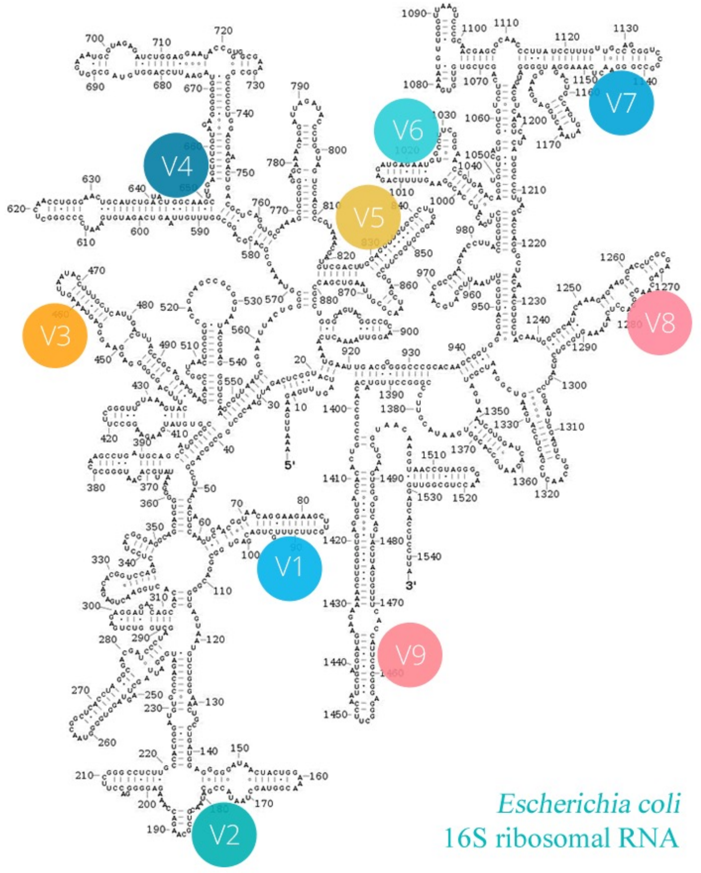
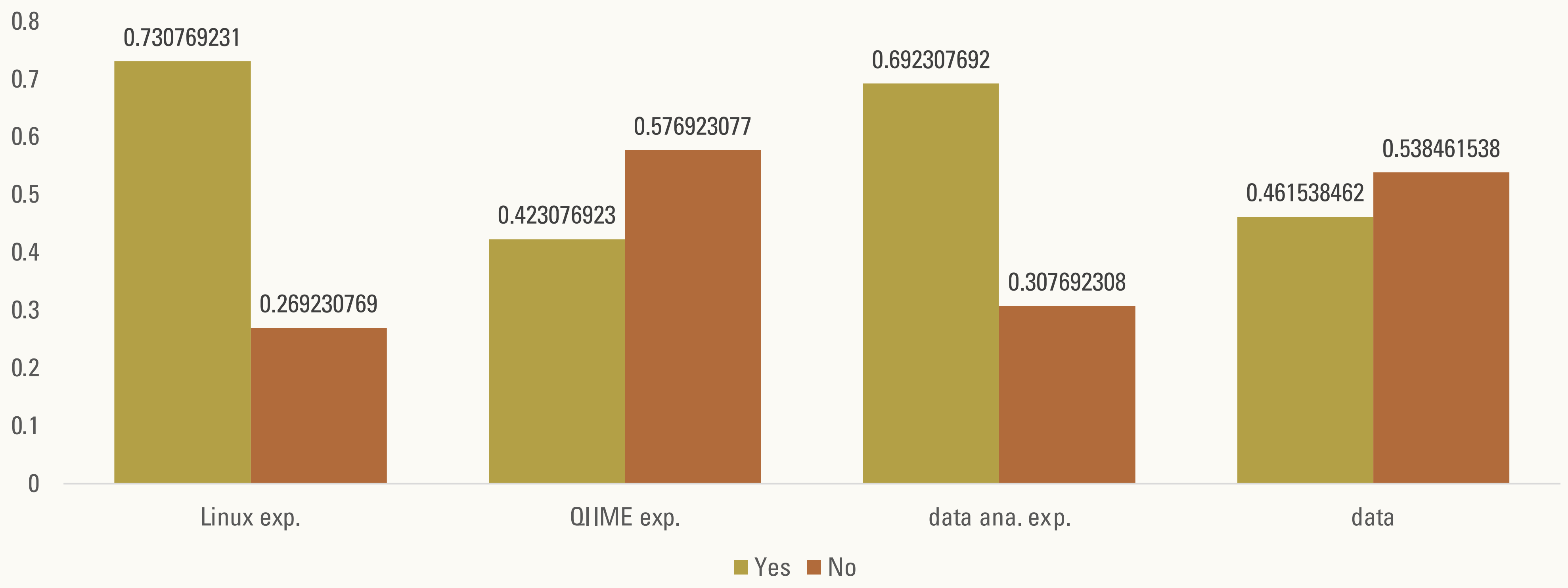
Microbiome pipelines : 16S sequencing pipeline

