南方科技大学
SOUTHERN UNIVERSITY OF SCIENCE AND TECHNOLOGY

海洋科学与工程系
Department of Ocean Science and Engineering

Practical Session 01

Shengwei Hou

Yuanhao Liu, Xunying Zhou, Qian Wen, Yuqian Mo

July-3-2025

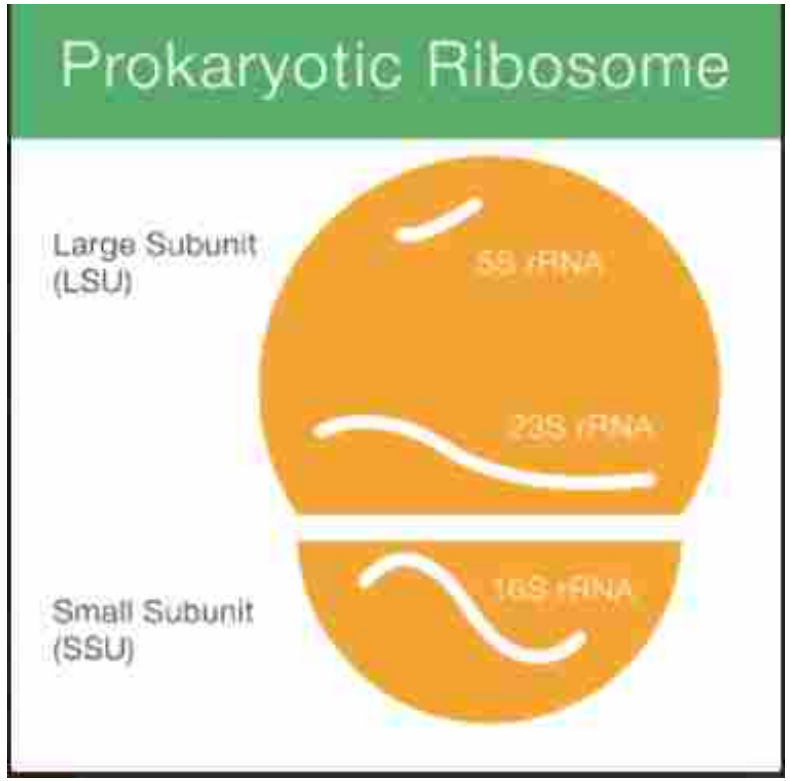
Part 1

Microbial Analysis based on 16S rRNA amplicon

Yuanhao Liu

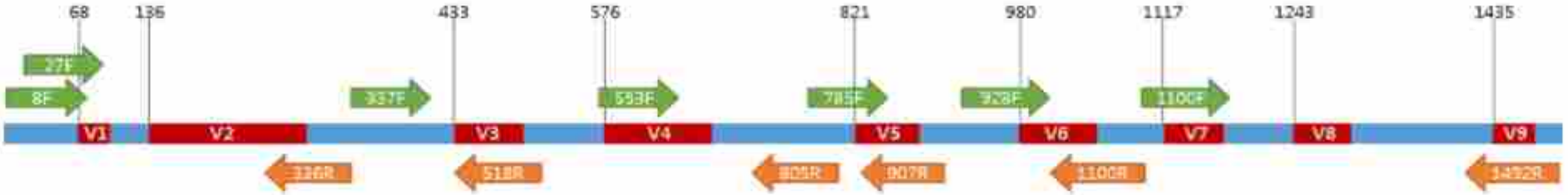


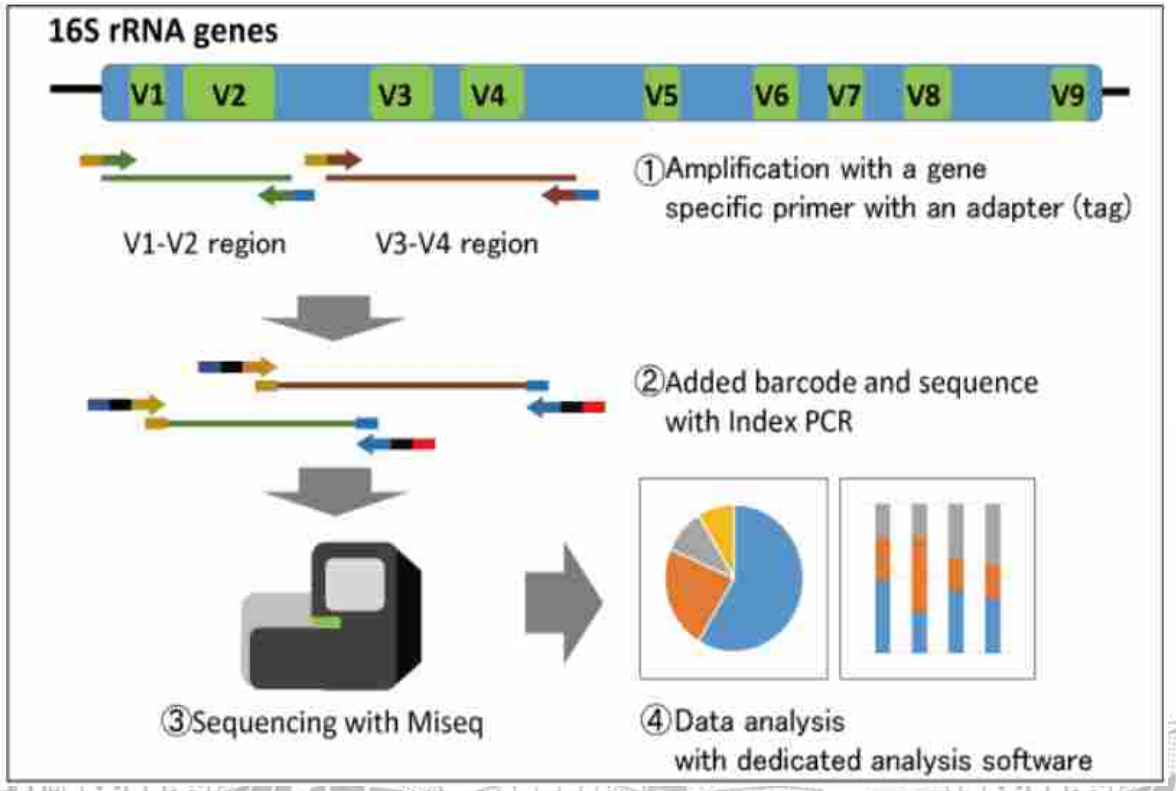
What is 16S rRNA?



- **16S rRNA** stands for 16S ribosomal ribonucleic acid (rRNA), where S (Svedberg) is a unit of measurement (sedimentation rate);
- This rRNA is an important constituent of the **small subunit (SSU)** of prokaryotic ribosomes as well as mitochondria and chloroplasts;
- DNA segment coding for rRNA is called either rRNA gene or **rDNA**;
- For the purposes of sequencing, sequence information is obtained from the rDNA because DNA is much easier to process and sequence than RNA.

16S rDNA sequencing





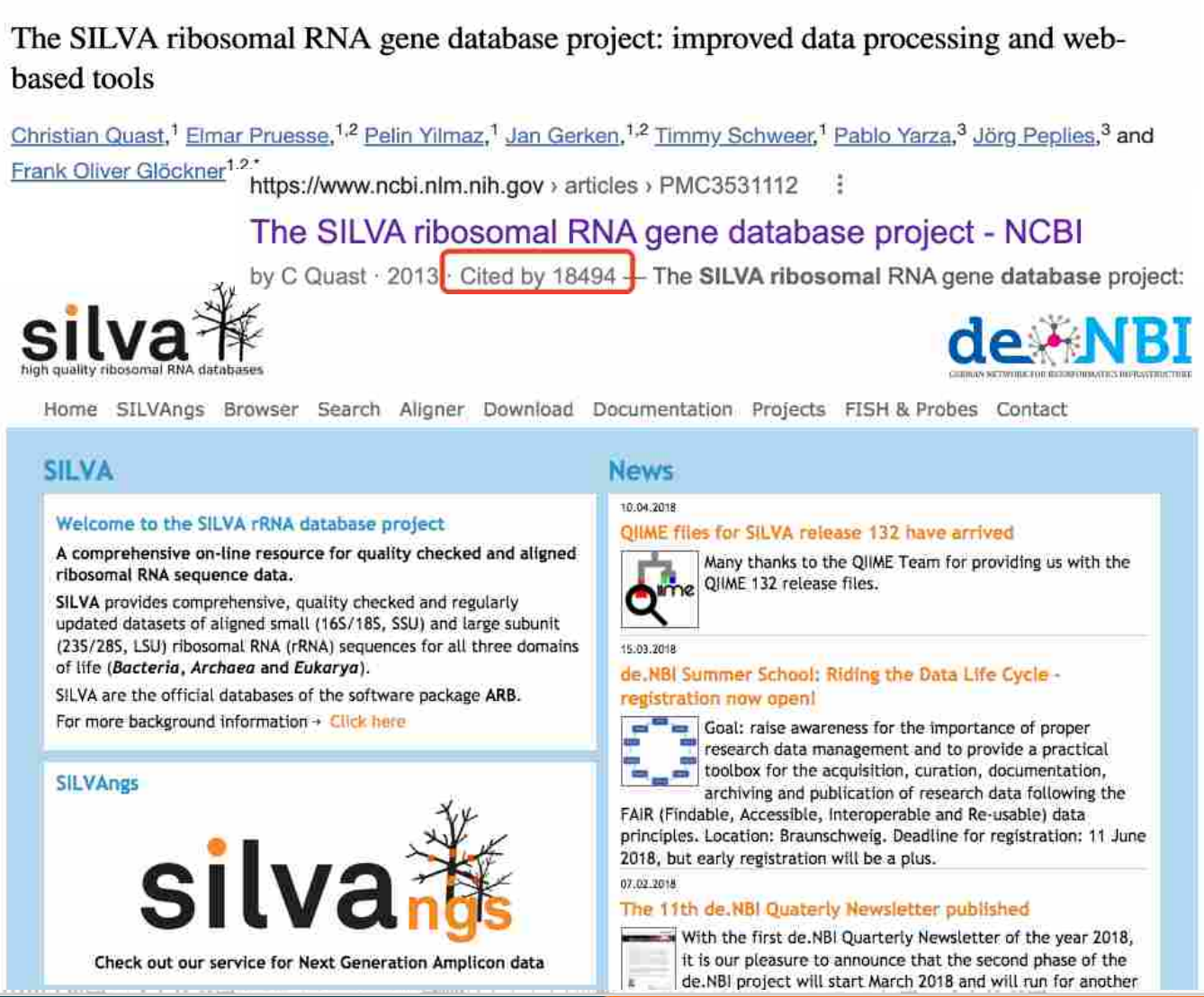
- 16S rDNA is about 1,500 bp long, the full-length 16S gene is usually amplified by the pair of primers 27F and 1492R, followed by Sanger DNA sequencing;
- Sequence variation among bacterial 16S rDNA is known to be not uniformly distributed, nine hypervariable regions were identified among Bacteria, which are named V1 to V9;
- 16S gene has multiple conserved regions that can be used as the priming sites. This becomes a significant advantage for NGS-based short-read sequencing;
- We have 16S sequence database holding almost all known species of Bacteria and Archaea;
- By searching 16S sequence against these databases, anyone, even without knowledge of serious taxonomy, can identify newly isolated strains.

16S ribosomal databases

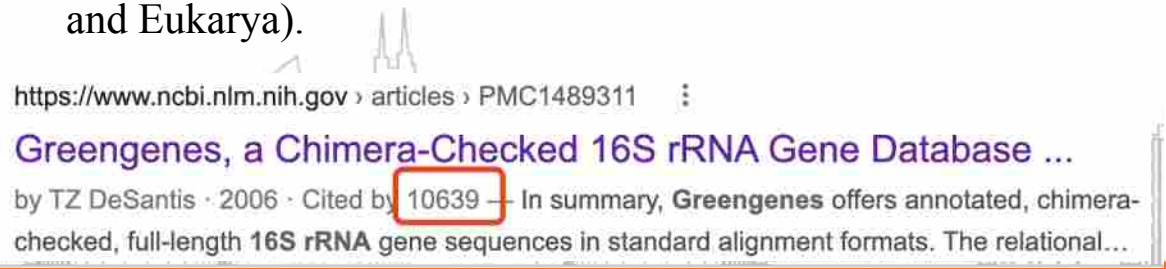
Nucleic Acids Res. 2013 Jan; 41(Database issue): D590–D596.
Published online 2012 Nov 27. doi: 10.1093/nar/gks1219
PMCID: PMC3531112
PMID: 23193283

The SILVA ribosomal RNA gene database project: improved data processing and web-based tools

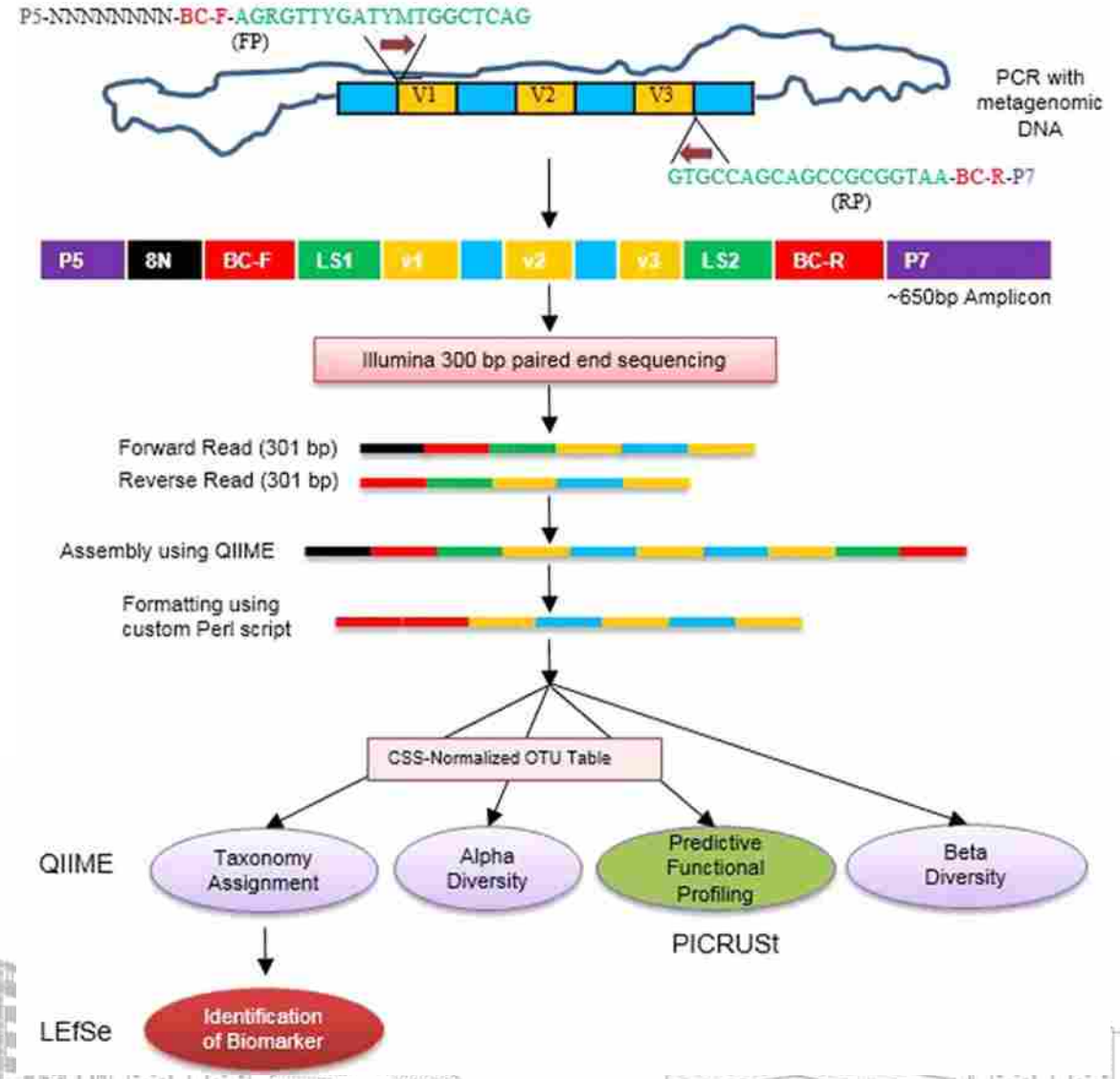
Christian Quast,[1] Elmar Pruesse,[1,2] Pelin Yilmaz,[1] Jan Gerken,[1,2] Timmy Schweer,[1] Pablo Yarza,[3] Jörg Peplies,[3] and Frank Oliver Glöckner[1,2,*]



- The available reference databases for annotating 16S rRNA gene include the Greengenes database, SILVA, etc.
- The **Greengenes 16S rRNA database** (http://greengenes.lbl.gov) addresses limitations of public repositories by providing chimera screening, standard alignment, and taxonomic classification using multiple published taxonomies;
- SILVA provides comprehensive, quality checked and regularly updated datasets of aligned small (16S/18S, SSU) and large subunit (23S/28S, LSU) ribosomal RNA (rRNA) sequences for all three domains of life as well as a suite of search, primer-design and alignment tools (Bacteria, Archaea and Eukarya).

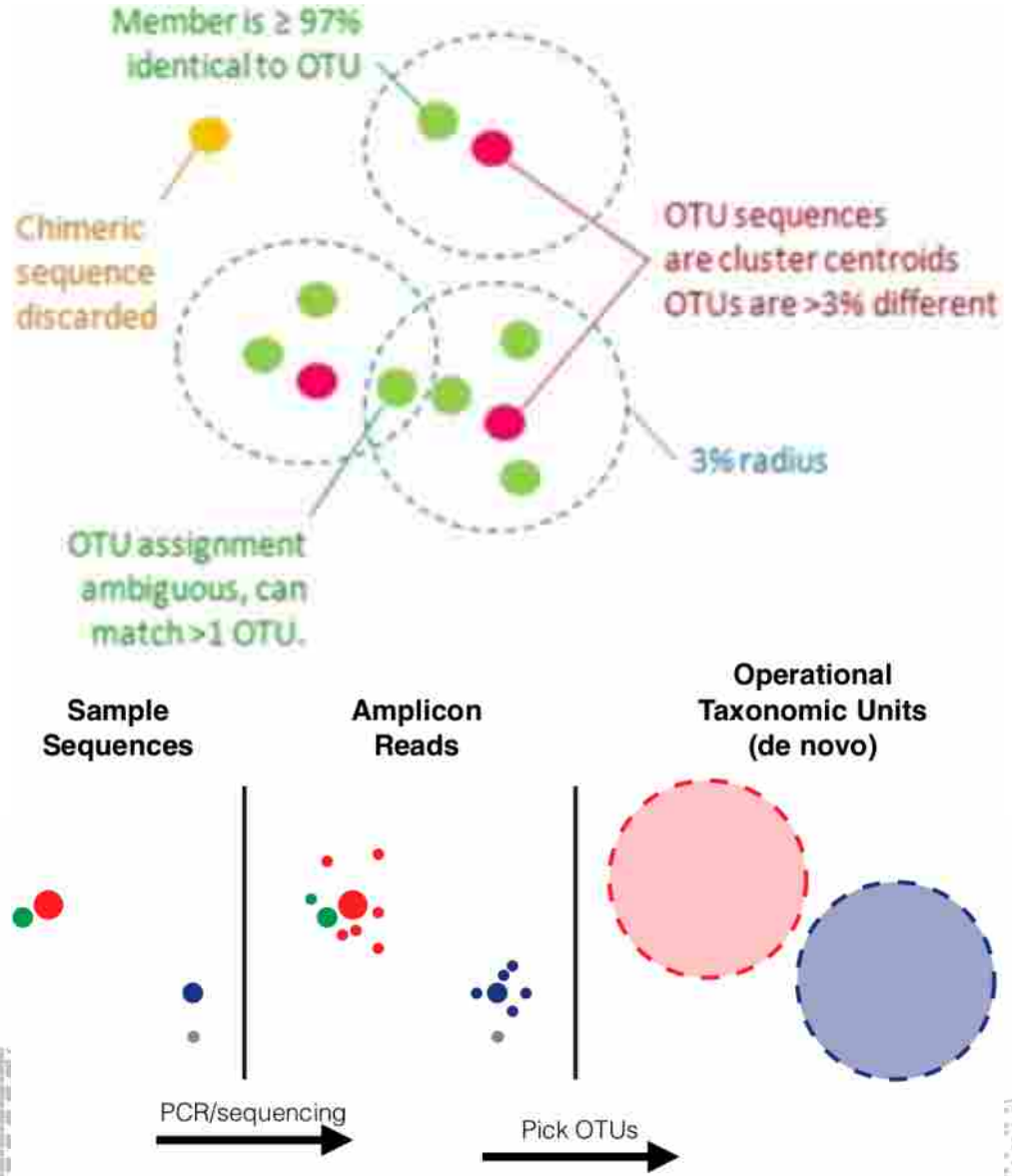


16S rDNA Amplicon Sequencing



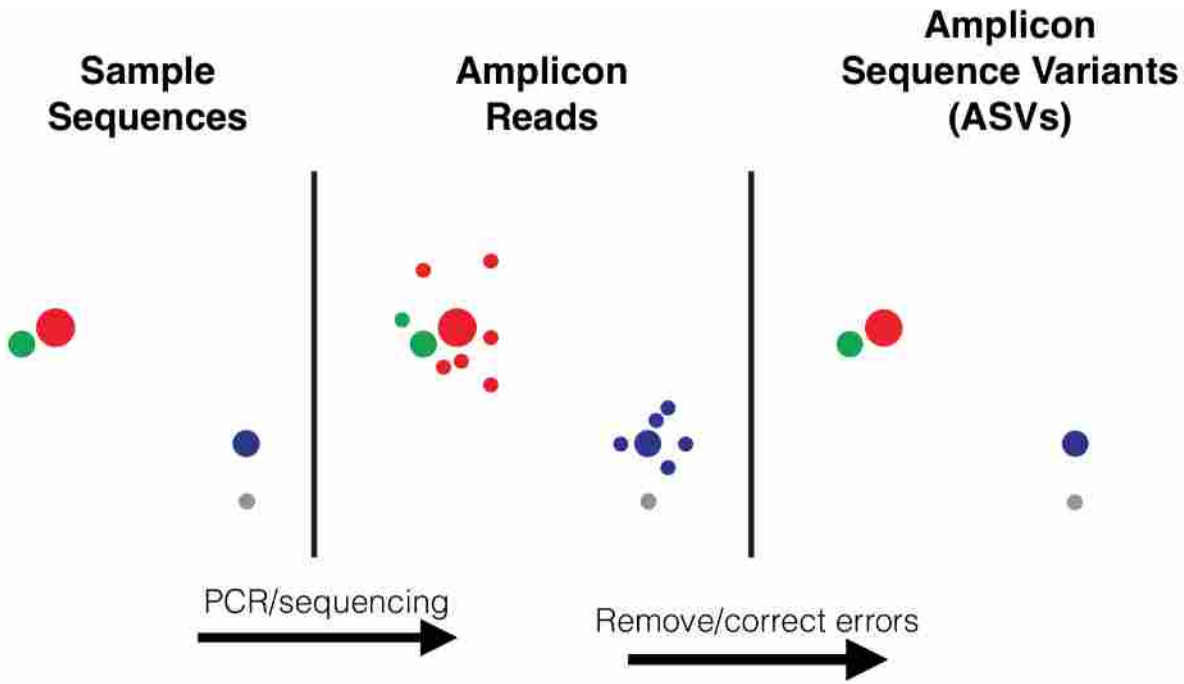
- 16S rRNA gene sequencing is commonly used for the identification, classification, and quantitation of microbes within complex biological mixtures such as environmental samples (e.g., marine water) and gut samples (e.g., human gut microbiome).
- 16S Amplicon Sequencing is an ultra-deep DNA sequencing method that focuses on sequencing specific target regions (amplicons).
- Short (<500 bp) hypervariable regions of conserved genes or intergenic regions are amplified by PCR and analyzed by next-generation sequencing (NGS) technology, to identify and differentiate multiple microbial species from complicated samples.
- Amplicon metagenomic sequencing is designated to sequence the target genes of 16S ribosomal RNA (rRNA) by universal primers, to describe and compare the phylogeny and taxonomy of prokaryotes.

