Erweiterbarer, Umweltfreundlicher, Leistungsfähiger ETH-Rechner (euler)

A Mini-Workshop on Bioinformatics for Microbial Metagenomics on the Euler

Feng Ju, Ph D
Department of Surface Waters, Microbial Ecology Group
Eawag Kastanienbaum, Switzerland

Date: 06.07.2018

Location: Raum-KB-Bootshaus

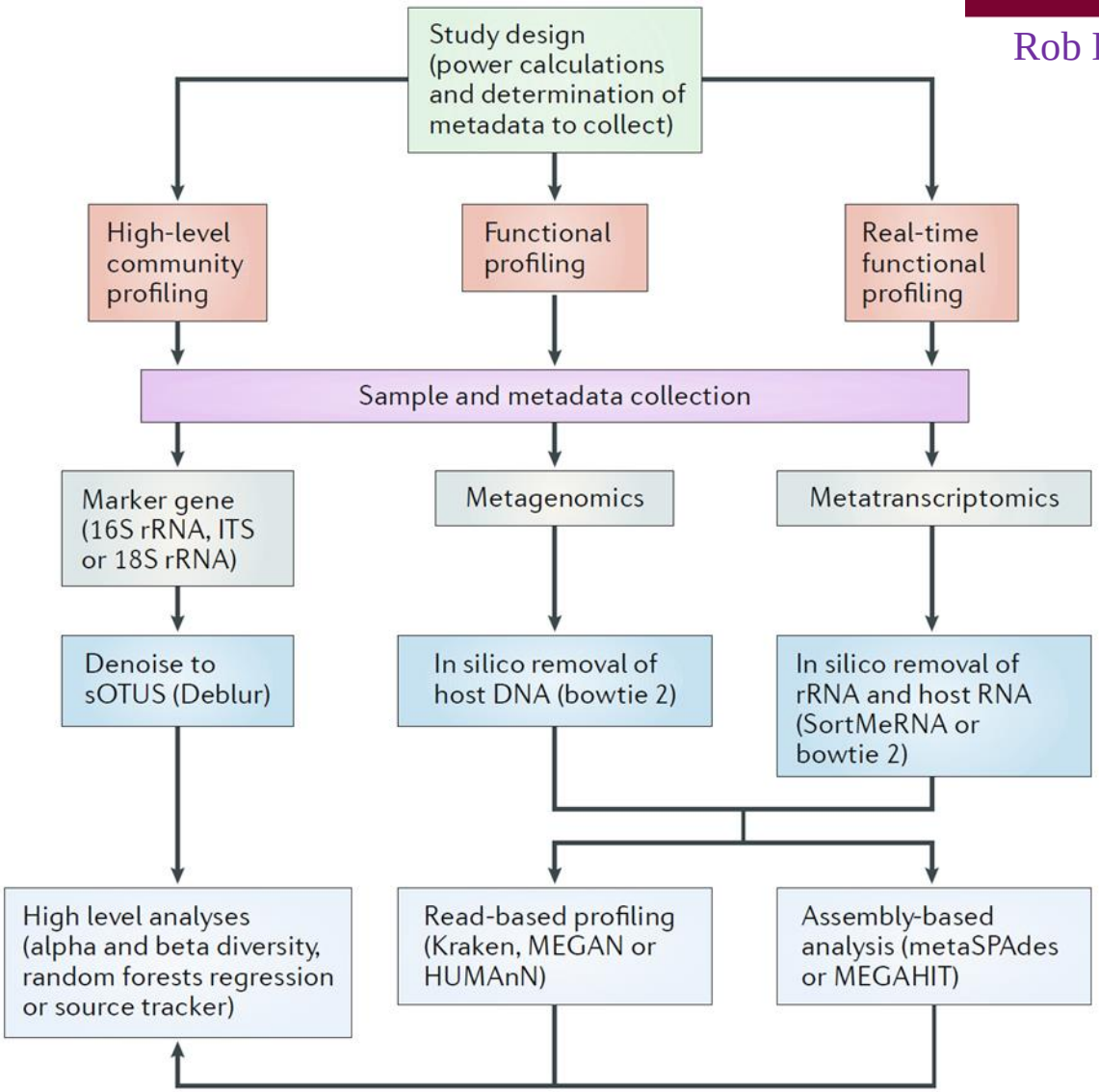
Outlines

- Background for microbial metagenomics
- Bioinformatics Practice on the euler

Best practices for analyzing microbiomes

nature REVIEWS MICROBIOLOGY

Rob Knight et al., 2018



Amplicon data analysis:
DADA2: amplicon sequence variants (ASVs), written in R
Deblur: sub-operational taxonomic units (sOTUs), 10 times faster
Written in python

Metagenome analysis:
Read-based strategy: Bowtie2, Diomond, Metaphylan2, etc.
Assembly-based strategy: IDBA_UD, MEGAHIT, metaSPAdes

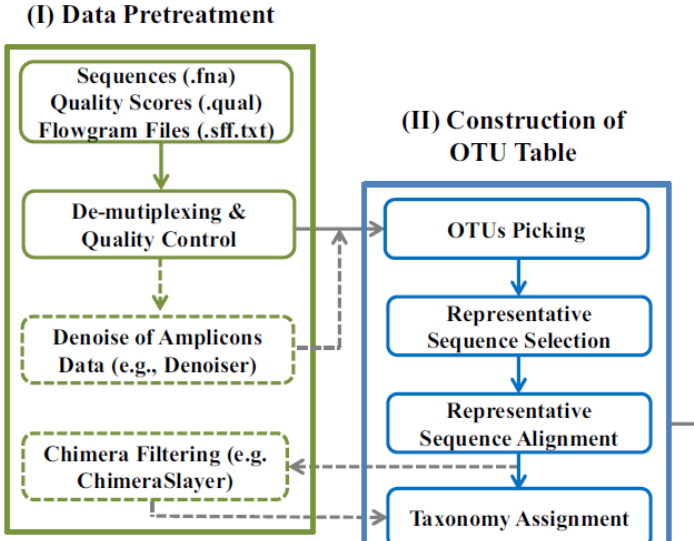
Metatranscriptome analysis:
Read-based strategy: Bowtie2, Metaphylan2, etc.
Assembly-based strategy: IDBA_UD, MEGAHIT, metaSPAdes

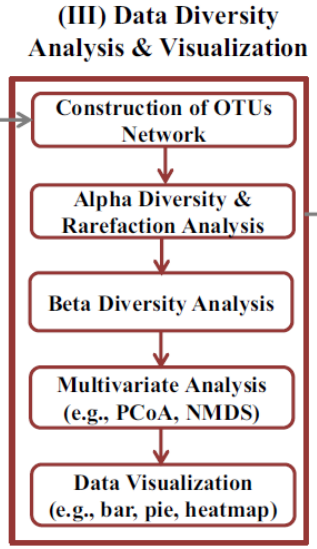
Pros and cons of genomic analyses for evaluating microbial communities

