Reference-guided metagenomic assembly and strain-level analyses

What is reference-guided assembly?

- An assembler with a *strong assumption* that genomes in your metagenome look a lot like those that are in a reference database.

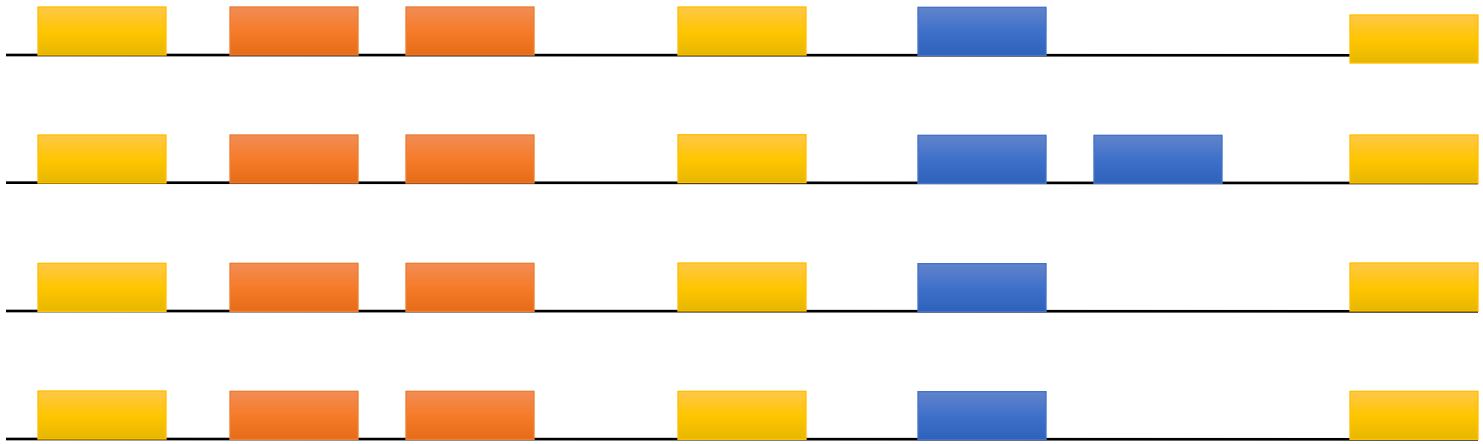
- If this is a reasonable assumption, **proceed with caution:**
  - Hint: rearrangements, horizonal gene transfer, and duplications are common!
- If this is not a reasonable assumption (viral genomes, soil samples), **think de novo assembly:**
  - Megahit
  - MetaSpades

Microbial genomes evolve over time

- *The presence of two or more homologous sequences within a single genome might reflect the acquisition of DNA sequence from a foreign source rather than the duplication of a resident gene.*

- Thus, since we do not know the origin a priori, we refer to these potential paralogs or xenologs as **synologs** (Lerat et al 2005).

Ubiquitous sequence fragments

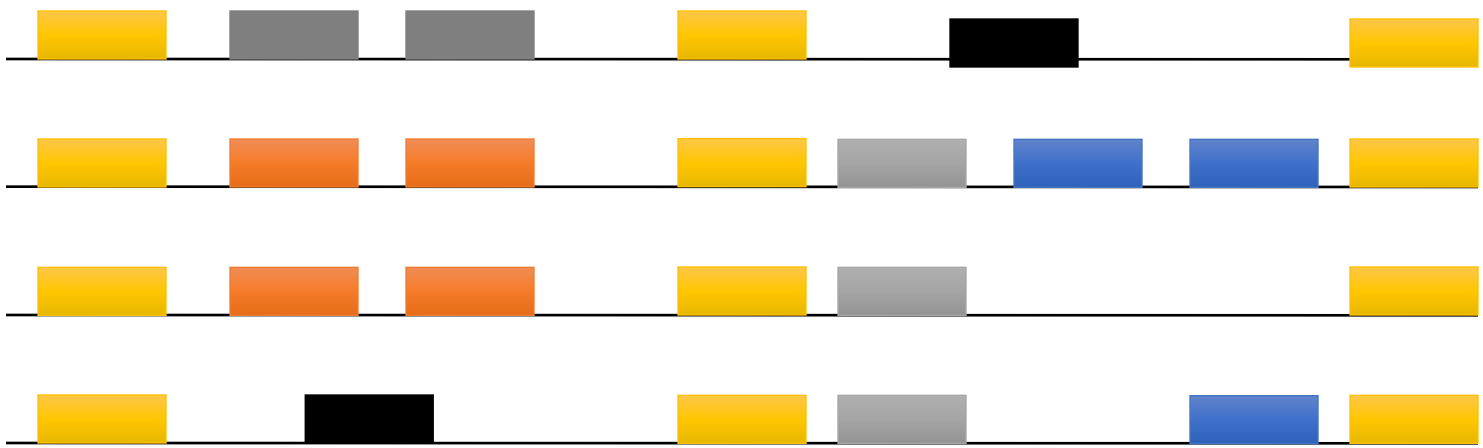


= ubiquitous with synologs (constant)

= core genome

= ubiquitous with synologs (variable)