Operational taxonomic unit (OTU)



- The determination of microbial diversity using sequencing data from 16S rRNA genes amplified from environmental DNA most often involves the clustering into **operational taxonomic units (OTUs)**, in which similar 16S rRNA sequences are considered to be from the same taxon (Edgar, 2013).
- One is to cluster these sequences into phylotypes based on their similarity to the reference database (**Closed-ref**), and the other is to cluster sequences into operational taxonomic units (OTUs) using a similarity threshold, only according to their similarity (**De novo**).
- An OTU is a pragmatic definition to group individuals by similarity, equivalent to but not necessarily in line with classical Linnaean taxonomy or modern evolutionary taxonomy;
- Sequences can be clustered according to their similarity to one another, and operational taxonomic units are defined based on the similarity threshold (usually 97% similarity; however 100% similarity is also common, known as single variants) set by the researcher;

Amplicon sequence variant (ASV)



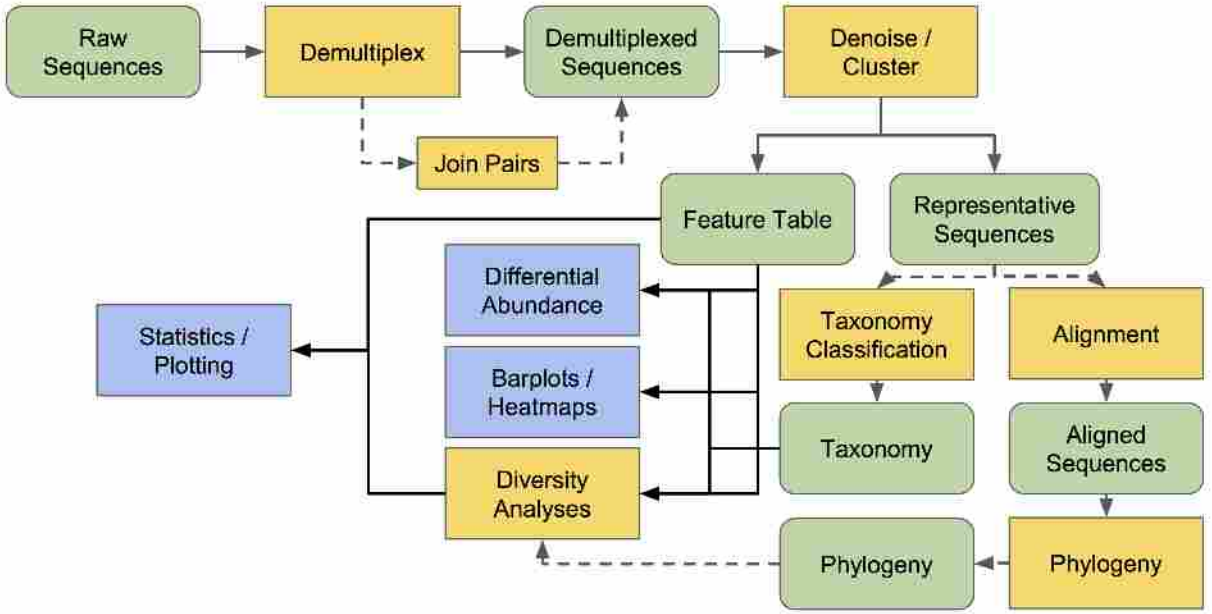
	ASVs	De novo	Closed-ref
Precise	✓	~	~
Tractable	✓	~	✓
Reproducible	✓	X	✓
Comprehensive	✓	✓	X

OTU-based

- An **amplicon sequence variant (ASV)** is any one of the inferred single DNA sequences recovered from a high-throughput analysis of marker genes;
- Because these analyses, also called "amplicon reads," are created following the removal of erroneous sequences generated during PCR and sequencing, using ASVs makes it possible to distinguish sequence variation by a single nucleotide change;
- The uses of ASVs include classifying groups of species based on DNA sequences, finding biological and environmental variation, and determining ecological patterns;
- OTUs and ASVs provided similar results, with ASVs enabling a slightly stronger detection of prokaryotic diversity.
- Species diversification can be measured more accurately with ASVs, the use of OTUs in well-constructed studies is generally valid to demonstrate diversification at broad scales.

QIIME2 Workflow

QIIME 2 is a powerful, extensible, and decentralized microbiome analysis package with a focus on data and analysis transparency.



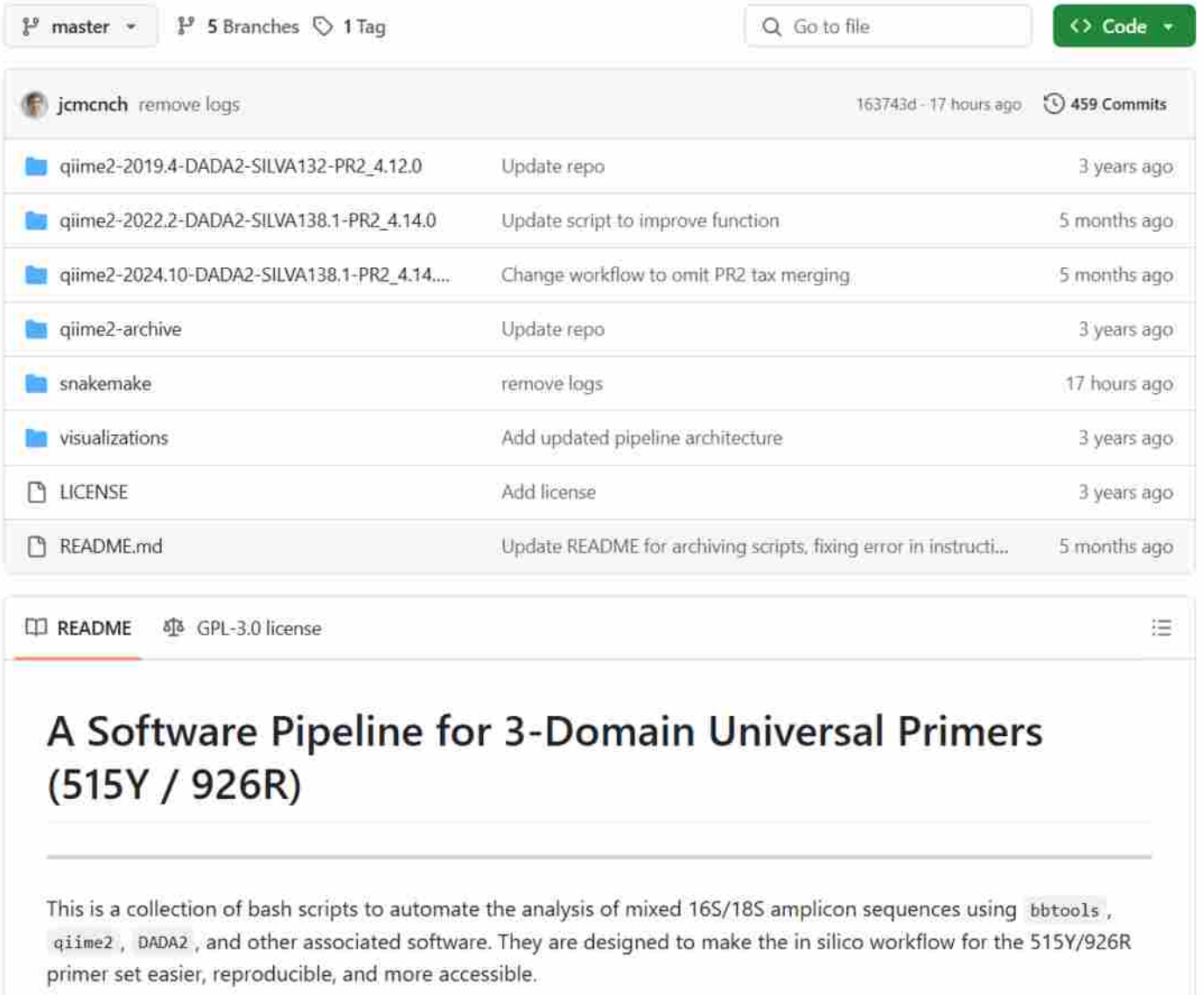
A conceptual overview of the various possible workflows for examining amplicon sequence data

- All amplicon sequencing experiments begin as raw sequence data;
- We demultiplex these reads to determine which sample each read came from;
- Reads should then be denoised into amplicon sequence variants (ASVs) or clustered into operational taxonomic units (OTUs);
- The resulting feature table and representative sequences are key pieces of data. A **feature table** is essentially a matrix of samples * observations, i.e., the number of times each "feature" (OTUs, ASVs, etc) is observed in each sample in a data set.
- Standard analyses with feature tables include:
  - ✓ Taxonomic classification of sequences (a.k.a., "what species are present?");
  - ✓ Alpha and beta diversity analyses, or measures of diversity within and between samples, respectively (a.k.a., "how similar are my samples?");
  - ✓ Many diversity analyses rely on the phylogenetic similarity between individual features;
  - ✓ Differential abundance measurements determine which features (OTUs, ASVs, taxa, etc) are significantly more/less abundant in different experimental groups.

QIIME2 Workflow

海洋科学与工程系
Department of Ocean Science and Engineering

We will use a tutorial from: https://github.com/jcmcnch/eASV-pipeline-for-515Y-926R



master · 5 Branches · 1 Tag · Go to file · Code

jcmcnch remove logs — 163743d · 17 hours ago · 459 Commits

qiime2-2019.4-DADA2-SILVA132-PR2_4.12.0	Update repo	3 years ago
qiime2-2022.2-DADA2-SILVA138.1-PR2_4.14.0	Update script to improve function	5 months ago
qiime2-2024.10-DADA2-SILVA138.1-PR2_4.14....	Change workflow to omit PR2 tax merging	5 months ago
qiime2-archive	Update repo	3 years ago
snakemake	remove logs	17 hours ago
visualizations	Add updated pipeline architecture	3 years ago
LICENSE	Add license	3 years ago
README.md	Update README for archiving scripts, fixing error in instructi...	5 months ago

README · GPL-3.0 license

A Software Pipeline for 3-Domain Universal Primers (515Y / 926R)

This is a collection of bash scripts to automate the analysis of mixed 16S/18S amplicon sequences using `bbtools`, `qiime2`, `DADA2`, and other associated software. They are designed to make the in silico workflow for the 515Y/926R primer set easier, reproducible, and more accessible.

About

This is a collection of scripts for analyzing mixed 16S/18S amplicon sequences using tools such as qiime2, DADA2, deblur, and bbtools

Readme
GPL-3.0 license
Activity
32 stars
2 watching
9 forks
Report repository

Releases 1

qiime2-2019.4 archive release — Latest
on Nov 21, 2022

Packages

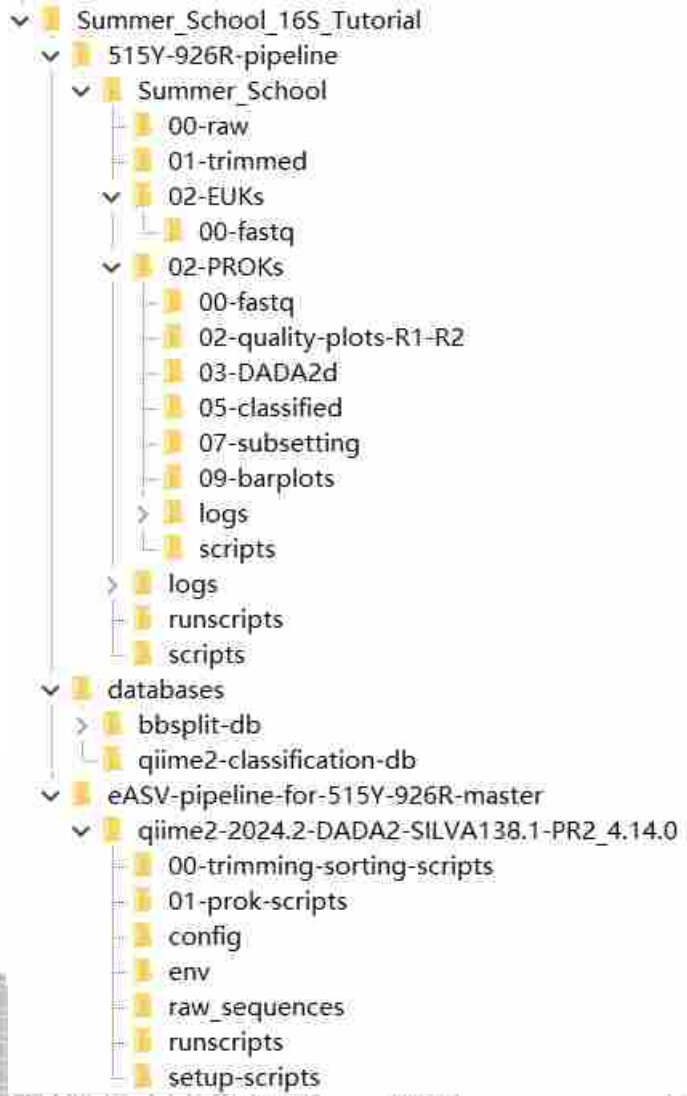
No packages published

Contributors 3

jcmcnch Jesse McNichol
MicrobialMelody Melody Aleman
fletchec99

QIIME2 Workflow

Modified tutorial has been upload to: https://pan.quark.cn/s/132729f16038



515Y-926R-pipeline: Store input .fastq files and all output files

Summer_School: A name of a working directory

databases: Store classification databases

bbsplit-db is used to seperate 16S and 18S sequences

qiime2-classification-db stores two classifiers based on SILVA and PR2 databases

eASV-pipeline-for-515Y-926R-master: Store scripts we will used in 16S amplicon analysis

00-trimming-sorting-scripts, 01-prok-scripts, runscripts: Scripts for data processing

config: Store a configuration file that stores initial parameters

env: .yaml files used to install necessary environments

raw_sequences: Example of input file

setup-scripts: Scirpts used to construct environments

Miniconda3: Manage Bioinformatics Software Easily

What is Miniconda?

Miniconda is a lightweight version of Anaconda, used to:

- Install Bioinformatics Tools (e.g., QIIME 2, FastQC).
- Avoid Dependency Conflicts – Isolate environments for different tools.
- Reproduce Analyses – Export environments (conda env export > env.yml).

To start our tutorial, go to this folder:

```
cd ~/eASV-pipeline-for-515Y-926R-master/qiime2-2024.2-DADA2-SILVA138.1-PR2_4.14.0
```

How to Install Miniconda?

```
bash ./setup-scripts/00-install-conda.sh
source ~/.bashrc
```

```
#!/bin/bash -i
#wget -c https://repo.anaconda.com/miniconda/Miniconda3-latest-Linux-x86_64.sh
bash env/Miniconda3-latest-Linux-x86_64.sh
~/miniconda3/bin/conda init
source ~/.bashrc
```

QIIME2 Workflow

- **Install QIIME2:**

```
bash ./setup-scripts/01-install-qiime2-amplicon-2024.2.sh
```

```
#!/bin/bash -i

#update conda to avoid weird problems
conda update conda

#install mamba which is way faster than conda at solving environments
conda install -c conda-forge mamba

#download yaml config file from qiime2 peeps and install, now in a one-liner!
#wget https://data.qiime2.org/distro/amplicon/qiime2-amplicon-2024.2-py38-linux-conda.yml
conda env create -n qiime2-amplicon-2024.2 --file env/qiime2-amplicon-2024.2-py38-linux-conda.yml
```

- **Install cutadapt, bbmap and blast (their configuration files have been downloaded in ./env folder)**

```
bash ./setup-scripts/02-install-conda-envs.sh
```

```
#!/bin/bash -i

#needed for step 1: primer trimming
mamba env create -n cutadapt-env --file env/cutadapt-env.yaml

#needed for step 2: splitting 16S from 18S
mamba env create -n bbmap-env --file env/bbmap.yaml

#OPTIONAL, for checking mock denosing fidelity
mamba env create -n blast-env --file env/blast-env.yaml
```

QIIME2 Workflow


```
name: blast-env
channels:
  - bioconda
  - defaults
dependencies:
  - _libgcc_mutex=0.1
  - _openmp_mutex=4.5
  - blast=2.12.0
  - bzip2=1.0.8
  - c-ares=1.17.1
  - ca-certificates=2021.9.30
  - curl=7.78.0
  - entrez-direct=15.6
  - krb5=1.19.2
  - libcurl=7.78.0
  - libedit=3.1.20210714
  - libev=4.33
  - libgcc-ng=9.3.0
  - libgomp=9.3.0
  - libnghttp2=1.41.0
  - libssh2=1.9.0
  - libstdcxx-ng=9.3.0
  - ncurses=6.2
  - openssl=1.1.1l
  - pcre=8.45
  - perl=5.26.2
  - perl-archive-tar=2.32
  - perl-carp=1.38
  - perl-common-sense=3.74
  - perl-compress-raw-bzip2=2.087
  - perl-compress-raw-zlib=2.087
  - perl-exporter=5.72
  - perl-exporter-tiny=1.002001
  - perl-extutils-makemaker=7.36
  - perl-io-compress=2.087
  - perl-io-zlib=1.10
  - perl-json=4.02
  - perl-json-xs=2.34
  - perl-list-moreutils=0.428
  - perl-list-moreutils-xs=0.428
  - perl-pathtools=3.75
  - perl-scalar-list-utils=1.52
  - perl-types-serialiser=1.0
  - perl-xsloader=0.24
  - zlib=1.2.11
```

```
name: cutadapt-env
channels:
  - conda-forge
  - bioconda
  - defaults
dependencies:
  - _libgcc_mutex=0.1
  - _openmp_mutex=4.5
  - ca-certificates=2021.1.19
  - certifi=2020.12.5
  - cutadapt=3.2
  - dnaio=0.5.0
  - isa-l=2.30.0
  - ld_impl_linux-64=2.35.1
  - libffi=3.3
  - libgcc-ng=9.3.0
  - libgomp=9.3.0
  - libstdcxx-ng=9.3.0
  - ncurses=6.2
  - openssl=1.1.1i
  - pigz=2.5
  - pip=21.0.1
  - python=3.8.6
  - python-isal=0.4.0
  - python_abi=3.8
  - readline=8.1
  - setuptools=52.0.0
  - sqlite=3.34.0
  - tk=8.6.10
  - wheel=0.36.2
  - xopen=1.1.0
  - xz=5.2.5
  - zlib=1.2.11
```