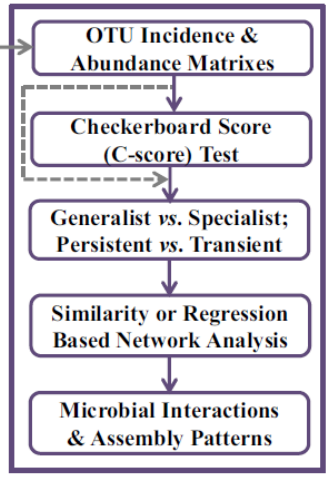
Method	Pros	Cons
Marker gene analysis	• Quick, simple and inexpensive sample preparation and analysis[55,59] • Correlates well with genomic content[37–41] • Amenable to low-biomass and highly host-contaminated samples • Large existing public data sets for comparison[16,55,160]	• No live, dead or active discrimination • Subject to amplification biases[34] • Choice of primers and variable region magnifies biases[33,54,159] • Requires a priori knowledge of microbial community[36] • Resolution typically limited to genus level at best • Appropriate negative controls required • Functional information is limited[39,40]
Whole metagenome analysis	• Can directly infer the relative abundance of microbial functional genes; microbial taxonomic and phylogenetic identity to species and strains level is attainable for known organisms[42] • Does not assume knowledge of microbial community (that is, captures phages, viruses, plasmids, microbial eukaryotes, etc.) • No PCR-related biases • Can estimate in situ growth rates for target organisms with sequenced genomes[161] • Can allow assembly of population-averaged microbial genomes[43,162] • Can be mined for novel gene families	• Relatively expensive, laborious and complex sample preparation and analysis • Contamination from host-derived DNA and organelles may obscure microbial signatures • Viruses and plasmids are not typically well annotated by default pipelines • Deep sequencing depths are typically required relative to other methods • No live, dead or active discrimination • Population-averaged microbial genomes tend to be inaccurate owing to assembly artefacts
Metatranscriptome analysis	• Can estimate which microorganisms in a community are actively transcribing when paired with marker gene analysis • Inherently discriminates between active live organisms versus dormant or dead microorganisms and extracellular DNA • Captures dynamic intra-individual variation[51] • Directly evaluates microbial activity, including responses to intervention and event exposure[52]	• Most expensive, laborious and complex sample preparation and analysis[163] • Host mRNA contamination and rRNA must be removed[48,164,165] • Requires careful sample collection and storage • Data are biased towards organisms with high transcription rates • Requires paired DNA sequencing to decouple transcription rates from bacterial abundance changes

Rob Knight et al., 2018.
Nature Reviews Microbiology

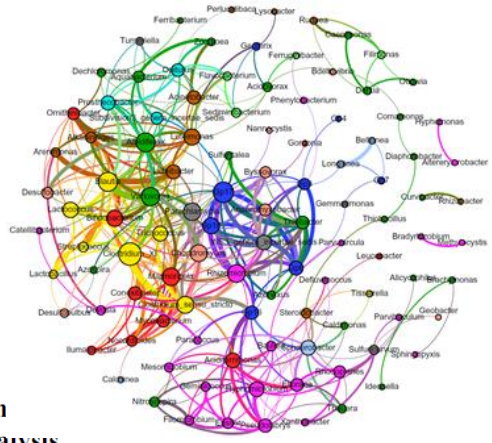
16S rRNA gene amplicon NGS data mining of microbial diversity & interactions



