Complement assemblage procaryotes

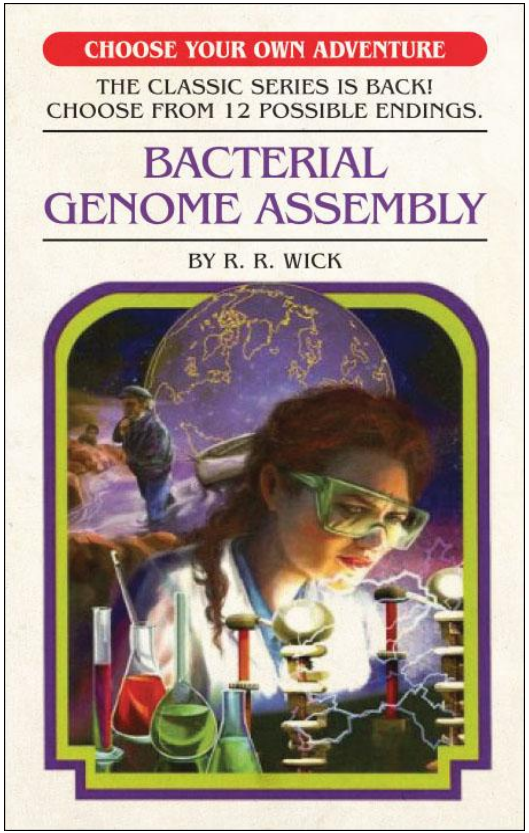
Valentin Loux
Guillaume Gautreau

co-rédacteur : Olivier Rué, Cédric Midoux :
https://documents.migale.inrae.fr/posts/training-materials/2024-03-18-module24/slides/#/title-slide

March 18, 2024

Isolate genome assembly

- Sequencing strategy ?
  - long read only
  - Mix ?
- Assembly strategy ?
  - short + long
  - long + short



https://github.com/rrwick/Trycycler/wiki/Guide-to-bacterial-genome-assembly

Short read or low depth hybrid

Short read only OR hybrid with low depth (< 100x) long reads

- **Illumina only & hybrid** :
  - short-read first assembly with SPAdes
  - use long read to scaffold
  - filter low depth reads
  - handle plasmids
  - circularize & choose “start”
- **Long read only**
  - Long read only with Miniasm (assembly)
  - Polishing with Racoon

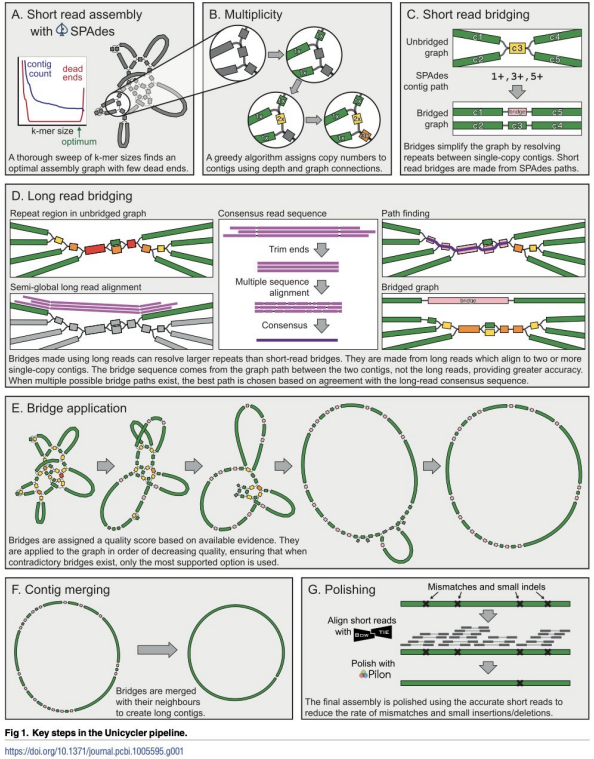


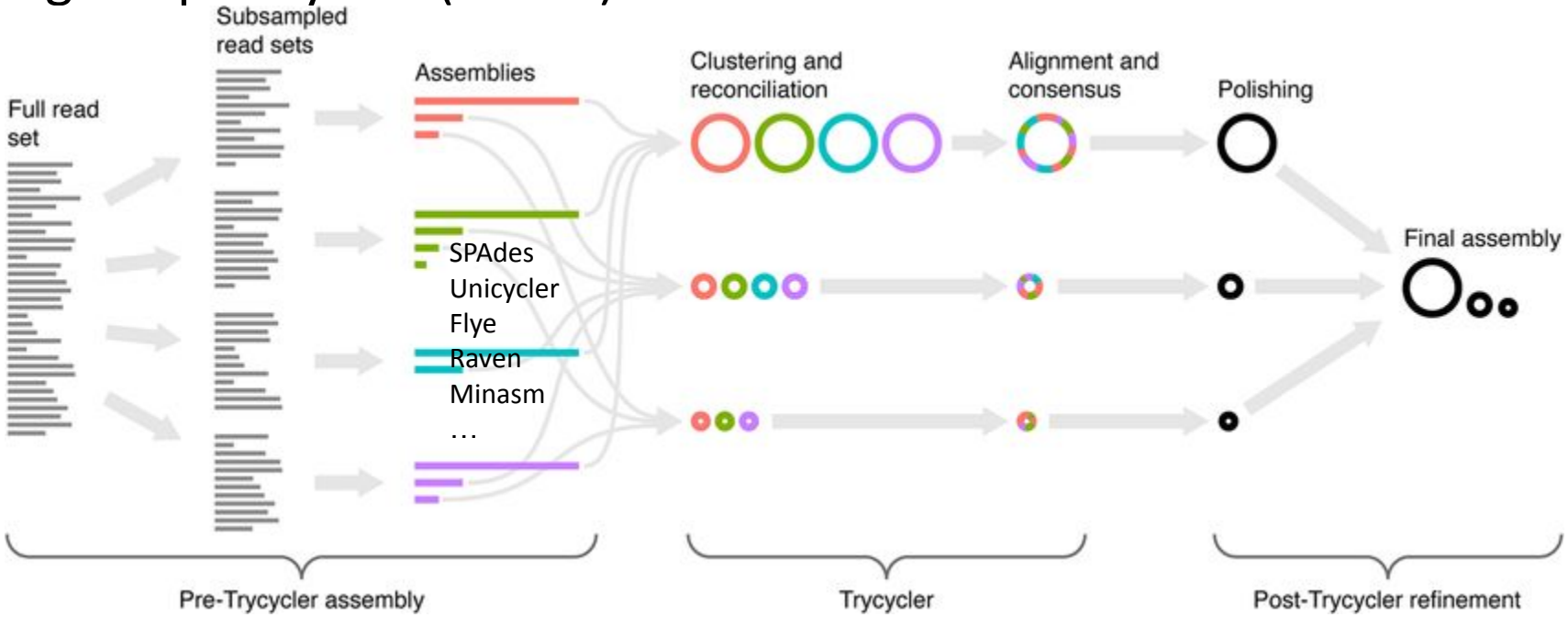
Fig 1. Key steps in the Unicycler pipeline.

https://doi.org/10.1371/journal.pcbi.1005595.g001

Wick RR, Judd LM, Gorrie CL, Holt KE. Unicycler: resolving bacterial genome assemblies from short and long sequencing reads. *PLoS Comput Biol* 2017.

Trycycler : Long read only or hybrid

High depth hybrid (>100x) with or without short reads



Wick RR, Judd LM, Cerdeira LT, Hawkey J, Méric G, Vezina B, Wyres KL, Holt KE. Trycycler: consensus long-read assemblies for bacterial genomes. Genome Biology. 2021. doi:10.1186/s13059-021-02483-z.

Trycycler : detailed view (1)

Step 1: Generating assemblies for Trycycler

Assembly A:
```
contig_1: TCGGCGTGTGGTCTAAAGACTCCGGATGGGGCGTCATGGTTGATTCATCGATAATTTTC
contig_2: AGCGTTGTACG
```

Assembly B:
```
contig_1: GACGCCCCCATCCGGAGTCTTTAGGACCACTCGCCGAGAAAAATTATCGATGAATCACCA
contig_2: TTGTAGCGAGCG
contig_3: AAAAAA
```

Assembly C:
```
contig_1: GCCGAGAAAAATTATCGATGAATCAACCATGACGCCCCATCCGGAGTCTTTAGACCACCGCC
```

Assembly D:
```
contig_1: GATCCGGATGGGGCGTCATGGTTGATTCATCGATAATTTTTCTCGGCGGGTGGTCTAAA
contig_2: AACGCCGCTACAAC
```

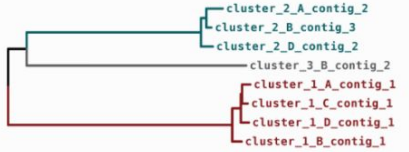
As input, Trycycler takes multiple different assemblies of the same genome. These can be generated using different assemblers and/or different read subsets.

Step 2: Clustering contigs

Cluster 1:
```
A_contig_1: TCGGCGTGTGGTCTAAAGACTCCGGATGGGGCGTCATGGTTGATTCATCGATAATTTTC
B_contig_1: GACGCCCCCATCCGGAGTCTTTAGGACCACTCGCCGAGAAAAATTATCGATGAATCACCA
C_contig_1: GCCGAGAAAAATTATCGATGAATCAACCATGACGCCCCATCCGGAGTCTTTAGACCACCGCC
D_contig_1: GATCCGGATGGGGCGTCATGGTTGATTCATCGATAATTTTTCTCGGCGGGTGGTCTAAA
```

Cluster 2:
```
A_contig_2: AGCGTTGTACG
B_contig_2: TTGTAGCGAGCG
D_contig_2: AACGCCGCTACAAC
```

Cluster 3:
```
B_contig_3: AAAAAA
```



Contigs from all assemblies are clustered based on their *k*-mer content. Trycycler makes a tree of the contig relationships to help users distinguish good clusters (which represent completely assembled replicons) vs bad clusters (which contain spurious, fragmented or incorrectly assembled sequences).

Step 3: Reconciling contigs

Normalise strands and fix circularisation:

Cluster 1:
```
A_contig_1: GAAAATTATCGATGAATCAACCATGACGCCCCATCCGGAGTCTTTAGACCACACGCCGA
B_contig_1: GACGCCCCCATCCGGAGTCTTTAGGACCACTCGCCGAGAAAAATTATCGATGAATCACCAT
C_contig_1: GCCGAGAAAAATTATCGATGAATCAACCATGACGCCCCATCCGGAGTCTTTAGACCACC
D_contig_1: TTTAGACCACCCGCCGAGAAAAATTATCGATGAATCAACCATGACGCCCCATCCGGATC
```

Cluster 2:
```
A_contig_2: CGTACAACGCT
B_contig_2: CGCTCGCTACAA
D_contig_2: AACGCCGCTAC
```

Contig sequences are flipped to their reverse complement as necessary to ensure that all sequences within each cluster are on the same strand. For circular clusters, sequences are aligned to each other to repair circularisation issues: trimming overlapping bases or adding missing bases.