Non-Ubiquitous sequence fragments



= core genome

= genome specific

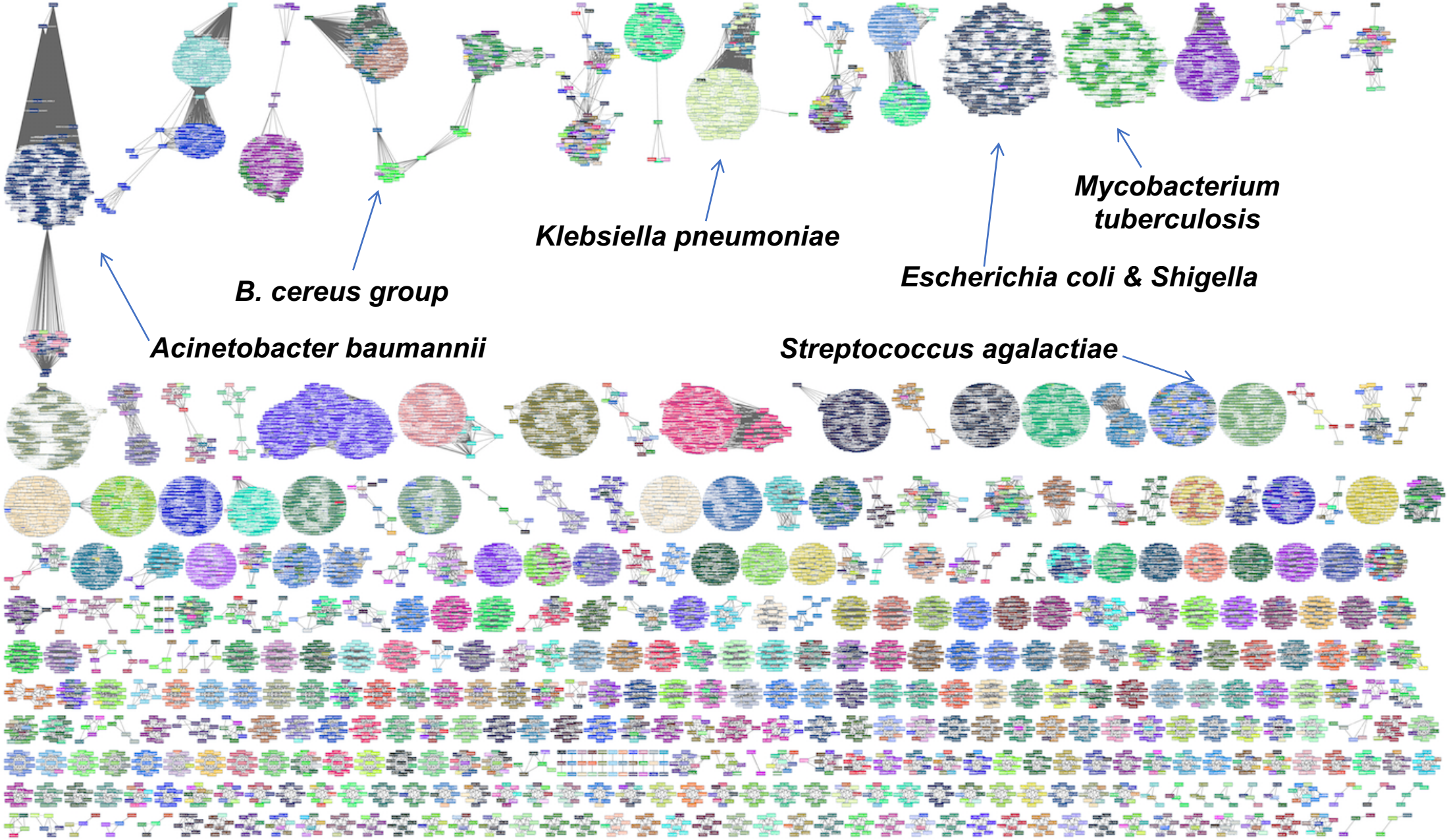
= singletons

= non-ubiquitous with synologs (constant)

= non-ubiquitous with synologs (variable)

= non-ubiquitous without synologs

Powdered water -- to drink, just add water.

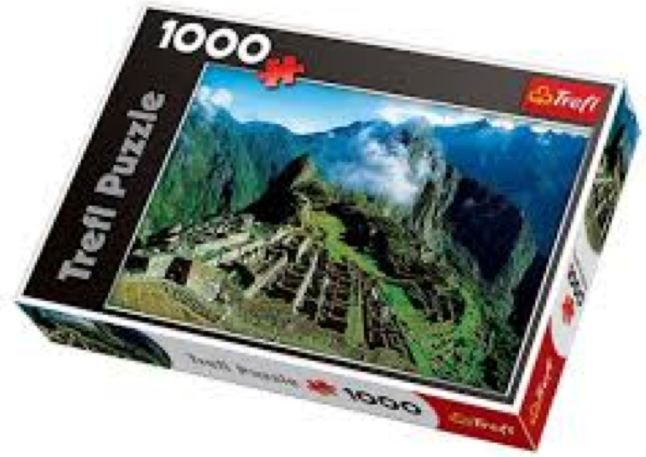

Mycobacterium tuberculosis
Klebsiella pneumoniae
Escherichia coli & Shigella
B. cereus group
Acinetobacter baumannii
Streptococcus agalactiae

Reference-guided metagenomic assembly

Ready, set, go!

Reference-guided genome assembly

- Reconstructing the original DNA sequence aligning reads to a genome.
- Intuitively like a puzzle
- But we have the box!