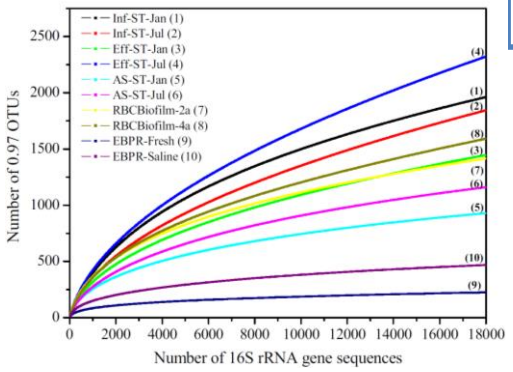




Microbial co-occurrence network

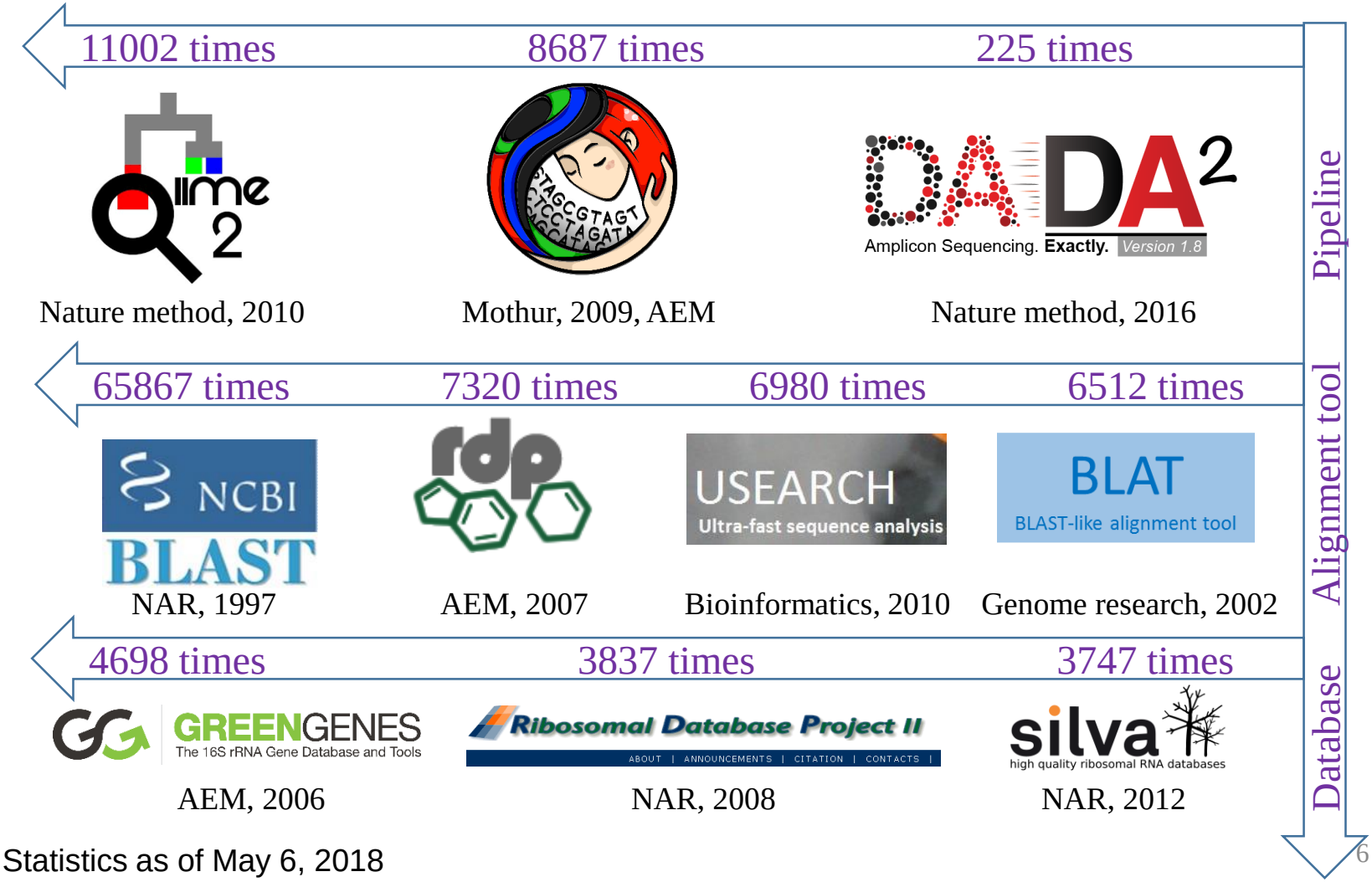


Microbial richness in WWTP habitats



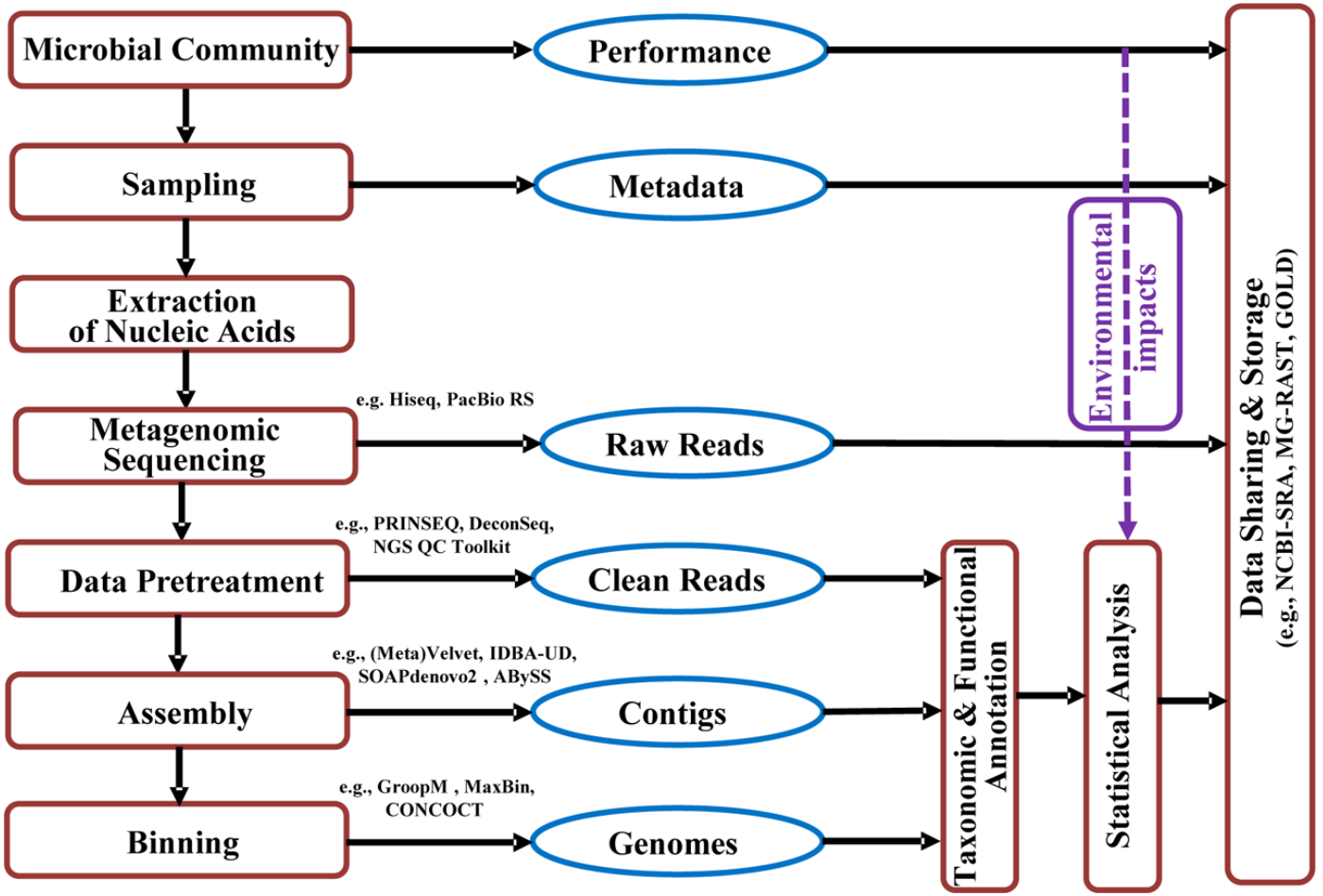
Ju F and Zhang T, 2015. 16S rRNA gene high-throughput sequencing data mining of microbial diversity and interactions, AMB. 99(10): 4119–4129

Google scholar citation of 16S amplicon NGS data analysis tools and databases



Statistics as of May 6, 2018

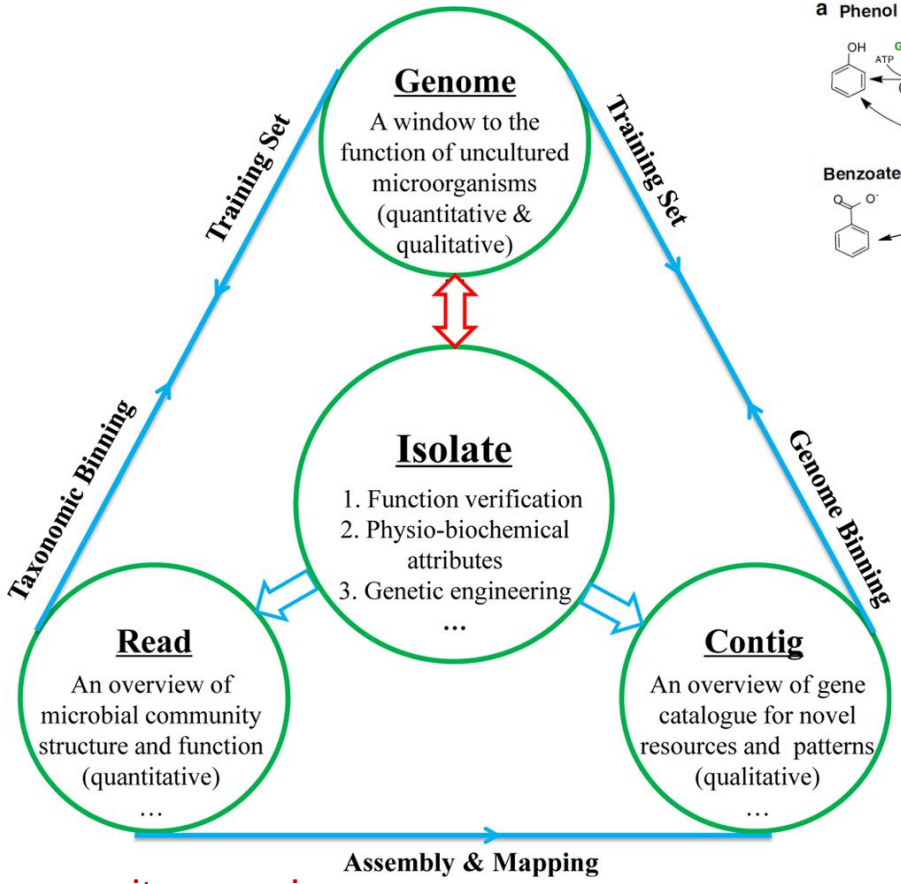
NGS-based shot-gun metagenomic survey of microbial ecosystem function



Reference: Ju F, Zhang T*. 2015. Experimental design and bioinformatics analysis for the application of metagenomics in environmental sciences and biotechnology. Environmental Science & Technology. 49(21), 12628-12640