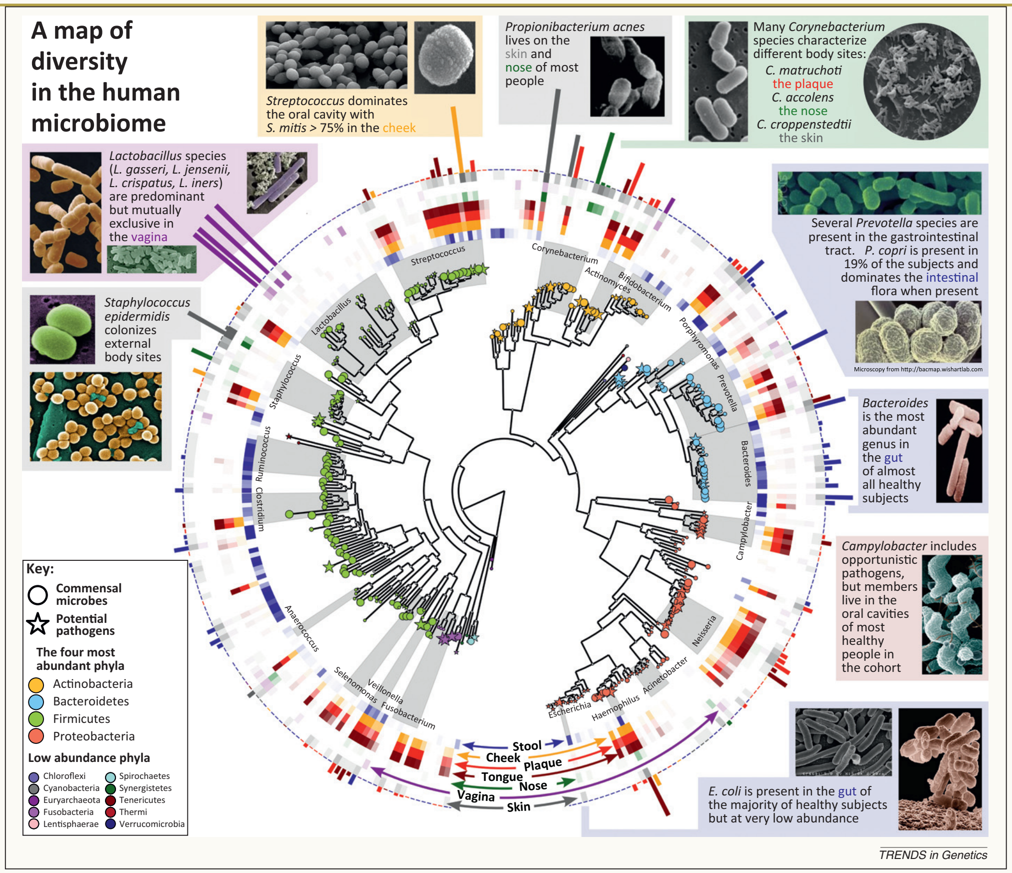
jianhong/16s_pipeline



stats

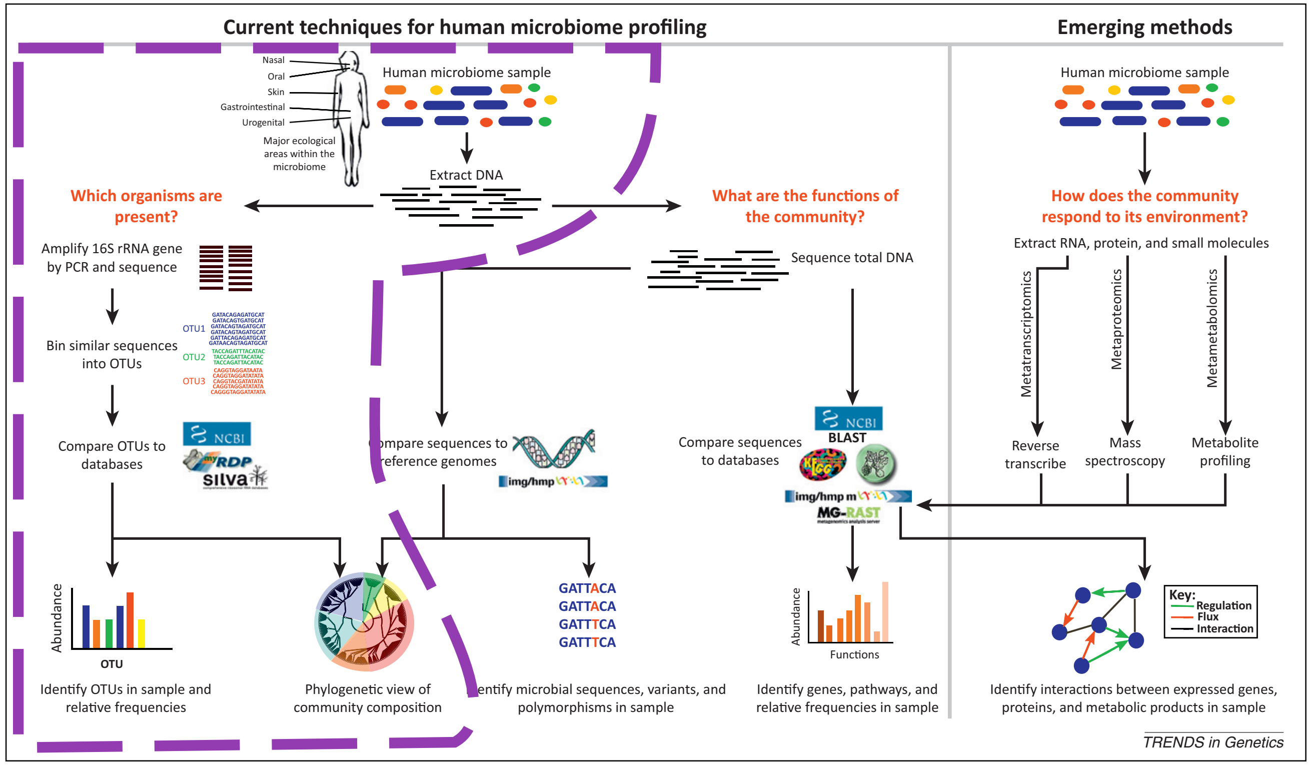

0.8
0.7
0.6
0.5
0.4
0.3
0.2
0.1
0
0.730769231
0.269230769
0.423076923
0.576923077
0.692307692
0.307692308
0.461538462
0.538461538
Linux exp.
QIIME exp.
data ana. exp.
data
Yes
No

Metagenomics


A map of diversity in the human microbiome
Streptococcus dominates the oral cavity with S. mitis > 75% in the cheek
Propionibacterium acnes lives on the skin and nose of most people
Many Corynebacterium species characterize different body sites:
C. matruchoti the plaque
C. accolens the nose
C. croppenstedtii the skin
Lactobacillus species (L. gasseri, L. jensenii, L. crispatus, L. iners) are predominant but mutually exclusive in the vagina
Several Prevotella species are present in the gastrointestinal tract. P. copri is present in 19% of the subjects and dominates the intestinal flora when present
Microscopy from http://bacmap.wishartlab.com
Staphylococcus epidermidis colonizes external body sites
Bacteroides is the most abundant genus in the gut of almost all healthy subjects
Campylobacter includes opportunistic pathogens, but members live in the oral cavities of most healthy people in the cohort
E. coli is present in the gut of the majority of healthy subjects but at very low abundance
Streptococcus
Corynebacterium
Actinomyces
Bifidobacterium
Porphyromonas
Prevotella
Bacteroides
Campylobacter
Neisseria
Acinetobacter
Haemophilus
Escherichia
Fusobacterium
Veillonella
Selenomonas
Anaerococcus
Clostridium
Ruminococcus
Staphylococcus
Lactobacillus
Stool
Cheek
Plaque
Tongue
Nose
Vagina
Skin
Key:
Commensal microbes
Potential pathogens
The four most abundant phyla
Actinobacteria
Bacteroidetes
Firmicutes
Proteobacteria
Low abundance phyla
Chloroflexi
Cyanobacteria
Euryarchaeota
Fusobacteria
Lentisphaerae
Spirochaetes
Synergistetes
Tenericutes
Thermi
Verrucomicrobia
TRENDS in Genetics