QIIME2 Workflow

- **Download bbsplit-db and classifiers:**

```
#sudo apt install curl
bash ./setup-scripts/03-download-qiime2-classifiers.sh
```

```
#!/bin/bash -i

currdir=$PWD

mkdir -p ~/databases/qiime2-classification-db ; cd ~/databases/qiime2-classification-db

#download database files from OSF, raw artifacts, and scripts
for item in w32ac nvgru ; do curl -O -J -L https://osf.io/$item/download ; done

cd $currdir
```

```
bash ./setup-scripts/04-make-bbsplit-db.sh
```

```
#!/bin/bash -i

currdir=$PWD

mkdir -p ~/databases/bbsplit-db/ ; cd ~/databases/bbsplit-db

#download database files from OSF
for item in kv3xp eux4r npb2k 4qtev s5j6q 5jmkv eahds ; do curl -O -J -L https://osf.io/$item/download ; done

#make databases
chmod u+x make-dbs-bbsplit.sh ; ./make-dbs-bbsplit.sh

cd $currdir
```

QIIME2 Workflow

- **Setup-new-analysis-directory:**

```
bash ./setup-scripts/05-setup-new-analysis-directory.sh Summer_School
```

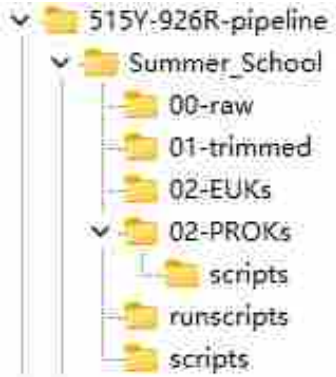
```
#!/bin/bash -i

studyName=$1
studyDir=~/515Y-926R-pipeline/$studyName
mkdir -p $studyDir/scripts $studyDir/runscripts $studyDir/01-trimmed $studyDir/02-PROKs/scripts $studyDir/02-EUKs $studyDir/00-raw

cp 00-trimming-sorting-scripts/* $studyDir/scripts
cp 01-prok-scripts/* $studyDir/02-PROKs/scripts
cp runscripts/* $studyDir/runscripts
cp raw_sequences/* $studyDir/00-raw
cp config/qiime2-2024.2-DADA2-SILVA138.1-PR2_4.14.0.cfg $studyDir/515Y-926R.cfg
echo "studyName=$studyName" >> $studyDir/515Y-926R.cfg

chmod -R +x $studyDir
```

- **A folder for analyzing prokaryotes will be created:**



QIIME2 Workflow

- **All configurations are stored in 515Y-926R.cfg:**

```
#!/bin/bash -i
#this configuration file needs to be edited with the appropriate paths and parameters before running the workflow

qiime2version=qiime2-amplicon-2024.2

#the suffix you want stripped off from your raw files, leaving only the sample identification information
#this will then become the sample names in the pipeline if not otherwise changed in the manifest.csv file

rawFileEndingR1=.R1.fastq.gz
rawFileEndingR2=.R2.fastq.gz

bbsplitdb=~/databases/bbsplit-db/EUK-PROK-bbsplit-db/

#using SILVA for everything for the moment, may include PR2 in future versions

SILVAdb=~/databases/qiime2-classification-db/silva-138.1-ssu-nr99-515FY-926R-classifier.qza

trimR1=210
trimR2=191

studyName=Summer_School
```

QIIME2 Workflow

- **Enter your new working directory:**

```
cd ~/515Y-926R-pipeline/Summer_School
```

- **We can complete data processing with just one script:**

```
bash runscripts/00-denoising-workflow.sh
```

```
#!/bin/bash -i
#Raw data has been put in a folder named "00-raw"
./scripts/00-run-cutadapt.sh
./scripts/01-sort-16S-18S-bbsplit.sh
#files have now been split into 16S and 18S pools, and can be denoised separately
cd 02-PROKs
./scripts/P00-create-manifest.sh
./scripts/P01-import.sh
./scripts/P02-visualize-quality_R1-R2.sh
./scripts/P03-DADA2.sh
#needed for merging script
./scripts/P04-export-DADA2-results.sh
./scripts/P05-classify-eASVs.sh
./scripts/P06-make-sample-metadata-file.sh
#exclude sequences of Chloroplast and Mitochondria
./scripts/P07-exclude-mito-chloro.sh
./scripts/P08-export-exclude-results.sh
./scripts/P09-make-barplot.sh
```

QIIME2 Workflow

- **Trimming off primers**

`./scripts/00-run-cutadapt.sh`

```
#!/bin/bash -i

source 515Y-926R.cfg
conda activate cutadapt-env
mkdir -p 01-trimmed logs/01-trimmed/

for item in `ls 00-raw/*$rawFileEndingR1`

        do

        filestem=`basename $item $rawFileEndingR1`
        R1=00-raw/${filestem}$rawFileEndingR1
        R2=00-raw/${filestem}$rawFileEndingR2

       cutadapt --no-indels --pair-filter=any --error-rate=0.2 --discard-untrimmed \
        -g ^GTGYCAGCMGCCGCGGTAA -G ^CCGYCAATTYMTTTRAGTTT \
        -o 01-trimmed/${filestem}.R1.trimmed.fastq \
        -p 01-trimmed/${filestem}.R2.trimmed.fastq $R1 $R2 \
        2>&1 | tee -a logs/01-trimmed/${filestem}.cutadapt.stderrout

done

conda deactivate
```

QIIME2 Workflow

- **Separate 16S and 18S sequences**

./scripts/01-sort-16S-18S-bbsplit.sh

```
#!/bin/bash -i
conda activate bbmap-env
source 515Y-926R.cfg

mkdir -p 02-PROKs/00-fastq
mkdir -p 02-EUKs/00-fastq
mkdir logs/02-bbsplit

for item in `ls 01-trimmed/*.R1.trimmed.fastq`
          do

          filestem=`basename $item .R1.trimmed.fastq`
          R1in=01-trimmed/$filestem.R1.trimmed.fastq
          R2in=01-trimmed/$filestem.R2.trimmed.fastq

          bbsplit.sh usequality=f qtrim=f minratio=0.30 minid=0.30 pairedonly=f threads=20 -Xmx100g \
          path=$bbsplitdb \
          in=$R1in in2=$R2in basename=$filestem.trimmed.%_#.fastq \
          2>&1 | tee -a logs/02-bbsplit/$filestem.bbsplit_log

done

mv *EUK*fastq 02-EUKs/00-fastq
mv *PROK*fastq 02-PROKs/00-fastq

rm -rf ref

conda deactivate
```

QIIME2 Workflow

- **Trimming off primers**

```
./scripts/00-run-cutadapt.sh
./scripts/01-sort-16S-18S-bbsplit.sh
```

G1.R1.fastq.gz	G1.R1.trimmed.fastq	G1.trimmed.SILVA_132_PROK.cdhit95pc_1.fastq
G1.R2.fastq.gz	G1.R2.trimmed.fastq	G1.trimmed.SILVA_132_PROK.cdhit95pc_2.fastq
G2.R1.fastq.gz	G2.R1.trimmed.fastq	G2.trimmed.SILVA_132_PROK.cdhit95pc_1.fastq
G2.R2.fastq.gz	G2.R2.trimmed.fastq	G2.trimmed.SILVA_132_PROK.cdhit95pc_2.fastq
G3.R1.fastq.gz	G3.R1.trimmed.fastq	G3.trimmed.SILVA_132_PROK.cdhit95pc_1.fastq
G3.R2.fastq.gz	G3.R2.trimmed.fastq	G3.trimmed.SILVA_132_PROK.cdhit95pc_2.fastq
G4.R1.fastq.gz	G4.R1.trimmed.fastq	G4.trimmed.SILVA_132_PROK.cdhit95pc_1.fastq
G4.R2.fastq.gz	G4.R2.trimmed.fastq	G4.trimmed.SILVA_132_PROK.cdhit95pc_2.fastq
G5.R1.fastq.gz	G5.R1.trimmed.fastq	G5.trimmed.SILVA_132_PROK.cdhit95pc_1.fastq
G5.R2.fastq.gz	G5.R2.trimmed.fastq	G5.trimmed.SILVA_132_PROK.cdhit95pc_2.fastq
G6.R1.fastq.gz	G6.R1.trimmed.fastq	G6.trimmed.SILVA_132_PROK.cdhit95pc_1.fastq
G6.R2.fastq.gz	G6.R2.trimmed.fastq	G6.trimmed.SILVA_132_PROK.cdhit95pc_2.fastq
G7.R1.fastq.gz	G7.R1.trimmed.fastq	G7.trimmed.SILVA_132_PROK.cdhit95pc_1.fastq
G7.R2.fastq.gz	G7.R2.trimmed.fastq	G7.trimmed.SILVA_132_PROK.cdhit95pc_2.fastq
G8.R1.fastq.gz	G8.R1.trimmed.fastq	G8.trimmed.SILVA_132_PROK.cdhit95pc_1.fastq
G8.R2.fastq.gz	G8.R2.trimmed.fastq	G8.trimmed.SILVA_132_PROK.cdhit95pc_2.fastq
G9.R1.fastq.gz	G9.R1.trimmed.fastq	G9.trimmed.SILVA_132_PROK.cdhit95pc_1.fastq
G9.R2.fastq.gz	G9.R2.trimmed.fastq	G9.trimmed.SILVA_132_PROK.cdhit95pc_2.fastq

QIIME2 Workflow

- **Importing data**

`./scripts/P00-create-manifest.sh`

```
#!/bin/bash -i

cutME=".trimmed.SILVA_132_PROK.cdhit95pc_1.fastq"
printf "sample-id   forward-absolute-filepath     reverse-absolute-filepath\n" > manifest.tsv

for item in `ls 00-fastq/*1.fastq` ; do

        sampleID=`basename $item $cutME | sed 's/_/-/g'` #remove underscores before importing
        sampleIDstring=`basename $item $cutME`
        R1=`ls 00-fastq/$sampleIDstring.trimmed.SILVA_132_PROK.cdhit95pc_1.fastq`
        R2=`ls 00-fastq/$sampleIDstring.trimmed.SILVA_132_PROK.cdhit95pc_2.fastq`

        printf "$sampleID   \$PWD/$R1 \$PWD/$R2\n" >> manifest.tsv

done
```

sample-id	forward-absolute-filepath	reverse-absolute-filepath
G1	$PWD/00-fastq/G1.trimmed.SILVA_132_PROK.cdhit95pc_1.fastq	$PWD/00-fastq/G1.trimmed.SILVA_132_PROK.cdhit95pc_2.fastq
G2	$PWD/00-fastq/G2.trimmed.SILVA_132_PROK.cdhit95pc_1.fastq	$PWD/00-fastq/G2.trimmed.SILVA_132_PROK.cdhit95pc_2.fastq
G3	$PWD/00-fastq/G3.trimmed.SILVA_132_PROK.cdhit95pc_1.fastq	$PWD/00-fastq/G3.trimmed.SILVA_132_PROK.cdhit95pc_2.fastq
G4	$PWD/00-fastq/G4.trimmed.SILVA_132_PROK.cdhit95pc_1.fastq	$PWD/00-fastq/G4.trimmed.SILVA_132_PROK.cdhit95pc_2.fastq
G5	$PWD/00-fastq/G5.trimmed.SILVA_132_PROK.cdhit95pc_1.fastq	$PWD/00-fastq/G5.trimmed.SILVA_132_PROK.cdhit95pc_2.fastq
G6	$PWD/00-fastq/G6.trimmed.SILVA_132_PROK.cdhit95pc_1.fastq	$PWD/00-fastq/G6.trimmed.SILVA_132_PROK.cdhit95pc_2.fastq
G7	$PWD/00-fastq/G7.trimmed.SILVA_132_PROK.cdhit95pc_1.fastq	$PWD/00-fastq/G7.trimmed.SILVA_132_PROK.cdhit95pc_2.fastq
G8	$PWD/00-fastq/G8.trimmed.SILVA_132_PROK.cdhit95pc_1.fastq	$PWD/00-fastq/G8.trimmed.SILVA_132_PROK.cdhit95pc_2.fastq
G9	$PWD/00-fastq/G9.trimmed.SILVA_132_PROK.cdhit95pc_1.fastq	$PWD/00-fastq/G9.trimmed.SILVA_132_PROK.cdhit95pc_2.fastq

QIIME2 Workflow

- **Visualizing quality of reads**

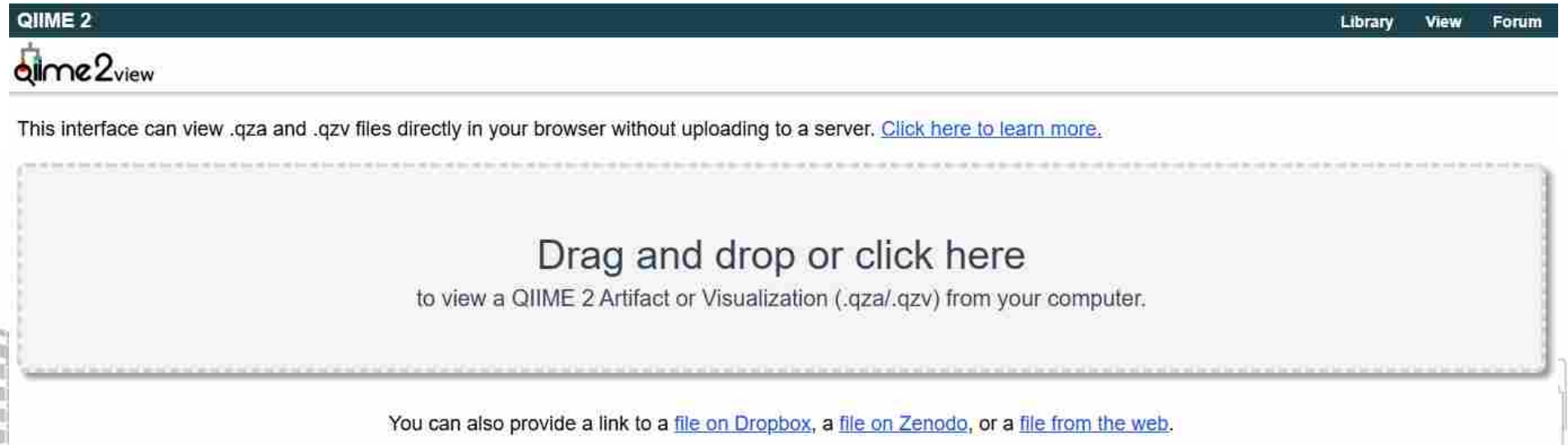
`./scripts/P02-visualize-quality_R1-R2.sh`

```
#!/bin/bash -i
source ../515Y-926R.cfg
conda activate $qiime2version

qiime demux summarize \
--i-data 16s.qza \
--output-dir 02-quality-plots-R1-R2 \
--verbose

conda deactivate
```

- **Drag the output result visualization.qzv to the webpage https://view.qiime2.org/**



QIIME2 Workflow

- **We will determine the trimming length of sequences based on the results in the visualization.qzv**

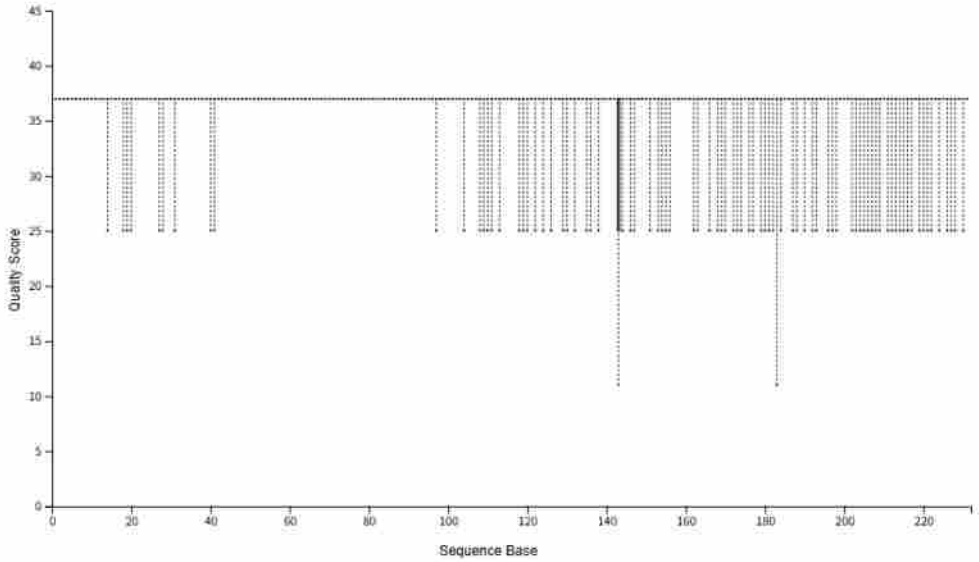
QIIME 2 | Library | View | Forum

qiime2view | File: visualization.qzv | Visualization | Citations | Provenance | Metadata

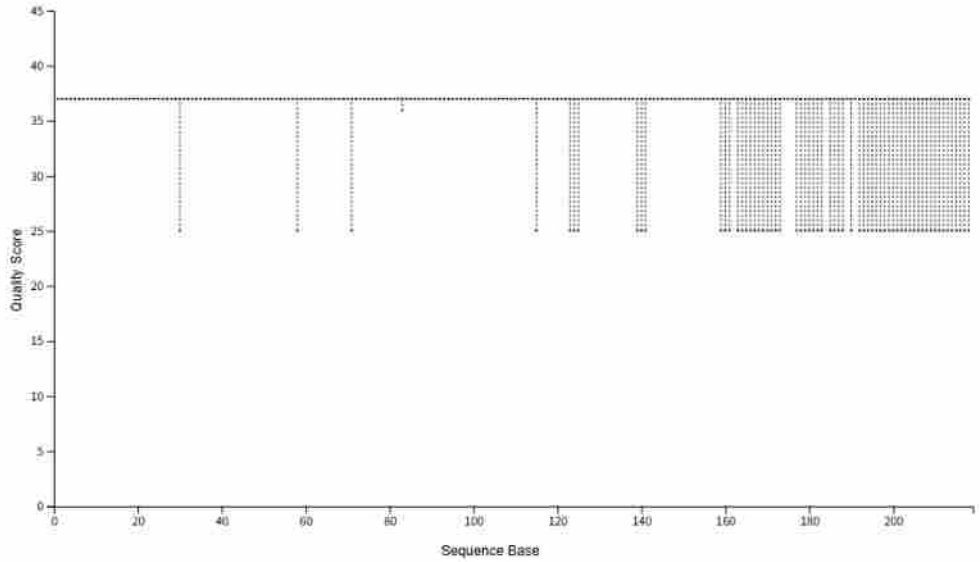
Overview | Interactive Quality Plot

Click and drag on plot to zoom in. Double click to zoom back out to full size. Hover over a box to see the parametric seven-number summary of the quality scores at the corresponding position.

Forward Reads



Reverse Reads



The plot at position 211 was generated using a random sampling of 10000 out of 2133485 sequences without replacement. The minimum sequence length identified during subsampling was 228 bases. Outlier quality scores are not shown in box plots for clarity.

Parametric seven-number summary for **position 211**

Box plot feature	Percentile	Quality score
(Not shown in box plot)	2nd	11
Lower Whisker	9th	25
Bottom of Box	25th	37

The plot at position 192 was generated using a random sampling of 10000 out of 2133485 sequences without replacement. The minimum sequence length identified during subsampling was 215 bases. Outlier quality scores are not shown in box plots for clarity.