A self-accelerating data mining circle in an era of NGS-based metagenomcis

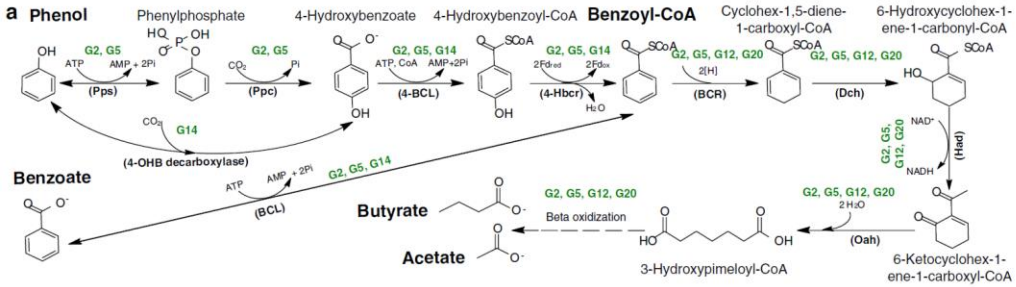
Organism overview



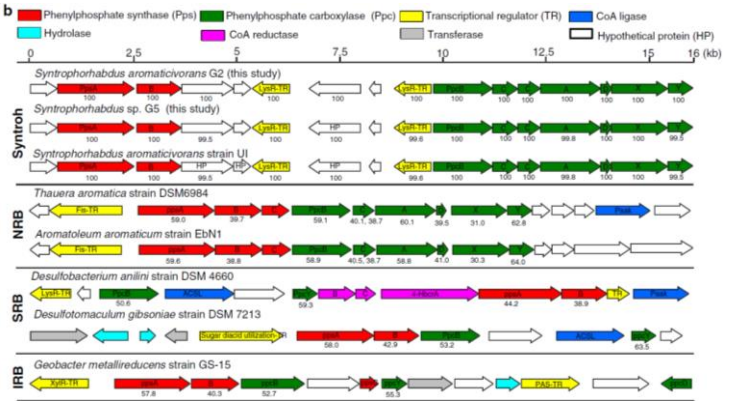
Community overview

Gene overview

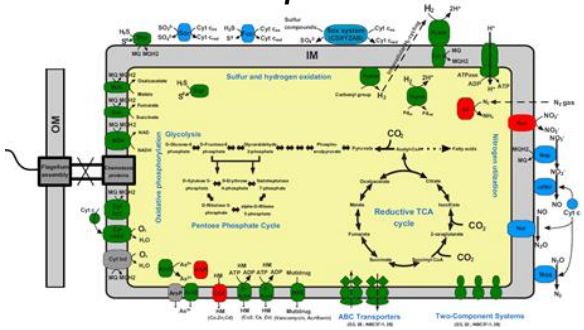
a) Phenol-degrading pathways in uncultured Syntrophorhabdus & Cryptanaerobacte



b) Pps-Ppc operons in anaerobic degraders of phenolic compounds



c) Central metabolism of *Sulfurovum*-like ε-Proteobacterium G1.



Ju F and Zhang T. 2015. Environmental Science & Technology. 51 (7), 3982-3992; Ju F et al., 2018. Biotechnology for Biofuels. 11:135

Shot-gun metagenomic data analysis platforms, tools and database resources

Table 1. Platforms and Software Tools Available for the Bioinformatics Analysis of Metagenomes

data	platform/software	description
Pretreatment	MG-RAST	DMP, QC, DR
	CLC bio	ALR, DMP, QC, DR, OL
	IMG/M system	QC, DR
	PRINSEQ	QC, DR, summary statistics
	NGS QC Toolkit	ALR, QC
	DeconSeq	DNA contamination removal
	FASTX-Toolkit	ALR, DMP, QC
Assembly	Velvet	genome assembler
	ABySS	
	SOAPdenovo2	
	CLC bio	genome/metagenome assembler
	IDBA-UD	metagenome assembler
	MetaVelvet	
	Ray Meta	
	Omega	
	MEGAHIT	
Binning	GroopM	genome reconstruction
	CONCOCT	
	MaxBin	
	METABAT	
	PhyloPythiaS	composition-based taxonomic binning/assignment
	TETRA	
	CompostBin	
	TACAO	
	MetaPhlAn2	homology-based taxonomic binning/assignment
	MetaPhyler	
	PhymmBL	composition & homology-based taxonomic binning/assignment
	MetaCluster	

- ExPASy: Bioinformatics Resource Portal: https://www.expasy.org/
- OBRC: Online Bioinformatics Resources Collection: https://www.hsls.pitt.edu/obrc/
- Nucleic Acids Research Database: http://www.oxfordjournals.org/our_journals/nar/database/c/
- Nucleic Acids Research Web Server Issue: https://academic.oup.com/nar/issue/43/W1
- OmicsTools: https://omictools.com/
- GenomeNet: https://www.genome.jp/en/gn_tools.html

Ju F and Zhang T. 2015. Experimental design and bioinformatics analysis for the application of metagenomics in environmental sciences and biotechnology. Environmental Science & Technology. 51 (7), 3982-3992

Outline for Bioinformatics Practice on the euler

- 1) Make data subsets of lake water sample D10
- 2) Make subfolders and sample lists for the datasets
- 3) Check and activate modules and commands
- 4) Data QC for DNA-seq and RNA-seq
- 5) *De novo* assembly of metagenomes
- 6) In-house scripts for assemblies statistics
- 7) Gene prediction from contigs
- 8) Reads mapping to contigs and genes with Bowtie2
- 9) Genes/Reads annoation using BLAST
- 10) Run a jobarray using BLAST as an example

Software to install for euler access

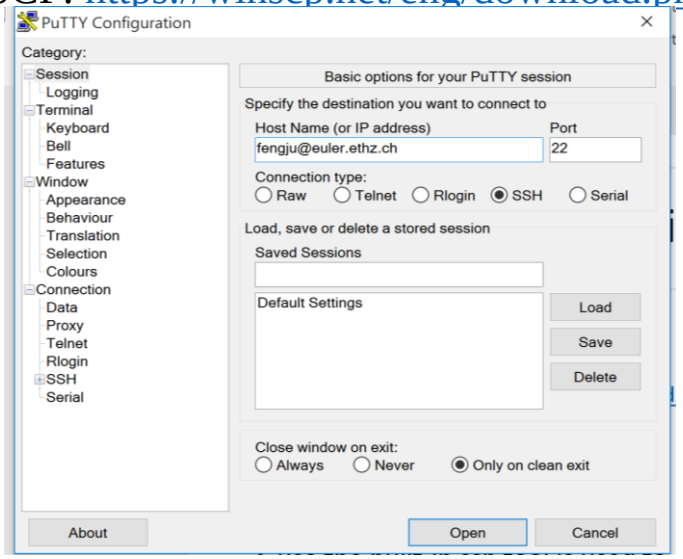
- **1. VPN connection from eawag network to the ETH Zurich**

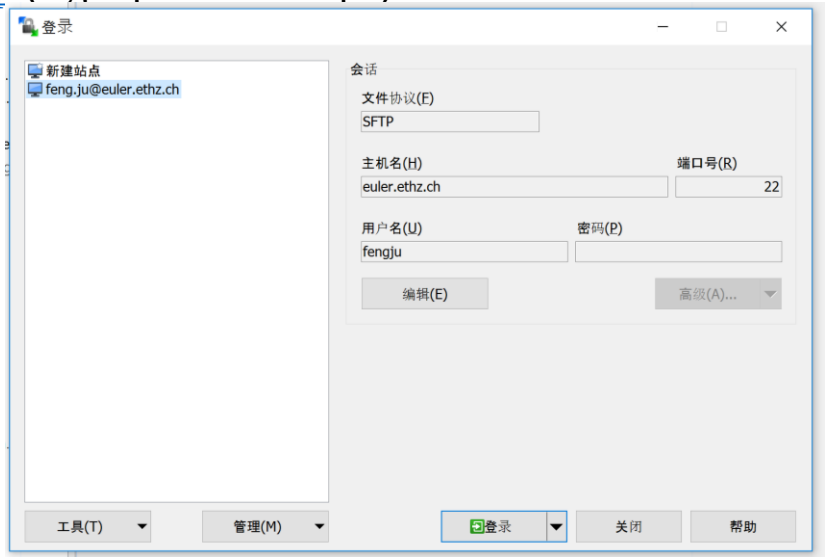
instructions here: https://www.ethz.ch/services/de/it-services/katalog/netzwerke-verbindungen/remote.html

- **2. For windows user, download and install:**

Putty: https://putty.org/ (left plot for setups)