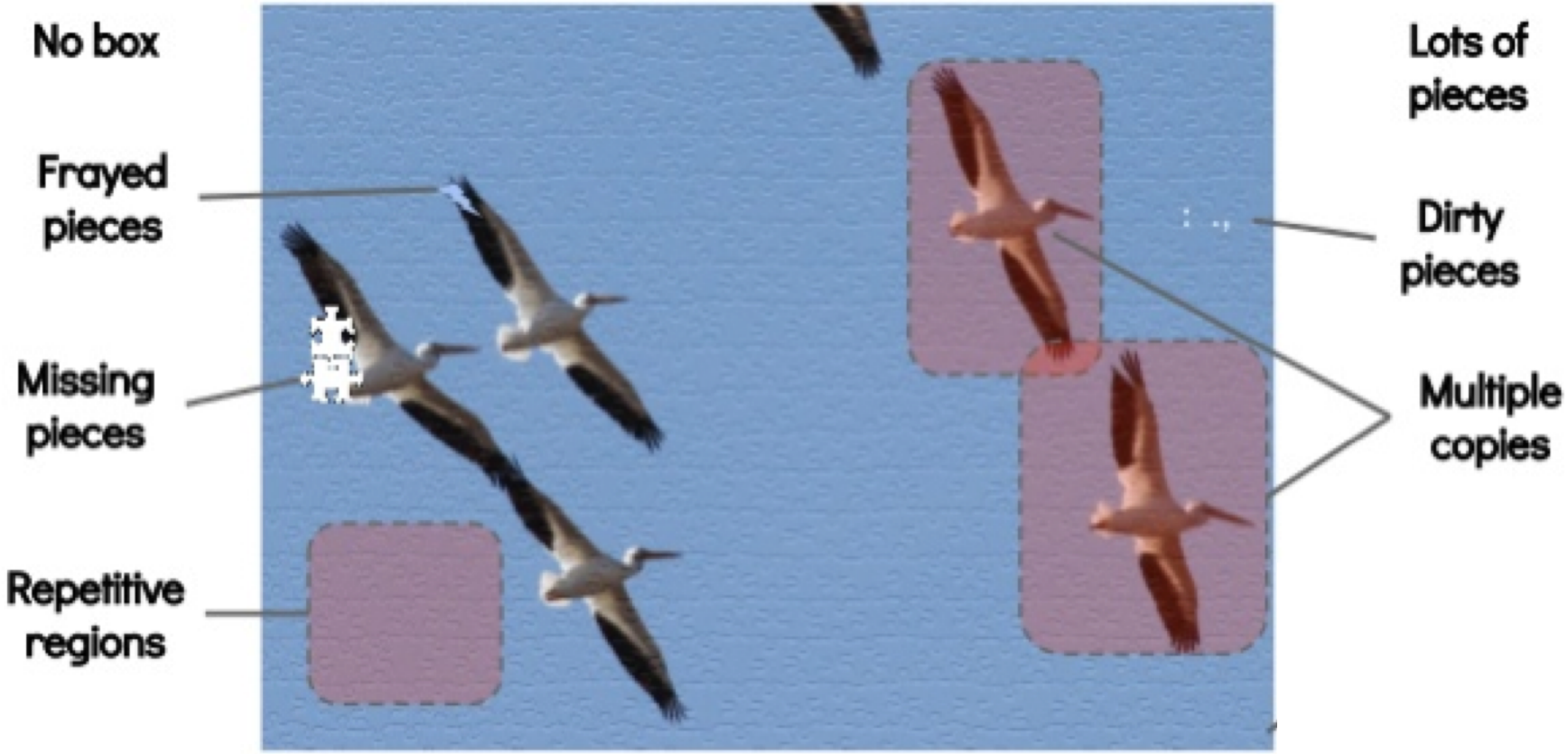
Reference-guided metagenome assembly

- Reconstructing original DNA sequences aligning reads to a set of genomes.
- Intuitively like multiple puzzles
- But we need to find the boxes!