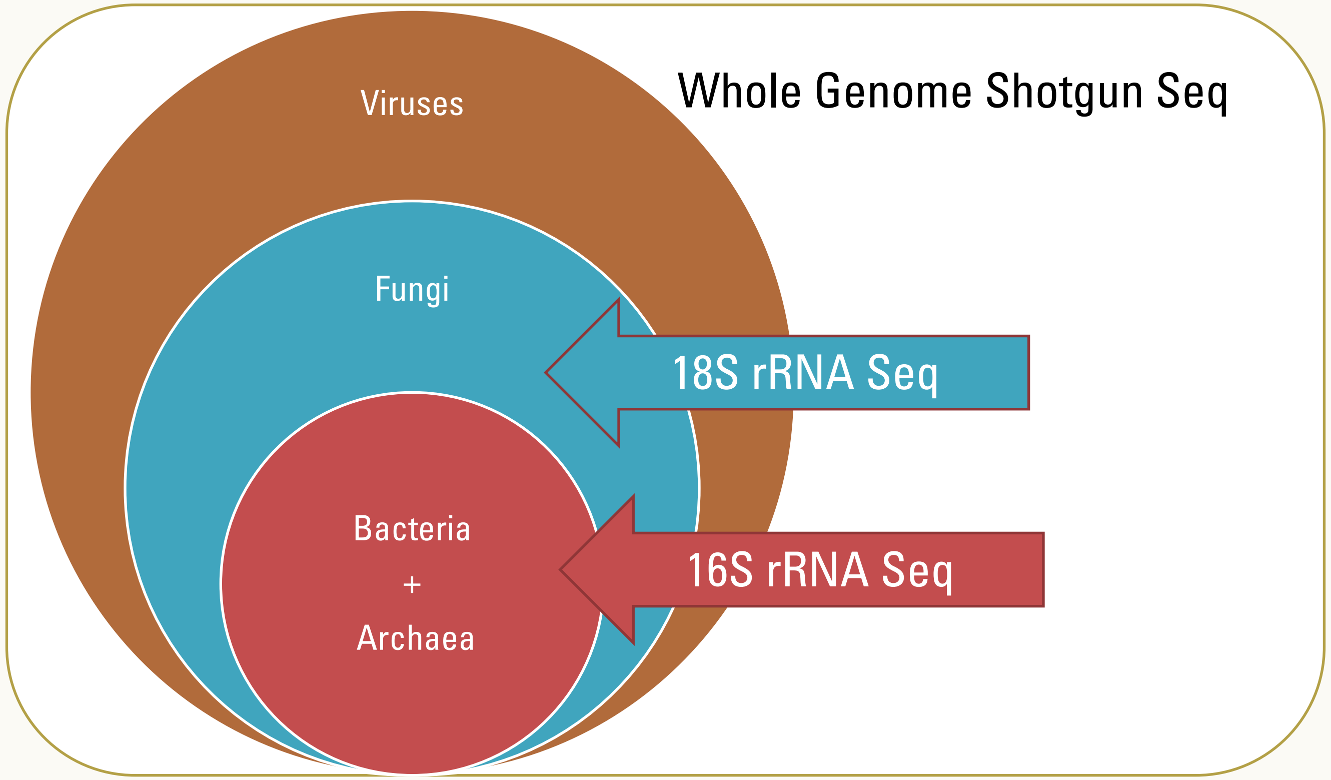
TRENDS IN GENETICS 2013 2951-58DOI: (10.1016/J.TIG.2012.09.005)

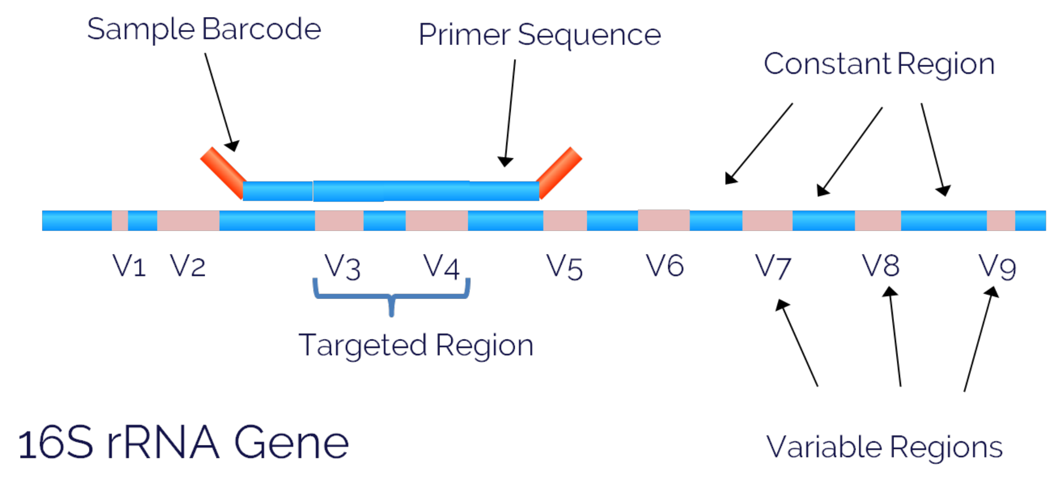
Microbiome profiling


Current techniques for human microbiome profiling
Emerging methods
Human microbiome sample
Nasal
Oral
Skin
Gastrointestinal
Urogenital
Major ecological areas within the microbiome
Extract DNA
Which organisms are present?
What are the functions of the community?
How does the community respond to its environment?
Extract RNA, protein, and small molecules
Amplify 16S rRNA gene by PCR and sequence
Sequence total DNA
Bin similar sequences into OTUs
OTU1
OTU2
OTU3
Metatranscriptomics
Metaproteomics
Metametabolomics
Compare OTUs to databases
Compare sequences to reference genomes
Compare sequences to databases
NCBI
RDP
SILVA
img/hmp
BLAST
MG-RAST
Reverse transcribe
Mass spectroscopy
Metabolite profiling
Abundance
OTU
GATTACA
GATTACA
GATTTCA
GATTTCA
Functions
Key:
Regulation
Flux
Interaction
Identify OTUs in sample and relative frequencies
Phylogenetic view of community composition
Identify microbial sequences, variants, and polymorphisms in sample
Identify genes, pathways, and relative frequencies in sample
Identify interactions between expressed genes, proteins, and metabolic products in sample
TRENDS in Genetics

TRENDS IN GENETICS 2013 2951-58DOI: (10.1016/J.TIG.2012.09.005)

