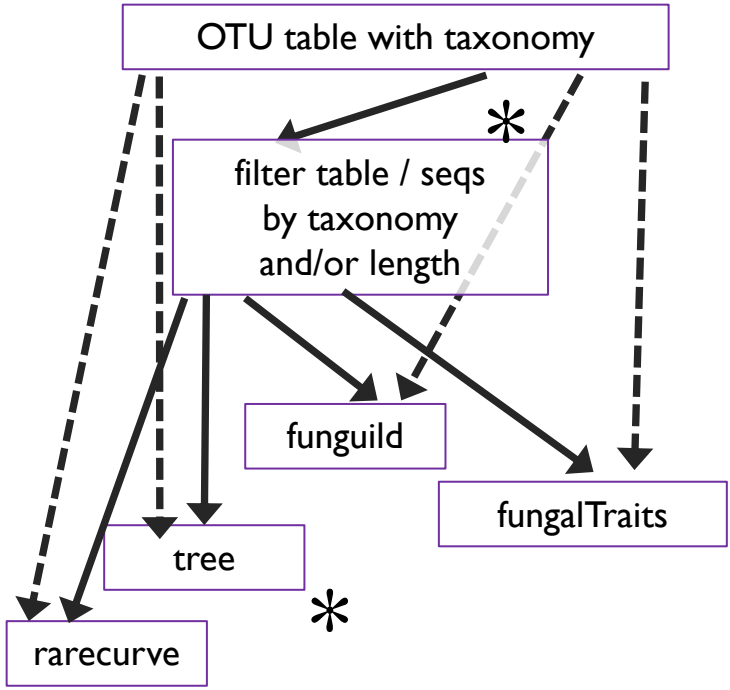

raw
primers
dada
dada with chimera removal
dada without chimera removal
ITSx
mothur
dada
DECIPHER
blast
OTU table
*
taxonomy
OTU table with taxonomy
OTU table without taxonomy
*
filter table / seqs by taxonomy and/or length
*
filter table / seqs by length
funguild
fungalTraits
rarecurve
tree
tree
*
rarecurve
*
ASV sequences
tax4fun2
post

Workflows



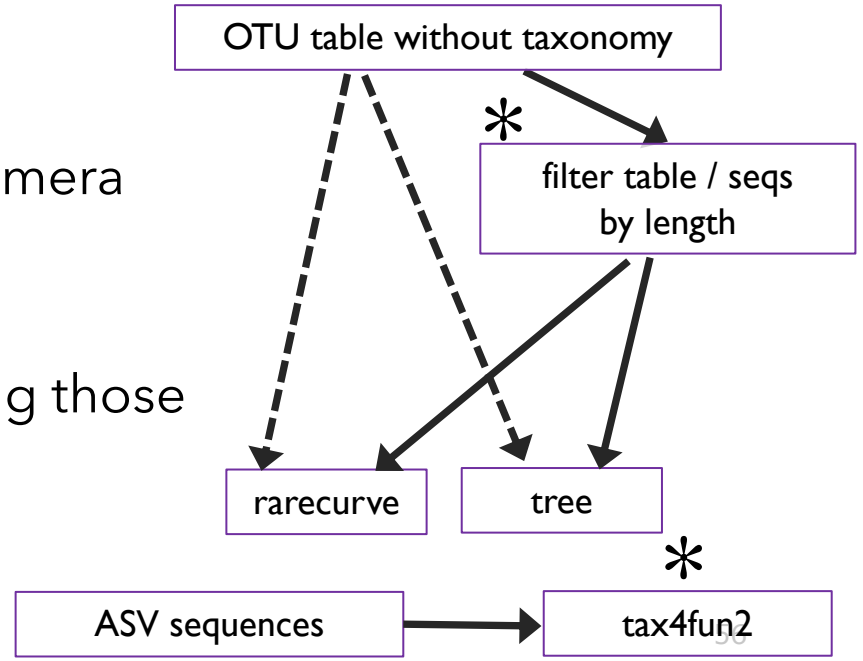
*hand-off to .biom:

- OTU table,
- if chimera removed, after chimera removal
- if taxonomy is done, including those results



*hand-off as phyloseq-object:

- OTU table,
- if chimera removed, after chimera removal
- if filtered, after filtering
- if taxonomy is done, including those results
- if tree, including tree



The config file: general info on sequencing run

- default is paired-end sequencing

- both primers have to be given
- can have degenerate positions

- name is just for your reference

```
primers:
  fwd:
    sequence: GTGYCAGCMGCCGCGGTAA
    name: 515F
  rvs:
    sequence: GGACTACNVGGGTWTCTAAT
    name: 806R
paired: true
```

The config file: settings for primer removal

- sequencing_direction can be unknown, fwd_1, rvs_1
  - fwd_1: read1 contains the fwd primer
  - rvs_1: read1 contains the rvs primer
  - unknown: you don't know or it's mixed
- primer_cutting are cut-adapt settings

- in this steps all primers are cut

```
paired: true
sequencing_direction: "unknown"
primer_cutting:
  overlap: 10
  count: 2
  filter_if_not_match: any
  perc_mismatch: 0.2
  indels: "--no-indels"
```