Rotate to consistent start:

Cluster 1:
```
A_contig_1: ATGAATCAACCATGACGCCCCATCCGGAGTCTTTAGACCACACGCCGAGAAAATTATCG
B_contig_1: ATGAATCACCATGACGCCCCATCCGGAGTCTTTAGGACCACTCGCCGAGAAAAATTATCG
C_contig_1: ATGAATCAACCATGACGCCCCATCCGGAGTCTTTAGACCACCGCCGAGAAAAATTATCG
D_contig_1: ATGAATCAACCATGACGCCCCATCCGGATCTTTAGACCACCCGCCGAGAAAAATTATCG
```

Cluster 2:
```
A_contig_2: GCTCGTACAAC
B_contig_2: GCTCGCTACAAC
D_contig_2: GCCGCTACAAC
```

For each circular cluster, a starting sequence is identified (using a standard coding sequence, if possible) and the sequences are rotated to have a consistent start/end. Each cluster's sequences are now ready for global multiple sequence alignment.

Trycycler : detailed view (2)

Step 4: Multiple sequence alignment

Cluster 1:

```
A_contig_1: ATGAATCAACCATGACGCCCC-ATCCGGAGTCTTTAG-ACCACACGCCGAGAAAA-TTATCG
B_contig_1: ATGAATC-ACCATGACGCCCCCATCCGGAGTCTTTAGGACCACTCGCCGAGAAAAATTATCG
C_contig_1: ATGAATCAACCATGACGCCCC-ATCCGGAGTCTTTAG-ACCAC-CGCCGAGAAAAATTATCG
D_contig_1: ATGAATCAACCATGACGCCCC-ATCCGGA-TCTTTAG-ACCACCCGCCGAGAAAAATTATCG
```

Cluster 2:

```
A_contig_2: GCTCG-TACAAC
B_contig_2: GCTCGCTACAAC
D_contig_2: GC-CGCTACAAC
```

Trycycler uses MUSCLE to produce a global multiple sequence alignment for each of the clusters.

Step 5: Partitioning reads

All reads:

```
CTCGCC AATTAT AGAAAA CTCGCT
GAGAAA TTAGAC AACGCT TCGCTA
AGACCA CGAGAA CCGCCG GACCAC
TCTTTA CACTCG CGGAGT CGCTCG
ATCAAC GCTCGC GAAAAA AACCAT
GTCTTT CCGCTA GTACAA CACCAT
ACCACA TACAAC TGACGC CCCATC
ATGACG CGCCGA CTACAA ACGCCG
TCCGGA AAAAAT GCTACA GGAGTC
CATGAC GCCCCA ACAACG GATGAA
```

Cluster 1 reads:

```
CTCGCC AATTAT AGAAAA GAGAAA
TTAGAC AGACCA CGAGAA CCGCCG
GACCAC TCTTTA CACTCG CGGAGT
ATCAAC GAAAAA AACCAT GTCTTT
CACCAT ACCACA TGACGC CCCATC
ATGACG CGCCGA TCCGGA AAAAAT
GGAGTC CATGAC GCCCCA GATGAA
```

Cluster 2 reads:

```
CTCGCT AACGCT TCGCTA CGCTCG
GTACAA GCTCGC CCGCTA TACAAC
CTACAA ACGCCG GCTACA ACAACG
```

Reads are aligned to each contig sequence and assigned to the cluster to which they best align.

Step 6: Generating a consensus

Divide alignment into chunks:

Cluster 1:

Cluster 2:

The multiple sequence alignment is divided into chunks: “same” chunks where the sequences agree and “different” chunks where there are multiple possible options.

Choose best option for each chunk:

Cluster 1:

Cluster 2:

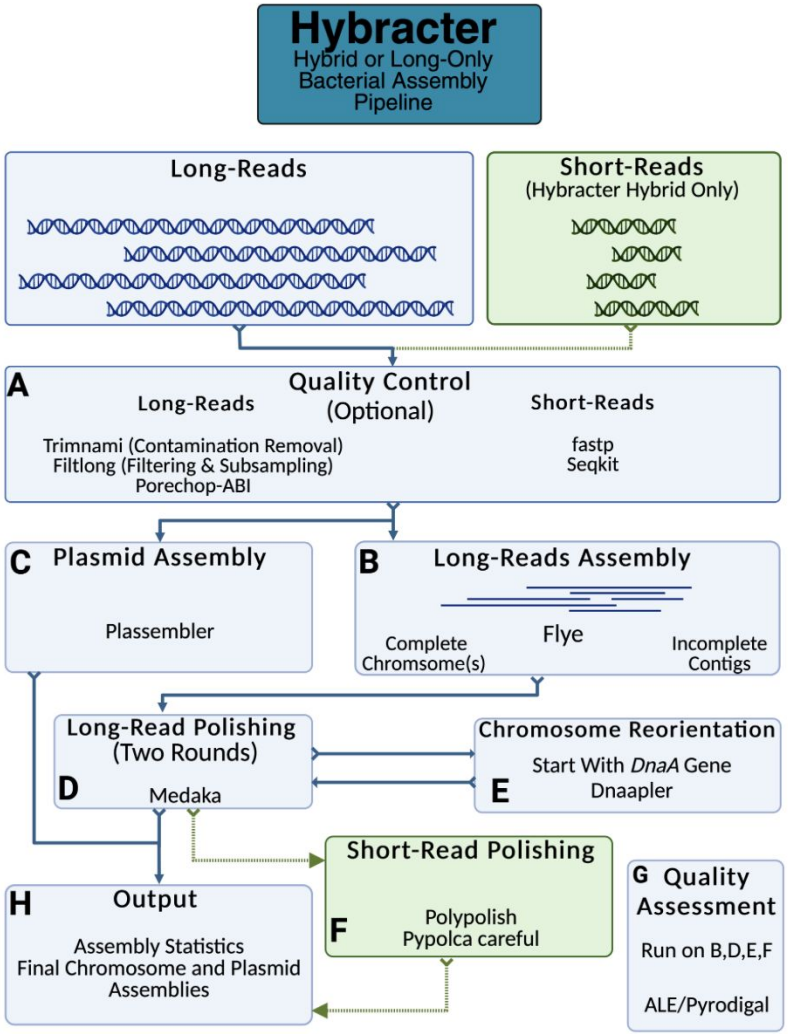
For each “different” chunk, the most popular option is chosen (as defined by the minimum total Hamming distance to other options). When there is a tie, reads are aligned to each alternative to decide which option to keep (the one with the best total read alignment score).

Step 7: Polishing after Trycycler

Platform-specific long-read polishing (e.g. Medaka for ONT sequencing or GenomicConsensus for PacBio sequencing) can reduce the number of small-scale errors in the Trycycler assembly. If available, short-read polishing (e.g. with Pilon) can further reduce small-scale errors.

Hybracter

- Automated snakemake workflow
- “As easy as Unycycler”
- Assembly + Plasmid



George Bouras, Ghais Houtak, Ryan R Wick, Vijini Mallawaarachchi, Michael J. Roach, Bhavy Papudeshi, Louise M Judd, Anna E Sheppard, Robert A Edwards, Sarah Vreugde - Hybracter: Enabling Scalable, Automated, Complete and Accurate Bacterial Genome Assemblies. (2024) *Microbial Genomics* doi: https://doi.org/10.1099/mgen.0.001244.