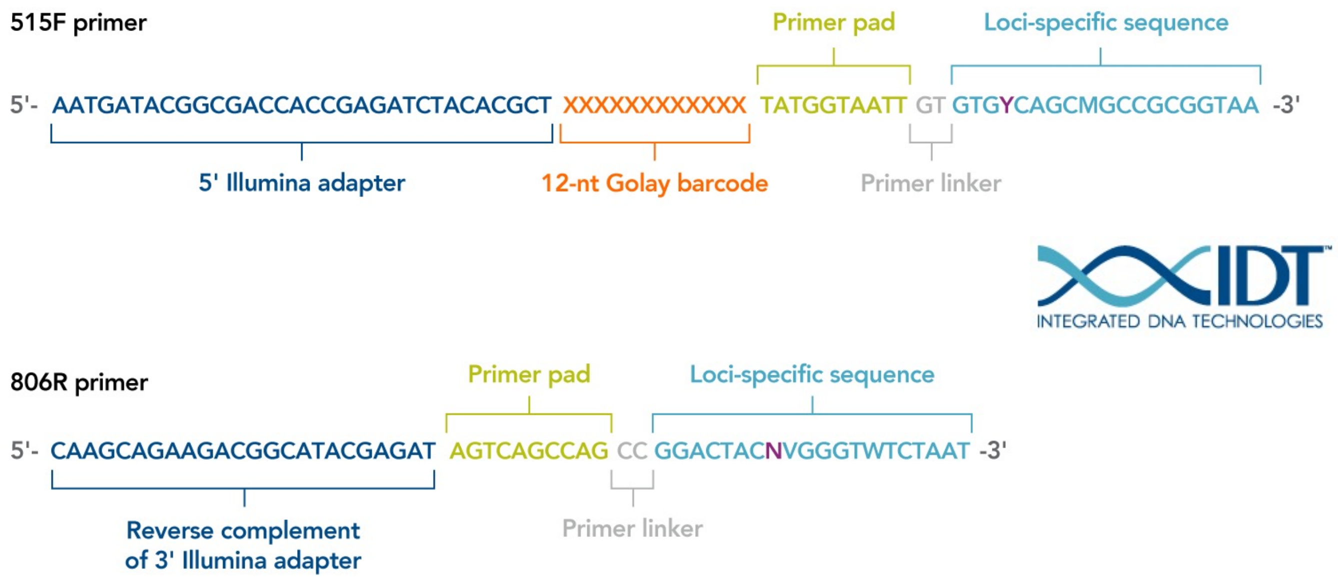
Microbiome

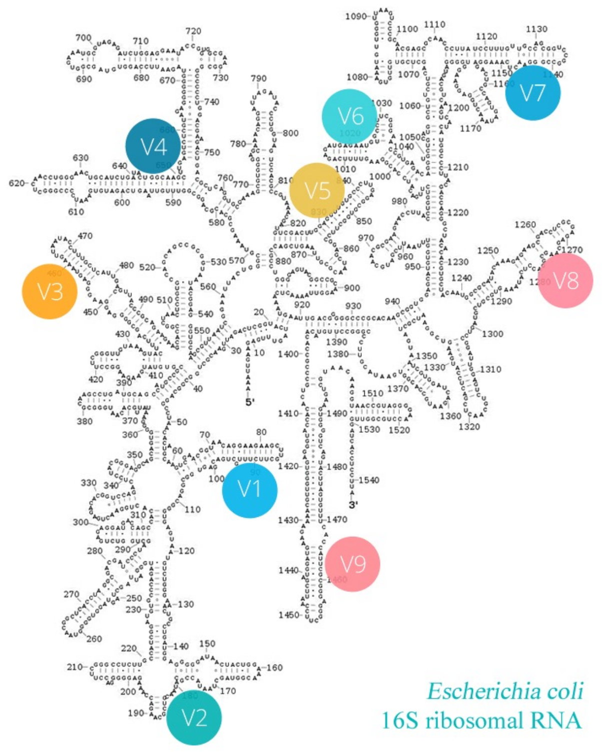

Viruses
Whole Genome Shotgun Seq
Fungi
18S rRNA Seq
Bacteria
+
Archaea
16S rRNA Seq



16S rRNA-seq

- Template relatively short
  - 1.5 kb
- Highly conserved and well studied
  - Amplified by same primers
- Genus-level resolution
  - Closely-related species have a high sequence similarity across the 16S gene


515F primer
Primer pad
Loci-specific sequence
5'- AATGATACGGCGACCACCGAGATCTACACGCT XXXXXXXXXXXX TATGGTAATT GT GTGYCAGCMGCCGCGGTAA -3'
5' Illumina adapter
12-nt Golay barcode
Primer linker
806R primer
Primer pad
Loci-specific sequence
5'- CAAGCAGAAGACGGCATACGAGAT AGTCAGCCAG CC GGACTACNVGGGTWTCTAAT -3'
Reverse complement of 3' Illumina adapter
Primer linker



HTTPS://WWW.IDTDNA.COM/PAGES/EDUCATION/DECODED/ARTICLE/16S-RRNA-INDEXED-PRIMERS-AMPLIFY-PHYLOGENIC-MARKERS-FOR-MICROBIOME-SEQUENCING-ANALYSIS

primers

Primer name	Primer sequence	Reference
515f Original	GTGCCAGCMGCCGCGGTAA	Caporaso et al.
806r Original	GGACTACHVGGGTWTCTAAT	Caporaso et al.
515f Modified	GTG**Y**CAGCMGCCGCGGTAA	Parada et al.
806r Modified	GGACTAC**N**VGGGTWTCTAAT	Apprill et al.
926r	CCGYCAATTYMTTTRAGTTT	Parada et al.
ITS1f	CTTGGTCATTTAGAGGAAGTAA	Gardes and Bruns
ITS2	GCTGCGTTCTTCATCGATGC	White et al.

4/27/22

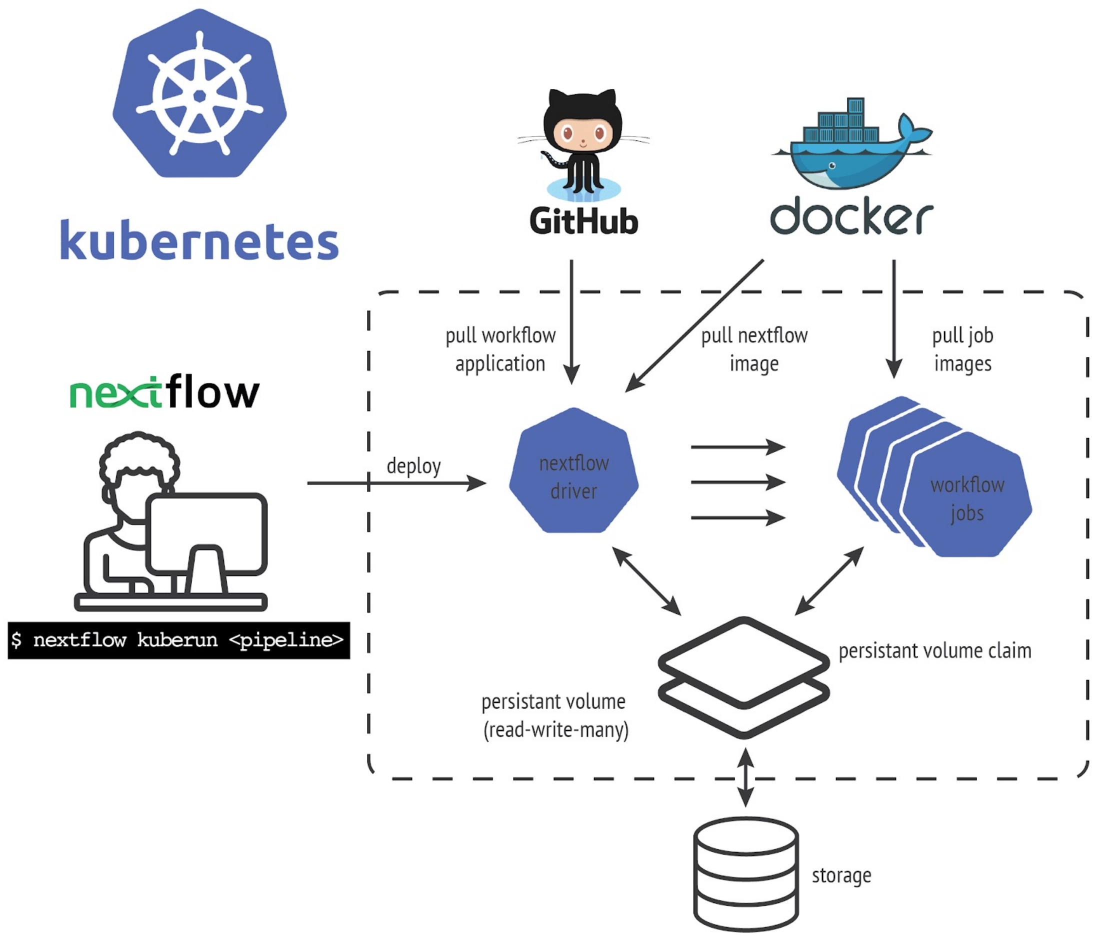
https://doi.org/10.1128/mSystems.00009-15

Note

- There are multiple different pipelines, such as mothur, QIIME, DADA2, and etc.
- Basic steps: sequence trimming -> demultiplexing -> Chimera filtering -> Sequence classification
- There are multiple available 16S databases, and the used database will affect the taxonomic classification.