Parametric seven-number summary for **position 192**

Box plot feature	Percentile	Quality score
(Not shown in box plot)	2nd	11
Lower Whisker	9th	25
Bottom of Box	25th	37

QIIME2 Workflow

- **Denoising with DADA2**

`./scripts/P03-DADA2.sh`

```
#!/bin/bash -i
source ../515Y-926R.cfg
conda activate $qiime2version

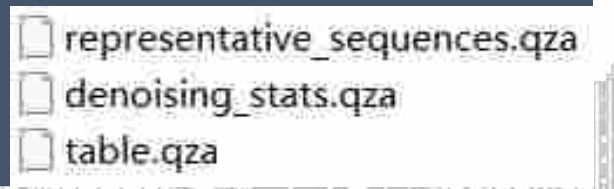
#trimR1=210
#trimR2=191

mkdir -p logs/03-DADA2/

echo "DADA2 truncation length for forward reads = $trimR1 bp" > DADA2_trunclengths.txt
echo "DADA2 truncation length for reverse reads = $trimR2 bp" >> DADA2_trunclengths.txt

qiime dada2 denoise-paired \
  --i-demultiplexed-seqs 16s.qza \
  --p-trim-left-f 0 \
  --p-trim-left-r 0 \
  --p-trunc-len-f $trimR1 \
  --p-trunc-len-r $trimR2 \
  --output-dir 03-DADA2d \
  --p-n-threads 4 \
  --verbose 2>&1 | tee -a logs/03-DADA2/DADA2.stderrout

conda deactivate
```



QIIME2 Workflow

- **Convert the resulting .qza files to readable formats**

`./scripts/P04-export-DADA2-results.sh`

```
#!/bin/bash -i
source ../515Y-926R.cfg
conda activate $qiime2version

timestamp=`date +"%y%m%d-%H%M"`

qiime tools export --input-path 03-DADA2d/representative_sequences.qza --output-path 03-DADA2d/
qiime tools export --input-path 03-DADA2d/denoising_stats.qza --output-path 03-DADA2d/
qiime tools export --input-path 03-DADA2d/table.qza --output-path 03-DADA2d/
biom convert -i 03-DADA2d/feature-table.biom -o 03-DADA2d/feature-table.biom.tsv --to-tsv

for item in 03-DADA2d/*fasta 03-DADA2d/*tsv ; do

        mv $item 03-DADA2d/$timestamp.$studyName.16S.`basename $item`

done

conda deactivate
```

QIIME2 Workflow

- **Results of conversion**

`dna-sequences.fasta`

```
>262ff06c345b8811f587ab7942e1487c
AACAGAGAGGACTTACGTTACTCTAGTATTTGGCGTAAAGTGTATTTAGATTTTTTATGAAGTTTTTTA
TTAAAAATTAATTTTAATTTTAAAGTAATTTAATACTTCTTTAATTATTGTGTGAACTTAAAGTTAATA
GAATTTTTAGTGGAGGTGTGAAAACCAATGAGATTAAAAGGAACATCAAAGGCGAAGGCAATTTTCTTA
AGTTCTCAGACATCAATATACGGAAGTATAGGTAGCAAACTGGATTAGAGACCCAGGTAGTCTATACCG
TCAATTATGGATGTTATTTTTTAGTTTAGTATATTAGACTAAAACTACAGCTAACGCGTTAAACATCTC
GCCTAGGAACTACGATCGCAAGATTA
>614ec59255863501e947c7002b2f7352
AACAGAGCCCCCGAGCGTTATCCGGATTTATTGGGCGTAAAGCGTCTGTAGCCGGTCTCGCAAGTCTTT
AGTTAAAGCCCGAGAGCTCAACTTTCGGAATGCTAAGGAAACTGCAAGACTAGAGGGCATTAGGGGCTG
ATGGAACTCTTGGTGGAGGGGTGAAATCCGTTGATATCAAGAGGAACACCGAAAGCGAAGGCATTCAGC
TGGGGTGACCCTGACGGTGAGAGACGAAAGCGTGGGTAGCGAAAAGGATTAGATACCCTTGTAGTCCAC
GCTGTAAACGATGGATACTAGCTATGTGAAGTGTCGACCCTTCATGTGGCGAAGCTAACGCGTTAAGTA
TCCCGCCTGGGAAGTACGGCCGCAAGGCTA
>0a584f45b26dfe36b9650ad32e95ec28
AACAGAGGATGCAAGCGTTATCCGGATTGACTGGGCGTAAAGCGTCTGGAGGTGGTTTTCTAAGTCTTA
TGTTAAATATCAGAGCTTAACTTTGACTCGGCATAAGAAACTTGATAACTTGAGTGCGATAGAAGCAAT
AAGAATTCCCAGTGTAGCGGTGAAATGCGCAGAGCTTGGGAGGAATACCAGAAGCGAAAGCGTGTTGCT
AGGTCGTTACTGACACTCAAGGACGAAAGCTAGGGTAGCAAATGGGATTAGATACCCCAGTAGTCTTAG
CTCTAAACGATGGAAACTAGGTCTTTTTTCTTTCTGATTCAGAATTATGGAAGAGGATTGAAGCTAACG
CGTTAAGTTTCCCGCCTGGGGAGTATGTTTGCAAGAATG
>6776c9dd4a09b79470cd552f68439dd5
AACAGAGGATGCAAGCGTTATCCGGATTGACTGGGCGTAAAGCGTCTGGAGGTGGTTTTCTAAGTCTTA
TGTTAAATATCAGAGCTTAACTTTGACTCGGCATAAGAAACTTGATAACTTGAGTGCGATAGAAGCAAT
AAGAATTCCCAGTGTAGCGGTGAAATGCGCAGAGCTTGGGAGGAATACCAGAAGCGAAAGCGTGTTGCT
AGGTCGTTACTGACACTCAAGGACGAAAGCTAGGGTAGCAAATGGGATTAGATACCCCAGTAGTCTTAG
CTCTAAATGATGGATACTAGACGTAGGAACTGGATTGACGGTTTCTGTGTCGCAGCTAACGCGTTAAGT
ATCCCGCCTGGGGAGTACGGTCGCAAGACTA
```

`feature-table.biom.tsv`

```
# Constructed from biom file
#OTU ID G1      G2      G3      G4      G5      G6      G7      G8      G9
262ff06c345b8811f587ab7942e1487c        0.0     0.0     0.0     0.0     3.0     0.0     0.0     0.0     0.0
614ec59255863501e947c7002b2f7352        0.0     0.0     0.0     0.0     0.0     0.0     0.0     0.0     2.0
0a584f45b26dfe36b9650ad32e95ec28        13.0    0.0     0.0     0.0     7.0     0.0     0.0     0.0     0.0
6776c9dd4a09b79470cd552f68439dd5        3.0     0.0     0.0     0.0     0.0     0.0     0.0     0.0     0.0
0b876fa6a483f777f7f4c5696d949bbf        8.0     0.0     0.0     0.0     0.0     0.0     0.0     0.0     0.0
49dcbd9f199db1ac43d905b9c11e9cfe        0.0     0.0     0.0     5.0     0.0     0.0     0.0     0.0     0.0
4aedbc84894bcf293da4c965ae5c7352        0.0     0.0     0.0     0.0     0.0     1.0     0.0     0.0     0.0
55867a351e79a4e37e950379be85a8a4        0.0     0.0     0.0     0.0     0.0     0.0     5.0     0.0     0.0
0ef4b8980e2d67d775158150b58483e9        0.0     103.0   0.0     0.0     0.0     0.0     0.0     0.0     0.0
236e02721027c14a7829578d4304c942        0.0     27.0    0.0     0.0     0.0     42.0    0.0     0.0     0.0
625cabfbd66e4f32de52ec86cb441744        0.0     0.0     31.0    9.0     13.0    0.0     33.0    44.0    0.0
e8913379568f59623818e298d387df01        0.0     0.0     0.0     0.0     0.0     0.0     0.0     33.0    0.0
2fe221c35ac82b91677784341e97e33e        13.0    0.0     14.0    0.0     0.0     0.0     0.0     5.0     0.0
014df9fe1ea26f69886fdf906a357eb4        0.0     20.0    0.0     0.0     0.0     20.0    19.0    0.0     0.0
abfd7e57892614230eb088348aebc73d        0.0     0.0     0.0     0.0     0.0     0.0     2.0     0.0     0.0
510b0e077836f29c6b44bd1f3b4d961a        0.0     0.0     0.0     0.0     0.0     0.0     0.0     7.0     0.0
8242a652d47b21381591c70d6d77b8ce        0.0     0.0     0.0     0.0     0.0     0.0     0.0     2.0     0.0
1ef057389bd992a2c4c85cb0e5f77497        0.0     0.0     0.0     0.0     0.0     0.0     0.0     9.0     0.0
068387f20f9981e2ad2bacda2882607f        23.0    9.0     9.0     6.0     10.0    5.0     11.0    3.0     22.0
4a8a237bbb37c6080ed6f1bece32d04d        0.0     0.0     0.0     0.0     0.0     0.0     0.0     3.0     0.0
58df06f752ba019818bd1026bfdfef9b        0.0     0.0     0.0     0.0     0.0     0.0     0.0     6.0     0.0
26c0cf382b28c643f8f4cac66625391c        0.0     0.0     0.0     0.0     2.0     0.0     0.0     0.0     4.0
afe7f73acf92f1a204def942aeb9d1f8        0.0     0.0     0.0     0.0     0.0     0.0     0.0     7.0     0.0
c0647328a1626b4336bf2dfee94e1b74        0.0     0.0     0.0     0.0     2.0     0.0     0.0     0.0     0.0
673d0ebfecd7cc219f03c840289f9fd8        0.0     0.0     0.0     0.0     0.0     0.0     0.0     0.0     2.0
77335bea331993675692cf9422a94f8d        0.0     8.0     2.0     0.0     4.0     4.0     0.0     0.0     0.0
caff1ef0a0649ca043fd671ac1a178ce        0.0     0.0     0.0     0.0     0.0     0.0     0.0     2.0     0.0
0d7a191dfadb6f4342ff80ca2e52a153        2.0     0.0     0.0     0.0     0.0     0.0     0.0     0.0     0.0
118d3394d0199c3805eaa4412596620b        0.0     0.0     0.0     0.0     0.0     0.0     4.0     21.0    0.0
d07d3e2f427d96797d98530dad36f7f0        0.0     0.0     2.0     0.0     0.0     0.0     0.0     0.0     0.0
186b38c898e1cecb82133db49dc76013        0.0     0.0     0.0     0.0     0.0     0.0     0.0     0.0     7.0
24ef43b2bb865b6d545be17c4a112753        0.0     0.0     0.0     0.0     0.0     0.0     0.0     2.0     0.0
156831489ed4daa62b060d3c76b98780        0.0     0.0     0.0     3.0     0.0     0.0     0.0     0.0     0.0
1d40f358f65ba9f1919f165b1d6f1d3d        6.0     0.0     0.0     0.0     0.0     0.0     0.0     0.0     0.0
71fd6731acf7f61d3cd990f71650fc5f        0.0     0.0     0.0     0.0     0.0     0.0     0.0     4.0     0.0
3e8cb5b52cf06683159efd71cec947d9        0.0     0.0     0.0     0.0     0.0     0.0     0.0     2.0     0.0
f7e1abb790ddacdcc6bb8182561debe        0.0     0.0     0.0     0.0     0.0     0.0     0.0     2.0     0.0
```

QIIME2 Workflow

- **Assign taxonomy**

`./scripts/P05-classify-eASVs.sh`

```
#!/bin/bash -i
source ../515Y-926R.cfg
conda activate $qiime2version

qiime feature-classifier classify-sklearn \
  --i-classifier $SILVAdb \
  --i-reads 03-DADA2d/representative_sequences.qza \
  --output-dir 05-classified

conda deactivate
```

- **(Option) Make sample-metadata file**

`./scripts/P06-make-sample-metadata-file.sh`

```
#!/bin/bash -i

#print headers
printf "sample-idExample1_categorical_change_me    Example2_numeric_change_me\n" > sample-metadata.tsv
printf "#q2:typescategorical      numeric\n" >> sample-metadata.tsv

#get sample IDs from manifest file
cut -f1 manifest.tsv | tail -n+2 >> sample-metadata.tsv
```

QIIME2 Workflow

P06-make-sample-metadata-file.sh will now generate a blank sample-metadata.tsv file which you can fill in with your metadata. Such metadata is very useful for making qiime2 plots that can be sorted and organized by your metadata.

sample-id	Example1_categorical_change_me	Example2_numeric_change_me
#q2:types	categorical	numeric
G1		
G2		
G3		
G4		
G5		
G6		
G7		
G8		
G9		

QIIME2 Workflow

- **Delete chloroplast and mitochondria sequences**

`./scripts/P07-exclude-mito-chloro.sh`

```
#!/bin/bash -i
source ../515Y-926R.cfg
conda activate $qiime2version

mkdir -p 07-subsetting

#Filter Chloroplast sequences out from representative_sequences using default SILVA classification
qiime taxa filter-seqs \
  --i-sequences 03-DADA2d/representative_sequences.qza \
  --i-taxonomy 05-classified/classification.qza \
  --p-exclude "f__Mitochondria,o__Chloroplast" \
  --o-filtered-sequences 07-subsetting/exclude_o__Chloroplast_exclude_f__Mitochondria_filtered_seqs.qza

#Create Mitochondria and Chloroplast-free table
qiime taxa filter-table \
    --i-table 03-DADA2d/table.qza \
    --i-taxonomy 05-classified/classification.qza \
    --p-exclude "f__Mitochondria,o__Chloroplast" \
    --o-filtered-table 07-subsetting/exclude_o__Chloroplast_exclude_f__Mitochondria_filtered_table.qza

conda deactivate
```

QIIME2 Workflow

- **Convert the resulting .qza files to readable formats**

`./scripts/P08-export-exclude-results.sh`

```
#!/bin/bash -i
source ../515Y-926R.cfg
conda activate $qiime2version

timestamp=`date +"%y%m%d-%H%M"`

qiime tools export \
    --input-path 07-subsetting/exclude_o__Chloroplast_exclude_f__Mitochondria_filtered_seqs.qza \
    --output-path 07-subsetting/
qiime tools export \
    --input-path 07-subsetting/exclude_o__Chloroplast_exclude_f__Mitochondria_filtered_table.qza \
    --output-path 07-subsetting/

biom convert -i 07-subsetting/feature-table.biom -o 07-subsetting/feature-table.biom.tsv --to-tsv

for item in 07-subsetting/*fasta 07-subsetting/*tsv ; do

        mv $item 07-subsetting/$timestamp.$studyName.16S.`basename $item`

done

conda deactivate
```

QIIME2 Workflow

- **Visualization of prokaryotic community structure**

`./scripts/P09-make-barplot.sh`

```
#!/bin/bash -i
source ../515Y-926R.cfg
conda activate $qiime2version

timestamp=`date +"%y%m%d-%H%M"`

qiime taxa barplot \
  --i-table 07-subsetting/exclude_o__Chloroplast_exclude_f__Mitochondria_filtered_table.qza \
  --i-taxonomy 05-classified/classification.qza \
  --m-metadata-file sample-metadata.tsv \
  --output-dir 09-barplots

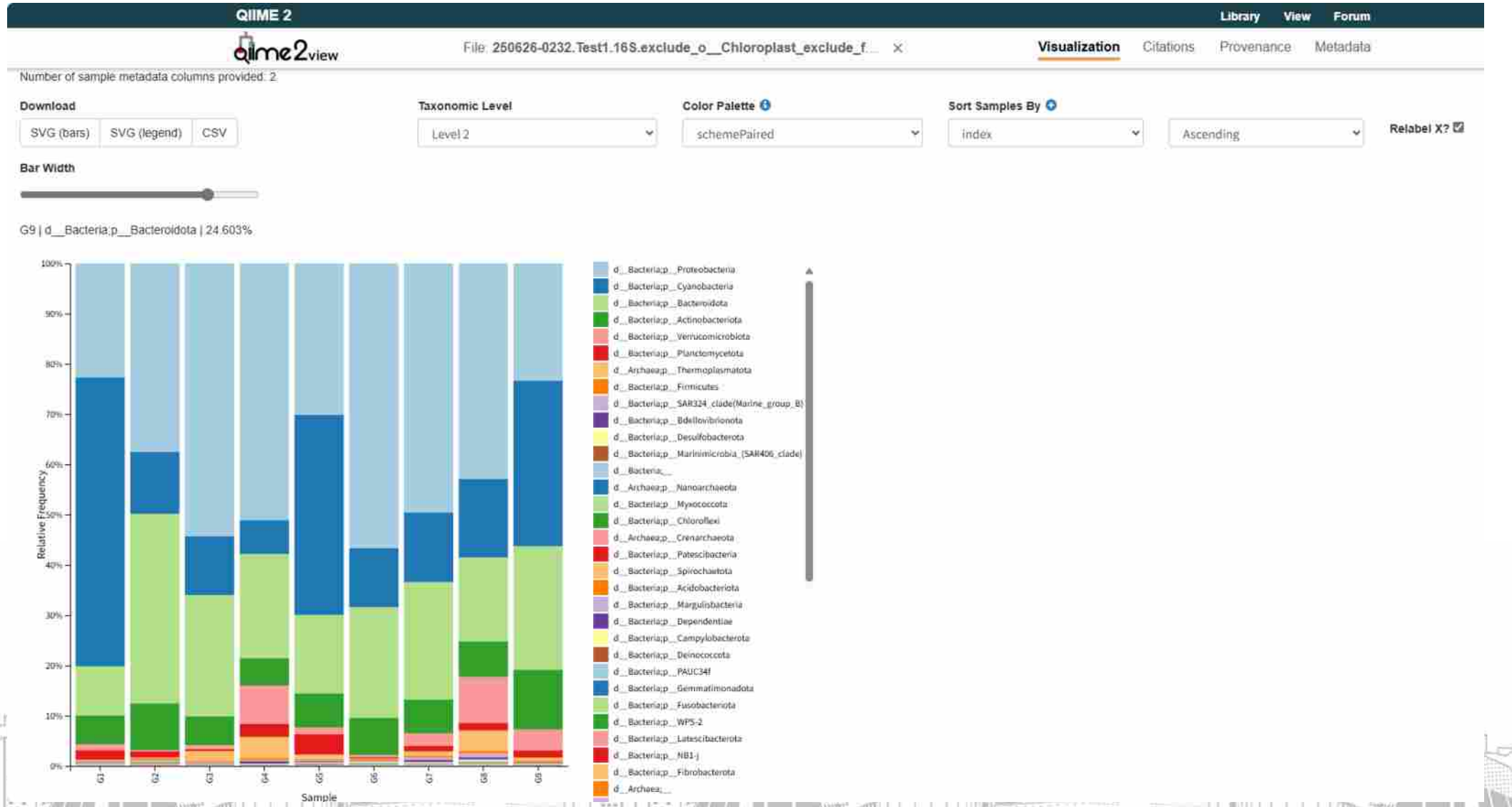
mv 09-barplots/visualization.qzv \
    09-barplots/$timestamp.$studyName.16S.exclude_o__Chloroplast_exclude_f__Mitochondria_filtered_barplot.qzv

conda deactivate
```

- **Drag the output result visualization.qzv to the webpage https://view.qiime2.org/**

QIIME2 Workflow

- **View Results**



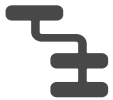
Part 2

Metagenomic Analysis

Xunying Zhou



Metagenomic Data Analysis Workflow




Assembly → **Binnin g** → **Quality Assessment** → **Taxonomic classification** → **Functional Annotatio**

Reconstruct contigs from short reads

- MEGAHIT
- metaSPAdes

Group contigs into bins representing genomes

- MetaBAT2
- MaxBin 2.0
- CONCOCT
- BASALT

Evaluate completeness and contamination of bins

- CheckM

Assign taxonomic labels to MAGs

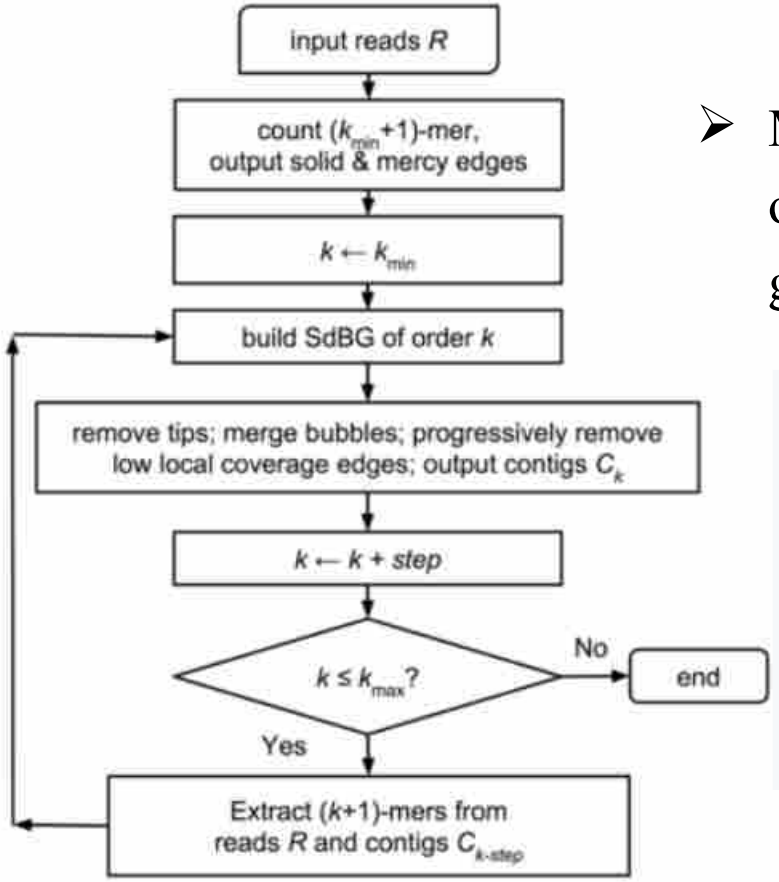
- Gtdb-Tk

Annotate gene functions and Phylogenetic analysis

- Prokka
- METABOLIC

Step1: MEGAHIT

◆ Assembly



The workflow of MEGAHIT

➢ MEGAHIT is an ultra-fast and memory-efficient NGS assembler. It is optimized for metagenomes, but also works well on generic single genome assembly (small or mammalian size) and single-cell assembly.

```
megahit \
  -1 pe_1.fq\  # R1 paired-end reads input
  -2 pe_2.fq \  # R2 paired-end reads input
  --presets meta-sensitive \  # High-sensitivity metagenomic
preset
  --min-contig-len 1000 \  # Minimum contig length: 1000 bp
  -t 16 \  # Use 16 CPU threads
  -m 0.9 \  # Use 90% of available memory
  -o "$OUTPUT_DIR"  # Output directory for results
```