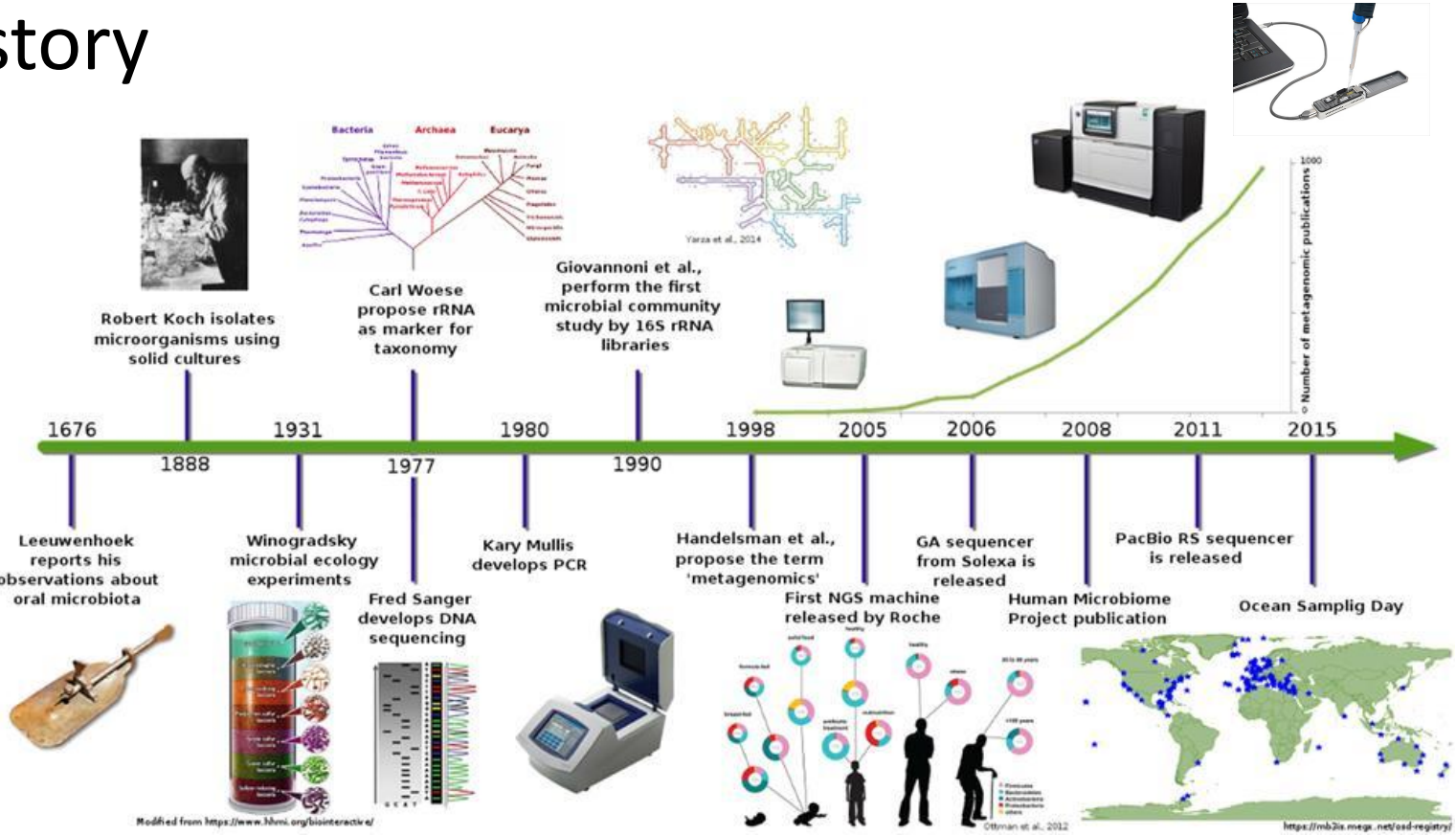
Introduction to shotgun metagenomics

Guillaume Gautreau
Valentin Loux

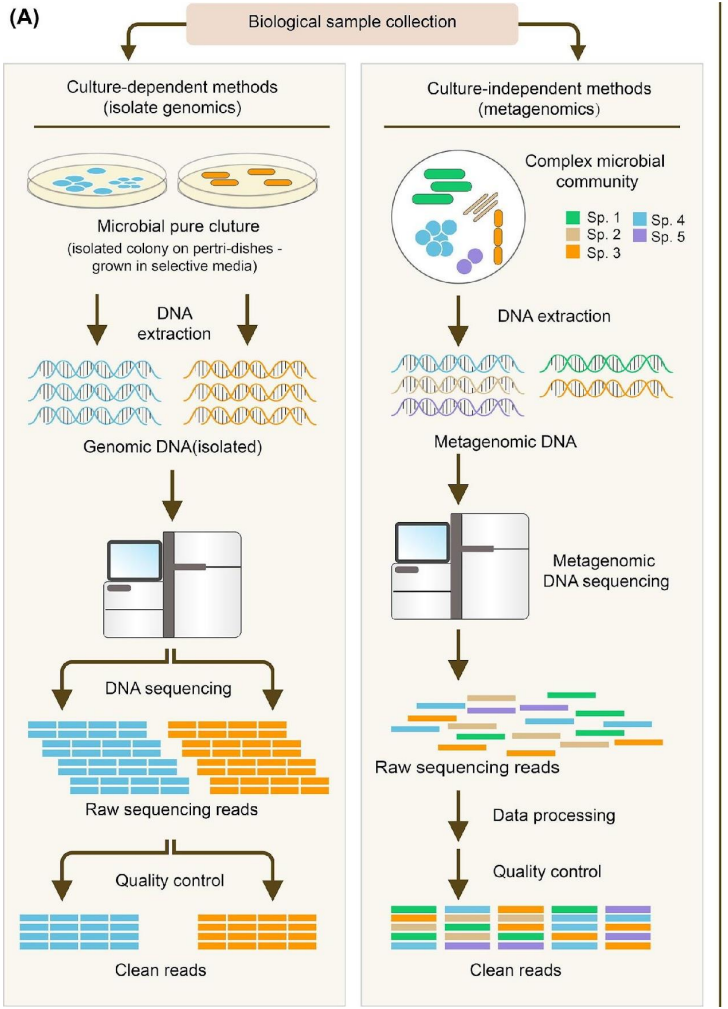
co-rédacteur : Olivier Rué, Cédric Midoux :
https://documents.migale.inrae.fr/posts/training-materials/2024-03-18-module24/slides/#/title-slide

March 18, 2024

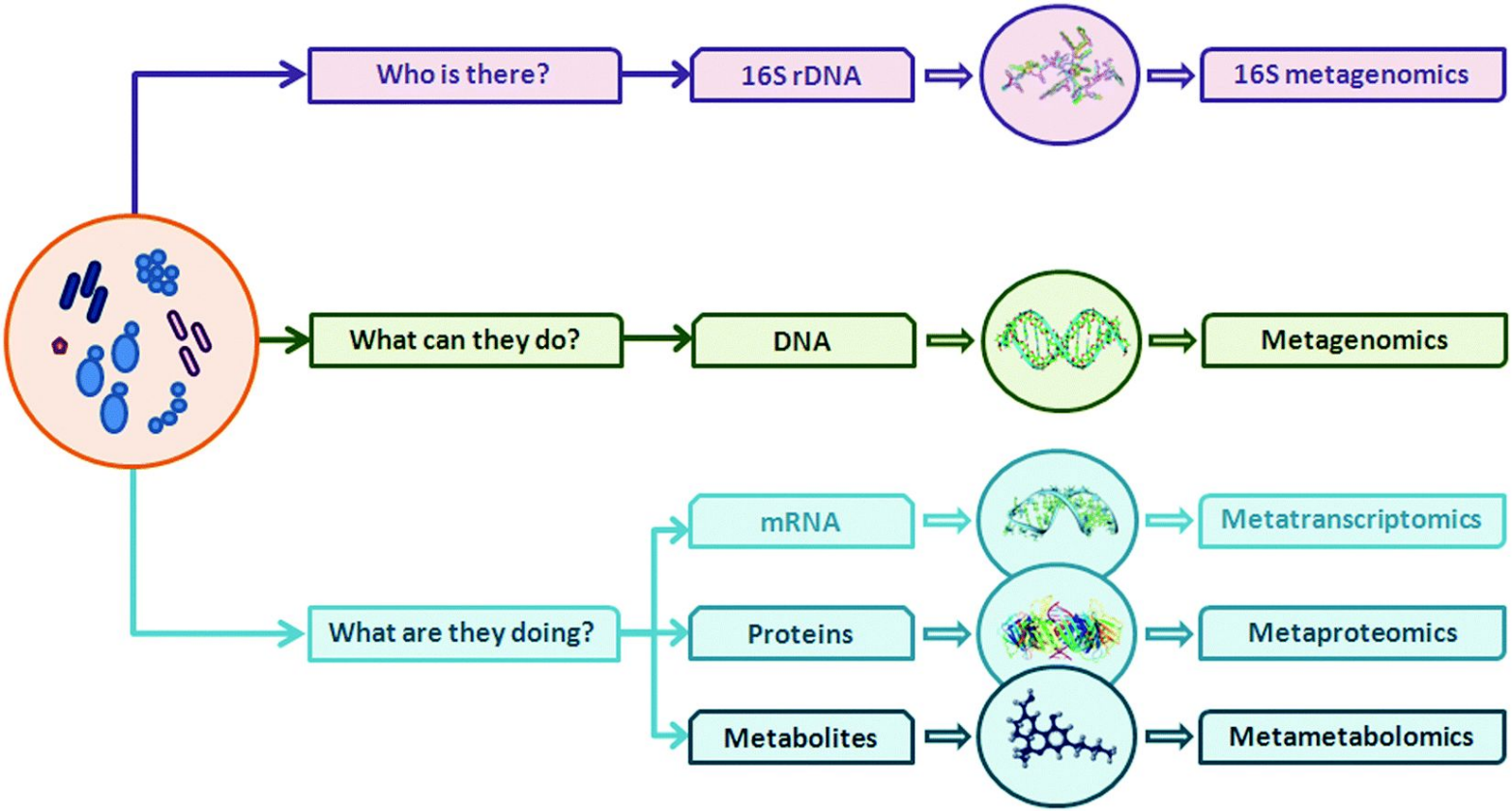
History


Bacteria
Archaea
Eucarya
Yarza et al., 2014
Number of metagenomic publications
1000
0
1676
1888
1931
1977
1980
1990
1998
2005
2006
2008
2011
2015
Leeuwenhoek reports his observations about oral microbiota
Robert Koch isolates microorganisms using solid cultures
Winogradsky microbial ecology experiments
Carl Woese propose rRNA as marker for taxonomy
Fred Sanger develops DNA sequencing
Kary Mullis develops PCR
Giovannoni et al., perform the first microbial community study by 16S rRNA libraries
Handelsman et al., propose the term 'metagenomics'
First NGS machine released by Roche
GA sequencer from Solexa is released
Human Microbiome Project publication
PacBio RS sequencer is released
Ocean Samplig Day
Modified from https://www.hhmi.org/biointeractive/
Ottman et al., 2012
https://mb3is.megx.net/osd-registry/