Current pipeline

BCL2fastq	• bcl2fastq
Trimming	• Trimmomatic
demultiplex	• fastq_pair_filter.py • QIIME2::demux
Filter reads	• DADA2
classification	• DADA2



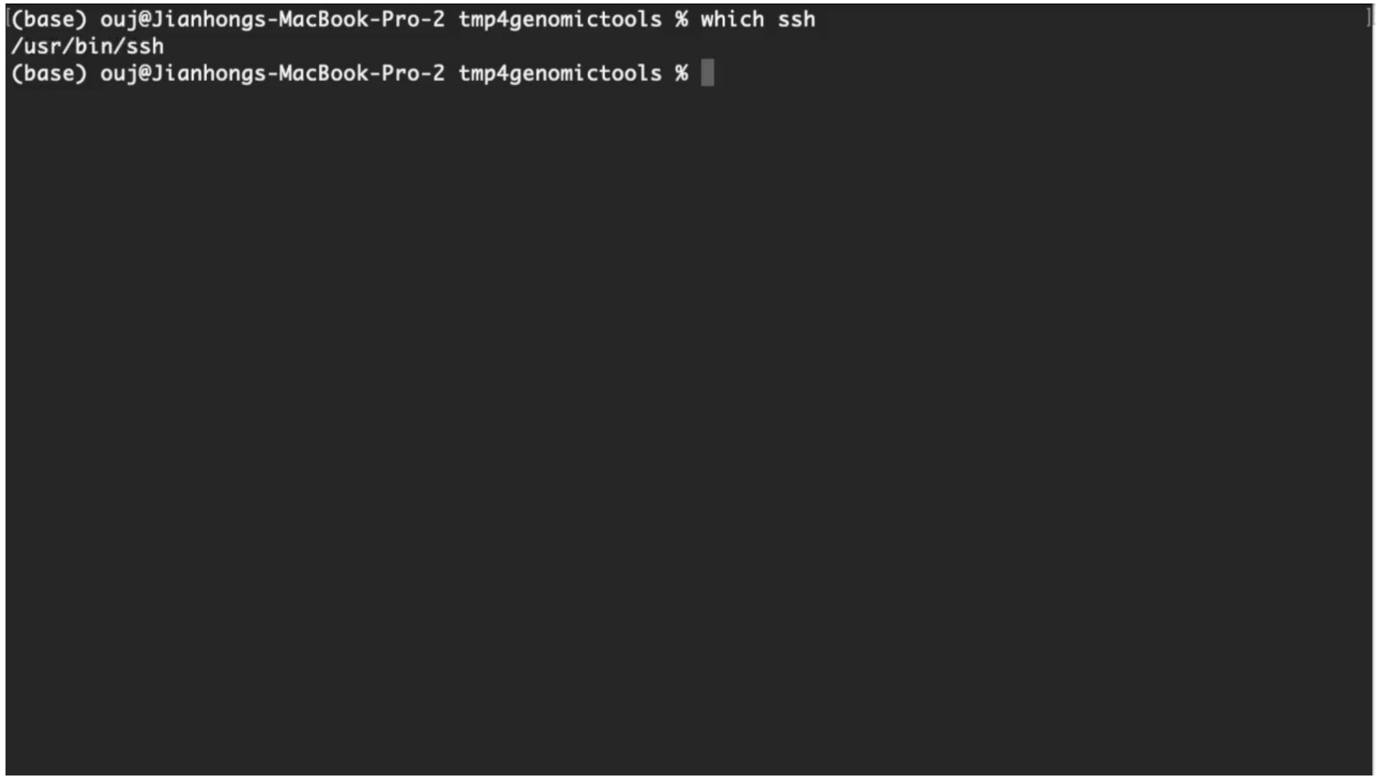
```
$ nextflow kuberun <pipeline>
```

nextflow

- Track progress tracking
  - Troubleshooting
  - Version control
- Module based design
- Portable and scalable
- Highly reproducible
- Compatible with cluster/cloud computation

Why CONDA®

- Conda is an open-source package management system and environment management system
- It does not require root privilege
- Cons:
  - Not stable as Docker/Singularity
  - Version conflicts
  - Storage space



```
(base) ouj@Jianhongs-MacBook-Pro-2 tmp4genomictools % which ssh
/usr/bin/ssh
(base) ouj@Jianhongs-MacBook-Pro-2 tmp4genomictools %
```

3 Steps to Run jianhong/16S_pipeline

- Install `conda`
- Install `nextflow`
- Run pipeline

https://youtu.be/XiVnM5iptUI

Install `MiniConda`

- wget -O minicoda.sh "https://repo.anaconda.com/miniconda/Miniconda3-latest-Linux-x86_64.sh"
- bash miniconda.sh
- vim ~/.condarc (The order is important)

```
channels:
  - conda-forge
  - bioconda
  - defaults
```

Test command are available at: /work/jo117/16S_pipeline/test.run.command.txt

Create `nextflow` environment

- conda create -y --name nextflow bioconda::nextflow=21.10.6

Test run

- Create an interactive job in DCC
  - srun --mem 10G -c 2 --pty bash -i
  - conda activate nextflow
  - nextflow run jianhong/16S_pipeline -r main -profile conda,test
  - To save time for testing:
    - --silva_nr99 '/work/jo117/16S_pipeline/silva_nr99_v138.1_train_set.fa.gz'
    - --silva_tax ' /work/jo117/16S_pipeline/silva_species_assignment_v138.1.fa.gz’

Test run by submit it as a job

- Create profile config file named as profile.config

```
// submit by slurm
process.executor = "slurm"
process.clusterOptions = "-J ProjectName"
params {
max_cpus =2
max_memory = '6.GB'

// Input data
input = "${projectDir}/assets/test_data”
skip_bcl2fastq = true
barcodes = "${projectDir}/assets/barcodes.tsv"
metadata = "${projectDir}/assets/metadata.csv"
// report email
email = 'your@email.addr'
}
```

- Create a slurm script file named as microbiome.sh

```
#!/bin/bash
#SBATCH -J 16S_submitter #jobname
#SBATCH -o microbiome.out.%A_%a.txt
#SBATCH -e microbiome.err.%A_%a.txt
#SBATCH --mem-per-cpu=10G #memory for the job submission
node
#SBATCH -c 1 # 1 CPU is good enough


source ${HOME}/.bashrc
conda activate nextflow

nextflow run jianhong/16S_pipeline -r main -profile conda -c
profile.config -resume
```

```
sbatch microbiome.sh
```

Run pipeline for your own data

- Create profile config file named as profile.config

```
// submit by slurm
process.executor = "slurm"
process.clusterOptions = "-J ProjectName"
params {
// Input data
input = 'path/to/your/initialFiles' // replace it by your own folder contain
Intensities folder.
barcodes = 'path/to/your/barcodes.tsv'
metadata = 'path/to/your/metadata.csv'

// report email
email = 'your@email.addr'
}
```

- Create a slurm script file named as microbiome.sh

```
#!/bin/bash
#SBATCH -J 16S_submitter #jobname
#SBATCH -o microbiome.out.%A_%a.txt
#SBATCH -e microbiome.err.%A_%a.txt
#SBATCH --mem-per-cpu=20G #memory for the job submission node
#SBATCH -c 1 # 1 CPU is good enough

mkdir -p tmp
export TMPDIR=${PWD}/tmp
export TMP=${PWD}/tmp
export TEMP=${PWD}/tmp
source ${HOME}/.bashrc
conda activate nextflow
module load bcl2fastq/2.20

nextflow run jianhong/16S_pipeline -r main -profile conda -c profile.config
-resume
```

```
sbatch microbiome.sh
```

Parameters: input/output

- --input '[path to raw reads files]'
  - If fastq files are supplied, the names must follow the pattern: _[RI][12]_[0-9]+, eg: sample1_R1_001.fastq.gz
- --barcodes '[path to barcodes tsv file]'
  - Barcodes will be used by QIIME2::demux
  - Sample file can be found at https://github.com/jianhong/16S_pipeline/blob/main/assets/barcodes.tsv
- --metadata '[path to metadata csv file]'
  - Metadata will be used by Bioconductor::phyloseq package, see help https://bioconductor.org/packages/phyloseq/
  - Sample file can be found at https://github.com/jianhong/16S_pipeline/blob/main/assets/metadata.csv