The config file: settings for filtering

- two general principles:
  - ➢ fixed length + local or total quality threshold
  - ➢ or: flexible length by quality cut-off + length threshold

```
filtering:
  trunc_length:
    fwd: 0
    rvs: 0
  trunc_qual:
    fwd: 2
    rvs: 2
  max_EE:
    fwd: Inf
    rvs: Inf
  minLen:
    fwd: 20
    rvs: 20
  maxLen:
    fwd: Inf
    rvs: Inf
  minQ:
    fwd: 0
    rvs: 0
  maxN: 0
  rm_phix: true
  trim_left:
    fwd: 0
    rvs: 0
```

The config file: settings for downsampling

- set do to true to downsample reads after quality filtering and before DADA2 clustering:

➢ you can set the number of reads to keep

➢ samples with less will be treated as empty

➢ set a seed to keep consistent in re-runs

```
downsampling:
  do: false
  number: 50000
  min: true
  seed: 123
```

The config file: settings for dada

- error_seed for reproducibility
- dada settings only need to be changed for non-Illumina

- pair merging: flexible overlap and mismatches

➢ just concatenate only works with fixed length filtering

- chimera removal is done after OTU table

➢ based on whole table

➢ or: per sample

```
error_seed: 100
dada:
  band_size: 16
  homopolymer_gap_penalty: NULL
pair_merging:
  min_overlap: 12
  max_mismatch: 0
  just_concatenate: false
  trim_overhang: true
chimeras:
  remove: true
  method: consensus
```

The config file: settings for taxonomy

- DECIPHER and/or dada's and/or mothur's classifier can be used (switched on by `do`)
- can be done before or after ITSx

- **you have to choose the databases**

```yaml
taxonomy:
  dada:
    do: false
    post_ITSx: false
    db_path: "/zfs/omics/projects/metatools/DB/amplicon/dada2_format"
    refFasta: "silva_nr99_v138_train_set.fa.gz"
    ref_dbs_full: ""
    db_short_names: "silva_v138_nr99"
    minBoot: 50
    tryRC: false
    look_for_species: false
    seed: 101
    spec_db: "/zfs/omics/projects/metatools/DB/amplicon/dada2_format/silva_species_assignment_v138.fa.gz"
  decipher:
    do: false
    post_ITSx: false
    db_path: "/zfs/omics/projects/metatools/DB/amplicon/decipher"
    tax_db: "SILVA_SSU_r138_2019.RData"
    ref_dbs_full: ""
    db_short_names: "SILVA_SSU_r138"
    threshold: 60
    strand: top
    bootstraps: 100
    seed: 100
    look_for_species: false
    spec_db: "/zfs/omics/projects/metatools/DB/amplicon/dada2_format/silva_species_assignment_v138.fa.gz"
  mothur:
    do: true
    post_ITSx: false
    db_path: "/zfs/omics/projects/metatools/DB/amplicon/mothur_format"
```

The config file: settings for taxonomy

- DECIPHER and/or dada's and/or mothur's classifier can be used (switched on by `do`)
- can be done before or after ITSx

- **you have to choose the databases**

```
taxonomy:
  mothur:
    do: true
    db_path: "/zfs/omics/projects/metatools/DB/amplicon/mothur_format"
    ref_dbs_full: "/zfs/omics/projects/metatools/DB/amplicon/mothur_format/SILVA_138_SSURef_NR99_spec_prok.515F.785R
/zfs/omics/projects/metatools/DB/amplicon/mothur_format/GTDB_202.515.806"
    db_short_names: "SILVA_138 GTDB_r202"
```

The config file: settings for taxonomy

- **you have to choose the databases**
- **you can use several databases by including their path and name in the ref_dbs_full, separated by a space**
- **you have to enter the same number of names in the db_short_names**

```
taxonomy:
  mothur:
    do: true
    db_path: "/zfs/omics/projects/metatools/DB/amplicon/mothur_format"
    ref_dbs_full: "/zfs/omics/projects/metatools/DB/amplicon/mothur_format/SILVA_138_SSURef_NR99_spec_prok.515F.785R
/zfs/omics/projects/metatools/DB/amplicon/mothur_format/GTDB_202.515.806"
    db_short_names: "SILVA_138 GTDB_r202"
```

The config file: settings for taxonomy

- blast is run on the sequences that have no taxonomic annotation

➢ can be all sequences, if no classifier is run

- you need to provide the path to the database and the name

- this version of blast usually expects to have taxonomic information

- all: set to true to blast all ASVs, set to false to blast only ASVs without annotation

```
blast:
  do: false
  db_path: "/zfs/omics/projects/metatools/DB/amplicon/blast_format/ncbi_16S_ribosomal_RNA"
  tax_db: 16S_ribosomal_RNA
  e_val: 0.01
  tax2id: ""
  all: true
  max_targets: 10
```