WinSCP: https://winscp.net/eng/download.php (right plot for setups)





- **3. For Macs user, open terminal to type in:**

ssh <USER>@euler.ethz.ch

1) Make data subsets of sample D10

Change directory (cd) to the path with paired-end DNA-seq and RNA-seq

- cd /cluster/project/gdc/people/fengju/example (right click on WinSCP to get path)

Extract the first 1 million lines of DNA-seq

- **gzip -h** ### always type in '-h' or '—help' to learn about a command
- gzip -cd D10.R1_1.fq.gz | head -400000 > D10.R1_1.fq
- gzip -cd D10.R1_2.fq.gz | head -400000 > D10.R1_2.fq

Extract the last 1 million lines of DNA-seq

- gzip -cd D10.R2_1.fq.gz | tail -400000 > D10.R2_1.fq && gzip -cd D10.R2_2.fq.gz | tail -400000 > D10.R2_2.fq

Likewise:

Extract the first 1 million lines of RNA-seq

- gzip -cd R10.R1_1.fq.gz | head -400000 > R10.R1_1.fq
- gzip -cd R10.R1_2.fq.gz | head -400000 > R10.R1_2.fq

Extract the last 1 million lines of RNA-seq

- gzip -cd R10.R2_1.fq.gz | tail -400000 > R10.R2_1.fq && gzip -cd R10.R2_2.fq.gz | tail -400000 > R10.R2_2.fq

2) Make subfolders and sample lists for the datasets

Create folder with name “D10” and move the DNA-seq data to “D10”

- mkdir D10
- mv -t D10 D10.R1_1.fq D10.R1_2.fq D10.R2_1.fq D10.R2_2.fq

Do the same for RNA-seq data with a folder name as “R10”

- mkdir R10 && mv -t R10 R10.R*.fq

Do the same for gzip files with a folder name as “rawdata”

- mkdir rawdata && mv -t rawdata *fq.gz

Make sample ID list for DNA-seq

- cd D10 && pwd
- ls *_1.fq | sed 's/_1.fq//' > sample.DNA.txt
- Do the same for RNA-seq
- cd ../R10 && pwd
- ls *_1.fq | sed 's/_1.fq//' > sample.RNA.txt

3) Check and activate modules and commands on the euler

#Check which modules are installed and shared by all the euler users

- module avail

#Load module for gdc users

- module load gdc
- module avail

#Load all the dependencies before run a module, say “fastqc/0.11.4”

- modepend fastqc/0.11.4

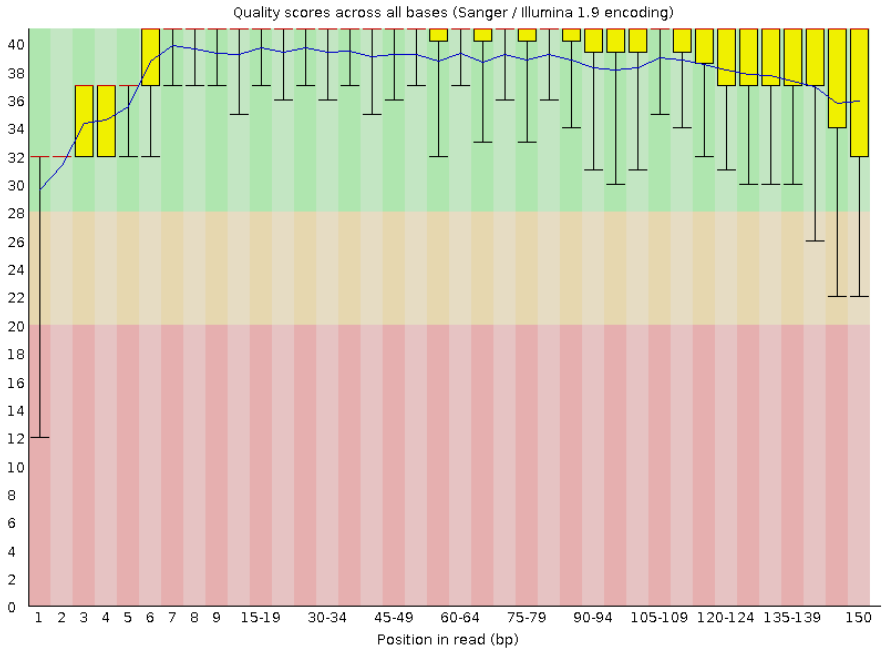
(tip: left click to select the terminal output and right click to automatically paste it into terminal)

- module load gcc/4.8.2 gdc java/1.8.0_73 fastqc/0.11.4
- fastqc -h

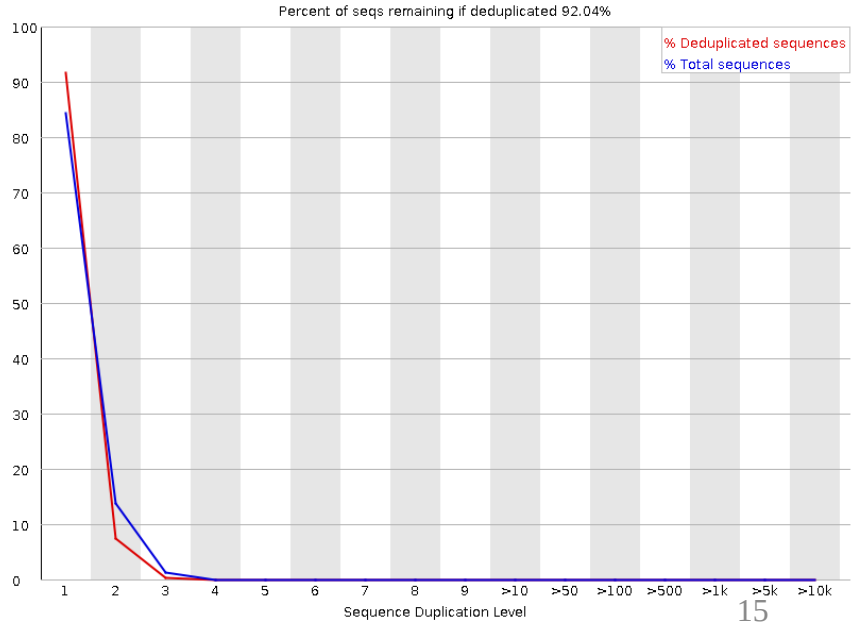
4) Data QC for DNA-seq and RNA-seq

Before data QC, a quick summary of raw data quality in an NGS library can be obtained using FASTQC:
http://www.bioinformatics.babraham.ac.uk/projects/fastqc/

Statistics and visualization of the quality of sequence data D10.R1 using fastqc
cd ../D10 && pwd && ls
fastqc D10.R1_1.fq



Base quality over 150 bps



Percentage of duplicated reads

4) Data QC for DNA-seq and RNA-seq

Two QC pipelines for shot-gun metagenomic reads:

prinseq: easy and rapid quality control and data preprocessing:
http://prinseq.sourceforge.net/

fastp: an alternative tool for fasta preprocessing for FastQ files:
https://github.com/OpenGene/fastp

One of the QC steps: whether and how to de-duplication

Natural/real or artificial? Unfortunately, artificial duplicates are difficult to distinguish from exactly overlapping reads that naturally occur within deep sequence samples

For high-complexity metagenomic samples lacking dominant species, natural duplicates only make up <1% of all duplicates. But for some other samples like transcriptomic samples, majority of the observed duplicates might be natural duplicates (Niu et al., 2010, BMC bioinformatics)

Deprelication (-derep) is recommended to be disabled when processing RNA-seq data

4) Data QC for DNA-seq and RNA-seq

Quality control (QC) of DNA-seq "D10.R1" using "prinseq"