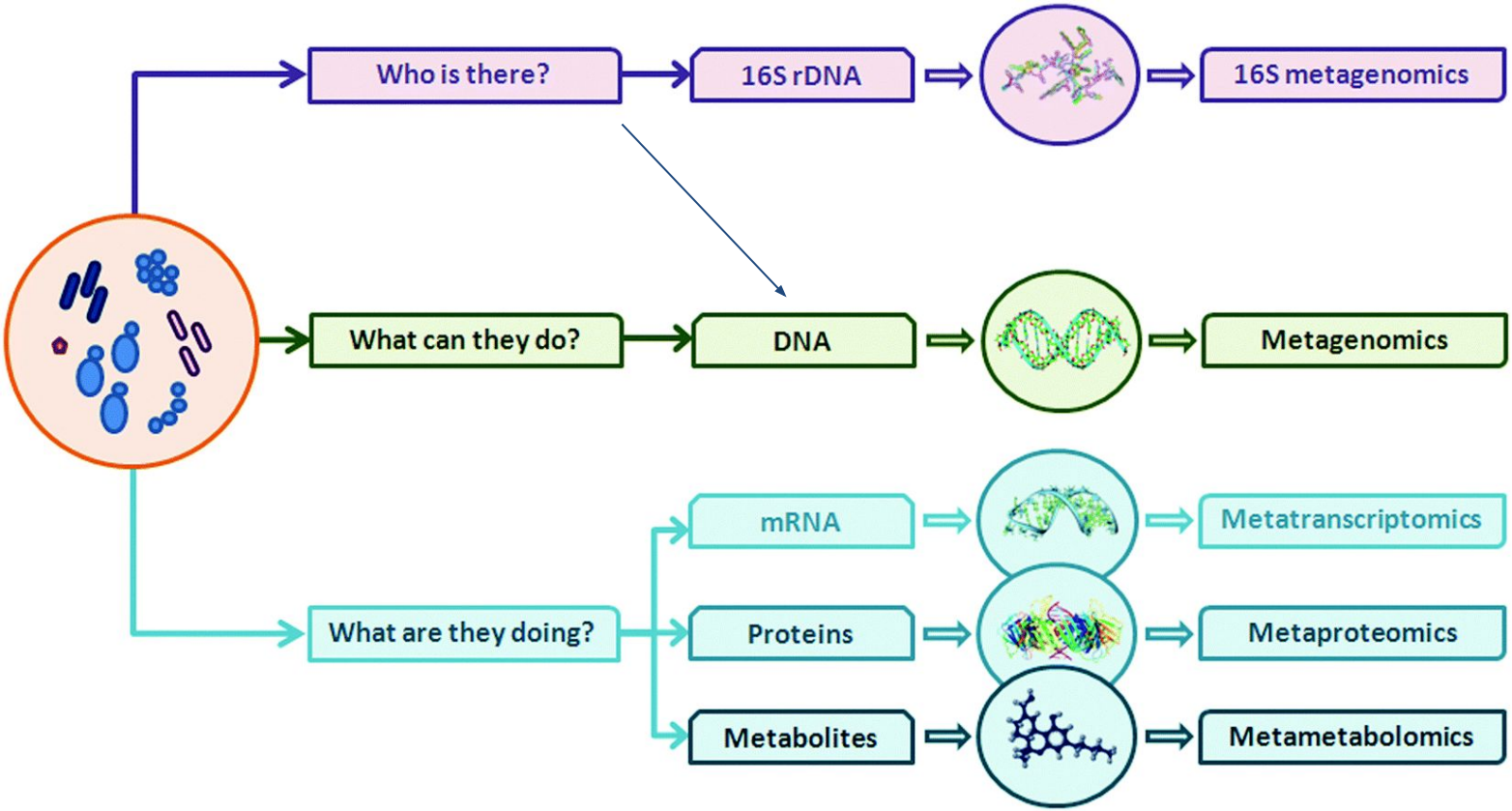
Isolate genomics versus metagenomics



Introduction



Introduction



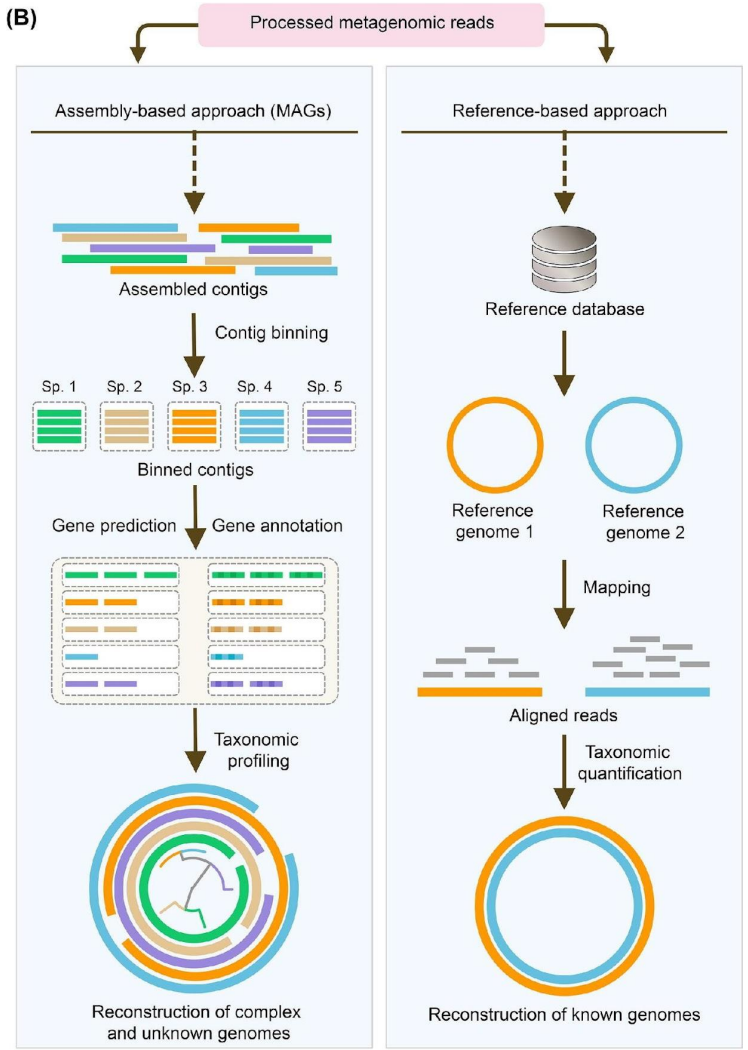
A review of methods and databases for metagenomic classification and assembly (2019)

Table 1. Metataxonomics, metagenomics and meta-transcriptomics strategies

Technique	Advantages and challenges	Main applications
Metataxonomics using amplicon sequencing of the 16S or 18S rRNA gene or ITS	+ Fast and cost-effective identification of a wide variety of bacteria and eukaryotes – Does not capture gene content other than the targeted genes – Amplification bias – Viruses cannot be captured	* Profiling of what is present * Microbial ecology * rRNA-based phylogeny
Metagenomics using random shotgun sequencing of DNA or RNA	+ No amplification bias + Detects bacteria, archaea, viruses and eukaryotes + Enables *de novo* assembly of genomes – Requires high read count – Many reads may be from host – Requires reference genomes for classification	* Profiling of what is present across all domains * Functional genome analyses * Phylogeny * Detection of pathogens
Meta-transcriptomics using sequencing of mRNA	+ Identifies active genes and pathways – mRNA is unstable – Multiple purification and amplification steps can lead to more noise	* Transcriptional profiling of what is active

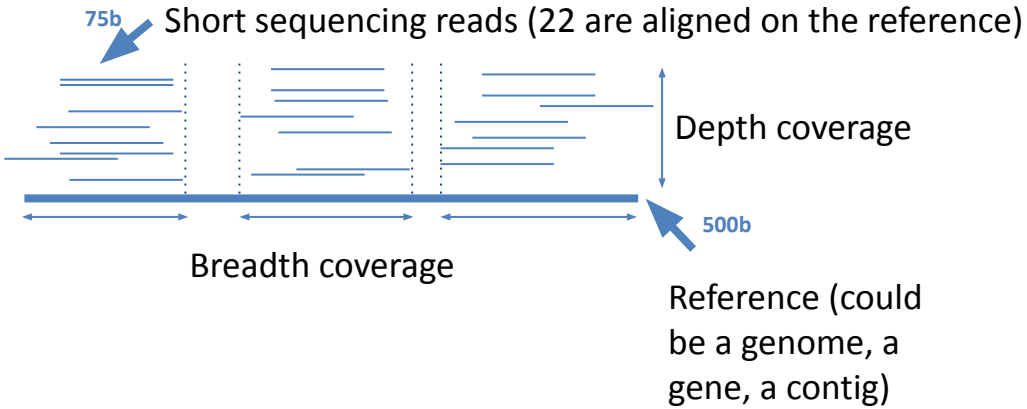
Challenges

- Complexity of the ecosystem
- Completeness of databases
- Sequencing depth
- Computational and storage resources required



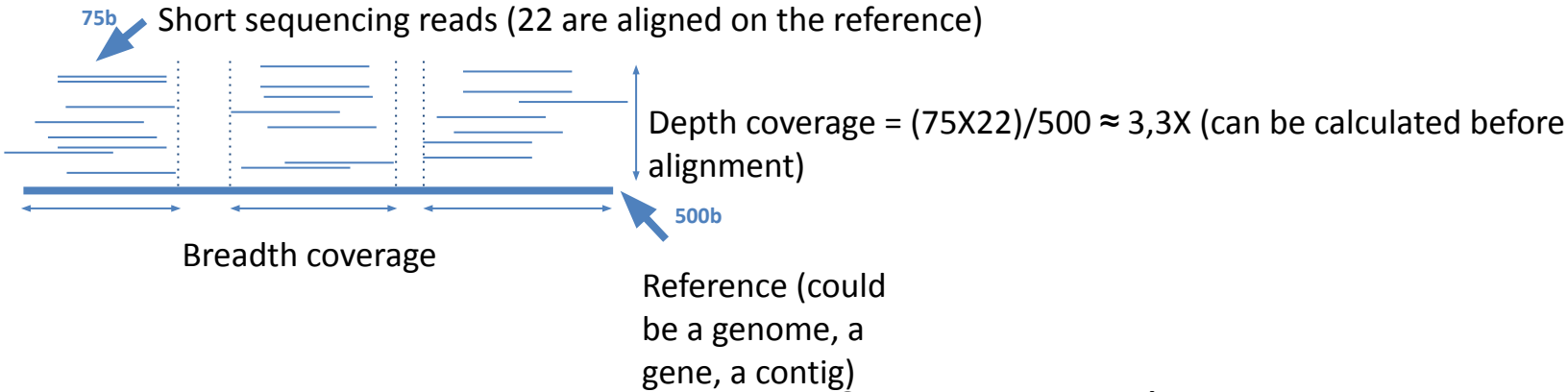
Coverage requirement (1/5)

- To detect a species based on marker genes ?
- To cover most of the genome to determine what part of the pangenome is covered by a sample?
- To perform an assembly from a metagenome ?



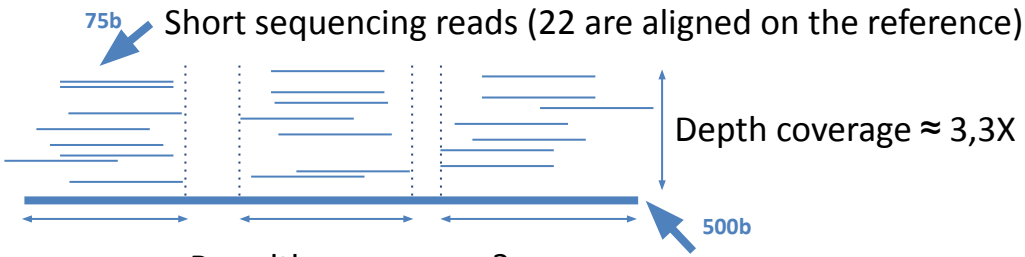
Coverage requirement (2/5)

- To detect a species based on marker genes ?
- To cover most of the genome to determine what part of the pangenome is covered by a sample ?
- To perform an assembly from a metagenome ?



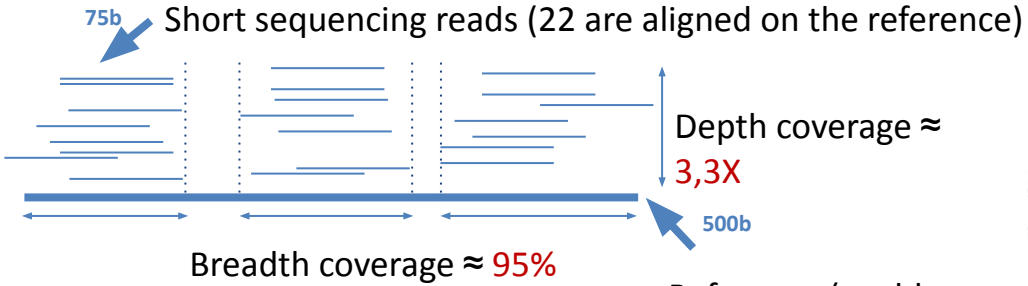
Coverage requirement (3/5)

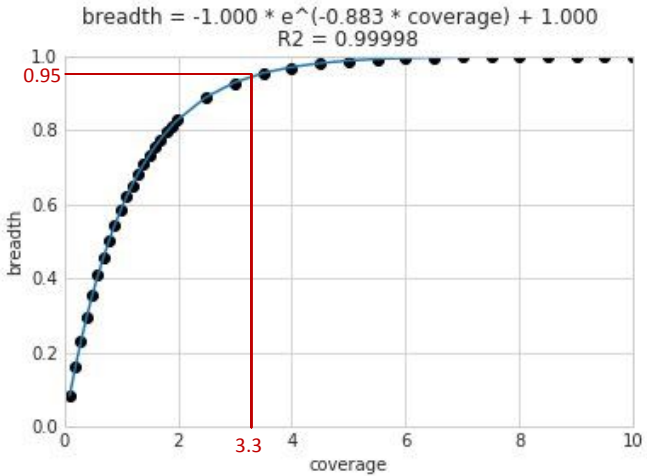
- To detect a species based on marker genes ?
- To cover most of the genome to determine what part of the pangenome is covered by a sample ?
- To perform an assembly from a metagenome ?