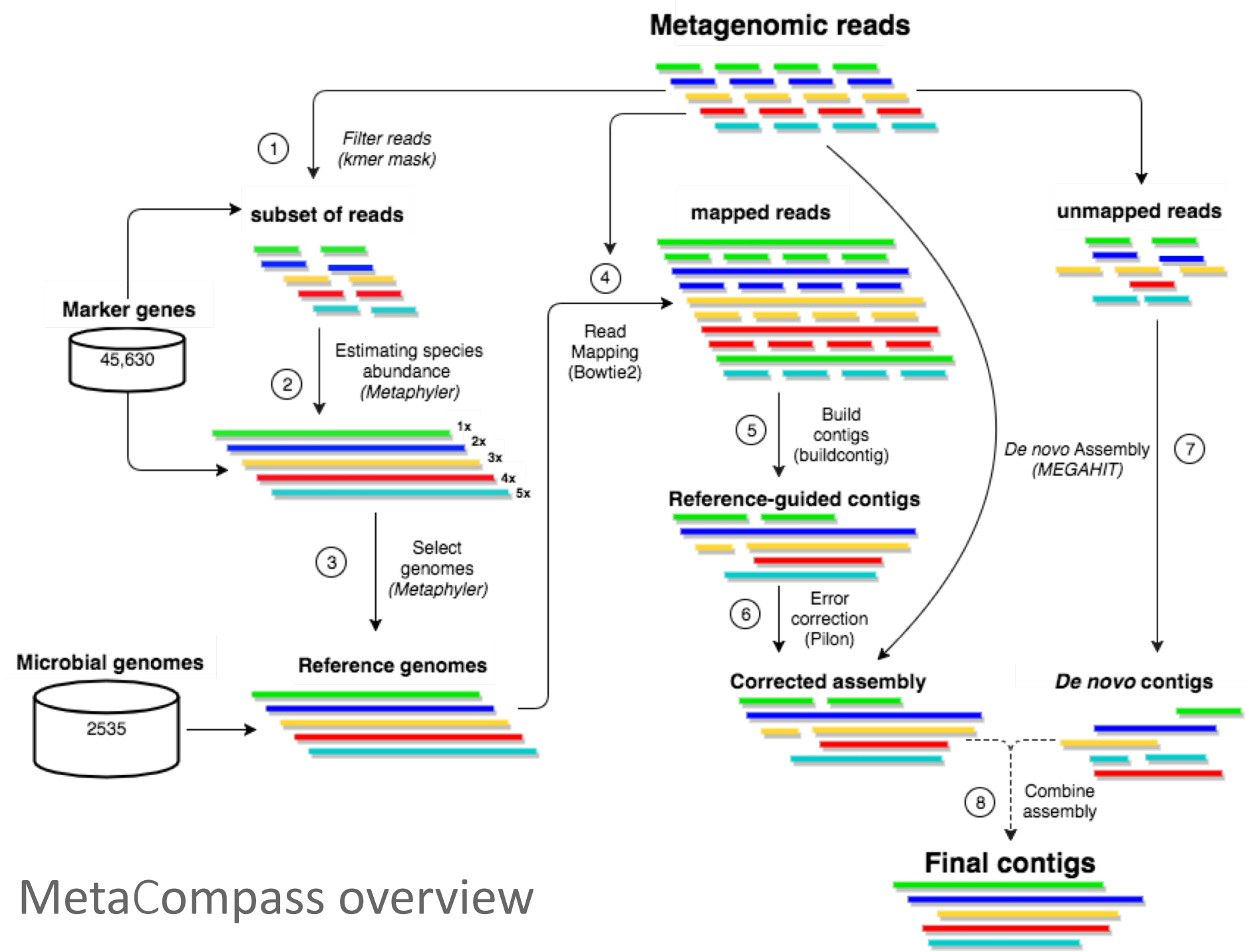
Reference-guided metagenome assembly

- Step 1: Find the puzzle boxes (reference selection)
- Step 2: Bin pieces into the right boxes (read mapping)
- Step 3: Solve each puzzle (assembly)

What makes a puzzle hard?