- modepend prinseq-lite/0.20.4
- module load gcc/4.8.2 gdc perl/5.18.4 prinseq-lite/0.20.4
- prinseq-lite.pl -h
- head -n 8 D10.R1_1.fq # Every four lines represent one read
- head -n 1 D10.R1_1.fq && head -n 1 D10.R1_2.fq #Ids of paired-end reads
- prinseq-lite.pl -verbose -fastq D10.R1_1.fq -fastq2 D10.R1_2.fq -out_good D10.R1.Good -ns_max_p 10 -min_qual_mean 20 -stats_dupl -derep 1

The same for DNA-seq "D10.R2"

- prinseq-lite.pl -verbose -fastq D10.R2_1.fq -fastq2 D10.R2_2.fq -out_good D10.R2.Good -ns_max_p 10 -min_qual_mean 20 -stats_dupl -derep 1
- rm *singletons.fastq ## remove files that will not be used here
- rm *_bad_* ## remove files that will not be used here

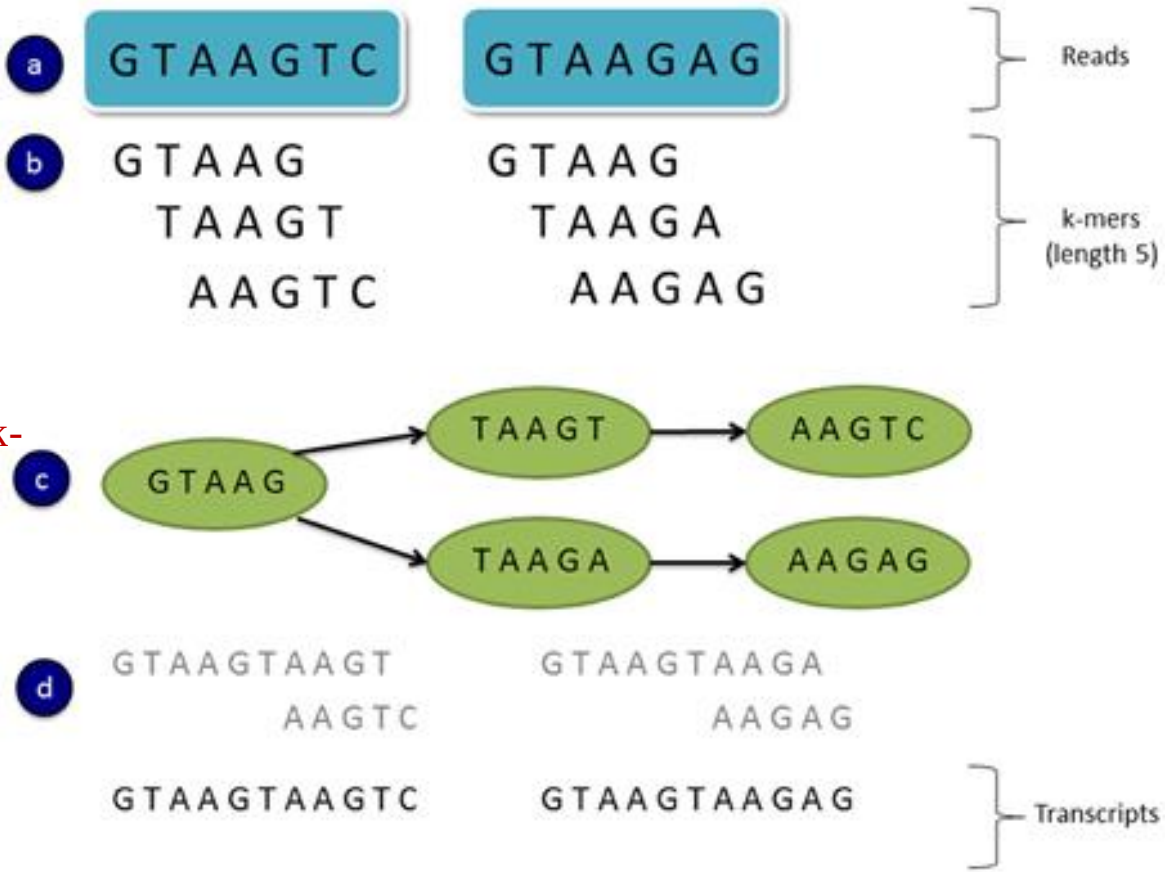
5) *De novo* assembly of metagenomes

k-mer: all the possible substrings of length k that are contained in a string (wiki)
Longer k-mer, the smaller contig size, conting number and probably more cohrecent assemblies

(A) Read sequences

(B) All subsequence k-mers of length 5 from the reads

(C) A De Bruijn graph constructed from unique k-mers as the nodes and overlapping k-mers connected by edges



- Check wiki for more about De Bruijn graph: https://en.wikipedia.org/wiki/De_Bruijn_graph

5) *De novo* assembly of metagenomes

Metagenome *de novo* assembly of D10.R1 using megahit

- modepend megahit/1.1.3
- module load gcc/4.8.2 gdc python/2.7.11 megahit/1.1.3
- megahit -h
- megahit --k-list 21,47,71,95,121 -t 2 --out-dir D10.R1_megahit --out-prefix D10.R1_contigs_megahit --min-contig-len 300 -1 D10.R1.Good_1.fastq -2 D10.R1.Good_2.fastq

Metagenome *de novo* assembly of D10.R2 using megahit

- megahit --k-list 21,47,71,95,121 -t 2 --out-dir D10.R2_megahit --out-prefix D10.R2_contigs_megahit --min-contig-len 300 -1 D10.R2.Good_1.fastq -2 D10.R2.Good_2.fastq

Homework 1: (if you like): using idba/1.1.1 for *de novo* assembly

Homework 2: (if you like): *de novo* assembly of raw reads vs. Post-QC reads to check how much it affect the assemblies statistics

6) In-house scripts for some statistics on the contig assemblies

- # Change to the “example” folder and build “contigs” folder
- cd .. && cd ..
- mkdir contigs
- find . -name '*megahit.contigs.fa' -exec mv -t contigs {} +

The N50 (N80) statistic is the length for which the collection of all contigs of that length or longer contains at least 50% (80%) of the sum of the lengths of all contigs

- # Calculate the N50, max., average and min. length of sequence in a file
- python scripts/N50_GC.py -h
- python scripts/N50_GC.py -i contigs -o contigs.summary.csv

7) Gene prediction from contigs

Load module for prodigal access on the euler

- module avail
- modepend prodigal/2.6.3
- module load gcc/4.8.2 gdc prodigal/2.6.3
- prodigal -h
- mkdir genes

#Gene prediction from contigs using prodigal

Start codon: usually ATG, GTG, or TTG; stop codon: usually TAA, TGA, or TAG)

- prodigal -i contigs/D10.R1_contigs_megahit.contigs.fa -a genes/D10.R1.megahit.prodigal.faa -d genes/D10.R1.megahit.prodigal.fna -p meta -f gff -o genes/D10.R1.megahit.prodigal.gff
- prodigal -i contigs/D10.R2_contigs_megahit.contigs.fa -a genes/D10.R2.megahit.prodigal.faa -d genes/D10.R2.megahit.prodigal.fna -p meta -f gff -o genes/D10.R2.megahit.prodigal.gff

•**Partial**: An indicator of if a gene runs off the edge of a sequence or into a gap. "11" indicates both edges are incomplete, and "00" indicates a complete gene with a start and stop codon.