Coverage requirement (4/5)

- To detect a species based on marker genes ?
- To cover most of the genome to determine what part of the pangenome is covered by a sample ?
- To perform an assembly from a metagenome ?

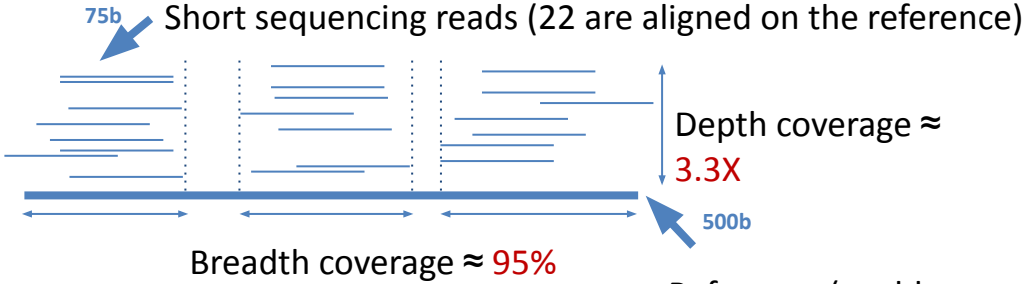


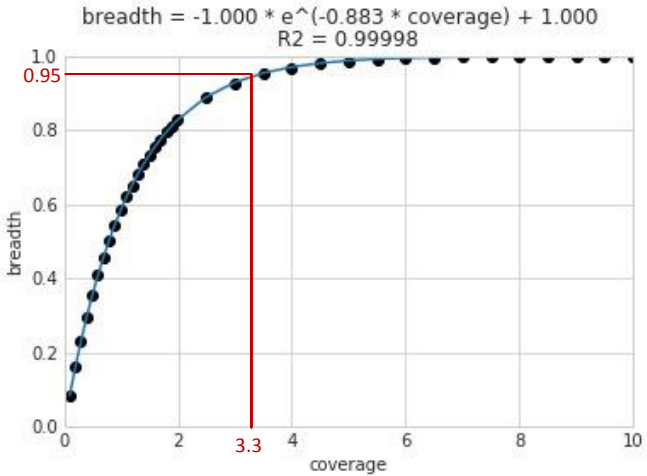


Lander and Waterman (1988)

Coverage requirement (5/5)

- To detect a species based on marker genes ? <0.1-3X
- To cover most of the genome to determine what part of the pangenome is covered by a sample ? 3X
- To perform an assembly from a metagenome ? 5-10X

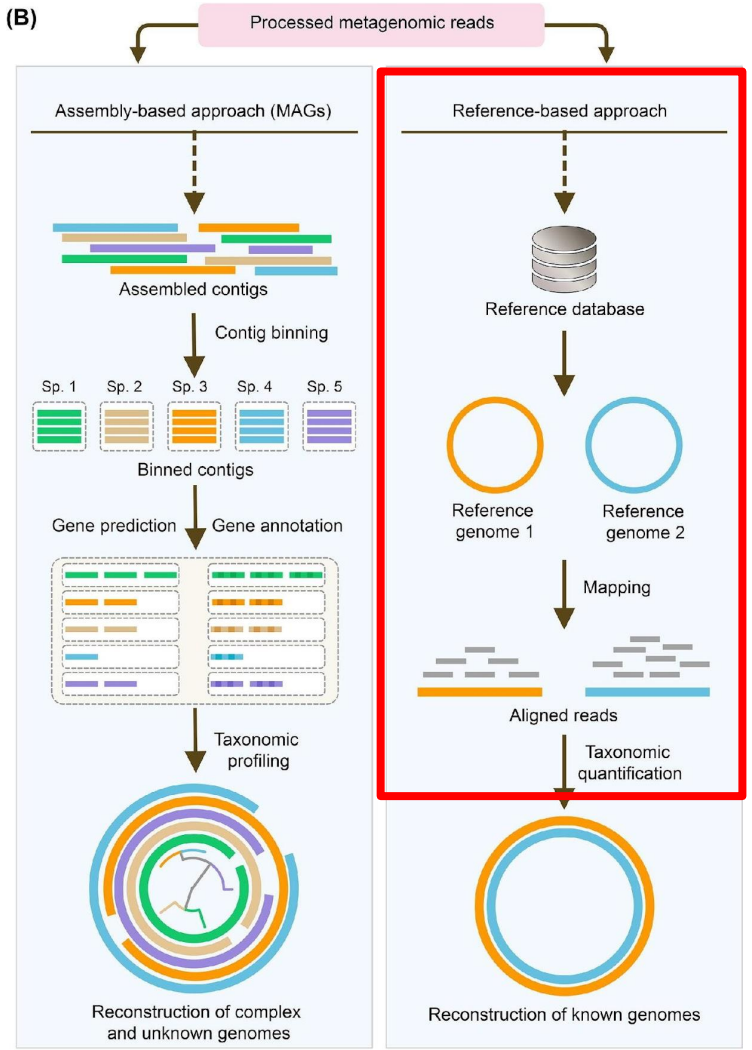




Lander and Waterman (1988)

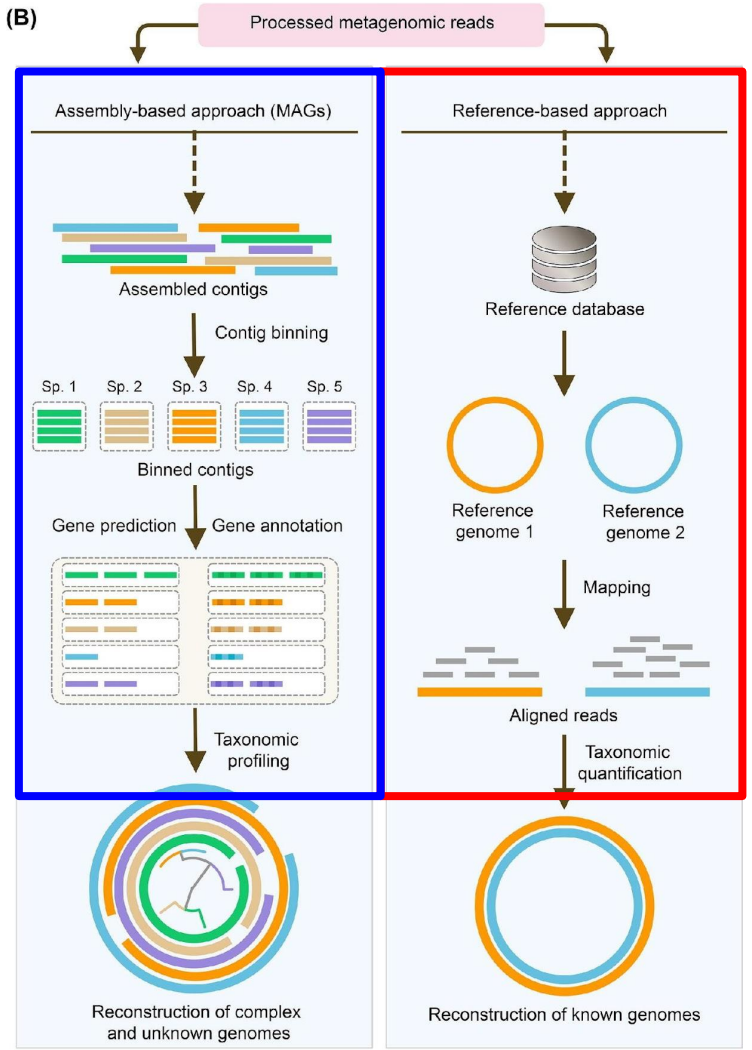
Challenges

- Complexity of the ecosystem
- Completeness of databases
- Sequencing depth
- Computational resources required



Challenges

- Complexity of the ecosystem
- Completeness of databases
- Sequencing depth
- Computational resources required



Taxonomic classification and quantification

Taxonomic classification caveats:

- Databanks
- K-mer choice (sensitivity / specificity)
- Allow a “fast” overview of your data
  - Contaminants?
  - Host reads?
  - unknown rate

2 kinds of approaches:

- kmer-based
- gene markers based

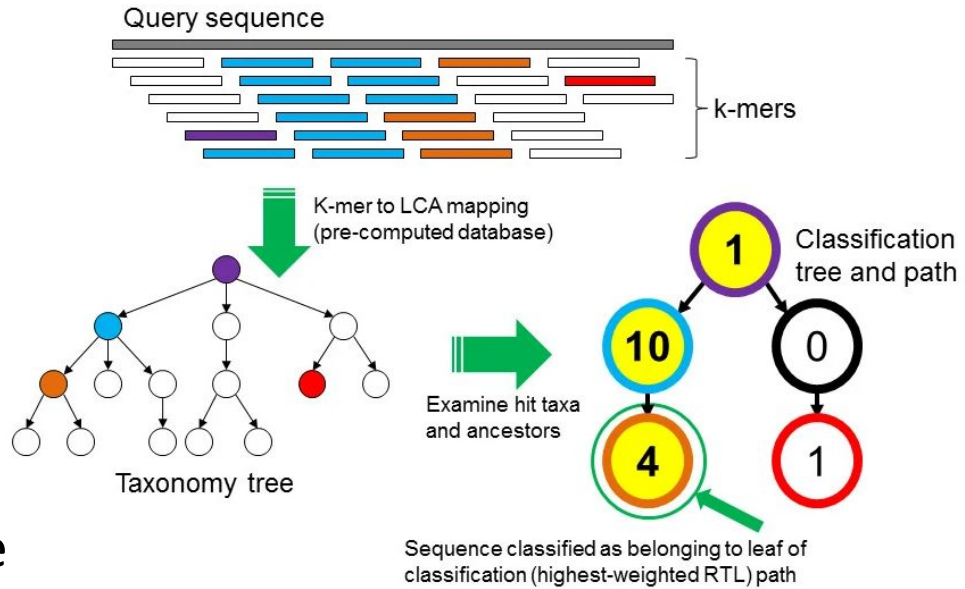
approachs	tools	galaxy	comments
kmer-based	Kraken2	✔	the reference, fast and efficient
	Bracken	✔	Bayesian Reestimation of Abundance from Kraken
	Centrifuge	✔	indexing scheme based on the Burrows-Wheeler transform (BWT) and the Ferragina-Manzini (FM) index, optimized specifically for the metagenomic classification problem
	Kaiju	✔	protein level
	Sylph	X	K-mer sketching. Work locally. Both fast and accurate
gene markers based	MetaPhlAn4	X version 2 :✔	MetaPhlAn relies on unique clade-specific marker genes identified from ~17,000 reference genomes (~13,500 bacterial and archaeal, ~3,500 viral, and ~110 eukaryotic)
	Meteor2	X	Based on environment specific gene catalogs (especially human gut).