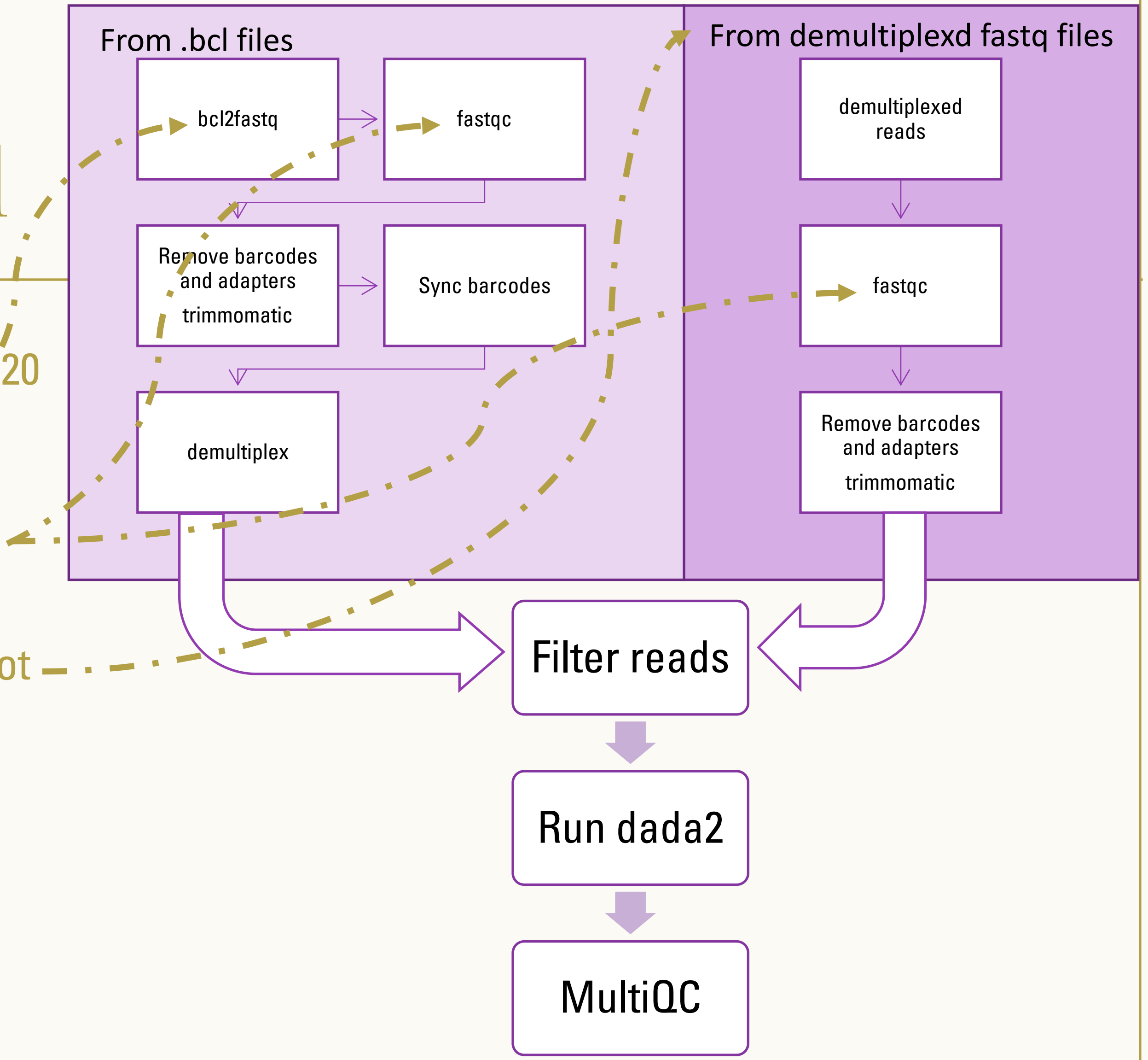
Parameters: 16S reference

- ❑ --silva_nr99: used by assignTaxonomy
- ❑ --silva_tax: used by addSpecies

- The latest release can be found at: https://www.arb-silva.de/download/arb-files/
- The defaults are set as files in https://zenodo.org/record/4587955#.YiZhRpNudJU
- The references will be used by DADA2::assignTaxonomy and addSpecies, see help at https://benjjneb.github.io/dada2/training.html
- They can be replaced by other resources such as RDP and UNITE.

Parameters: pipeline control

- --bcl2fastq: use DCC module bcl2fastq/2.20
- --skip_bcl2fastq: skip bcl2fastq or not
- --skip_fastqc: skip fastqc or not
- --skip_demultiplex: skip demultiplex or not


From .bcl files
bcl2fastq
fastqc
Remove barcodes and adapters trimmomatic
Sync barcodes
demultiplex
From demultiplexd fastq files
demultiplexed reads
fastqc
Remove barcodes and adapters trimmomatic
Filter reads
Run dada2
MultiQC

https://jianhong.github.io/16s_pipeline/parameters.html

Errors

- Exit code: 127: command not found
  - Check the conda environment installation
- Exit code: 137: out of memory
  - Increase the resource requirement
- Exit code: 139: fault installation
  - Check the conda environment installation

Increase the resource requirement

- https://jianhong.github.io/16S_pipeline/usage.html#resource-requests
- The maximal limitation
    - max_cpus, max_memory, max_time in profile.config file, eg: https://github.com/jianhong/16S_pipeline/blob/main/conf/test.config
- Change the requirement by process name in profile.config file.
    - process { withName: BCL2FASTQ { memory = 100.GB } }

Module specific parameters

- All default settings can be found at https://github.com/jianhong/16S_pipeline/blob/main/conf/modules.config
- Eg: for trimmomatic, see help at http://www.usadellab.org/cms/uploads/supplementary/Trimmomatic/TrimmomaticManual_V0.32.pdf

```
withName: REMOVE_PRIMERS {
        ext.args = 'ILLUMINACLIP:techseqs.fa:2:30:10'
        publishDir = [
                path: { "${params.outdir}/1_remove_primers" },
                mode: 'copy',
                saveAs: { filename -> filename.equals('versions.yml') ? null : filename }
        ]
}
```