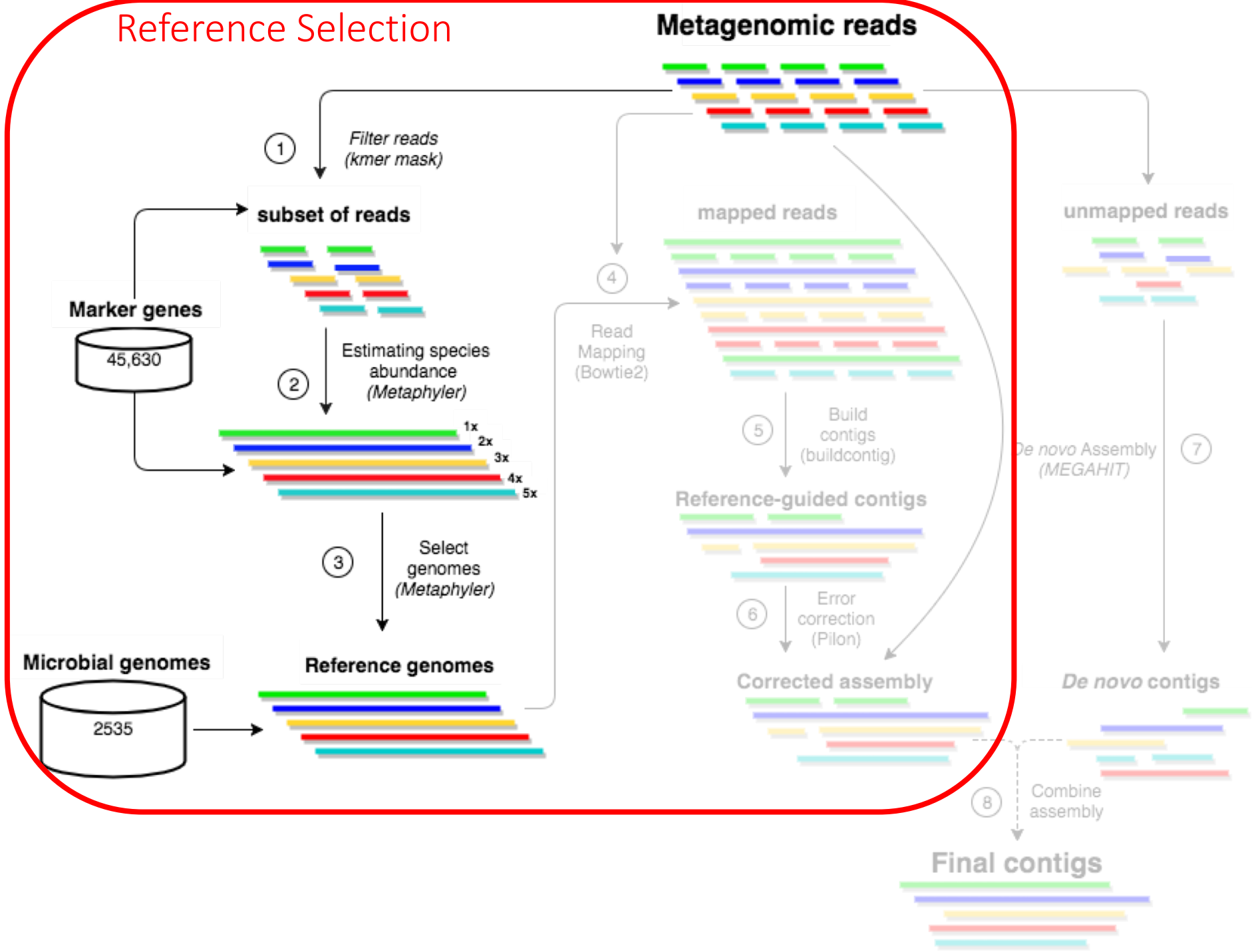


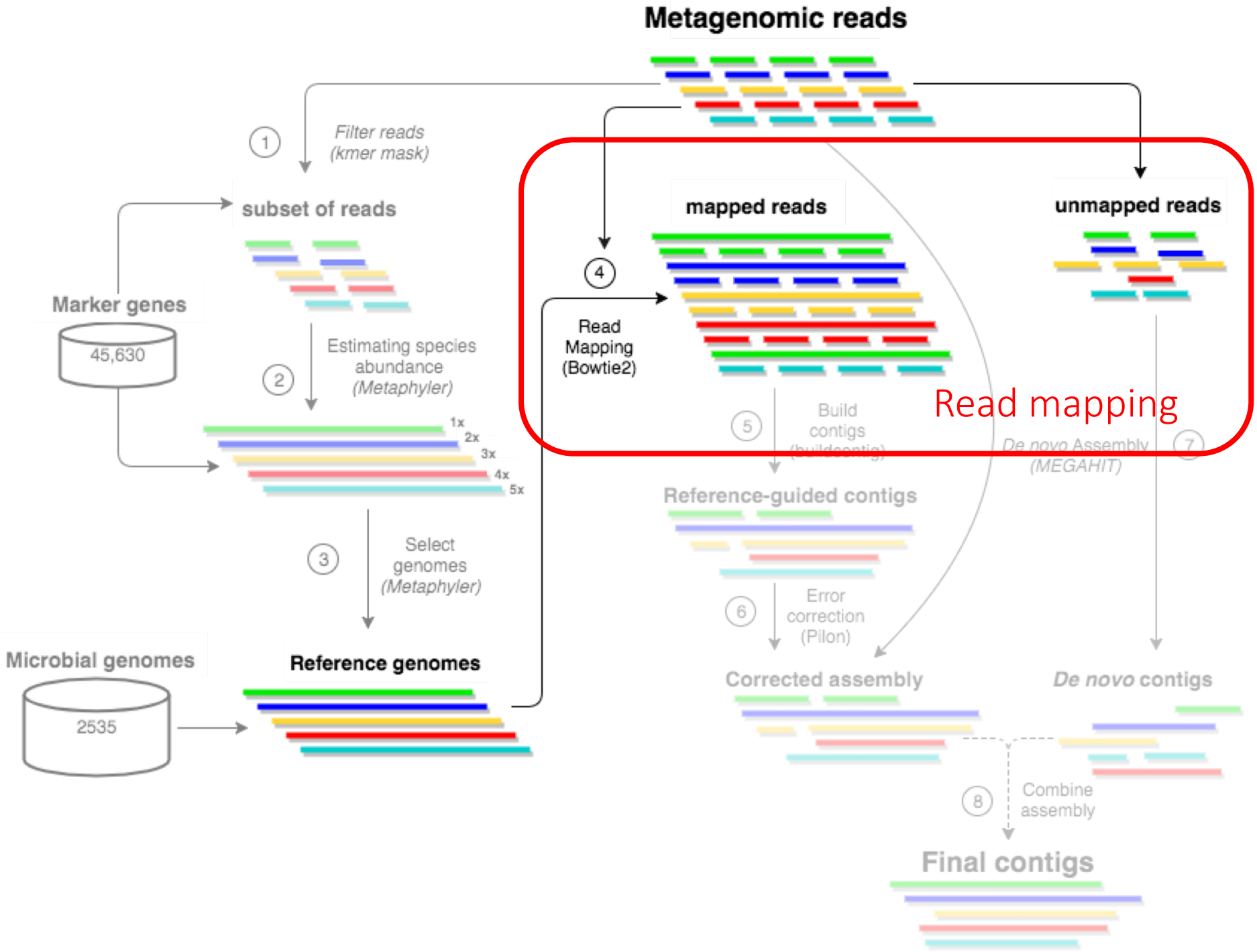
Slide credit → Torsten Seemann

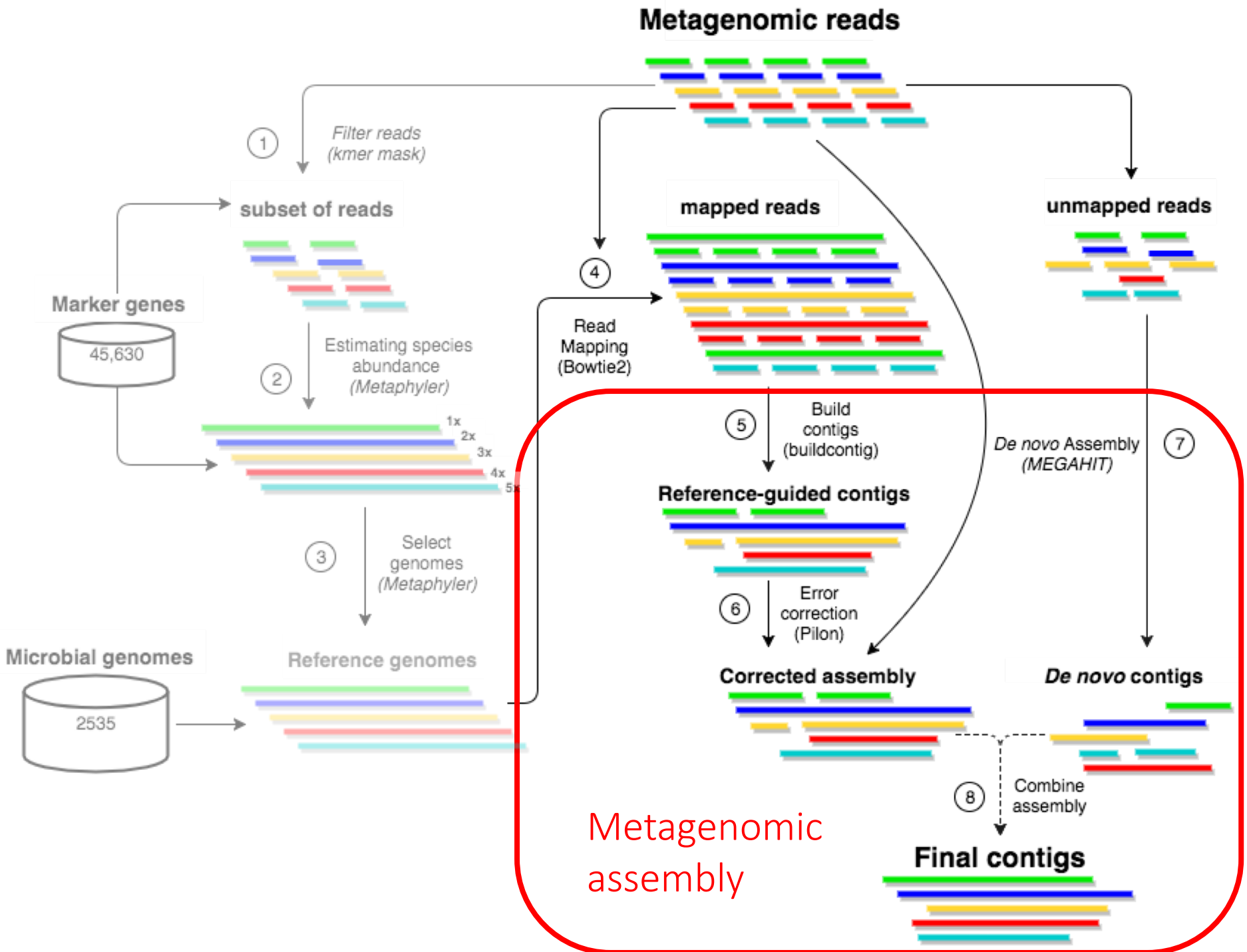


MetaCompass overview

Reference Selection




Metagenomic reads
1
Filter reads
(kmer mask)
subset of reads
Marker genes
45,630
2
Estimating species
abundance
(Metaphyler)
1x
2x
3x
4x
5x
3
Select
genomes
(Metaphyler)
Microbial genomes
2535
Reference genomes
4
Read
Mapping
(Bowtie2)
mapped reads
unmapped reads
Read mapping
5
Build
contigs
(buildcontig)
Reference-guided contigs
6
Error
correction
(Pilon)
Corrected assembly
De novo Assembly
(MEGAHIT)
7
De novo contigs
8
Combine
assembly
Final contigs


Metagenomic reads
1
Filter reads
(kmer mask)
subset of reads
Marker genes
45,630
2
Estimating species
abundance
(Metaphyler)
1x
2x
3x
4x
5x
3
Select
genomes
(Metaphyler)
Microbial genomes
2535
Reference genomes
4
Read
Mapping
(Bowtie2)
mapped reads
unmapped reads
5
Build
contigs
(buildcontig)
Reference-guided contigs
6
Error
correction
(Pilon)
Corrected assembly
De novo Assembly
(MEGAHIT)
7
De novo contigs
8
Combine
assembly
Final contigs
Metagenomic
assembly

Choose your own adventure: how shall we identify the reference genomes?

1. Universal marker gene- based approaches (MetaPhlan, MetaPhyler, etc)

2. MinHash based approaches (Mash, SourMash, etc)

3. Kmer + LCA based approaches (Clark, Kraken, etc)

- http://tiny.cc/stamps19

Liu *et al. BMC Genomics* 2011, **12**(Suppl 2):S4
http://www.biomedcentral.com/1471-2164/12/S2/S4

BMC Genomics

PROCEEDINGS Open Access

Accurate and fast estimation of taxonomic profiles from metagenomic shotgun sequences