8) Reads mapping to contigs and genes --- Bowtie2/BWA mapping

- # Load modules and create reference databases for mapping
- #module load gdc && module avail && modepend bowtie2/2.2.6
- module load gcc/4.8.2 gdc bowtie2/2.2.6
- bowtie2-build -h #always invoke help to check the software usage
- bowtie2-build contigs/D10.R1_contigs_megahit.contigs.fa D10.R1_contigs.bt.db
- mkdir databases && mv *.bt2 databases

Bowtie2 mapping of reads to reference sequences (e.g., contigs or genes)

- bowtie2 -h #always invoke help to help
- bowtie2 -x databases/D10.R1_contigs.bt.db -1 D10/D10.R1.Good_1.fastq -2 D10/D10.R1.Good_2.fastq --threads 1 -S D10.R1_contigs.sam -q
- mkdir D10_bowtie2 && mv D10.R1_contigs.sam D10_bowtie2

8) Reads mapping to contigs and genes --- Coverage calculation

Load modules to use samtools

- #module load gdc && module avail && modepend bowtie2/2.2.6
- module load gcc/4.8.2 gdc perl/5.18.4 samtools/1.3
- samtools view -bS D10_bowtie2/D10.R1_contigs.sam > D10_bowtie2/D10.R1_contigs.bam # file format conversion
- samtools sort D10_bowtie2/D10.R1_contigs.bam -o D10_bowtie2/D10.R1_contigs.sorted.bam # bam file sorting
- samtools depth D10_bowtie2/D10.R1_contigs.sorted.bam > D10_bowtie2/D10.R1_contigs.depth.txt # generate depth info.
- perl scripts/calc.coverage.in.bam.depth.pl -i D10_bowtie2/D10.R1_contigs.depth.txt -o D10.R1_contigs.coverage.csv

For space efficiency, remove or gzip temporary files

- rm D10_bowtie2/D10.R1_contigs.sam && rm D10_bowtie2/D10.R1_contigs.sorted.bam
- gzip D10_bowtie2/D10.R1_contigs.bam

9) Gene/Read annoation using BLAST: Basic Local Alignment Search Tool

Load modules to use NCBI's BLAST tools for nucleotide sequence search

- #module load gdc && module avail && modepend blast/2.2.30
- module load blast/2.2.30
- makeblastdb -h
- makeblastdb -in databases/91_otus.fasta -parse_seqids -dbtype nucl
- blastn -query genes/D10.R1.megahit.prodigal.fna -db databases/91_otus.fasta -out D10.R1_gene.GG91.blastp -evalue 1e-5 -max_target_seqs 10 -num_threads 1 -outfmt 6

Use blastp for protein sequence search

- makeblastdb -in databases/SARG_20170328.fasta -parse_seqids -dbtype prot
- blastp -query genes/D10.R1.megahit.prodigal.faa -db databases/SARG_20170328.fasta -out D10.R1_gene.SARG.blastp -evalue 1e-5 -max_target_seqs 1 -num_threads 1 -outfmt 6
- grep '>' genes/D10.R1.megahit.prodigal.faa | wc -l ### count sequence number in fasta

9) Gene/Read annoation using BLAST: blast outfmt 6 (tab-delimited)

1.	qseqid
2.	sseqid
3.	pident
4.	length
5.	mismatch
6.	gapopen
7.	qstart
8.	qend
9.	sstart
10.	send
11.	evalue
12.	bitscore

10 hits of 1 gene sequence in the 91% OTU set of GreenGene database

k121_369_1	806314	97.92	192	3	1	1	192	556	746	8e-91	331
k121_369_1	4296689	96.35	192	7	0	1	192	575	766	2e-86	316
k121_369_1	329744	95.83	192	8	0	1	192	567	758	1e-84	311
k121_369_1	329171	95.34	193	7	2	1	192	565	756	5e-83	305
k121_369_1	559843	94.27	192	11	0	1	192	568	759	1e-79	294
k121_369_1	4415092	94.21	190	9	2	4	192	550	738	5e-78	289
k121_369_1	877884	93.23	192	13	0	1	192	569	760	2e-76	283
k121_369_1	865748	93.12	189	13	0	4	192	554	742	1e-74	278
k121_369_1	1130640	92.63	190	12	2	4	192	526	714	5e-73	272
k121_369_1	667312	92.63	190	12	2	4	192	575	763	5e-73	272

The best hit for 20 gene sequences in the structured ARDB

k121_6_2	gi|610427609|gb|AHW76485.1|	40.00	90	54	0	1	90	390	479
k121_57_1	BAC58936	27.64	123	62	4	14	111	20	140	5e-06	42.0
k121_103_1	AAG07763	56.20	121	52	1	1	120	884	1004	3e-36	132
k121_157_1	U82085.gene.p01	43.06	72	40	1	66	136	5	76	4e-11	58.9
k121_157_1	U82085.gene.p01	35.14	74	43	2	61	133	288	357	6e-06	43.1
k121_185_1	gi|488156254|ref|WP_002227462.1|	35.71	112	72	0	3	114	7			
k121_223_2	gi|1011730119|ref|WP_062573757.1|	26.76	142	93	5	1	137	6			
k121_349_1	gi|445996732|ref|WP_000074587.1|	30.69	101	70	0	1	101	7			
k121_386_1	gi|542061059|gb|ERI11611.1|	36.84	95	53	2	6	99	144	232		
k121_573_1	AAL09826	37.93	87	53	1	14	100	49	134	9e-13	64.3
k121_707_2	gi|817122037|ref|WP_046494699.1|	66.67	27	9	0	1	27	570			
k121_758_1	X63451.gene.p01	32.34	235	128	5	4	211	166	396	9e-24	98.2
k121_758_1	X63451.gene.p01	48.33	60	31	0	4	63	467	526	6e-09	53.9
k121_798_1	CAA37477	41.25	80	44	2	9	88	73	149	2e-11	60.1
k121_816_2	NC_002951.3238224.p01	41.24	177	101	2	1	177	1	174	3e-39	
k121_836_1	gi|1004359922|gb|AMP42228.1|	32.14	84	54	1	4	87	340	420		
k121_859_1	gi|779850732|ref|WP_045348329.1|	41.49	94	55	0	9	102	110			
k121_876_1	FJ349556.1.gene2.p01	39.66	58	35	0	4	61	5	62	4e-08	
k121_898_1	AAL09826	40.19	107	57	3	12	116	49	150	5e-15	71.2
k121_1044_1	gi|446026113|ref|WP_000103968.1|	33.95	162	101	2	9	167	1			

Self-written scripts are needed to process the blast output tables and extract and match assignment information to the database annotations

10) Standard ways to submit your jobs on the euler --- single jobs and jobararrys

- # Use bsub to submit single jobs
- #module load gdc && module avail && modepend blast/2.2.30
- module load blast/2.2.30
- bsub -n1 -W 1:00 -R "rusage[mem=500]" -J "BLASTP" "blastp -query genes/D10.R1.megahit.prodigal.faa -db databases/SARG_20170328.fasta -out D10.R1_gene.SARG.blastp -evalue 1e-5 -max_target_seqs 1 -num_threads 1 -outfmt 6"

Use bsub to submit jobsarrarys

- cd genes/ && ls *.fna | sed 's/.fna//'> sample_list.txt
- bsub < scripts/submit.usearch16s.jobarrays.cmds.lsf # Open the .lsf to edit
- bjobs # check the states of the jobs

- More information about the "**7. Running jobs - The batch system**" are available GDC_Euler_manual.v01.09.2017.pdf

Final remarks and advice

- Basic bioinformatics precedures for microbial metagenomes may include data pretreatment, *de novo* assembly, gene prediction, and gene annotation
- Microbial metatranscriptomes can be annoated via RNA reads mapping to predicted genes and/or assembled contigs from paired metagenomes
- Microbial community structure and function could be explored based on both read-based and assembly-based annotation strategies

- You are the best teacher of your bioinformatics
- Always Google for published codes or de-bugging
- Learn from and share codes through open-source platforms (e.g., github)
- Make a electronic note/diary for your bioinformatics experiments
- ….