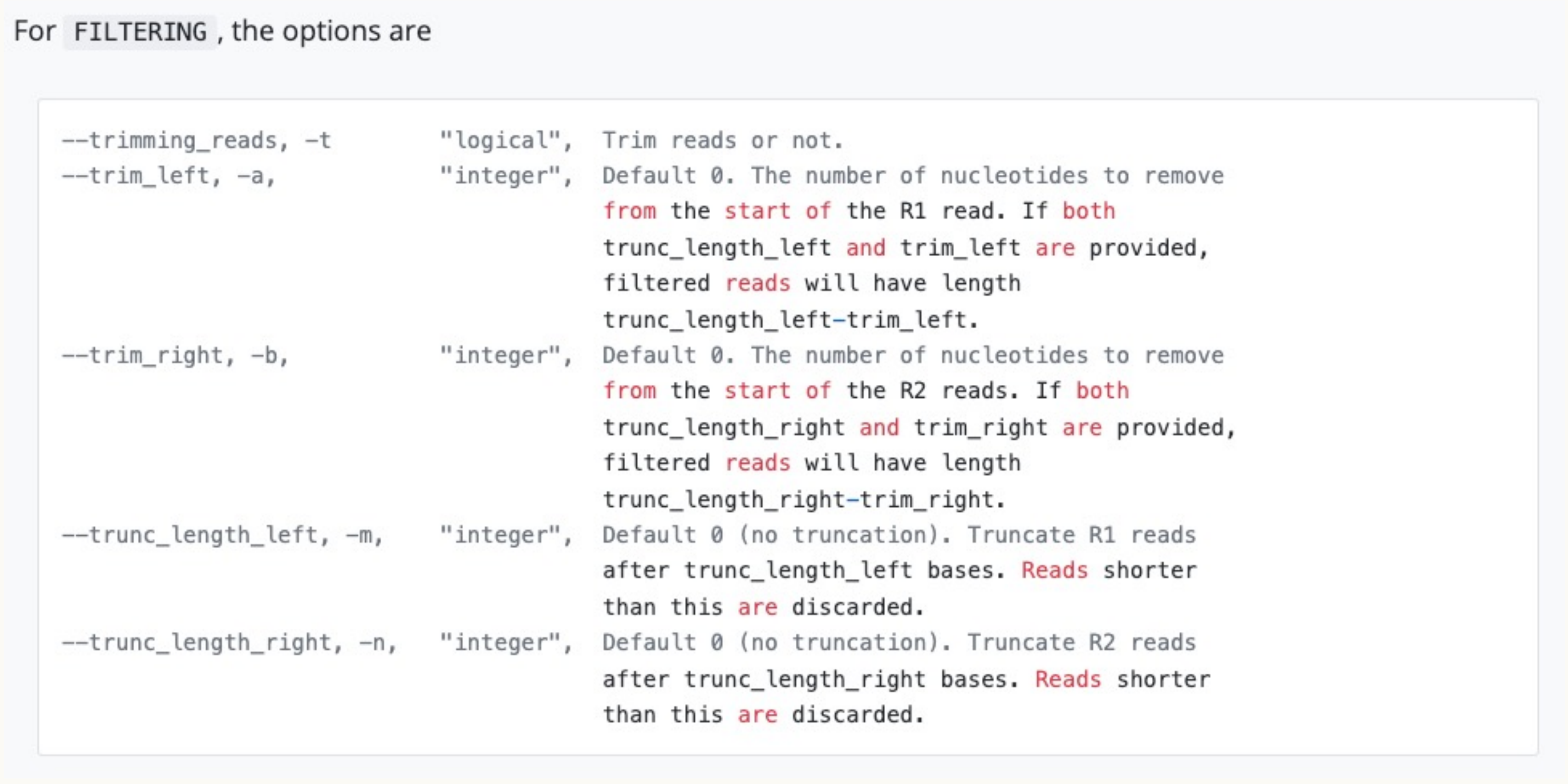
Parameters with demuxing

sample-id	barcode-sequence
#q2:types	categorical
Spinach1	TGTGCGATAACA
Spinach2	GATTATCGACGA
Spinach3	GCCTAGCCCAAT

```
withName: QIIME_DEMUX {
    ext.args   = '--m-barcodes-column barcode-sequence --p-rev-comp-mapping-barcodes '
    publishDir = [
      path: { "${params.outdir}/3_demultiplex" },
      mode: 'copy',
      saveAs: { filename -> filename.equals('versions.yml') ? null : filename }
    ]
  }
```

https://docs.qiime2.org/2022.2/plugins/available/demux/emp-paired/

Trim by dada2::filterAndTrim



For `FILTERING`, the options are

```
--trimming_reads, -t        "logical",  Trim reads or not.
--trim_left, -a,            "integer",  Default 0. The number of nucleotides to remove
                                        from the start of the R1 read. If both
                                        trunc_length_left and trim_left are provided,
                                        filtered reads will have length
                                        trunc_length_left-trim_left.
--trim_right, -b,           "integer",  Default 0. The number of nucleotides to remove
                                        from the start of the R2 reads. If both
                                        trunc_length_right and trim_right are provided,
                                        filtered reads will have length
                                        trunc_length_right-trim_right.
--trunc_length_left, -m,    "integer",  Default 0 (no truncation). Truncate R1 reads
                                        after trunc_length_left bases. Reads shorter
                                        than this are discarded.
--trunc_length_right, -n,   "integer",  Default 0 (no truncation). Truncate R2 reads
                                        after trunc_length_right bases. Reads shorter
                                        than this are discarded.
```

dada2 workflow

1. check PAIRED END by length(filtered forward) == length(filtered reverse)
2. Learn Error Rates by a subset data (36 samples) (Another way learnErrors)
   1. Dereplication by dada2::derepFastq
   2. Sample Inference by dada2::dada(dereps, err=NULL, selfConsist=TRUE, multithread=NCORE)
3. Run `dada` by the error model for all samples
   1. derepFastq → dada(sample, err=errModel, multithread=NCORE)
4. Merge Paired Reads by mergePairs
5. Construct Sequence Table: makeSequenceTable and Trim sequences by sequence lengths.
6. Remove Chimeras: removeBimeraDenovo(seqtab, method='consensus', multithread=NCORE)
7. Assign Taxonomy:
   1. TaxTable <- assignTaxonomy(sequences, TRAIN_SET, tryRC=TRYRC, multithread=NCORE)
   2. addSpecies(TaxTable, SPECIES_ASSIGNMENT, tryRC=TRYRC, verbose=TRUE)

SOURCE: HTTP://BENJJNEB.GITHUB.IO/DADA2_PIPELINE_MV/BIGDATA.HTML

Parameter - - tryRC

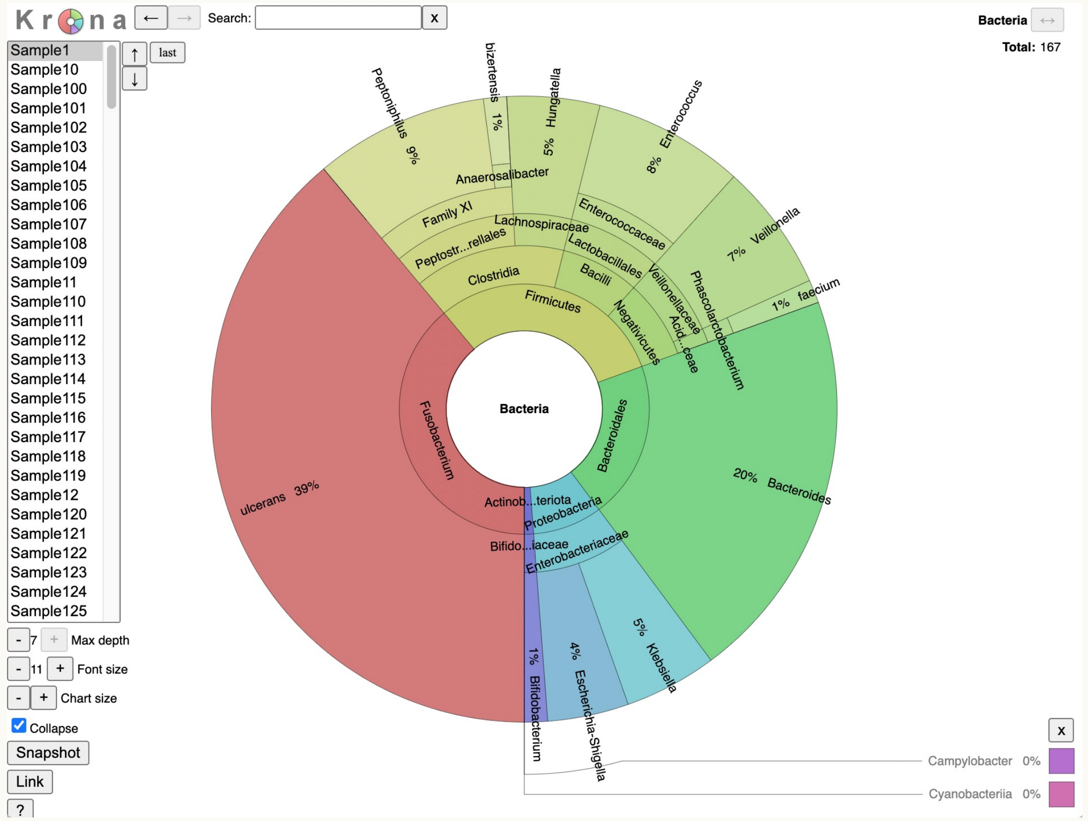
- If you have unmerged reverse-complement reads issue, you may want to try the parameter --tryRC
- It will try to merge the reverse-complement reads counts.
- Please note that partial of this operation is not in DADA2 package. It is used for the reads that not proper demultiplexed.