The config file: settings for taxonomy

- blast is run on the sequences that have no taxonomic annotation

- blast results are simplified by BASTA
- there is no one-size-fits-all to the settings
- also check the detailed BASTA output

```
blast:
  do: false
  db_path: "/zfs/omics/projects/metatools/DB/amplicon/blast_format/ncbi_16S_ribosomal_RNA"
  tax_db: 16S_ribosomal_RNA
  e_val: 0.01
  tax2id: ""
  all: true
  max_targets: 10
  run_basta: false
  basta_path: "/zfs/omics/projects/metatools/TOOLS/BASTA/bin/basta"
  basta_db: "/zfs/omics/projects/metatools/DB/amplicon/blast_format/ncbi_taxonomy"
  basta_e_val: 0.00001
  basta_alen: 100
  basta_number: 0
  basta_min: 3
  basta_id: 80
  basta_besthit: true
  basta_perchits: 99
```

The config file: ITSx settings for taxonomy

- ITSx settings, including number of regions, which regions and e-value

- you can now choose which taxa to search against (`query_taxa`)
- you can choose which sequences to return (`target_taxon`)

```
ITSx:
  do: false
  min_regions: 1
  region: ITS2
  e_val: 1e-5
  query_taxa: .
  target_taxon: F
```

- in this example, all taxa are searched and only sequences with a best hit in fungi are returned

The config file: settings for post-processing

- filtering by taxonomy and/or length is done first
  - ➢ by taxonomy only works, if a classifier was run (not on blast result)
  - ➢ if several classifiers were run, taxonomy filter keeps ASV, if any of the classifiers identified it

- `keep_target_taxa` should accept regular expressions (R)

```
final_table_filtering:
  do: true
  keep_target_taxa: "."
  target_min_length: 0
  target_max_length: Inf
```

The config file: settings for post-processing

- the other steps are done after filtering (if filtering is enabled)

- rarefaction curve: plots a set of rarefaction curves
- treeing: calculates a multiple alignment and a phylogenetic tree

➢ only advised for size-consistent markers (e.g. 16S)

➢ not advised for large datasets (more than 10,000 ASVs)

➢ uses clustal omega and fasttreeMP

```
postprocessing:
  rarefaction_curve: true
  funguild:
    do: false
    funguild_db: "/zfs/omics/projects/metatools/DB/amplicon/Functions/funguild_db.json"
    classifier: mothur
  fungalTraits:
    do: false
    db: "/zfs/omics/projects/metatools/DB/amplicon/Functions/FungalTraits_1.2_ver_16Dec_2020_V.1.2.tsv"
    classifier: mothur.SILVA_138_SSURef_NR99_cut
  tax4fun2:
    do: false
    db: "/zfs/omics/projects/metatools/DB/amplicon/Functions/Tax4Fun2_ReferenceData_v2"
    database_mode: "Ref99NR"
    normalize_by_copy_number: true
    min_identity_to_reference: 0.97
    user_data: false
    user_dir: "/zfs/omics/projects/metatools/DB/amplicon/Functions/GTDB_202_tax4fun2"
    user_db: "GTDB_fun"
  treeing:
    do: true
    fasttreeMP: "export OMP_NUM_THREADS={threads}\n/zfs/omics/projects/metatools/TOOLS/FastTreeMP"
```

The config file: settings for post-processing

- the other steps are done after filtering (if filtering is enabled)

- add functional annotations for the taxa:

➢ don't expect this to work if you don't have taxonomic information

➢ you can choose funguild or FungalTraits for fungi

➢ and Tax4Fun2 for bacteria

```yaml
postprocessing:
  rarefaction_curve: true
  funguild:
    do: false
    funguild_db: "/zfs/omics/projects/metatools/DB/amplicon/Functions/funguild_db.json"
    classifier: mothur
  fungalTraits:
    do: false
    db: "/zfs/omics/projects/metatools/DB/amplicon/Functions/FungalTraits_1.2_ver_16Dec_2020_V.1.2.tsv"
    classifier: mothur.SILVA_138_SSURef_NR99_cut
  tax4fun2:
    do: false
    db: "/zfs/omics/projects/metatools/DB/amplicon/Functions/Tax4Fun2_ReferenceData_v2"
    database_mode: "Ref99NR"
    normalize_by_copy_number: true
    min_identity_to_reference: 0.97
    user_data: false
    user_dir: "/zfs/omics/projects/metatools/DB/amplicon/Functions/GTDB_202_tax4fun2"
    user_db: "GTDB_fun"
  treeing:
    do: true
    fasttreeMP: "export OMP_NUM_THREADS={threads}\n/zfs/omics/projects/metatools/TOOLS/FastTreeMP"
```

The config file: settings for post-processing

- the other steps are done after filtering (if filtering is enabled)

➢ tax4fun2 also accepts user databases
➢ I have built a KO database for all reference genomes in the GTDB (~40,000 genomes)

```
tax4fun2:
  do: false
  db: "/zfs/omics/projects/metatools/DB/amplicon/Functions/Tax4Fun2_ReferenceData_v2"
  database_mode: "Ref99NR"
  normalize_by_copy_number: true
  min_identity_to_reference: 0.97
  user_data: false
  user_dir: "/zfs/omics/projects/metatools/DB/amplicon/Functions/GTDB_202_tax4fun2"
  user_db: "GTDB_fun"
```