https://github.com/voutcn/megahit

Step1: MEGAHIT

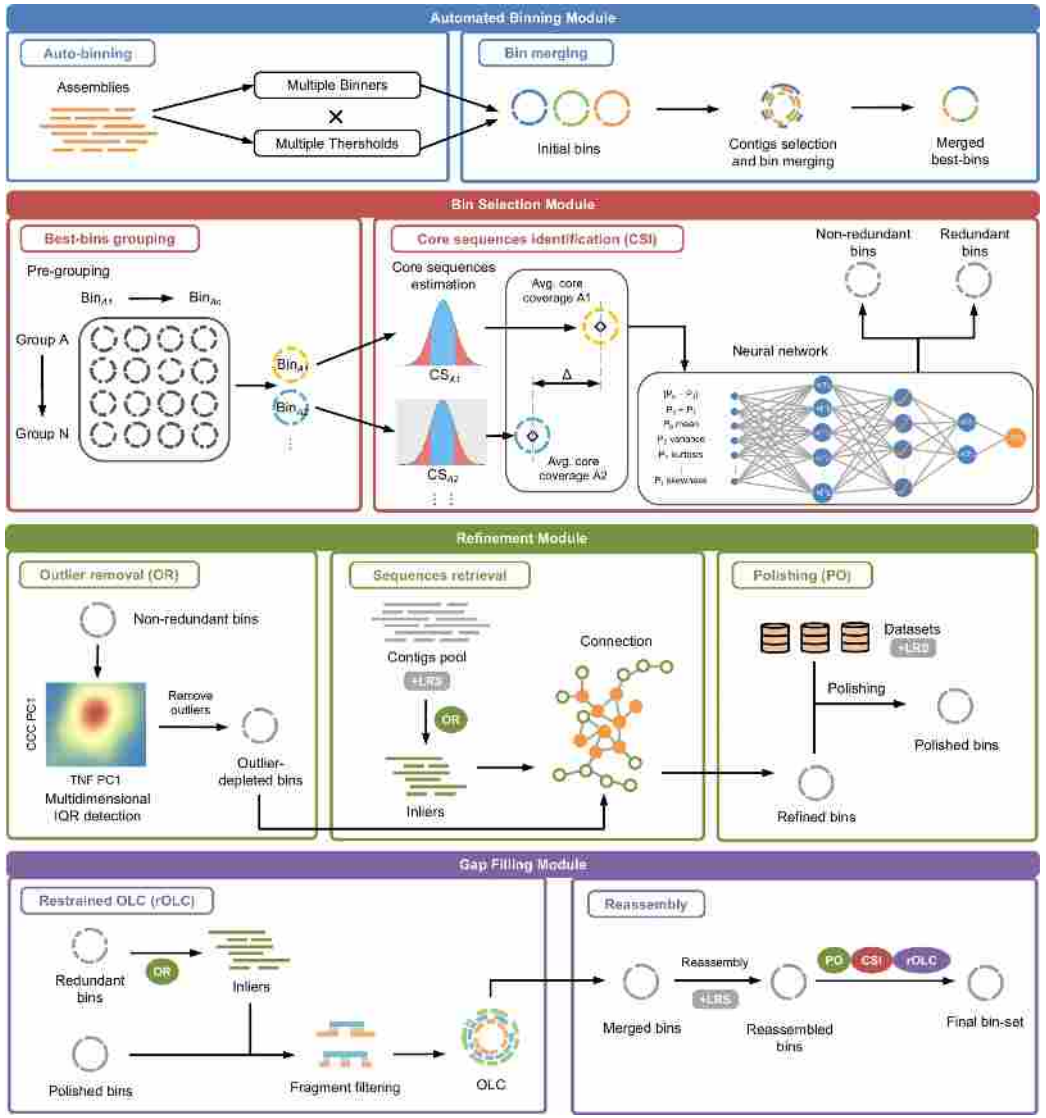
◆ **Assembly**

```
     414 6月   25 03:14 checkpoints.txt
       0 6月   25 03:14 done
54966190 6月   25 03:14 final.contigs.fa
    4096 6月   25 03:14 intermediate_contigs/
  237789 6月   25 03:14 log
    1180 6月   25 01:32 options.json
```

```
xunying@pacific:/home/xunying/data/summer_course_MG/01_megahit/intermediate_contigs$ ls
k109.addi.fa                k129.bubble_seq.fa          k21.final.contigs.fa        k39.local.fa                k69.addi.fa                 k89.bubble_seq.fa
k109.addi.fa.info           k129.bubble_seq.fa.info     k21.final.contigs.fa.info   k39.local.fa.info           k69.addi.fa.info            k89.bubble_seq.fa.info
k109.bubble_seq.fa          k129.contigs.fa             k21.local.fa                k49.addi.fa                 k69.bubble_seq.fa           k89.contigs.fa
k109.bubble_seq.fa.info     k129.contigs.fa.info        k21.local.fa.info           k49.addi.fa.info            k69.bubble_seq.fa.info      k89.contigs.fa.info
k109.contigs.fa             k129.final.contigs.fa       k29.addi.fa                 k49.bubble_seq.fa           k69.contigs.fa              k89.final.contigs.fa
k109.contigs.fa.info        k129.final.contigs.fa.info  k29.addi.fa.info            k49.bubble_seq.fa.info      k69.contigs.fa.info         k89.final.contigs.fa.info
k109.final.contigs.fa       k129.local.fa               k29.bubble_seq.fa           k49.contigs.fa              k69.final.contigs.fa        k89.local.fa
k109.final.contigs.fa.info  k129.local.fa.info          k29.bubble_seq.fa.info      k49.contigs.fa.info         k69.final.contigs.fa.info   k89.local.fa.info
k109.local.fa               k141.addi.fa                k29.contigs.fa              k49.final.contigs.fa        k69.local.fa                k99.addi.fa
k109.local.fa.info          k141.addi.fa.info           k29.contigs.fa.info         k49.final.contigs.fa.info   k69.local.fa.info           k99.addi.fa.info
k119.addi.fa                k141.bubble_seq.fa          k29.final.contigs.fa        k49.local.fa                k79.addi.fa                 k99.bubble_seq.fa
k119.addi.fa.info           k141.bubble_seq.fa.info     k29.final.contigs.fa.info   k49.local.fa.info           k79.addi.fa.info            k99.bubble_seq.fa.info
k119.bubble_seq.fa          k141.contigs.fa             k29.local.fa                k59.addi.fa                 k79.bubble_seq.fa           k99.contigs.fa
k119.bubble_seq.fa.info     k141.contigs.fa.info        k29.local.fa.info           k59.addi.fa.info            k79.bubble_seq.fa.info      k99.contigs.fa.info
k119.contigs.fa             k141.final.contigs.fa       k39.addi.fa                 k59.bubble_seq.fa           k79.contigs.fa              k99.final.contigs.fa
k119.contigs.fa.info        k141.final.contigs.fa.info  k39.addi.fa.info            k59.bubble_seq.fa.info      k79.contigs.fa.info         k99.final.contigs.fa.info
k119.final.contigs.fa       k21.addi.fa                 k39.bubble_seq.fa           k59.contigs.fa              k79.final.contigs.fa        k99.local.fa
k119.final.contigs.fa.info  k21.addi.fa.info            k39.bubble_seq.fa.info      k59.contigs.fa.info         k79.final.contigs.fa.info   k99.local.fa.info
k119.local.fa               k21.bubble_seq.fa           k39.contigs.fa              k59.final.contigs.fa        k79.local.fa
k119.local.fa.info          k21.bubble_seq.fa.info      k39.contigs.fa.info         k59.final.contigs.fa.info   k79.local.fa.info
k129.addi.fa                k21.contigs.fa              k39.final.contigs.fa        k59.local.fa                k89.addi.fa
k129.addi.fa.info           k21.contigs.fa.info         k39.final.contigs.fa.info   k59.local.fa.info           k89.addi.fa.info
```

https://github.com/voutcn/megahit

Step2: BASALT



Qiu, Z., Yuan, L., Lian, CA. *et al.* BASALT refines binning from metagenomic data and increases resolution of genome-resolved metagenomic analysis. *Nat Commun* **15**, 2179 (2024).

◆ Binning

BASALT is an efficient tool for metagenome binning and post-binning refinement with the following key features:

Multiple Assembly Input & Dereplication

- Supports single and co-assembly files, automatically removes redundant bins.
- Efficient processing of multiple assemblies, improving MAG quality.

Standalone Refinement Module

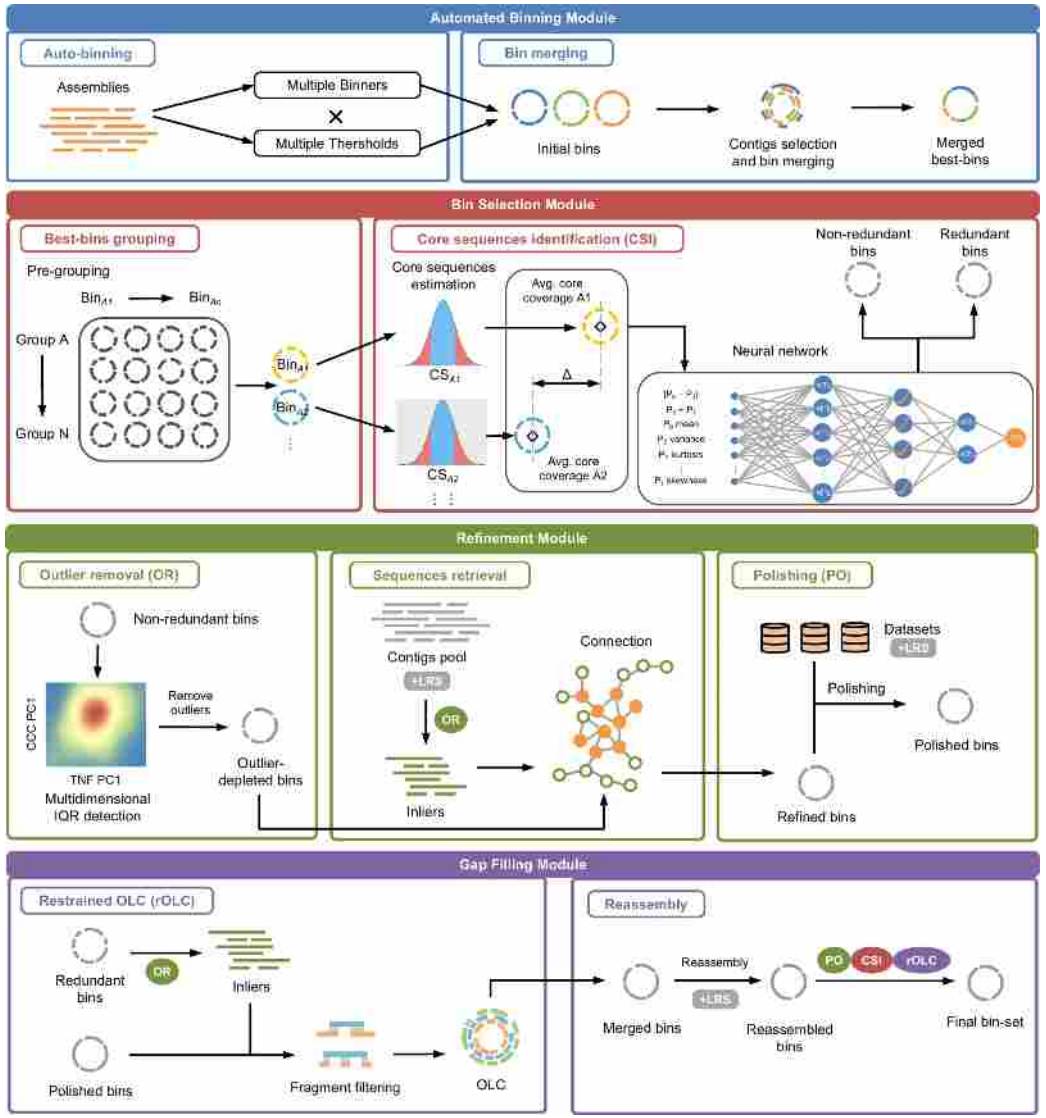
- Removes contamination sequences using neural networks, can be used independently.

Efficient Use of Long-Read Sequences (LRS)

- Maximizes LRS utility in refinement and reassembly, reducing computation time.

https://github.com/EMBL-PKU/BASALT

Step2: BASALT



Qiu, Z., Yuan, L., Lian, CA. *et al.* BASALT refines binning from metagenomic data and increases resolution of genome-resolved metagenomic analysis. *Nat Commun* **15**, 2179 (2024).

◆ Binning

BASALT is an efficient tool for metagenome binning and post-binning refinement.

```
BASALT \
  -a final.contigs.fa \  # Input assembly file
  -s
PTR30_S1.R1.fq,PTR30_S1.R2.fq/PTR3A2B_S16.R1.fq,PT
R3A2B_S16.R2.fq \  # Paired-end short-read files
  -t 16 \  # Use 16 CPU threads
  -m 500 \  # Maximum memory usage (500 GB)
  --max-ctn 10 \  # Maximum contamination cutoff
(10%)
  --min-cpn 50 \  # Minimum completeness cutoff
(50%)
  --mode continue \  # Continue previous run from
checkpoint
```

https://github.com/EMBL-PKU/BASALT

Step2: BASALT

◆ Binning

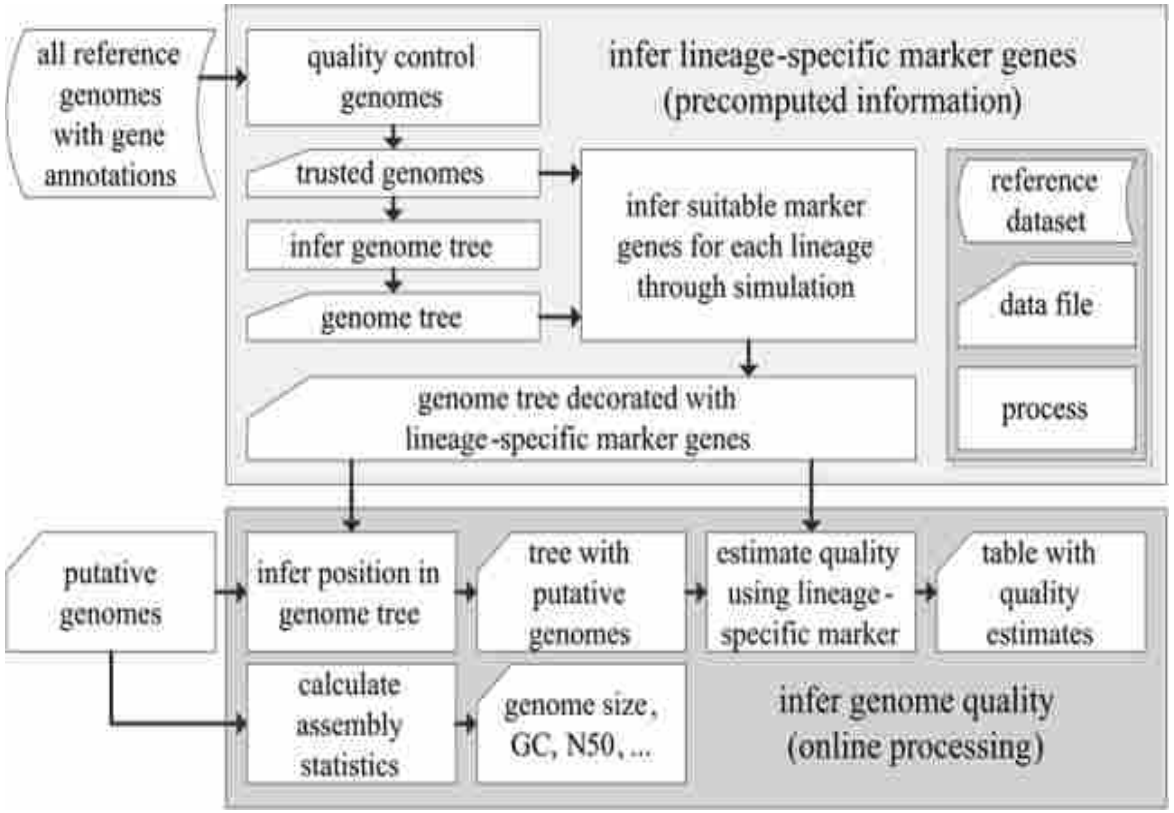
```
202001016 6月   25 10:16 1_DNA-1.bam
229609817 6月   25 10:17 1_DNA-2.bam
       85 6月   25 11:46 Assembly_status.txt
       30 6月   25 10:13 BASALT_command.txt
      530 6月   25 11:36 BASALT_log.txt
154627867 6月   25 11:57 Binsets_backup.tar.gz
  1680226 6月   25 11:54 Coverage_depth_connection_SimilarBin_files_backup.tar.gz
     4096 6月   25 11:54 Final_bestbinset/
       30 6月   25 09:23 final.contigs.fa -> ../01_megahit/final.contigs.fa
 22183541 6月   25 11:54 Group_Bestbinset.tar.gz
 49919400 6月   25 11:54 Group_checkm.tar.gz
    55233 6月   25 11:54 Group_comparison_files.tar.gz
439790005 6月   25 11:56 Group_genomes.tar.gz
 16313534 6月   25 11:40 idba.fa
   106894 6月   25 11:37 Not_mapped_reads.txt
       31 6月   25 09:25 PTR30_S1.R1.fq -> ../00_clean_data/PTR30_S1.R1.fq
       31 6月   25 09:25 PTR30_S1.R2.fq -> ../00_clean_data/PTR30_S1.R2.fq
       34 6月   25 09:25 PTR3A2B_S16.R1.fq -> ../00_clean_data/PTR3A2B_S16.R1.fq
       34 6月   25 09:25 PTR3A2B_S16.R2.fq -> ../00_clean_data/PTR3A2B_S16.R2.fq
      322 6月   25 11:36 Record_error_bin.txt
        0 6月   25 11:36 Total_contig_eliminated_from_deep-refinment.txt
```

```
bin1.1_re.fa
bin2_SPAdes_re-assembly_contigs.1_re.fa
bin3_SPAdes_re-assembly_contigs.fa
bin4.fa
OLC_bin_stats_ext_o.tsv
OLC_bin_stats_ext.tsv
```

Qiu, Z., Yuan, L., Lian, CA. *et al.* BASALT refines binning from metagenomic data and increases resolution of genome-resolved metagenomic analysis. *Nat Commun* **15**, 2179 (2024).

https://github.com/EMBL-PKU/BASALT

Step3: CheckM2



Parks DH, Imelfort M, Skennerton CT, Hugenholtz P, Tyson GW. CheckM: assessing the quality of microbial genomes recovered from isolates, single cells, and metagenomes. Genome Res. 2015 Jul;25(7):1043-55.