Modula names and outputs

Modules	program	outputs
BCL2FASTQ	bcl2fastq	0_data_raw/*.fastq.gz
FASTQC	fastqc	fastqc/*_fastqc.{html,zip}
REMOVE_PRIMERS	trimmomatic	1_remove_primers/*.fastq.gz
SYNC_BARCODES	sync_paired_end_reads.py	2_sync_barcodes/*.fastq.gz
QIIME_DEMUX	qiime demux	3_demultiplex/demuxd_reads/*.fastq.gz
FILTERING	dada2::filterAndTrim	4_filter/<sampleid>/*.fastq.gz
DADA2	dada2	5_data2/*
FHYLOSEQ	phyloseq	6_phyloseq/*
KRONA	Krona	7_Krona/*
MULTIQC	multiqc	multiqc/*

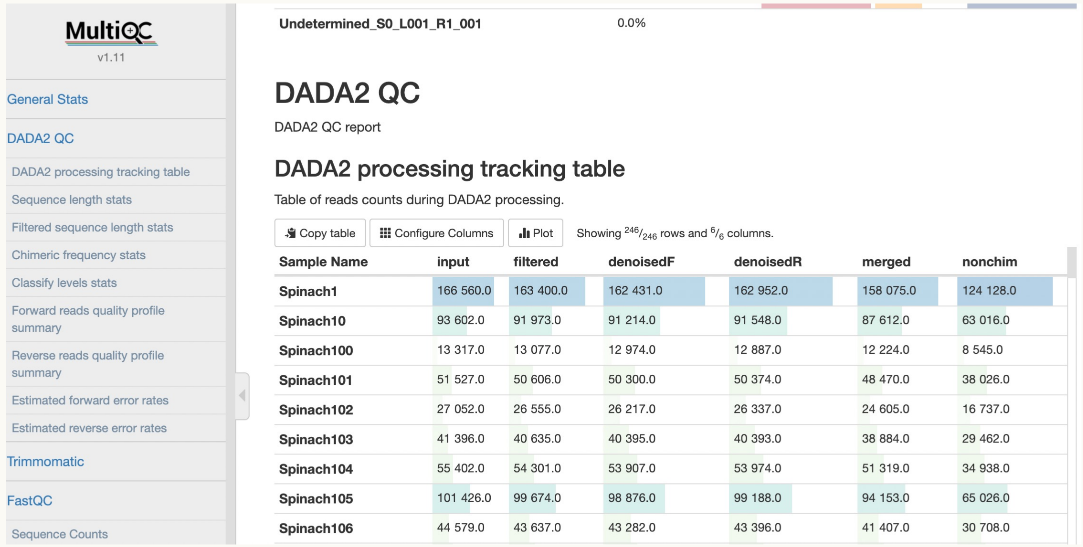
Krona visalization

- https://jianhong.github.io/16S_pipeline/krona.html
- https://github.com/marbl/Krona/wiki



QC reports

- https://jianhong.github.io/16S_pipeline/multiqc.html



MultiQC
v1.11
General Stats
DADA2 QC
DADA2 processing tracking table
Sequence length stats
Filtered sequence length stats
Chimeric frequency stats
Classify levels stats
Forward reads quality profile summary
Reverse reads quality profile summary
Estimated forward error rates
Estimated reverse error rates
Trimmomatic
FastQC
Sequence Counts
Copy table
Configure Columns
Plot

Undetermined_S0_L001_R1_001 0.0%

DADA2 QC

DADA2 QC report

DADA2 processing tracking table

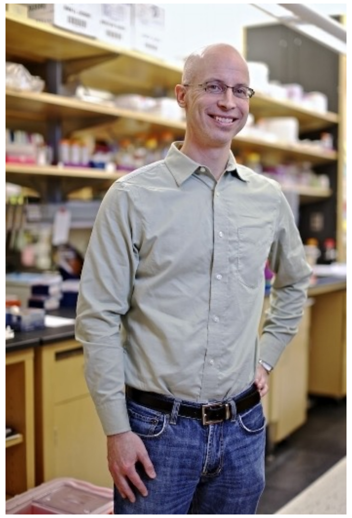
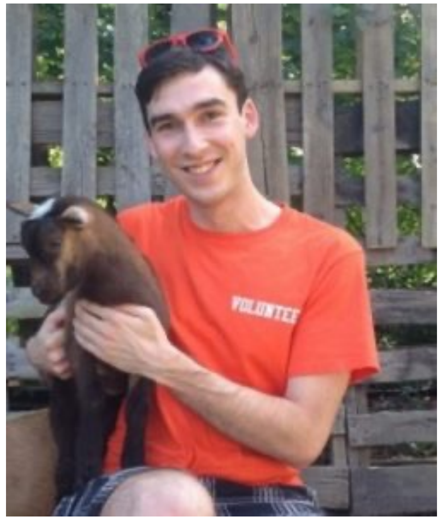
Table of reads counts during DADA2 processing.

Showing $^{246}/_{246}$ rows and $^{6}/_{6}$ columns.

Sample Name	input	filtered	denoisedF	denoisedR	merged	nonchim
Spinach1	166 560.0	163 400.0	162 431.0	162 952.0	158 075.0	124 128.0
Spinach10	93 602.0	91 973.0	91 214.0	91 548.0	87 612.0	63 016.0
Spinach100	13 317.0	13 077.0	12 974.0	12 887.0	12 224.0	8 545.0
Spinach101	51 527.0	50 606.0	50 300.0	50 374.0	48 470.0	38 026.0
Spinach102	27 052.0	26 555.0	26 217.0	26 337.0	24 605.0	16 737.0
Spinach103	41 396.0	40 635.0	40 395.0	40 393.0	38 884.0	29 462.0
Spinach104	55 402.0	54 301.0	53 907.0	53 974.0	51 319.0	34 938.0
Spinach105	101 426.0	99 674.0	98 876.0	99 188.0	94 153.0	65 026.0
Spinach106	44 579.0	43 637.0	43 282.0	43 396.0	41 407.0	30 708.0

Get help

- Online Doc: https://jianhong.github.io/16S_pipeline/
- Email: jianhong.ou@duke.edu
- Report an issue: https://github.com/jianhong/16S_pipeline/issues
- Dcc usage: https://oit-rc.pages.oit.duke.edu/rcsupportdocs/dcc/#cluster-shared-storage-resources-work-and-scratch

Acknowledgements

Duke University Regeneromics @ Regeneration Center

DUKE REGENERATION CENTER