Bo Liu[1,2], Theodore Gibbons[1,3], Mohammad Ghodsi[1,2], Todd Treangen[1], Mihai Pop[1,2,3*]

From IEEE International Conference on Bioinformatics and Biomedicine 2010
Hong Kong, P. R. China. 18-21 December 2010

Ondov *et al. Genome Biology* (2016) 17:132
DOI 10.1186/s13059-016-0997-x

Genome Biology

SOFTWARE Open Access

Mash: fast genome and metagenome distance estimation using MinHash

Brian D. Ondov[1], Todd J. Treangen[1], Páll Melsted[2], Adam B. Mallonee[1], Nicholas H. Bergman[1], Sergey Koren[3] and Adam M. Phillippy[3*]

Abstract

Mash extends the MinHash dimensionality-reduction technique to include a pairwise mutation distance and *P* value significance test, enabling the efficient clustering and search of massive sequence collections. Mash reduces large sequences and sequence sets to small, representative sketches, from which global mutation distances can be rapidly estimated. We demonstrate several use cases, including the clustering of all 54,118 NCBI RefSeq genomes in 33 CPU h; real-time database search using assembled or unassembled Illumina, Pacific Biosciences, and Oxford Nanopore data; and the scalable clustering of hundreds of metagenomic samples by composition. Mash is freely released under a BSD license (https://github.com/marbl/mash).

Keywords: Comparative genomics, Genomic distance, Alignment, Sequencing, Nanopore, Metagenomics

Wood and Salzberg *Genome Biology* 2014, **15**:R46
http://genomebiology.com/2014/15/3/R46

Genome Biology

METHOD Open Access

Kraken: ultrafast metagenomic sequence classification using exact alignments

Derrick E Wood[1,2*] and Steven L Salzberg[2,3]