◆ Quality Assessment

CheckM2: The main use of CheckM2 is to predict the completeness and contamination of metagenome-assembled genomes (MAGs) and single-amplified genomes (SAGs).

```
checkm2 predict \
  --threads 30 \  # Use 30 CPU threads
  --input Final_bestbinset \  # Input directory with bins
  --output-directory 03_Checkm2 \  # Output directory for results
  --force \  # Overwrite existing output directory
  -x fa  # Specify file extension for input bins
```

https://github.com/chklovski/CheckM2

Step3: CheckM2

◆ Quality Assessment

```
25 23:23 ./
29 17:20 ../
25 23:23 checkm2.log
25 23:22 diamond_output/
25 23:22 protein_files/
25 23:23 quality_report.tsv
```

Name	Completeness	Contamination	Completeness_Model_Used	Translati	Coding_Der	Contig_N5	Average_Ge	Genome_Siz	GC_Content	Total_Codi	Total_Cont	Max_Contig	Additional_Notes
bin1.1_re	91.12	4.45	Gradient Boost (General Model)	11	0.962	28077	280.83989	997393	0.34	1143	65	85768	None
bin2_SPAd	68.25	1.56	Gradient Boost (General Model)	11	0.961	14129	268.63422	815264	0.33	976	116	92258	None
bin3_SPAd	66.6	5.13	Gradient Boost (General Model)	11	0.965	5756	261.74895	774482	0.34	956	172	24622	None
bin4	86.45	3.83	Gradient Boost (General Model)	11	0.946	8470	329.34435	1891373	0.51	1815	287	29281	None

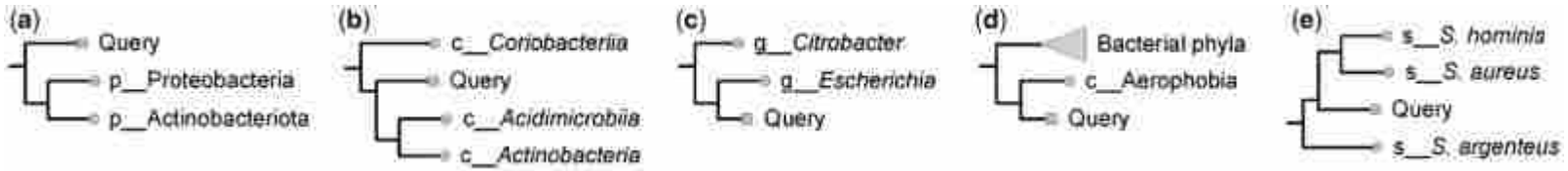
Parks DH, Imelfort M, Skennerton CT, Hugenholtz P, Tyson GW. CheckM: assessing the quality of microbial genomes recovered from isolates, single cells, and metagenomes. Genome Res. 2015 Jul;25(7):1043-55.

https://github.com/chklovski/CheckM2

Step4: GTDB-Tk

◆Taxonomic classification

GTDB-Tk: GTDB-Tk is a software toolkit for assigning objective taxonomic classifications to bacterial and archaeal genomes based on the Genome Database Taxonomy (GTDB). It is designed to work with recent advances that allow hundreds or thousands of metagenome-assembled genomes (MAGs) to be obtained directly from environmental samples. It can also be applied to isolate and single-cell genomes.




```
gtdbtk classify_wf \
  --genome_dir /data/genomes \  # Directory containing input genome files
  --out_dir /data/output \  # Directory where classification results will be
saved
  -x fa \  # Specify the file extension (.fa) for input genome files
  --cpus 16  # Specify the number of CPU threads (16) for parallel processing
```

https://github.com/Ecogenomics/GTDBTk

Step4: GTDB-Tk

◆Taxonomic classification

```
4096 6月   25 23:48 ./
4096 6月   29 17:20 ../
4096 6月   25 23:48 align/
4096 6月   25 23:48 classify/
  32 6月   25 23:39 gtdbtk.ar53.summary.tsv -> classify/gtdbtk.ar53.summary.tsv
  34 6月   25 23:48 gtdbtk.bac120.summary.tsv -> classify/gtdbtk.bac120.summary.tsv
5619 6月   25 23:48 gtdbtk.log
   0 6月   25 23:30 gtdbtk.warnings.log
4096 6月   25 23:48 identify/
```

user_genome	classification	fastani_re	fastani_re	fastani_ta	fastani_ar	fastani_af	closest_plc
bin4	d__Archaea;p__Thermoplasmatota;c__Poseidoniia;o__Poseidoniales;f__Poseidoniaceae;g__MGIIa-L1;s__	N/A	N/A	N/A	N/A	N/A	GCA_002502

user_genome	classification
bin1.1_re	d__Bacteria;p__Actinobacteriota;c__Acidimicrobiia;o__Actinomarinales;f__Actinomarinaceae;g__Actinomarina;s__Actinomarina sp002308095
bin2_SPAdes_re-assembly_contigs.1_re	d__Bacteria;p__Actinobacteriota;c__Acidimicrobiia;o__Actinomarinales;f__Actinomarinaceae;g__Actinomarina;s__
bin3_SPAdes_re-assembly_contigs	d__Bacteria;p__Actinobacteriota;c__Acidimicrobiia;o__Actinomarinales;f__Actinomarinaceae;g__Actinomarina;s__Actinomarina sp002683085

https://github.com/Ecogenomics/GTDBTk

Step5: Prokka

Tool (reference)	Features predicted
Prodigal (Hyatt 2010)	Coding sequence (CDS)
RNAmmer (Lagesen *et al.* , 2007)	Ribosomal RNA genes (rRNA)
Aragorn (Laslett and Canback, 2004)	Transfer RNA genes
SignalP (Petersen *et al.* , 2011)	Signal leader peptides
Infernal (Kolbe and Eddy, 2011)	Non-coding RNA

Feature prediction tools used by Prokka

Suffix	Description of file contents
.fna	FASTA file of original input contigs (nucleotide)
.faa	FASTA file of translated coding genes (protein)
.ffn	FASTA file of all genomic features (nucleotide)
.fsa	Contig sequences for submission (nucleotide)
.tbl	Feature table for submission
.sqn	Sequin editable file for submission
.gbk	Genbank file containing sequences and annotations
.gff	GFF v3 file containing sequences and annotations
.log	Log file of Prokka processing output
.txt	Annotation summary statistics

Description of Prokka output files

◆Functional Annotation

Prokka: Whole genome annotation is the process of identifying features of interest in a set of genomic DNA sequences, and labelling them with useful information. Prokka is a software tool to annotate bacterial, archaeal and viral genomes quickly and produce standards-compliant output files.

```
prokka $fa \
  --outdir ${out_prokka}/${sampleID} \  # Output directory for results
  --prefix ${sampleID} \  # Prefix for output file names
  --locustag ${sampleID} \  # Locus tag for genes
  --kingdom Bacteria \  # Annotates sequences as Bacteria
  --cpus $theads \  # Uses 16 CPU threads
  --compliant  # Ensures standard-compliant output
```

https://github.com/tseemann/prokka

Step5: Prokka

◆Functional Annotation

```
4096 6月   26 00:17 ./
4096 6月   29 17:20 ../
4096 6月   26 00:11 00_bin_arch/
4096 6月   26 00:21 00_bins_bac/
4096 6月   26 00:17 01_archaea_prokka/
4096 6月   26 00:22 02_bact_prokka/
```

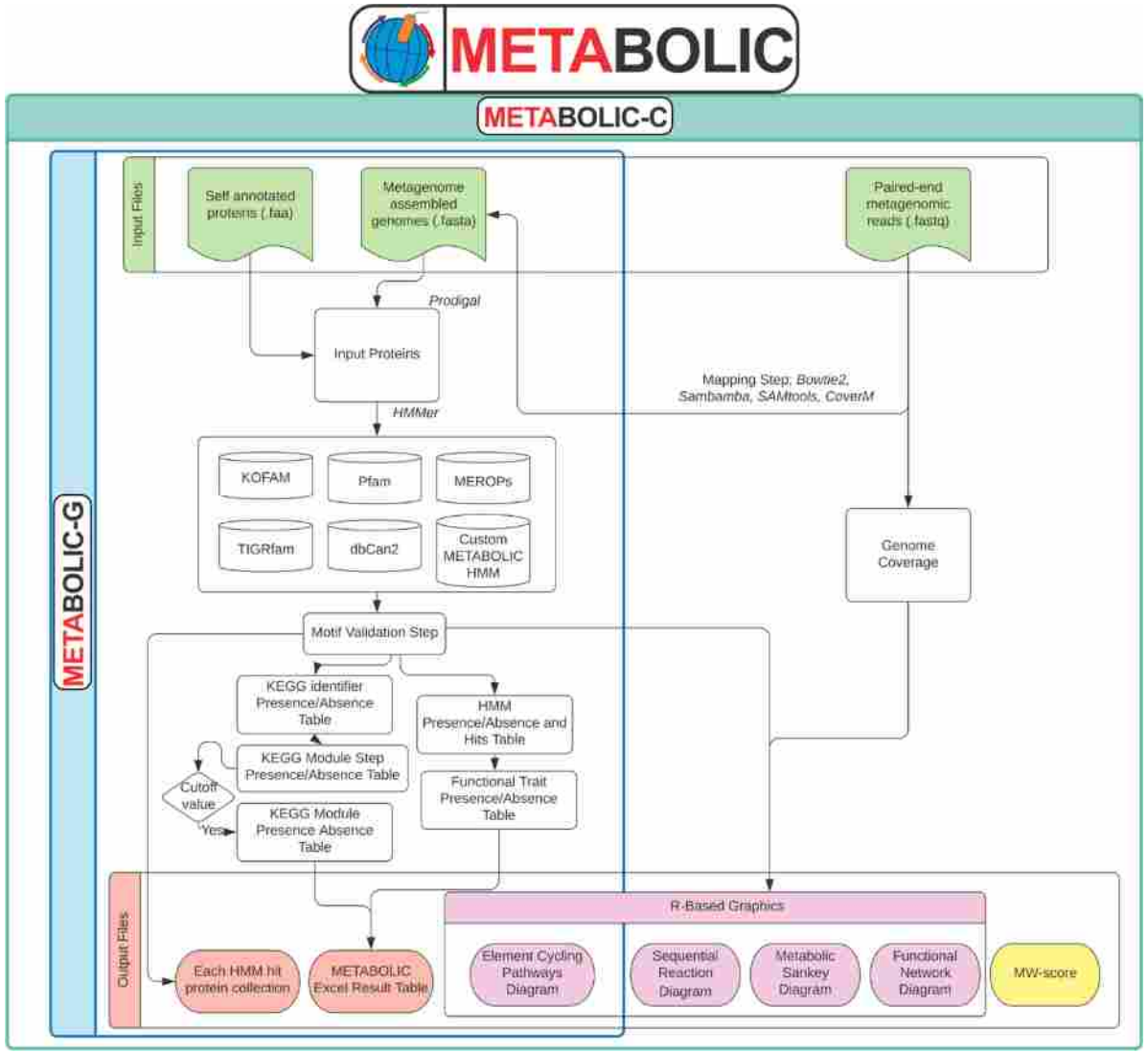
```
 337359 6月   26 00:17 bin4.err
 672955 6月   26 00:17 bin4.faa
1890285 6月   26 00:17 bin4.ffn
1928953 6月   26 00:17 bin4.fna
1943877 6月   26 00:17 bin4.fsa
4325886 6月   26 00:17 bin4.gbk
2629048 6月   26 00:17 bin4.gff
  19737 6月   26 00:17 bin4.log
7834448 6月   26 00:17 bin4.sqn
 433224 6月   26 00:17 bin4.tbl
 143094 6月   26 00:17 bin4.tsv
     90 6月   26 00:17 bin4.txt
```

```
4096 6月   26 00:22 ./
4096 6月   26 00:17 ../
4096 6月   26 00:21 bin1/
4096 6月   26 00:22 bin2/
4096 6月   26 00:22 bin3/
```

```
 208018 6月   26 00:22 bin3.err
 292809 6月   26 00:22 bin3.faa
 803683 6月   26 00:22 bin3.ffn
 790976 6月   26 00:22 bin3.fna
 799920 6月   26 00:22 bin3.fsa
1985462 6月   26 00:22 bin3.gbk
1195815 6月   26 00:22 bin3.gff
  19666 6月   26 00:22 bin3.log
3976340 6月   26 00:22 bin3.sqn
 264649 6月   26 00:22 bin3.tbl
  85067 6月   26 00:22 bin3.tsv
    104 6月   26 00:22 bin3.txt
```

https://github.com/tseemann/prokka

Step6: METABOLIC



◆Functional Annotation

METABOLIC: This software enables the prediction of metabolic and biogeochemical functional trait profiles to any given genome datasets. These genome datasets can either be metagenome-assembled genomes (MAGs), single-cell amplified genomes (SAGs) or isolated strain sequenced genomes.

Program Name	Program Description
METABOLIC-G.pl	Allows for classification of the metabolic capabilities of input genomes.
METABOLIC-C.pl	Allows for classification of the metabolic capabilities of input genomes, calculation of genome coverage, creation of biogeochemical cycling diagrams, and visualization of community metabolic interactions and contribution to biogeochemical processes by each microbial group.

Zhou, Z., Tran, P.Q., Breister, A.M. *et al.* METABOLIC: high-throughput profiling of microbial genomes for functional traits, metabolism, biogeochemistry, and community-scale functional networks. *Microbiome* 10, 33 (2022)

https://github.com/AnantharamanLab/METABOLIC

Step6: METABOLIC

◆Functional Annotation

```
4096 6月   26 10:03 ./
4096 6月   26 09:45 ../
7344 6月   26 09:58 Each_HMM_Amino_Acid_Sequence/
4096 6月   26 10:02 intermediate_files/
4096 6月   26 10:00 KEGG_identifier_result/
4096 6月   26 10:03 METABOLIC_Figures/
4096 6月   26 10:03 METABOLIC_Figures_input/
1031 6月   26 10:03 METABOLIC_log.log
4096 6月   26 10:03 METABOLIC_result_each_spreadsheet/
1448 6月   26 10:03 METABOLIC_result.xlsx
 367 6月   26 10:03 METABOLIC_run.log
```

Nutrient_Cycling_Diagrams

```
20464 6月   26 10:03 bin1.draw_carbon_cycle_single.pdf
 7138 6月   26 10:03 bin1.draw_nitrogen_cycle_single.pdf
12614 6月   26 10:03 bin1.draw_other_cycle_single.pdf
 7394 6月   26 10:03 bin1.draw_sulfur_cycle_single.pdf
20464 6月   26 10:03 bin2.draw_carbon_cycle_single.pdf
 7138 6月   26 10:03 bin2.draw_nitrogen_cycle_single.pdf
12629 6月   26 10:03 bin2.draw_other_cycle_single.pdf
 7394 6月   26 10:03 bin2.draw_sulfur_cycle_single.pdf
20464 6月   26 10:03 bin3.draw_carbon_cycle_single.pdf
 7138 6月   26 10:03 bin3.draw_nitrogen_cycle_single.pdf
12614 6月   26 10:03 bin3.draw_other_cycle_single.pdf
 7394 6月   26 10:03 bin3.draw_sulfur_cycle_single.pdf
20461 6月   26 10:03 bin4.draw_carbon_cycle_single.pdf
 7138 6月   26 10:03 bin4.draw_nitrogen_cycle_single.pdf
12614 6月   26 10:03 bin4.draw_other_cycle_single.pdf
 7406 6月   26 10:03 bin4.draw_sulfur_cycle_single.pdf
```

Zhou, Z., Tran, P.Q., Breister, A.M. *et al.* METABOLIC: high-throughput profiling of microbial genomes for functional traits, metabolism, biogeochemistry, and community-scale functional networks. *Microbiome* 10, 33 (2022)

https://github.com/AnantharamanLab/METABOLIC

Metagenomic Data Analysis Workflow

Assembly → Binning → Quality Assessment → Taxonomic classification → Functional Annotatio

```
  4096 6月   25 01:30 00_clean_data/
  4096 6月   25 03:14 01_megahit/
110592 6月   25 11:57 02_basalt/
  4096 6月   25 23:23 03_Checkm2/
  4096 6月   25 23:48 04_gtdbtk_classify/
  4096 6月   26 00:17 05_prokka/
  4096 6月   26 09:45 06_metabolic/
   639 6月   25 01:29 s01_megahit.sh
   289 6月   25 10:13 s02_basalt.sh
   246 6月   25 23:03 s03_checkm2.sh
   313 6月   25 23:29 s04_gtdb.sh
  1067 6月   26 14:18 s05_prokka.sh
   367 6月   26 09:45 s06_metabolic.sh
```

Others analysis : CoverM

◆Calculates coverage of genomes



CoverM: aims to be a configurable, easy to use and fast DNA read coverage and relative abundance calculator focused on metagenomics applications.

```
coverm genome \
--coupled ${fastq}/PTR30_S1.R1.fq  ${fastq}/PTR30_S1.R2.fq
${fastq}/PTR3A2B_S16.R1.fq  ${fastq}/PTR3A2B_S16.R2.fq  \
# Input fastq
--genome-fasta-files
/home/xunying/data/summer_course_MG/02_basalt/Final_bestbin
set/bin4.fa \ # input genome or MAG
-x fa #File extension of genomes
--methods rpkm tpm # method
```

```
/summer_course_MG/07_coupled_coverm$

0 23:52 ./
0 23:54 ../
1 00:17 coverm.tsv
```

```
Genome  PTR30_S1.R1.fq RPKM     PTR30_S1.R1.fq TPM      PTR3A2B_S16.R1.fq RPKM  PTR3A2B_S16.R1.fq TPM
bin4    0       0       257.43274       333517.1
bin1.1_re       1002.6138       1000000 514.4398        666482.8
coverm.tsv (END)
```

Aroney, S.T., Newell, R.J., Nissen, J.N., Camargo, A.P., Tyson, G.W. and Woodcroft, B.J., 2025. CoverM: Read alignment statistics for metagenomics, Bioinformatics 41:4, https://doi.org/10.1093/bioinformatics/btaf147.

https://github.com/wwood/CoverM?tab=readme-ov-file#usage

- **KofamScan：**

KofamScan is a gene function annotation tool based on KEGG Orthology and hidden Markov model. You need KOfam database to use this tool.

Usage

1. Download KOfam database from ftp://ftp.genome.jp/pub/db/kofam/ and decompress it. You will get profile HMMs in `profiles/` directory and `ko_list`.
2. Create `config.yml` in the same directory as `exec_annotation` script. See below for details.
3. Execute `exec_annotation`.

```
$ ./exec_annotation -o result.txt query.fasta
```

https://github.com/takaram/kofam_scan

Others analysis

海洋科学与工程系
Department of Ocean Science and Engineering

➢ **DiTing：**

DiTing is a Chinese mythical creature who knows everything when he puts ears on the earth's surface.

Parallelly, this program is developed to recognize biogeochemical cycles from environmental omic data accurately and efficiently.

Running

1. One step running

```
diting.py -r <clean_reads_dir> -o <output_dir>
diting.py -r <clean_reads_dir> -a <metagenomic_assembly> -o <output_dir>
```

Example reads run:

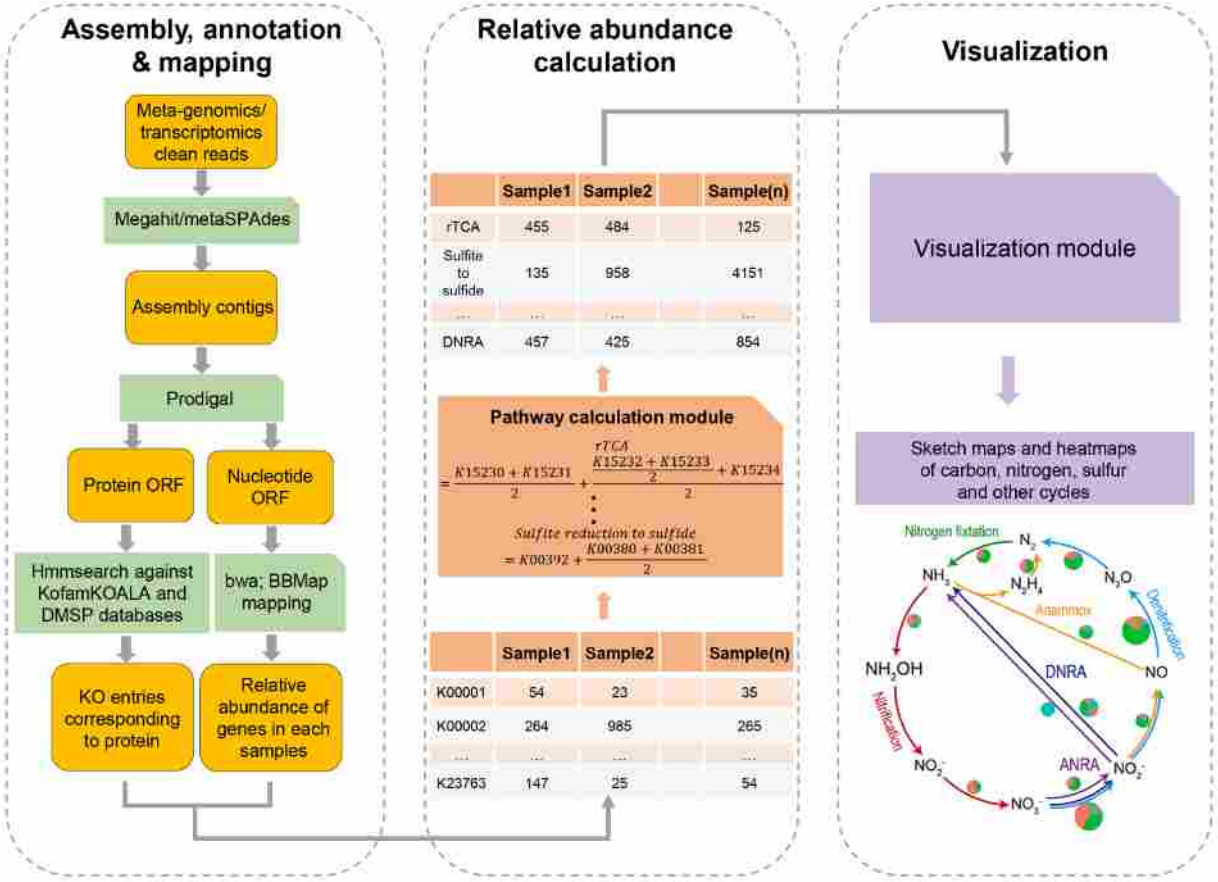
```
#download the example reads
Google Drive:
URL: https://drive.google.com/file/d/132605rtKuA-Xx--eh3aC7i5WIExNWl5k/view?usp=sharing
after download, run:
unzip Clean-reads_interleaved.zip

OR If you are in China, you can download from Baiduyun:
URL: https://pan.baidu.com/s/1gFtJnz1G3pdEqBSFnUqFJw
Password: diti

# run Example
diting.py -r Clean-reads_interleaved -o Clean-reads_interleaved.diting.out
```