Appendix I: commnd lines for genome binning on the euler

- Using metabat for automated coverage-composition-based binning of genome contigs using metabat/2.12.1
- module load gdc
- module avail
- modepend metabat/2.12.1
- module load gcc/4.9.2 gdc boost/1.55.0 python/2.7.11 metabat/2.12.1
- metabat2 -h

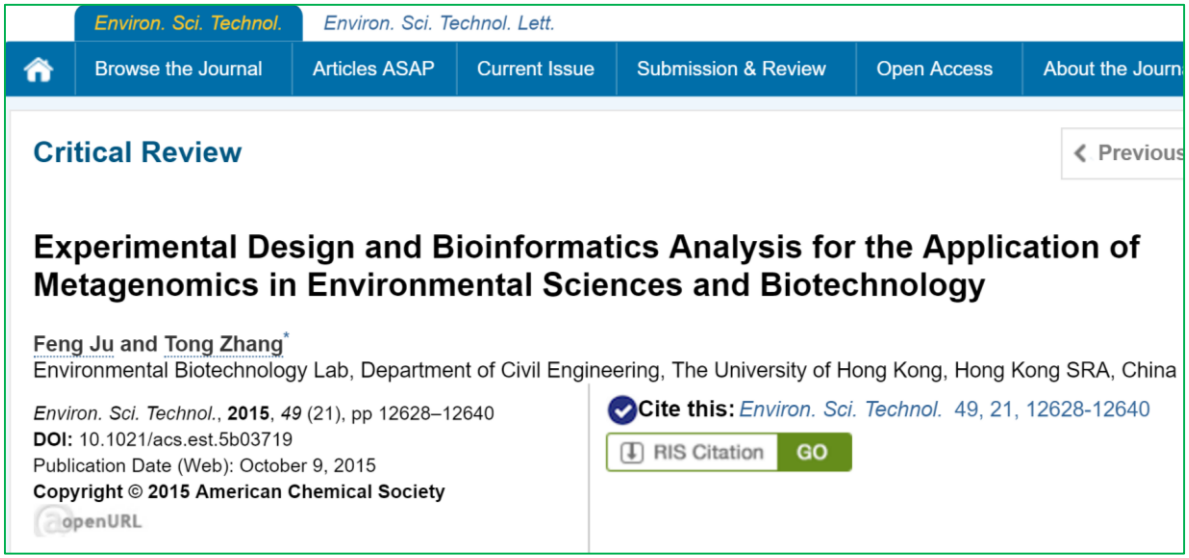
Appendix II:

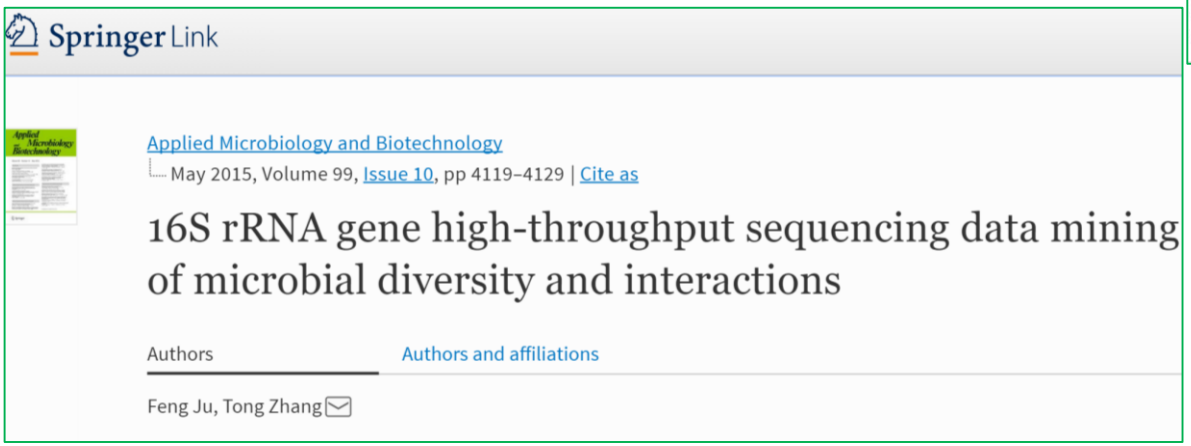
Quantitative meta-omics metrics of microbial gene and transcripts

Metric	Full definition	Definition	Reference
RPK	Reads Per Kilobase	Number of assigned reads divided by length of each gene or transcripts in kilobases	(Katz et al 2010)
PMSF	“Per million” scaling factor.	The sum of all the RPK values in a sample divided by one million	(Li and Dewey 2011)
TPM	Transcripts Per (Kilobase) Million	The RPK values of each transcripts divided by the PMSF of its sample	(Li and Dewey 2011)
GPM	Genes Per (Kilobase) Million	The RPK values of each gene divided by the PMSF of its sample	This study
RPK-16S	Reads per kilobase 16S rRNA gene	The sum of RPKs of all identified 16S rRNA genes	This study
16S%MG	16S rRNA gene percentage in a MetaGenome	Number of reads identified as 16S rRNA gene divided by total number of reads in a metagenome	This study
GP16S	Genes Per 16S rRNA gene	RPK of a give gene divided by RPK-16S	This study
16S-GCN	16S rRNA Gene Copy Number	Composition-weighted average 16S rRNA gene copy number per cell	(Yang et al 2016)
16S-GPL	16S rRNA Gene copies Per Liter	Copies of 16S rRNA gene per liter of sample (determined by qPCR)	This study
CPL	Cells Per Liter	16S-GPL divided by composition-weighted 16S-GCN	This study
GPL	Gene copies Per Liter	16S-GPL multiplied by GP16S of a given gene	This study
TPL	Transcript copies Per Liter	Read abundance of a transcript multiplied per-liter normalization factor (NF1) as determined by spiked RNA standard (Table S4)	This study
TPB	Transcript copies Per gram-of-biomass	Read abundance of a transcript multiplied by per-gram normalization factor (NF2) as determined by spiked RNA standard (Table S4)	This study
TPG	Transcript copies Per Gene copy (i.e., gene expression ratio)	TPL divided by GPL	This study
TPC	Transcript copies Per Cell	TPL divided by CPL	This study

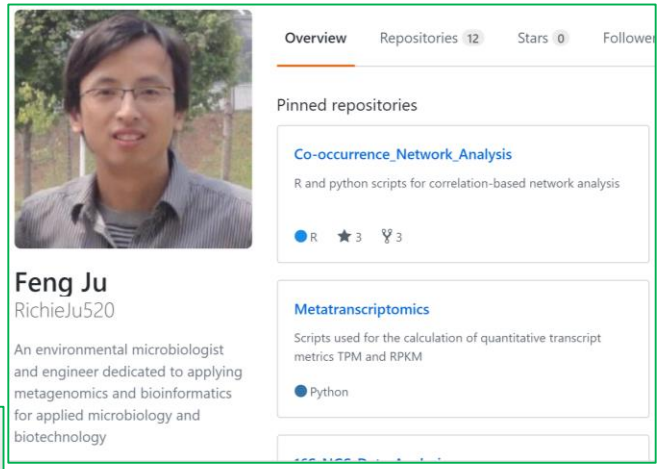
Table S2 from Ju F et al., 2018

Cite my papers and gihub link





My Github Profile



https://github.com/RichieJu520