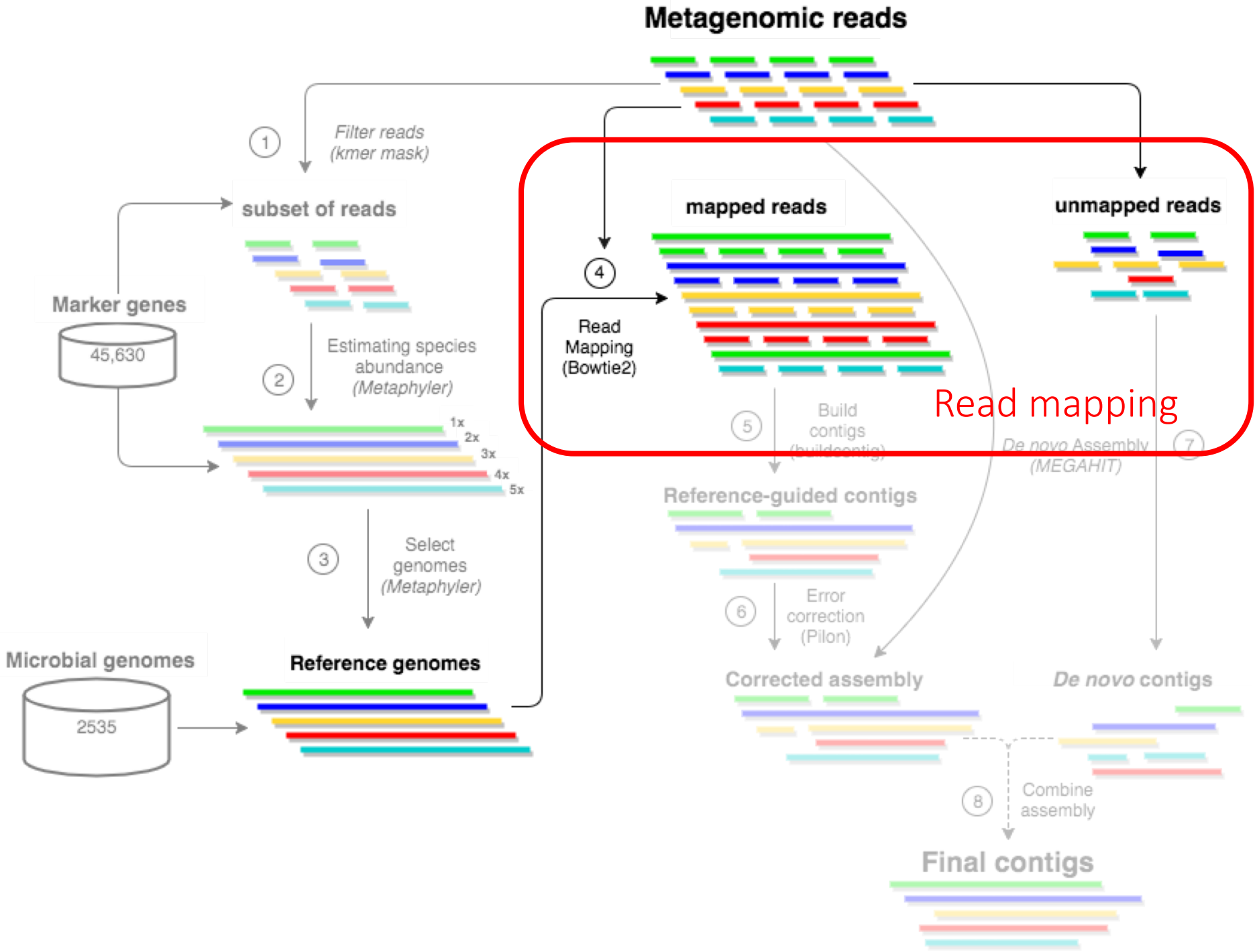
Abstract

Kraken is an ultrafast and highly accurate program for assigning taxonomic labels to metagenomic DNA sequences. Previous programs designed for this task have been relatively slow and computationally expensive, forcing researchers to use faster abundance estimation programs, which only classify small subsets of metagenomic data. Using exact alignment of *k*-mers, Kraken achieves classification accuracy comparable to the fastest BLAST program. In its fastest mode, Kraken classifies 100 base pair reads at a rate of over 4.1 million reads per minute, 909 times faster than Megablast and 11 times faster than the abundance estimation program MetaPhlAn. Kraken is available at http://ccb.jhu.edu/software/kraken/.

Keywords: metagenomics, sequence classification, sequence alignment, next-generation sequencing, microbiome

Survey results:

- https://ql.tc/axwUgB


Metagenomic reads
1
Filter reads
(kmer mask)
subset of reads
Marker genes
45,630
2
Estimating species
abundance
(Metaphyler)
1x
2x
3x
4x
5x
3
Select
genomes
(Metaphyler)
Microbial genomes
2535
Reference genomes
4
Read
Mapping
(Bowtie2)
mapped reads
unmapped reads
Read mapping
5
Build
contigs
(buildcontig)
Reference-guided contigs
6
Error
correction
(Pilon)
Corrected assembly
De novo Assembly
(MEGAHIT)
7
De novo contigs
8
Combine
assembly
Final contigs