Kraken2

- A very popular taxonomic affiliation tool.
- Very fast

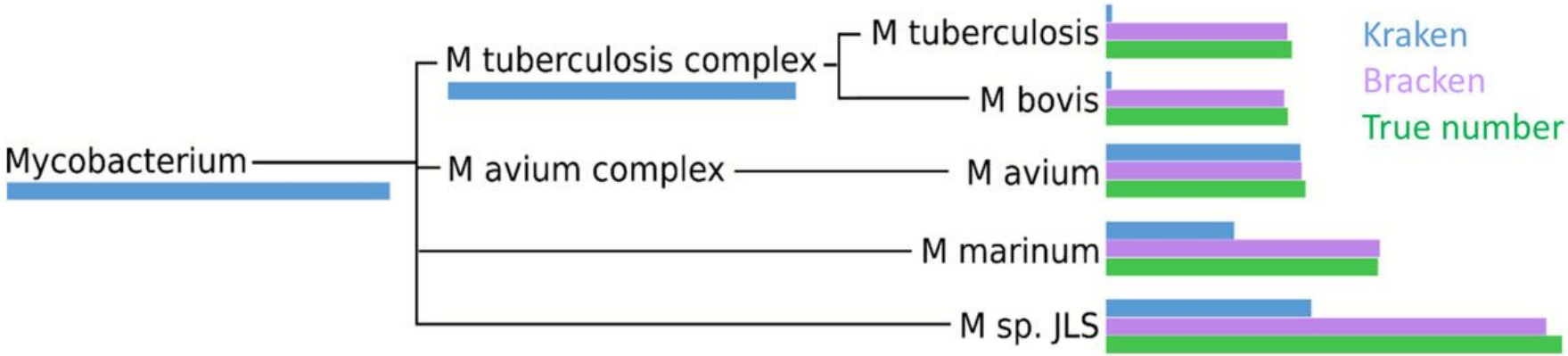
Method:

1. Chop genomes into k-mers and link to a taxonomic id.
2. Chop reads into k-mers and search for exact hits in database
3. Search for highest-weighted root-to-leaf paths and assign the taxonomic id of the lowest node to read



Braken

- Kraken classifies reads using the LCA approach
  - Some reads are shared
- Braken distributes abondancies from Kraken results using a Bayesian statistical method



Centrifuge

- Similar to Kraken but few differences :
  - Memory efficient (within species compression)
  - Allow multiple assignments per read
  - K-mer extension : a bit more accurate
  - Not as fast as Kraken

Kaiju

- An equivalent of Kraken, but with some particularities:
  - Database of proteic sequences
  - Supposed to be more sensitive
  - Translate reads in all six reading frames, split at stop codons

Kaiju databanks

Option	Description	Sequences*	RAM in GB (makedb)*
`refseq`	Completely assembled and annotated reference genomes of Archaea, Bacteria, and viruses from the NCBI RefSeq database.	63M	43 (55)
`progenomes`	Representative set of genomes from the proGenomes database and viruses from the NCBI RefSeq database.	41.8M	30 (35)
`viruses`	Only viruses from the NCBI RefSeq database.	0.37M	0.3 (0.3)
`plasmids`	Plasmid sequences from the NCBI RefSeq database.	2M	1.3 (2)
`fungi`	Fungi sequences from the NCBI RefSeq database.	3.2M	3 (4)
`nr`	Subset of NCBI BLAST *nr* database containing all proteins belonging to Archaea, Bacteria and Viruses.	196M	105 (175)
`nr_euk`	Like option `-s nr` and additionally include proteins from fungi and microbial eukaryotes, see taxon list in `bin/kaiju-taxonlistEuk.tsv`.	213M	117 (194)
`mar`	Protein sequences from all Mar databases. Subsets can be chosen by `mar_ref`, `mar_db`, or `mar_mag`.	32.6M	21 (27)
`rvdb`	Protein sequences from RVDB-prot	4.6M	4 (149)

Sylph

- Based on k-mer sketching to approximate ANI (Average Nucleotide Identify) calculation against reference genomes
- Easy to use
- Easy to install
- Works on a laptop
- Pretty accurate
- https://github.com/bluenote-1577/sylph
- Still on BioRiv : https://www.biorxiv.org/content/10.1101/2023.11.20.567879v2

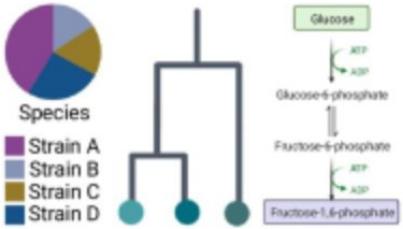
MetaPhlAn4

- Relies on :
  - 5.1M unique clade-specific marker genes identified
  - from ~1M microbial genomes
    - ~236,600 references
    - 771,500 metagenomic assembled genomes
  - spanning 26,970 species-level genome bins
  - 4,992 of them taxonomically unidentified at the species level
- associated to HUMAnN 3.0 for functional profiling (high coverage)
- StrainPhlAn for strain-level analyses (high coverage)
- PanPhlAn for pangenome-level analyses (high coverage)

Meteor2

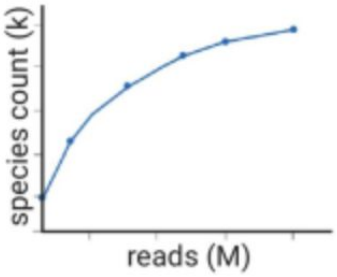
- Developed by MetaGenoPolis (INRAE)
- https://github.com/metagenopolis/meteor
- Relies on available gene catalogs :
    - human gut 10.4M of genes clustered in 1 990 species pangenome
    - human oral 8.4M of genes clustered in 853 species pangenome
    - cat gut 1.3M of genes clustered in 344 species pangenome
    - human skin 2.9M of genes clustered in 392 species pangenome
    - brown rat gut 5.9M of genes clustered in 1627 species pangenome
    - chicken gut 5.9M of genes clustered in 13.6M 2420 species pangenome
    - pig gut 9.3M of genes clustered in 1523 species pangenome
- Highly accurate quantification (unpublished)
- Able to remove host contaminations

Statistical analyses



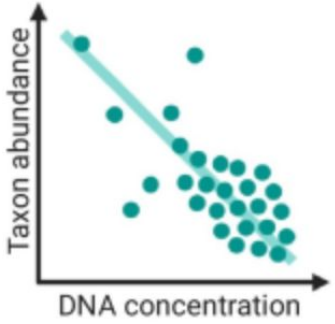
Taxonomic and functional profiling

Kraken2
Centrifuge
HUMAnN
MetaPhlAn
MEGAN
Kaiju



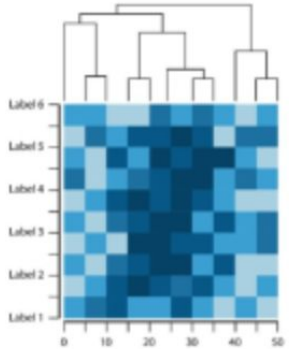
Post quality control

Sequencingeffort
Phyloseq



Contamination analysis

SourceTracker
Decontam



Statistical analysis

EdgeR
DESeq2
limma
LEfSe
MaAsLin
ANCOM-BC

⚠️ Contamination issues (1/2)

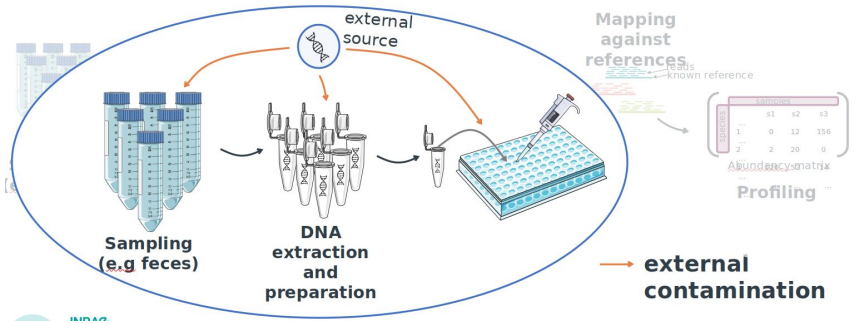
Host contaminations

- dilution effect (costly)
- ethical consideration (human)

tool	galaxy	comments
Kneaddata	X	remove rRNA and host (human and mouse) reads
SortmeRNA	✔️	remove rRNA reads, slow…

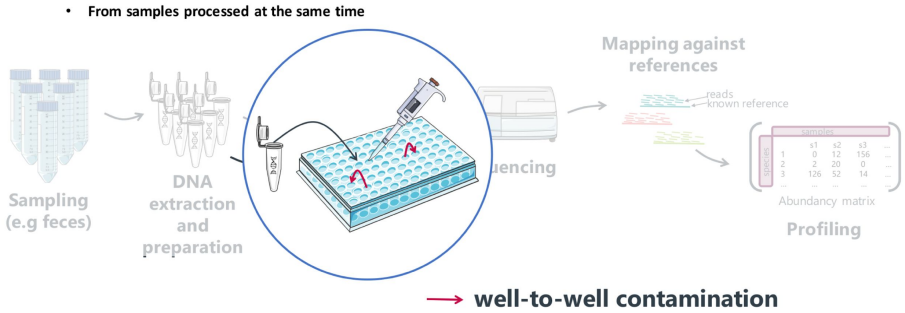
External contaminants QC:

- negative controls
- mapping on suspected contaminant
- taxonomic affiliation



⚠️ Contamination issues (2/2)

- **well-to-well contaminations :**
  - Overestimation of diversity
  - Can mute the main signal



  - **CroCoDeEL**
    - Find contamination pattern in gut microbiome study
    - Goulet et al. (JOBIM 2024, in preparation)
    - https://github.com/metagenopolis/CroCoDeEL
  - **SCRuB :**
    - Works across multiple ecosystems
    - Can decontaminate samples
    - Need blank controls

Metagenomics assembly

Metagenomics assembly

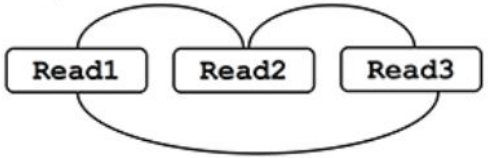
Objectives

- Reconstruct genes and organisms from complex mixtures
- Dealing with the ecosystem’s heterogeneity, multiple genomes at varying levels of abundance
- Limiting the reconstruction of chimeras

General assembly strategies

(a) Overlap, Layout, Consensus assembly

(i) Find overlaps



(ii) Layout reads

(iii) Build consensus

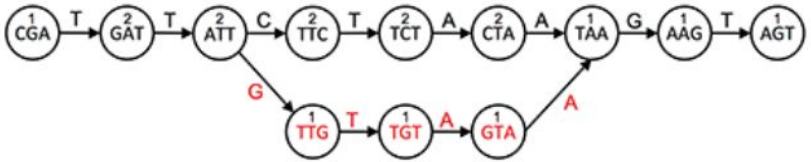
```
CGATTCTA
   TTCTAAGT
 GATTGTAA
---------
CGATTCTAAGT
```

(b) De Bruijn graph assembly

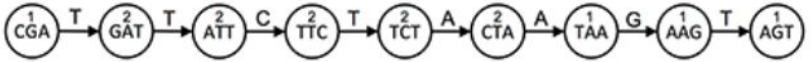
(i) Make kmers

```
Read1: TTCTAAGT    Read2: CGATTCTA    Read3: GATTGTAA
Kmers: TTC         Kmers: CGA         Kmers: GAT
        TCT                GAT                ATT
         CTA                ATT                TTG
          TAA                TTC                TGT
           AAG                TCT                GTA
            AGT                CTA                TAA
```

(ii) Build graph



(iii) Walk graph and output contigs



CGATTCTAAGT

Metagenome assembly specificity

- **Coverage** :
  - Widely different abundance levels of various species in a microbial sample result in a **highly nonuniform read coverage** across different genome
  - **Coverage** of most species in a typical metagenomic data set is **much lower**.
- **Interspecies repeats** :Various species within a microbial community often share highly conserved genomic regions in
- **Mixture** : many bacterial species in a microbial sample are represented by strain mixtures, that is, multiple related strains with varying abundances

Individual assembly or co-assembly ?