The input is the `<clean_reads_dir>` folder containing a group of paired-end metagenomic clean reads, looks like:

```
sample_one_1.fastq
sample_one_2.fastq
sample_two_1.fastq
sample_two_2.fastq
sample_three_1.fastq
sample_three_2.fastq
```



https://github.com/xuechunxu/DiTing

Part 3

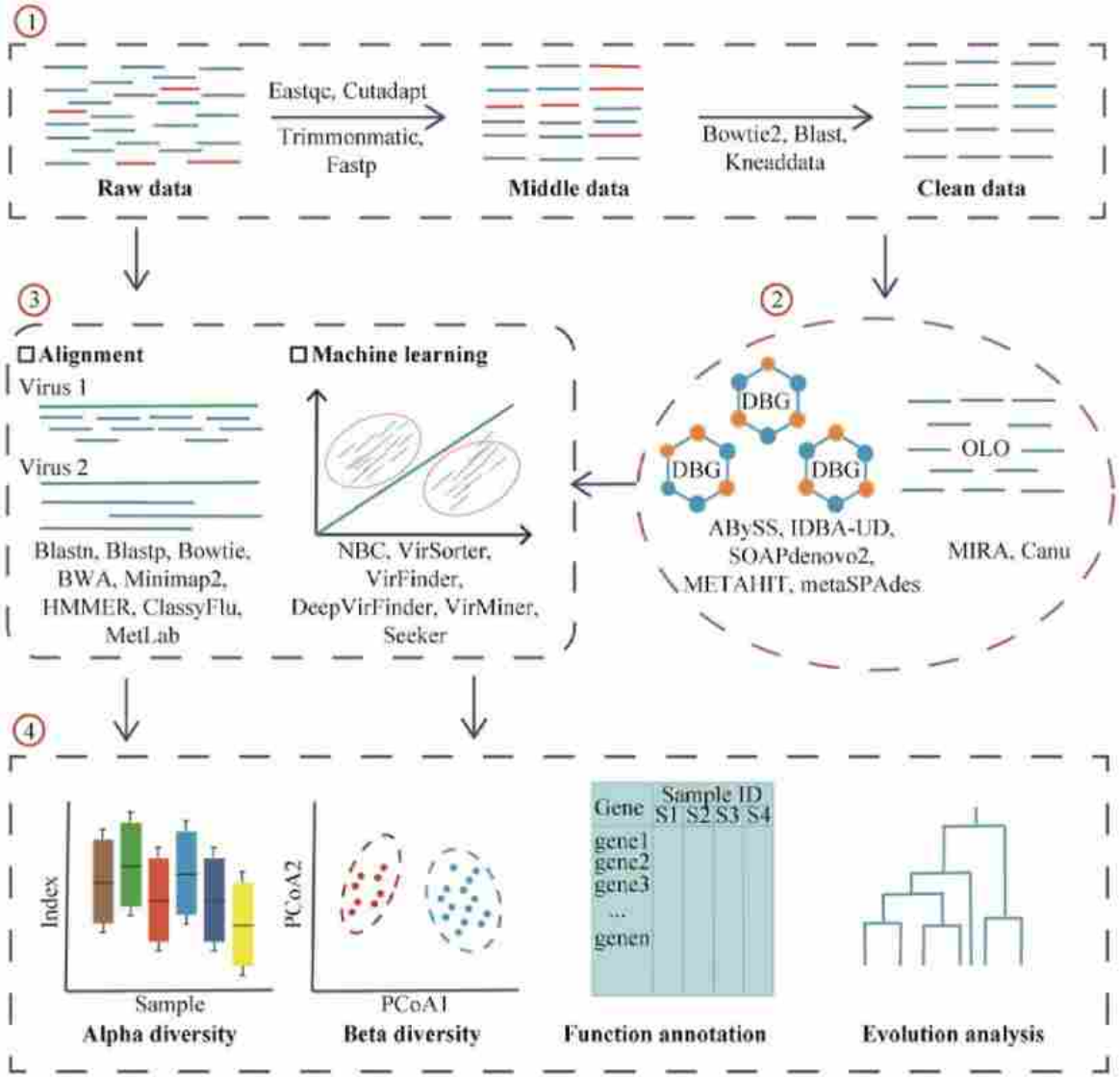
Viromic Analysis

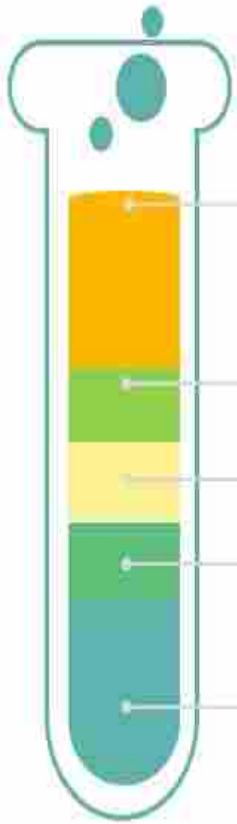
Qian Wen, Yuqian Mo



Work flow

Schematic pipeline of bioinformatic analysis for viromes





SAMPLE TYPE — DNA/RNA, Water/Sediments

DATA PROCESS — Quality control/ Assembly

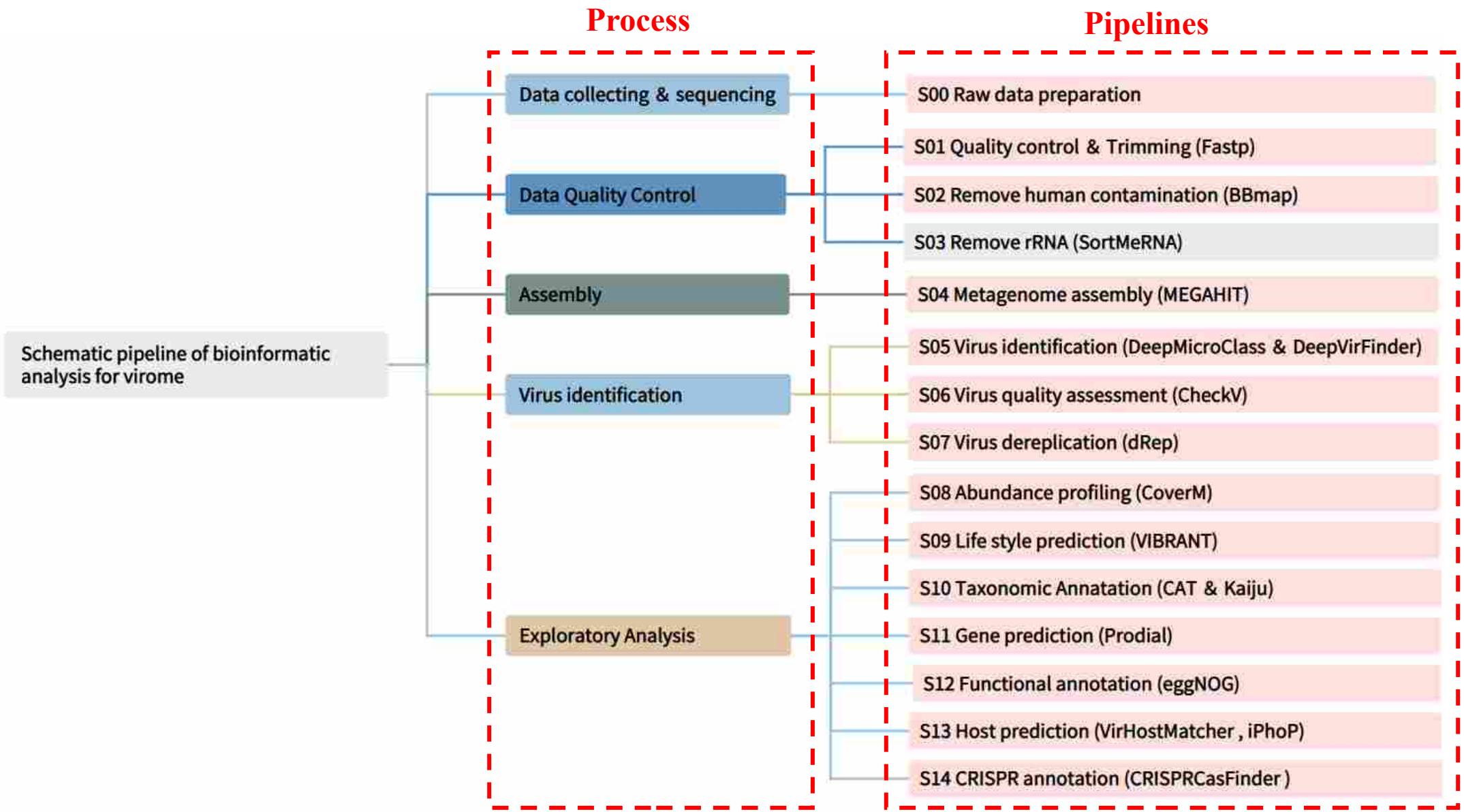
VIRUS IDENTIFICATION — Viral identification, checkV, dRep, Taxonomy annotation

VIRUS-HOST — Virus-host connection prediction, Viral life style

ECOLOGY FUNCTION — AMGs, Evolution

DOI: 10.13345/j.cjb.200372

Work flow


Process
Pipelines
Schematic pipeline of bioinformatic analysis for virome
Data collecting & sequencing
Data Quality Control
Assembly
Virus identification
Exploratory Analysis
S00 Raw data preparation
S01 Quality control & Trimming (Fastp)
S02 Remove human contamination (BBmap)
S03 Remove rRNA (SortMeRNA)
S04 Metagenome assembly (MEGAHIT)
S05 Virus identification (DeepMicroClass & DeepVirFinder)
S06 Virus quality assessment (CheckV)
S07 Virus dereplication (dRep)
S08 Abundance profiling (CoverM)
S09 Life style prediction (VIBRANT)
S10 Taxonomic Annatation (CAT & Kaiju)
S11 Gene prediction (Prodial)
S12 Functional annotation (eggNOG)
S13 Host prediction (VirHostMatcher , iPhoP)
S14 CRISPR annotation (CRISPRCasFinder)

1. Raw data quality control and genome assembly

Name	Description	Web address	Reference
FastQC	Quality control of fastq or fasta data	https://www.bioinformatics.babraham.ac.uk/projects/fastqc/	
Cutadapt	Removal of primer, adapter sequences and ployA tails	https://github.com/marcelm/cutadapt/	[52]
Trimmomatic	Filtering of low-quality sequences	http://www.usadellab.org/cms/?page=trimmomatic	[53]
Fastp	Removal of adapter sequences and filtering of low-quality sequences	https://github.com/OpenGene/fastp	[54]
KneadData	A tool designed to perform quality control on metagenomic sequencing data	https://github.com/biobakery/biobakery/wiki/kneaddata	
MIRA	Assembly of short-read sequences based on the overlap-layout-consensus method	http://www.chevreux.org/projects_mira.html	[58]
Canu	A tool designed for high-noise single-molecule sequencing	https://github.com/marbl/canu	[59]
ABySS	Assembly of short-read sequences based on the de-Brujin-graph method; faster parallel calculation support	https://www.bcgsc.ca/resources/software/abyss	[60]
IDBA-UD	Assembly of short-read sequences based on the de-Brujin-graph method; faster parallel calculation support	https://github.com/loneknightpy/idba	[61]
SOAPdenovo2	Assembly of short-read sequences based on the de-Brujin-graph method	https://github.com/aquaskyline/SOAPdenovo2	[62]
METAHIT	Assembly tools based on the de-Bruijn-graph method for single-cell and metagenomic data assembly	https://github.com/voutcn/megahit	[63]
metaSPAdes	Assembly tools based on the de-Bruijn-graph method for metagenomic data assembly; support for mixed assembly of short-read and long-read sequence	http://cab.spbu.ru/software/meta-spades/	[64]

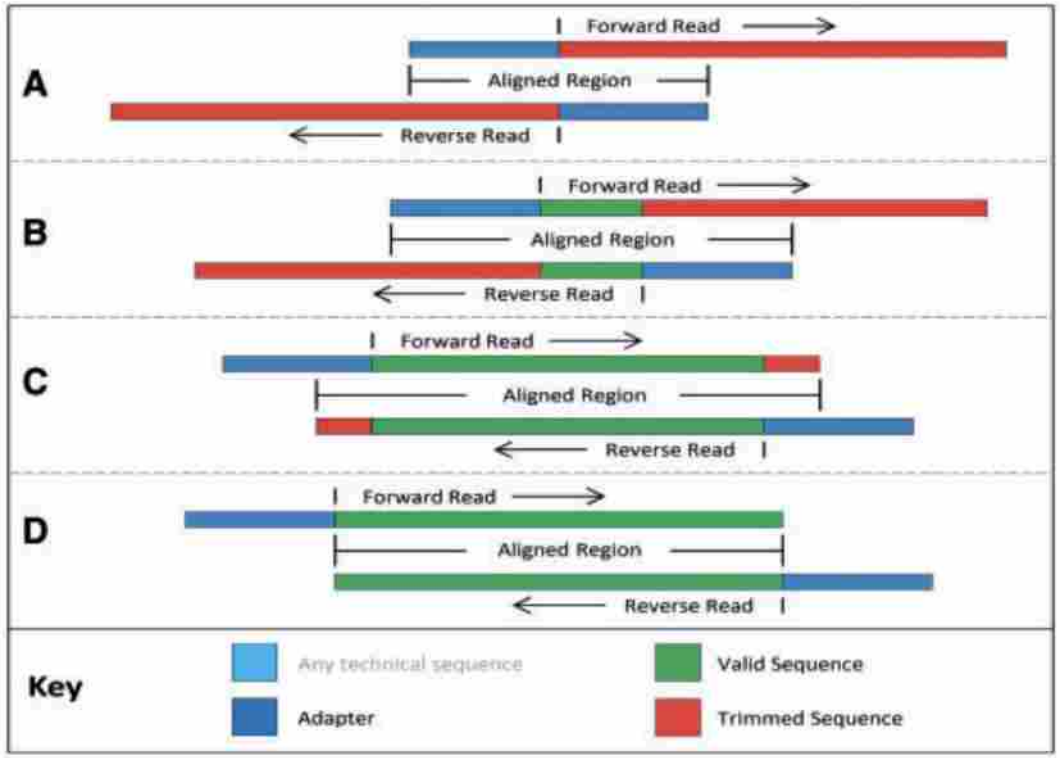
DOI: 10.13345/j.cjb.200372

◆ 1. Trimmomatic / Fastp

1) Remove paired reads with adapter

2) Remove duplicate reads generated by PCR amplification

3) Ren



◆2. bbmap (optional)

1) Download the reference genome

2) Host genome sequence indexing

3) The host genome is compared with the sample read

4) Remove the host sequence on the comparison pair

◆3. sortmerna (optional)

1) Remove rRNA contamination

➢ After Fastp

```
S1.QC.R1.clean.fq.gz  S1.QC.R2.clean.fq.gz
```

➢ After bbmap

```
S1.QC.nocon_nohuman_R1.fq.gz  S1.QC.nocon_nohuman_R2.fq.gz
```

◆MEGAHIT

MEGAHIT is an ultra-fast and memory-efficient NGS assembler. It is optimized for metagenomes, but also works well on generic single genome assembly (small or mammalian size) and single-cell assembly.

Assembly	Statistics	MEGAHIT	MetaSPAdes	IDBA-UD	Faucet
SRR769529+Simulation(5×)	Genome fraction (%)	89.38	92.08	85.81	12.73
	Duplication ratio	1.008	1.007	1.020	1.002
SRR769529+Simulation(20×)	Genome fraction (%)	98.80	99.13	98.96	97.09
	Duplication ratio	1.005	1.002	1.006	1.001
SRR769529+Simulation(50×)	Genome fraction (%)	98.81	99.21	99.06	97.54
	Duplication ratio	1.005	1.002	1.006	1
ERR2762185+Simulation(5×)	Genome fraction (%)	81.90	91.96	84.66	12.48
	Duplication ratio	1.010	1.005	1.018	1.002
ERR2762185+Simulation(20×)	Genome fraction (%)	99.62	99.53	99.56	98.37
	Duplication ratio	1.008	1.002	1.008	1.001
ERR2762185+Simulation(50×)	Genome fraction (%)	99.56	99.50	99.59	98.45
	Duplication ratio	1.008	1.002	1.008	1.003

```
 158 6月   27 20:34 checkpoints.txt
   0 6月   27 20:34 done
1.1K 6月   27 16:06 options.json
220M 6月   27 20:34 S1.QC.contigs.fa
107K 6月   27 20:34 S1.QC.log
```

The assembled contigs were obtained using MEGAHIT, resulting in the file contigs.fasta

2. Viral classification

海洋科学与工程系
Department of Ocean Science and Engineering

DeepMicroClass

DeepMicroClass, a deep learning based method that classifies metagenomic contigs into five sequence classes, e.g., viruses infecting prokaryotic or eukaryotic hosts, eukaryotic or prokaryotic chromosomes.

➢ S1.QG.contigs.fa_pred_one-hot_hybrid.tsv

Sequence Name	Eukaryote	EukaryoteVirus	Plasmid	Prokaryote	ProkaryoteVirus	best_choice	
k127_1490735	0.0007528962451033294	0.0021558087319135666		0.4392675459384918	0.34010493755340576	0.21771883964538574	3
k127_1250297	0.01594778522849083	0.006873198784887791		0.39570891857147217	0.5580115318298340	0.023458553478121758	4
k127_1106032	0.0002106780448229983 4	0.002083758357912302		0.013490971177816391	0.98258577877313 23	0.0016288774786517024	4

VirSorter2

VirSorter2 applies a multi-classifier, expert-guided approach to detect diverse DNA and RNA virus genomes

DeepVirFinder

DeepVirFinder predicts viral sequences using deep learning method. The method has good prediction accuracy for short viral sequences, so it can be used to predict sequences from the metagenomic data.

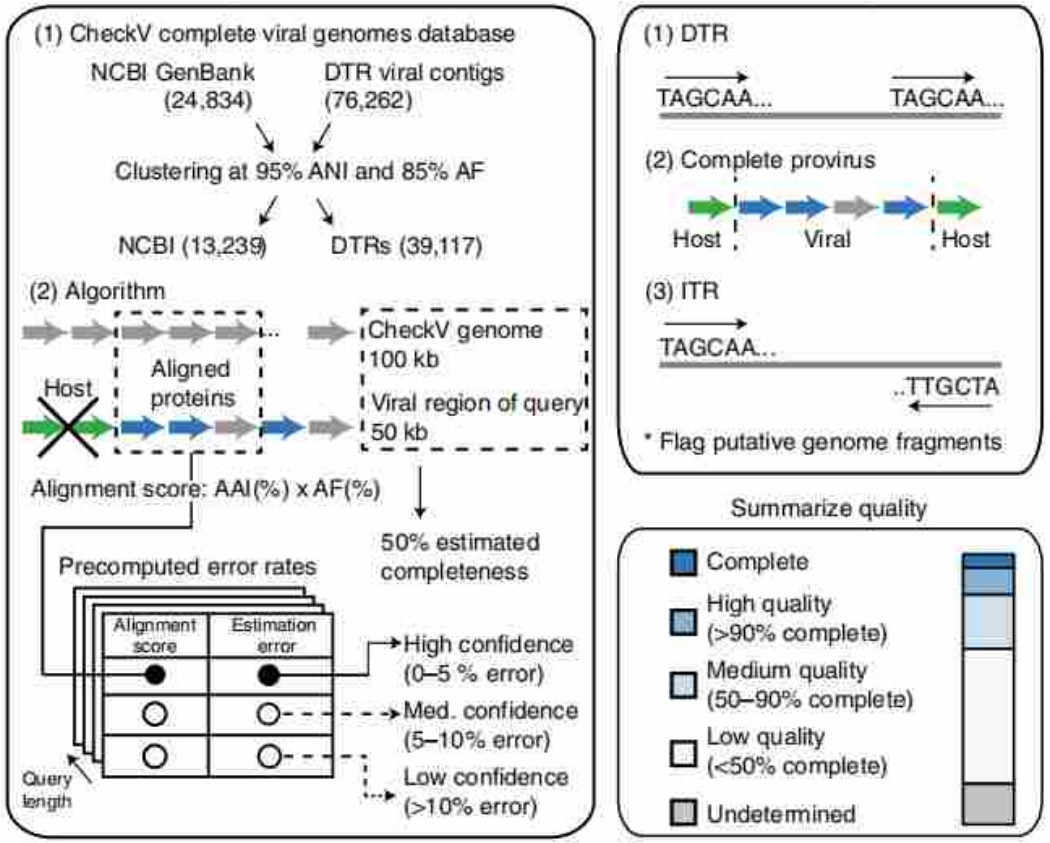
name	len	score	pvalue		
k127_673244 flag=1 multi=4.0000 len=1101	1101	0.0008362963562831283	0.5211118664551704		
k127_673242 flag=1 multi=2.9121 len=1037	1037	0.012337682768702507	0.3279515069113981		
k127_384723 flag=1 multi=2.0000 len=1020	1020	0.07369395345449448	0.20915476999773397		

S1.QG.contigs.fa_gt1000 bp_dvfpred.txt

genomad

geNomad's primary goal is to identify viruses and plasmids in sequencing data (isolates, metagenomes, and metatranscriptomes).

3. Check the quality and calculate the abundance



◆ CheckV

A readable way to validate data

◆ CoverM

CoverM aims to be a configurable, easy to use and fast DNA read coverage and relative abundance calculator focused on metagenomics applications.