Usefull to reduce differences in coverage between samples

Pros of co-assembly	Cons of co-assembly
More data	Higher computational overhead
Better/longer assemblies	Risk of shattering the assembly
Access to lower abundant organisms	Risk of increased contamination

Co-assembly

Co-assembly is reasonable if:

- Same samples
- Same sampling event
- Longitudinal sampling of the same site
- Related samples

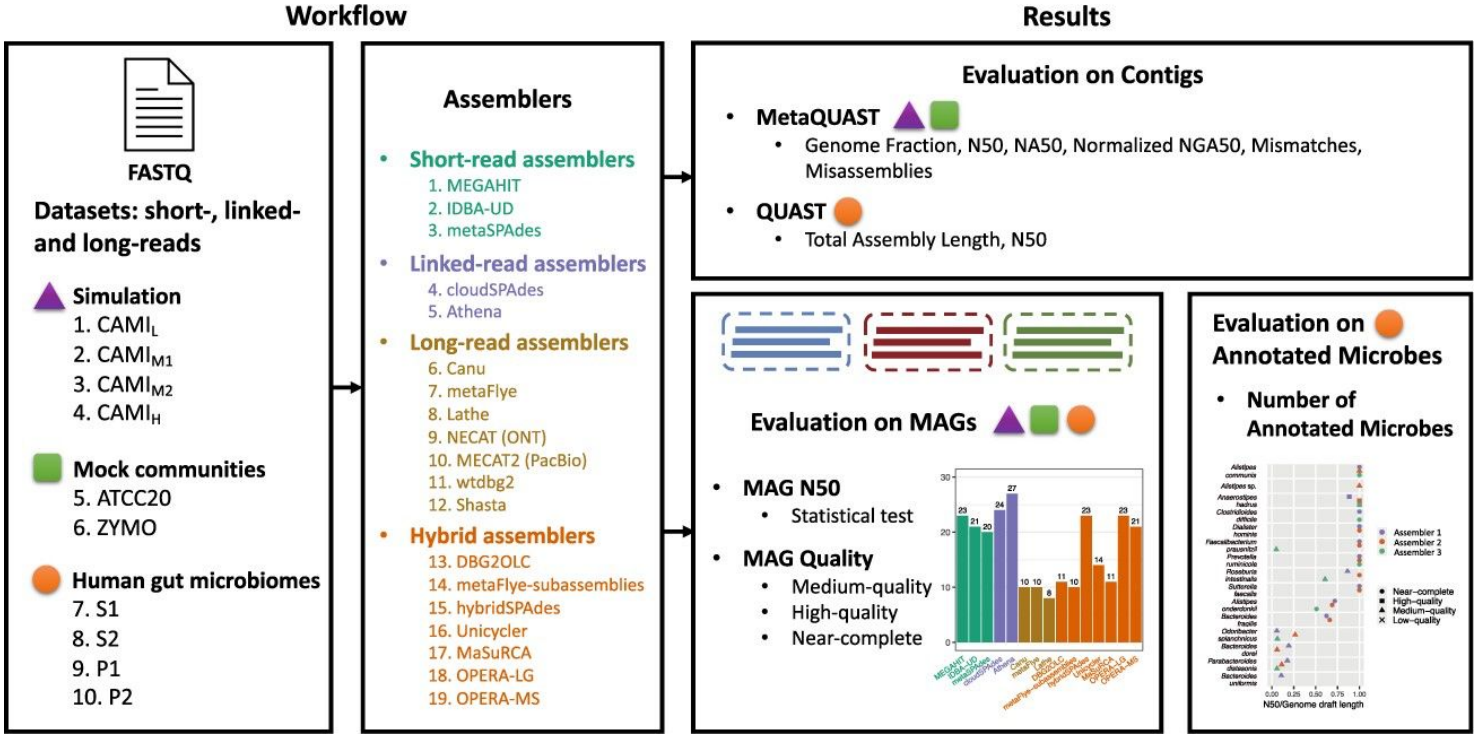
If it is not the case, individual assembly should be prefered. In this case, an extra step of **de-replication** should be used

Software

Metagenomic assembly software :

- Generic tool with a meta option :
  - SPAdes and metaSPAdes [Bankevich *et al.* 2012]
- Tools requiring less memory :
  - MEGAHIT [Li *et al.* 2015]
- Long read / Hybrid assemblies use different algorithms and strategies and are still a research question
  - metaFLYE, SPAdes …

Benchmark



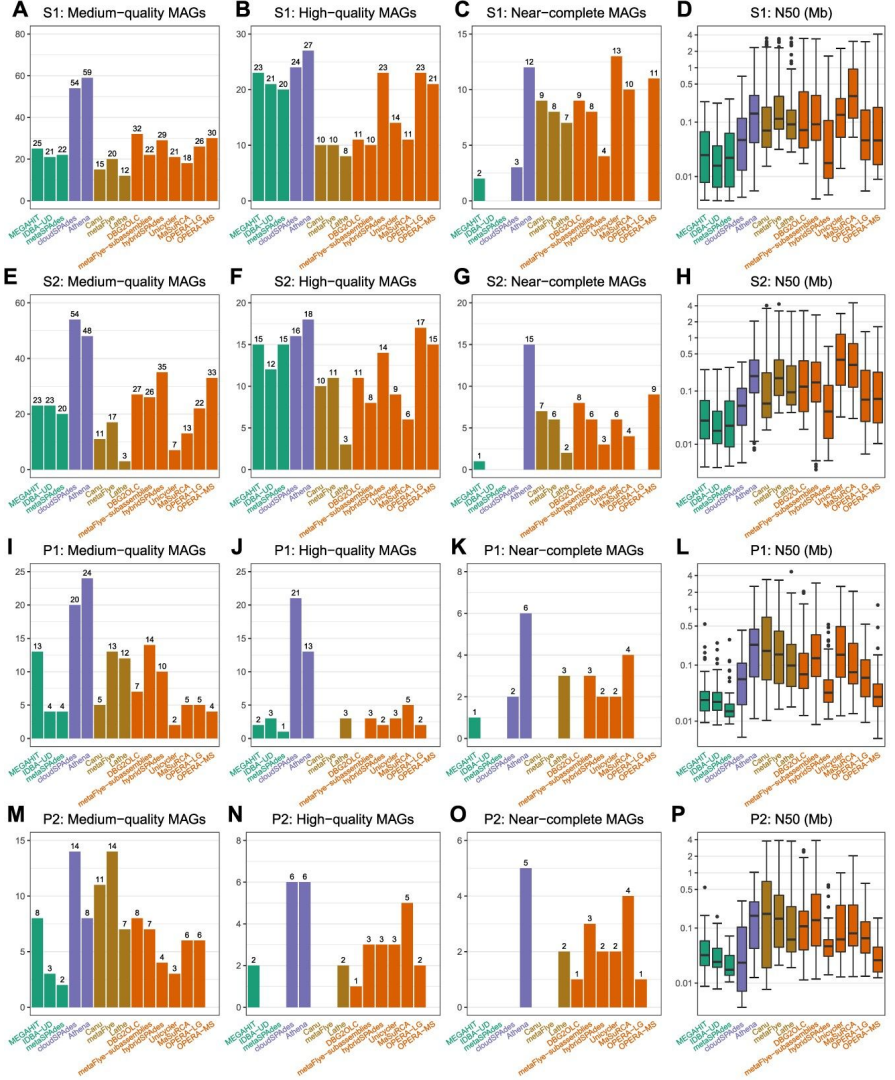
[Zhang *et al.* 2023]

Some results

*Our results showed that the **short-read assemblers** generated the **lowest contig contiguity and [Near Complete MAGs]**. **MEGAHIT** outperformed IDBA-UD and **metaSPAdes** on the **deeply sequenced datasets** (>100X), and **metaSPAdes** obtained better results than MEGAHIT and IDBA-UD on **low-complexity datasets** (depth < 100X).*

***Hybrid assemblies** demonstrated **higher** (or at least similar) **[Genome fraction] and [total assembly length]** than short- and long-read assemblies, and generated **higher [High Quality] and [Near Complete]** than long-read assemblies*

Short-read assemblers were unable to assemble any genomes of low-abundance microbes



Assessment of assembly quality

MetaQUAST [Mikheenko *et al.* 2015] to evaluate and compare metagenome assemblies

MetaQUAST :

- De novo metagenomic assembly evaluation
- [Optionally] identify reference genomes from the content of the assembly
- Reference-based evaluation
- Filtering so-called misassemblies based on read mapping
- Report and visualization

De novo metrics

Evaluation of the assembly based on:

- Number of contigs greater than a given threshold (0, 1kb, …)
- Total / thresholded assembly size
- Largest contig size
- N50 : the sequence length of the shortest contig at 50% of the total assembly length, equivalent to a median of contig lengths. (N75 idem, for 75%)
- L50 : the number of contigs at 50% of the total assembly length. (L75 idem, for 75%)

Reference-based metrics

- Metrics based on the comparison with reference genomes.
- Reference genomes are given by the user or automatically constitued by MetaQuast based on comparison of rRNA genes content of the assembly and a reference database (Silva).
- Complete genomes are then automatically downloaded.