CheckV
Virus assessment criteria

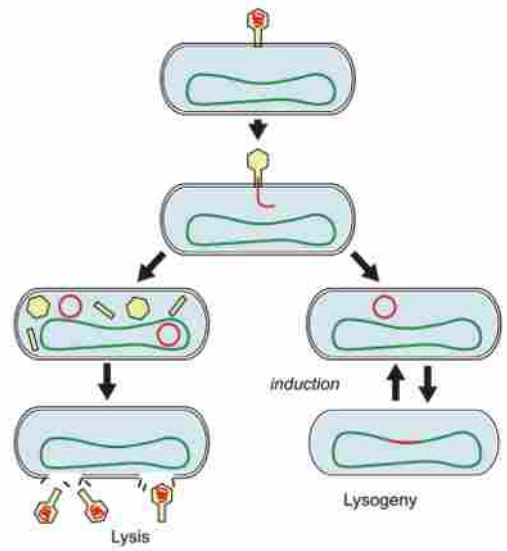
➢ quality_summary.tsv

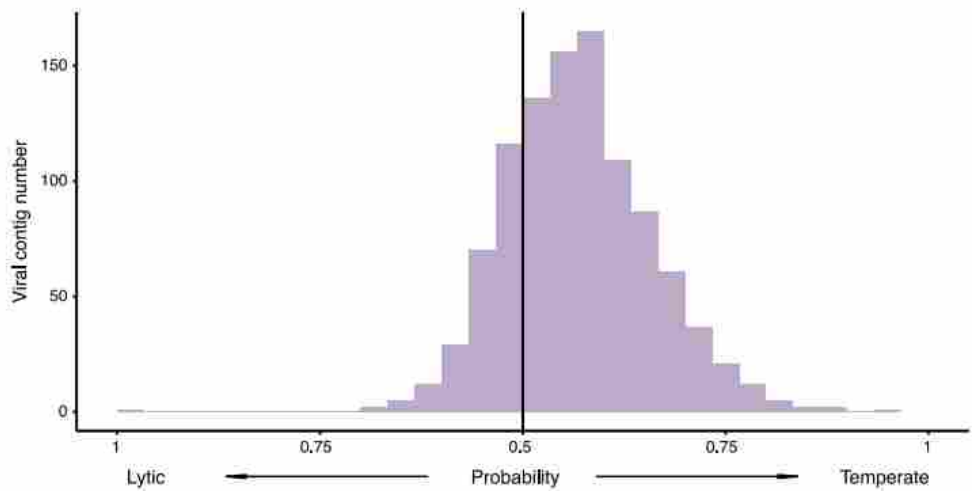
contig_id	contig_length	provirus	proviral_length	gene_count	viral_genes	host_genes	checkv_quality	miuvig_quality	completeness	co
mpleteness_method	contamination	kmer_freq	warnings							
k127_384731	1127	No	NA	2	0	2	Not-determined	Genome-fragment	NA	NA	0.0	1.0	no viral genes detected	
k127_721344	1121	No	NA	2	0	1	Not-determined	Genome-fragment	NA	NA	0.0	1.0	no viral genes detected	
k127_48112	2884	No	NA	3	0	1	Low-quality	Genome-fragment	1.6	AAI-based (medium-confidence)	0.0	1.0	no viral g
enes detected										
k127_336650	1122	No	NA	2	0	1	Not-determined	Genome-fragment	NA	NA	0.0	1.0	no viral genes detected	
k127_432853	1566	No	NA	3	0	1	Not-determined	Genome-fragment	NA	NA	0.0	1.0	no viral genes detected	

4. Life style prediction (for phages)

◆ VIBRANT

VIBRANT: automated recovery, annotation and curation of microbial viruses, and evaluation of viral community function from genomic sequences.





➢ The output folder of **VIBRANT**

```
...
VIBRANT_figures_viral/
VIBRANT_HMM_tables_parsed_viral/
VIBRANT_HMM_tables_unformatted_viral/
VIBRANT_log_viral.log
VIBRANT_phages_viral/
VIBRANT_results_viral/
viral.prodigal.faa
viral.prodigal.ffn
viral.prodigal.gff
```

10.1038/s41586-020-2192-1

5. Taxonomy Annotation of vOTUs

◆ Kaiju

Kaiju is a program for the taxonomic classification of high-throughput sequencing reads, e.g., Illumina or Roche/454, from whole-genome sequencing of metagenomic DNA. Reads are directly assigned to taxa using the NCBI taxonomy and a reference database of protein sequences from microbial and viral genomes.

Virus Taxonomy: 2024 Release

EC 56, Bari, Italy, August 2024

Email ratification February 2025 (MSL #40)

7 realms, 11 kingdoms, 22 phyla, 4 subphyla, 49 classes, 93 orders, 12 suborders, 368 families, 213 subfamilies, 3769 genera, 86 subgenera, 16215 species

➢ kaiju.vOTU.Taxonomy.txt

```
vOTU_21 2565501 Viruses; Uroviricota; Caudoviricetes; NA; NA;
vOTU_11 2881399 Viruses; Uroviricota; Caudoviricetes; NA; NA;
vOTU_28 0
vOTU_6  0
vOTU_4  0
vOTU_24 1978007 Viruses; Uroviricota; Caudoviricetes; Crassvir
```

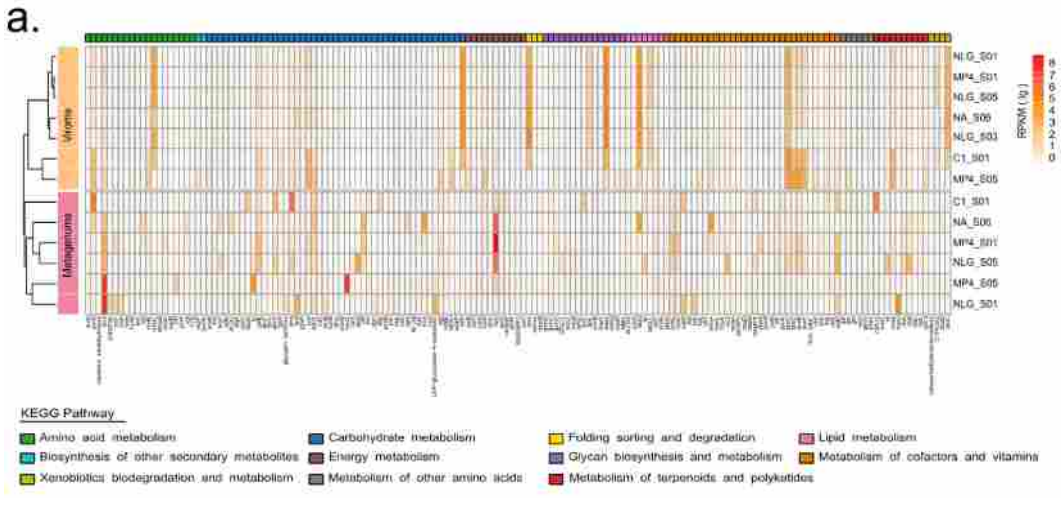
◆CAT

CAT: tools for taxonomic classification of contigs and metagenome-assembled genomes (MAGs) and for taxonomic profiling of metagenomes

➢ cat_taxonomy_ folder

```
euk_classification.txt
euk.out.BAT.alignment.diamond
euk.out.BAT.contig2classification.txt
euk.out.BAT.log
euk.out.BAT.ORF2LCA.txt
euk.out.BAT.predicted_proteins.faa
euk.out.BAT.predicted_proteins.gff
run.log
```

6. Viral function annotation



Prodigal

Fast, reliable protein-coding gene prediction for prokaryotic genomes.

```
viral.faa
viral.ffn
viral.gff
```

Prodigal Output Results

Example of viral.gff

```
##gff-version  3
# Sequence Data: seqnum=1;seqlen=21952;seqhdr="vOTU_1"
# Model Data: version=Prodigal.v2.6.3;run_type=Metagenomic;model="5|Anaplasma_marginale_Maries|B|49.8|11|1";gc_cont=49.80;transl_table=11;uses_sd=1
vOTU_1  Prodigal_v2.6.3 CDS     243     1421    134.3   +       0       ID=1_1;partial=00;start_type=ATG;rbs_motif=GGAG/GAGG;rbs_spacer=5-10bp;gc_cont=0.521;conf=
100.00;score=134.28;cscore=126.78;sscore=7.49;rscore=6.85;uscore=-3.03;tscore=3.68;
```

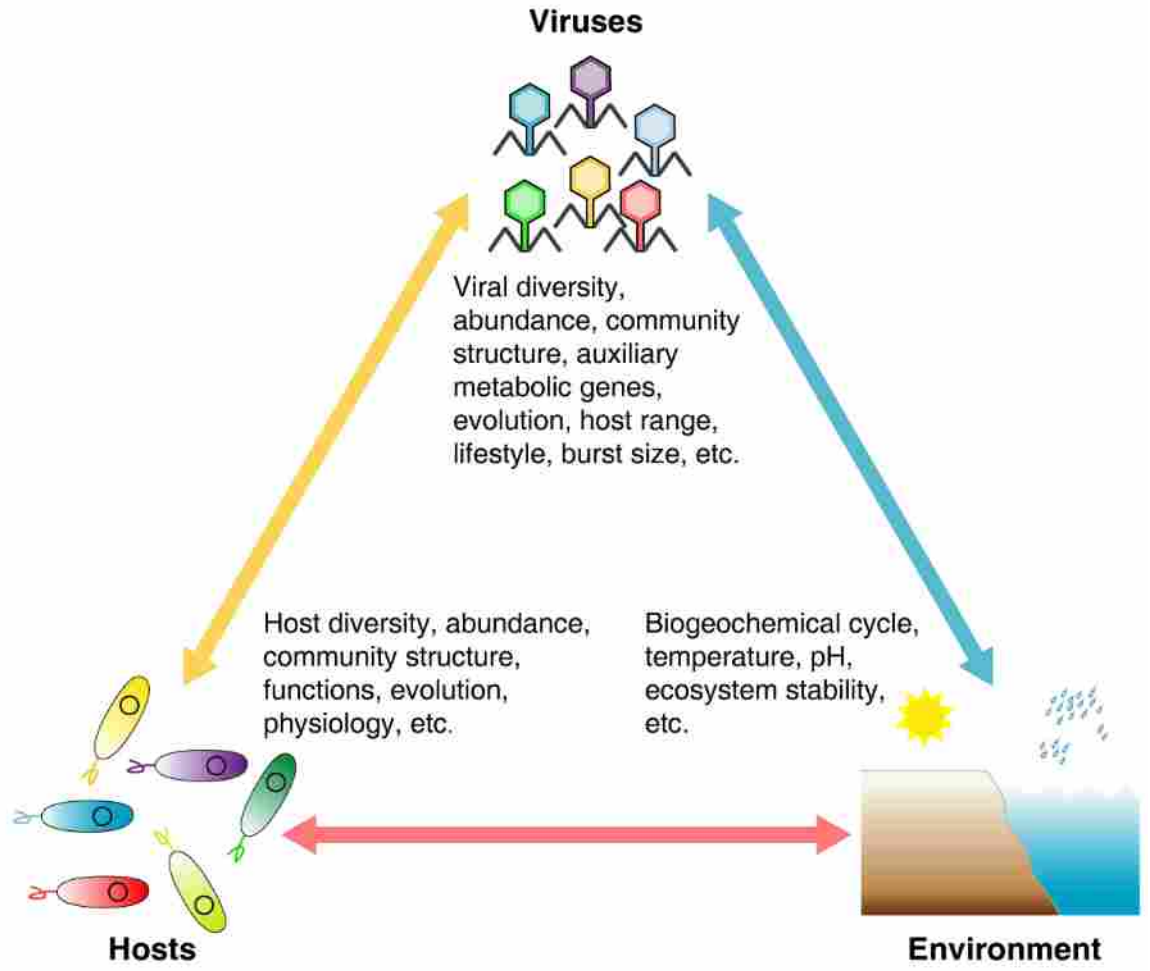
EggNOG

EggNOG-mapper is a tool for fast functional annotation of novel sequences. It uses precomputed orthologous groups and phylogenies from the eggNOG database (http://eggnog5.embl.de) to transfer functional information from fine-grained orthologs only.

S1.emapper.annotations

```
#query  seed_ortholog   evalue  score   eggNOG_OGs      max_annot_lvl   COG_category    Description     Preferred_name  GOs     EC      KEGG_ko KEGG_Pathway    KE
GG_Module       KEGG_Reaction   KEGG_rclass     BRITE   KEGG_TC CAZy    BiGG_Reaction   PFAMs
vOTU_1_2        1082932.ATCR1_06771     1.57e-15        84.0    28NCF@1|root,2ZBII@2|Bacteria,1R9KY@1224|Proteobacteria,2U1P7@28211|Alphaproteobacteria,4B747@8211
5|Rhizobiaceae  28211|Alphaproteobacteria        -       -       -       -       -       -       -       -       -       -       -       -       -       -
vOTU_1_10       1296415.JACC01000059_gene1025   2.38e-05        55.1    COG1874@1|root,COG3291@1|root,COG3420@1|root,COG3897@1|root,COG4257@1|root,COG4886@1|root,
COG1874@2|Bacteria,COG3291@2|Bacteria,COG3420@2|Bacteria,COG3897@2|Bacteria,COG4257@2|Bacteria,COG4886@2|Bacteria,4NMJ3@976|Bacteroidetes,1IKV6@117743|Flavobacter
iia,2YJTU@290174|Aquimarina     976|Bacteroidetes       G       leucine- rich repeat protein    -       -       -       -       -       -       -       -       --
-       -       ASH,DUF5011,Laminin_G_3
```

7. Virus-host connection



Methods	Principal	
Sequence similarity	Phages often integrate parts of their genomes into their hosts.	blastn
CRISPRs	The spacers in CIRSPR array may indicate previous invading phages.	CIRSPR-Cas Finder
Oligonucleotide profiling	Phages may leave specific gene elements (e.g., tRNA, oligonucleotide sequences) in host genomes during infection.	tRNAscan
Abundance-based profiling	This co-abundance of phages and Prokaryotes in the environment can be used to infer virus-host relationships	cytoscape

VirHostMatcher:

Matching hosts of viruses based on oligonucleotide frequency (ONF) comparison

hostNCBIName	hostName	hostSuperkingdom	hostPhylum	hostClass	hostOrder	hostFamily	hostGenus	hostSpecies
RS_1.bin.1.fa	Draconibacterium	Bacteria	Bacteroidota	Bacteroidia	Bacteroidales	Prolixibacteraceae	Draconibacterium	Draconibacterium
RS_1.bin.12.fa	DAOVZS01	Bacteria	Pseudomonadota	Gammaproteobacteria	Beggiatoales	Beggiatoaceae	DAOVZS01	DAOVZS01
RS_1.bin.15.fa	GRL18	Bacteria	Pseudomonadota	Gammaproteobacteria	GRL18	GRL18	GRL18	GRL18
RS_1.bin.20.fa	JAGLFE01	Bacteria	Pseudomonadota	Gammaproteobacteria	Beggiatoales	Beggiatoaceae	JAGLFE01	JAGLFE01
RS_1.bin.24.fa	JAGLFH01	Bacteria	JAFGBW01	JAFGBW01	JAFGJP01	JAFGJP01	JAGLFH01	JAGLFH01

virhostmatcher_host_taxa.txt_new.txt

iPHoP:

The Ultimate Host Prediction Tool for Microbial Viruses

Virus	Method	Host	Host taxonomy	Metric 1	Score 1	Metric 2	Score 2	Rank	Host representative
vOTU_1	iPHoP-RF	GB_GCA_017525505.1	d__Bacteria;p__Bacteroidota;c__Bacteroidia;o__Bacteroidales;f__WCHB1-69;g__RUC11690;s__RUC11690 sp017525505	Confidence_score	0.390	FDR	0.773	1	GB_GCA_017525505.1
vOTU_1	iPHoP-RF	MGYG000290379	d__Bacteria;p__Bacteroidota;c__Bacteroidia;o__Bacteroidales;f__WCHB1-69;g__RUC11690;s__	Confidence_score	0.390	FDR	0.773	1	MGYG000290379
vOTU_1	iPHoP-RF	GB_GCA_905233625.1	d__Bacteria;p__Bacteroidota;c__Bacteroidia;o__Bacteroidales;f__WCHB1-69;g__RUC11690;s__RUC11690 sp905233625	Confidence_score	0.378	FDR	0.782	2	GB_GCA_905233625.1

Detailed_output_by_tool.csv

CRISPRCasFinder

CRISPRCasFinder is an updated, improved, and integrated version of CRISPRFinder and CasFinder.

```
CasFinder/
CRISPRFinderProperties/
GFF/
Prodigal/
rawCRISPRs.fna
rawFASTA/
result.json
TSV/
```

CRISPRFinderProperties_output folder

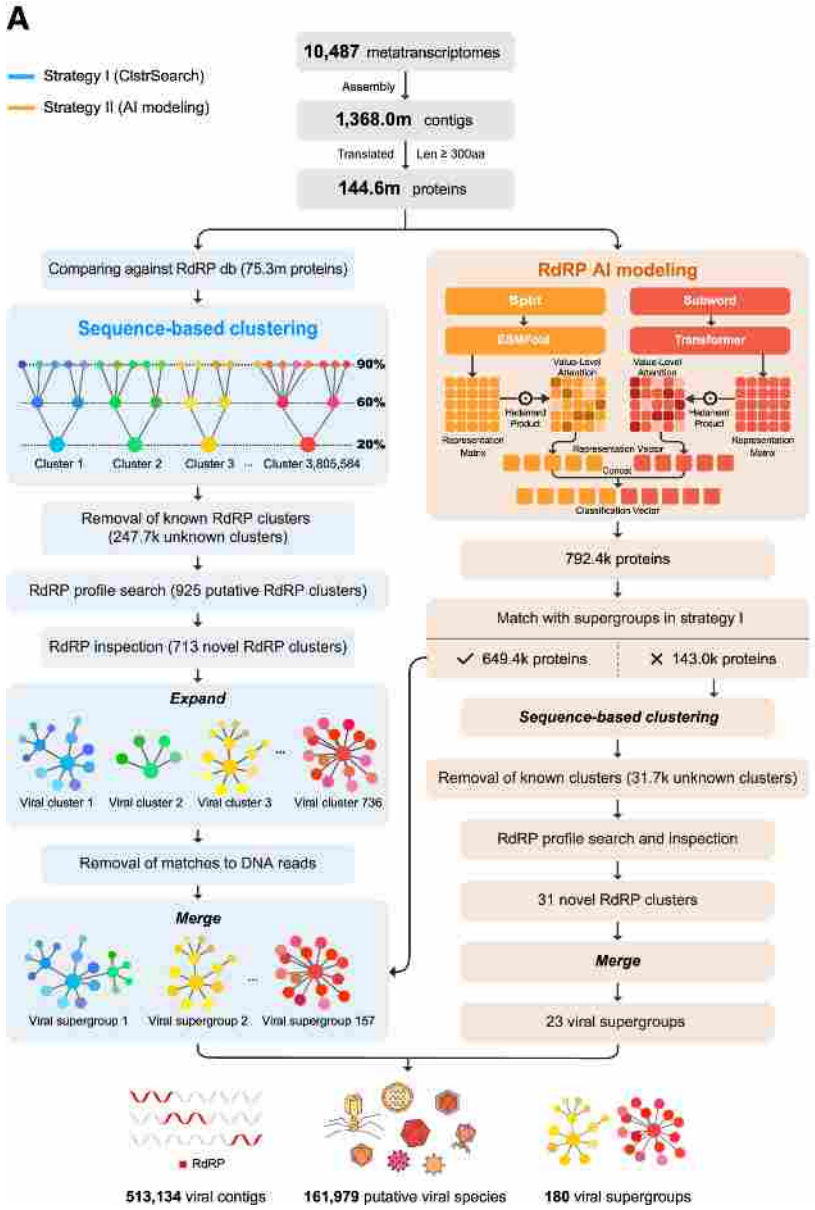
```
casfinder_vOTU_1   casfinder_vOTU_14  casfinder_vOTU_19  casfinder_vOTU_23  casfinder_vOTU_28  casfinder_vOTU_32  casfinder_vOTU_
casfinder_vOTU_10  casfinder_vOTU_15  casfinder_vOTU_2   casfinder_vOTU_24  casfinder_vOTU_29  casfinder_vOTU_33  casfinder_vOTU_
casfinder_vOTU_11  casfinder_vOTU_16  casfinder_vOTU_20  casfinder_vOTU_25  casfinder_vOTU_3   casfinder_vOTU_4   casfinder_vOTU_
casfinder_vOTU_12  casfinder_vOTU_17  casfinder_vOTU_21  casfinder_vOTU_26  casfinder_vOTU_30  casfinder_vOTU_5
casfinder_vOTU_13  casfinder_vOTU_18  casfinder_vOTU_22  casfinder_vOTU_27  casfinder_vOTU_31  casfinder_vOTU_6
```

CasFinder/

```
casfinder_vOTU_1   casfinder_vOTU_14  casfinder_vOTU_19  casfinder_vOTU_23  casfinder_vOTU_28  casfinder_vOTU_32  casfinder_vOTU_7
casfinder_vOTU_10  casfinder_vOTU_15  casfinder_vOTU_2   casfinder_vOTU_24  casfinder_vOTU_29  casfinder_vOTU_33  casfinder_vOTU_8
casfinder_vOTU_11  casfinder_vOTU_16  casfinder_vOTU_20  casfinder_vOTU_25  casfinder_vOTU_3   casfinder_vOTU_4   casfinder_vOTU_9
casfinder_vOTU_12  casfinder_vOTU_17  casfinder_vOTU_21  casfinder_vOTU_26  casfinder_vOTU_30  casfinder_vOTU_5
casfinder_vOTU_13  casfinder_vOTU_18  casfinder_vOTU_22  casfinder_vOTU_27  casfinder_vOTU_31  casfinder_vOTU_6
```

CRISPRFinderProperties/

8. RNA virus_ LocaProt/ LucaVirus



https://doi.org/10.1101/2025.06.14.659722

Files for omics practice

文件 / HKUST_Summer_school_HOU

URL to online drive： https://pan.quark.cn/s/132729f16038

Scan to download

＋ 上传文件 | 新建文件夹 | BT/磁力链下载

文件名	大小
Part01_Summer_School_16S_Tutorial.zip	2.4GB
Part00_VirtualBoxVM.zip	5.9GB
Part02_Summer_School_Metagenome_Tutorial.zip 上次	5.8GB
Part03_Summer_School_Virome_Test_Files.zip	24.6GB
Part03_Summer_School_Virome_Scripts.zip	10.6KB

Meet your classmates

海洋科学与工程系
Department of Ocean Science and Engineering



SUSTech Southern University of Science and Technology

THANK YOU !