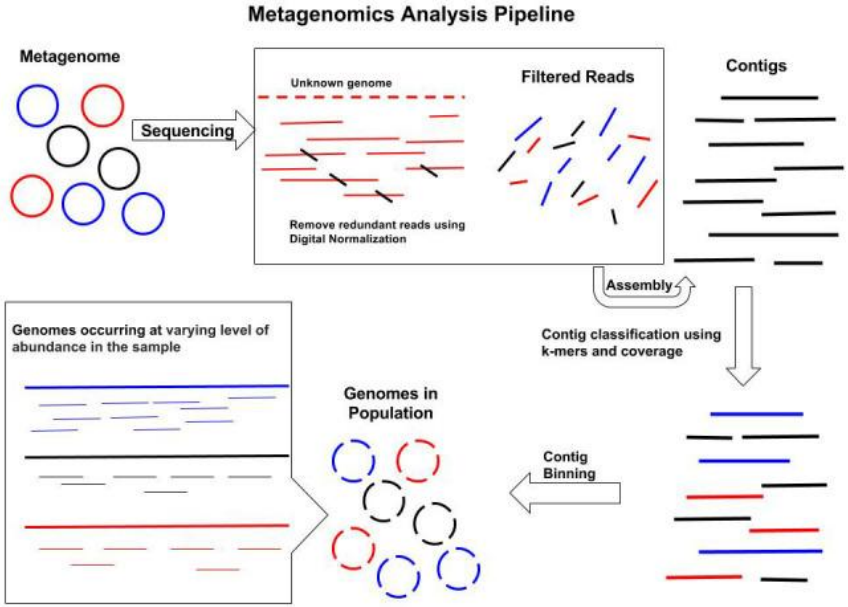
Binning

Binning :

- grouping *similar* contigs together into **metagenomic assembled genomes (MAG)**
- **In other words :**
  - **A MAG represents a microbial genome by a group of sequences from genome assembly with similar characteristics**

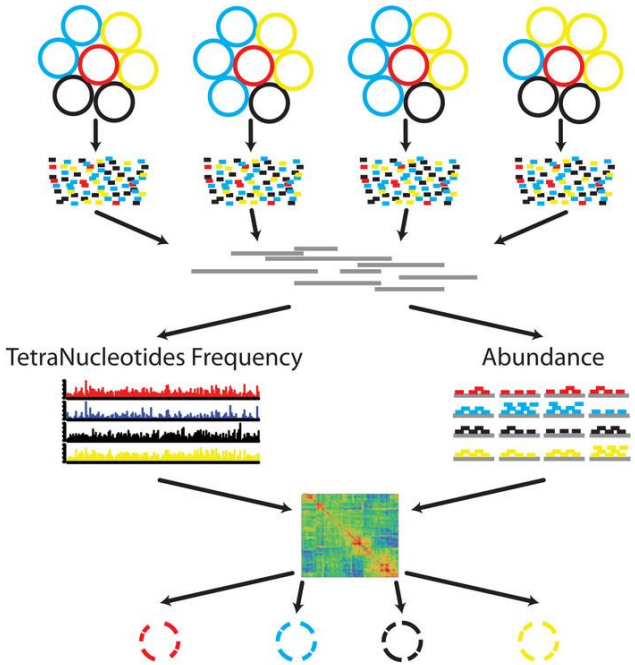
Binning is a good compromise when the assembly of whole genomes is not feasible.

Concoct, SemiBin



Approach

MetaBAT [Khang *et al.* 2019] is a tool for reconstructing genomes from complex microbial communities.



Preprocessing

1. Samples from multiple sites or times
2. Metagenome libraries
3. Initial de-novo assembly using the combined library

MetaBAT

4. Calculate TNF for each contig
5. Calculate Abundance per library for each contig
6. Calculate the pairwise distance matrix using pre-trained probabilistic models
7. Forming genome bins iteratively

Bins evaluation

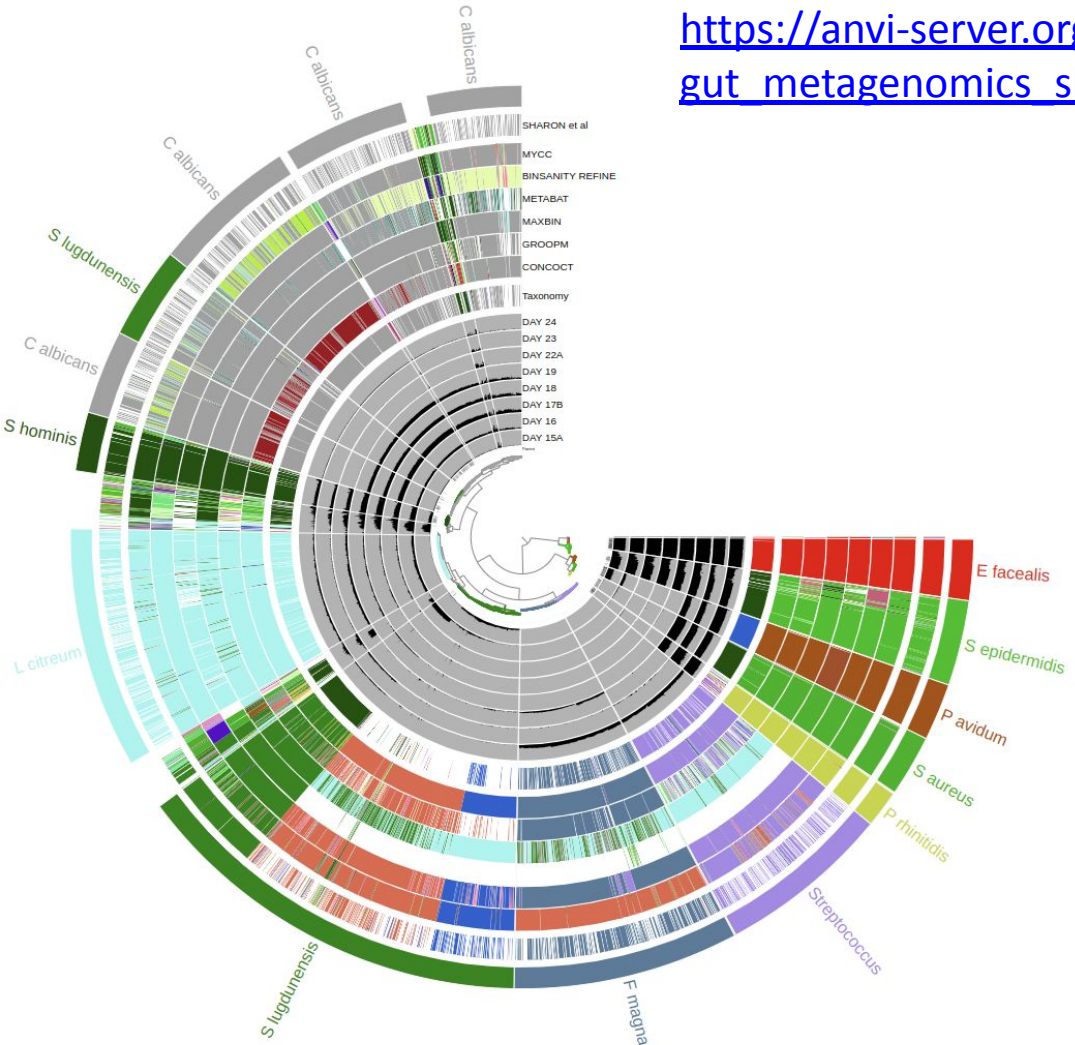
For the evaluation of bins, we will use *completeness* and *contamination* estimated by CheckM [Parks *et al.* 2015]

- Use of collocated sets of genes that are ubiquitous and single-copy within a phylogenetic lineage.
  - *completeness*: estimated completeness of genome as determined from the presence/absence of marker genes and the expected colocalization of these genes
  - *contamination*: estimated contamination of genome as determined by the presence of multi-copy marker genes and the expected colocalization of these genes
  - *strain heterogeneity*: estimated strain heterogeneity as determined from the number of multi-copy marker pairs which exceed a specified amino acid identity threshold (default = 90%). High strain heterogeneity suggests the majority of reported contamination is from one or more closely related organisms (i.e. potentially the same species), while low strain heterogeneity suggests the majority of contamination is from more phylogenetically diverse sources

Threshold depends on the type of assembly.
On metagenomics , usually : completeness >90% , < 5% conta, <= 0.5 hetereogenity
Pasolli *et al. 2019,Bowers et al., 2017*

Anvi'o

https://anvi-server.org/merenlab/infant_gut_metagenomics_sharon_et_al_2015



What’s next ?

Galaxy training on

- Assembly of metagenomics data
  - assembly, QC, QC with reference
- Binning of metagenomics data
  - Binning with metabat2

References

- Bankevich A, Nurk S, Antipov D, Gurevich AA, Dvorkin M, Kulikov AS, et al. SPAdes: A new genome assembly algorithm and its applications to single-cell sequencing. Journal of Computational Biology. 2012;19:455–77. doi:10.1089/cmb.2012.0021.
- Li D, Liu C-M, Luo R, Sadakane K, Lam T-W. MEGAHIT: An ultra-fast single-node solution for large and complex metagenomics assembly via succinct de bruijn graph. Bioinformatics. 2015;31:1674–6.
- Zhenmiao Zhang, Chao Yang, Werner Pieter Veldsman, Xiaodong Fang, Lu Zhang, Benchmarking genome assembly methods on metagenomic sequencing data, Briefings in Bioinformatics, Volume 24, Issue 2, March 2023, bbad087, https://doi.org/10.1093/bib/bbad087
- Mikheenko A, Saveliev V, Gurevich A. MetaQUAST: evaluation of metagenome assemblies. Bioinformatics. 2015;32:1088–90. doi:10.1093/bioinformatics/btv697.
- Kang DD, Li F, Kirton E, Thomas A, Egan R, An H, Wang Z. MetaBAT 2: an adaptive binning algorithm for robust and efficient genome reconstruction from metagenome assemblies. PeerJ. 2019 Jul 26;7:e7359. doi: 10.7717/peerj.7359. PMID: 31388474; PMCID: PMC6662567.
- 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 Research. 2015;25:1043–55. doi:10.1101/gr.186072.114.

Advances and challenges in metatranscriptomic analysis, 2019 [40]

Current concepts, advances, and challenges in deciphering the human microbiota with metatranscriptomics, 2023 [41]

Take home message

- Shotgun metagenomics is still an ongoing active bioinformatics research field
- Numerous software dedicated to assembly, binning, functional annotation are actively developed
- Depending on the ecosystem , one can have different approaches :
  - mapping on a reference database
  - assembly and mapping
- The biological question must determine the analysis

Need help?

- Any question at help-migale@inrae.fr
- Need tool? https://migale.inrae.fr/ask-tool
- Need more resources? https://migale.inrae.fr/ask-resources
- Need more help than just one question? https://migale.inrae.fr/ask-data-analysis

References

1. Escobar-Zepeda A, Vera-Ponce de León A, Sanchez-Flores A. The road to metagenomics: From microbiology to DNA sequencing technologies and bioinformatics. Frontiers in genetics. 2015;6:348.

2. Breitwieser FP, Lu J, Salzberg SL. A review of methods and databases for metagenomic classification and assembly. Briefings in bioinformatics. 2019;20:1125–36.

3. Yang C, Chowdhury D, Zhang Z, Cheung WK, Lu A, Bian Z, et al. A review of computational tools for generating metagenome-assembled genomes from metagenomic