Software | Open Access

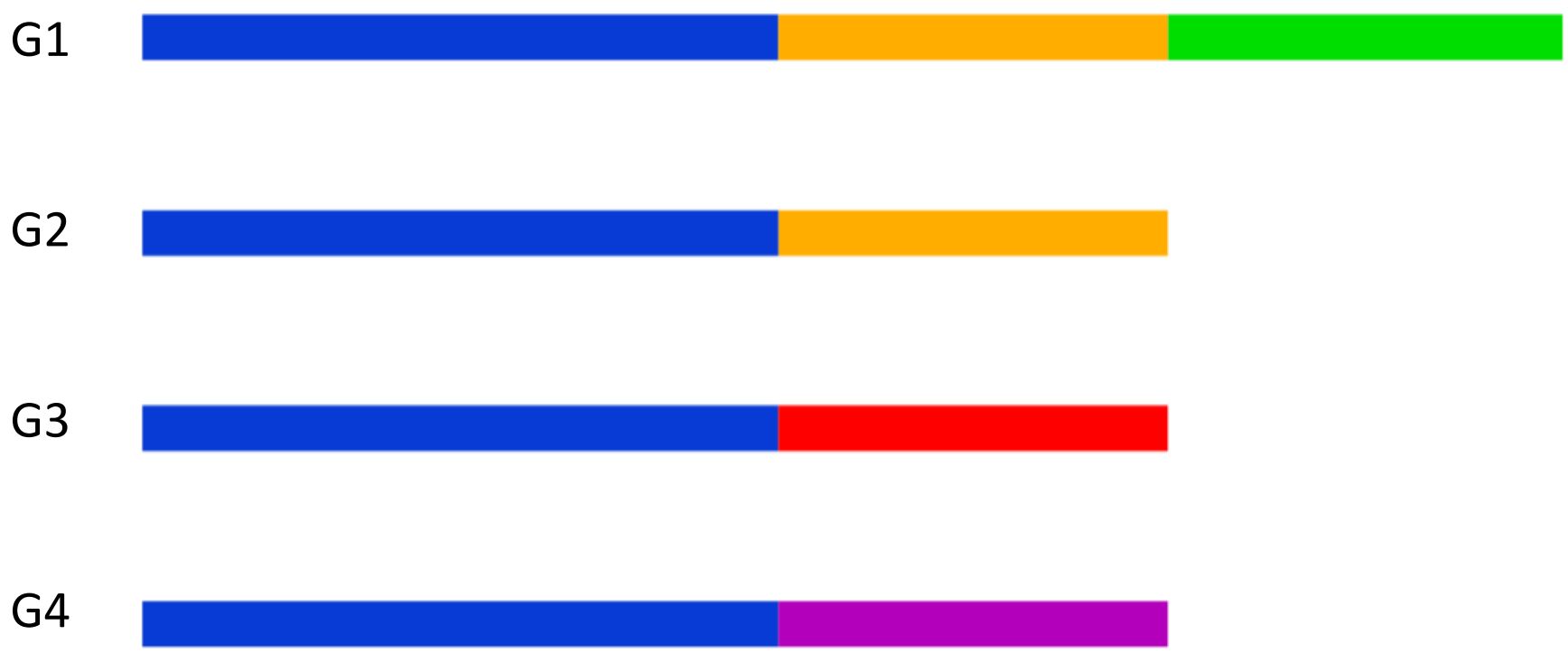
Ultrafast and memory-efficient alignment of short DNA sequences to the human genome

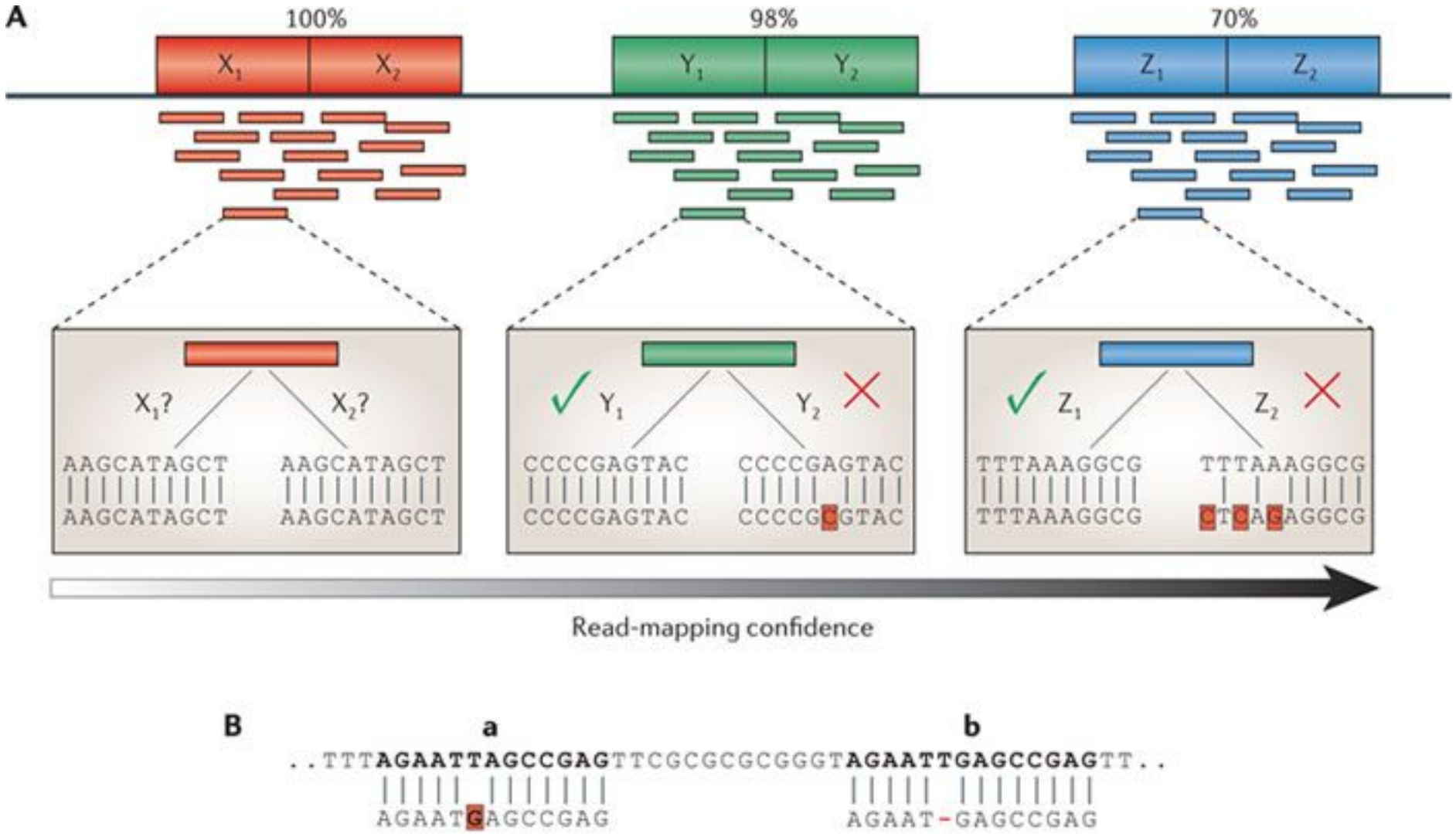
Ben Langmead ✉, Cole Trapnell, Mihai Pop and Steven L Salzberg

Genome Biology 2009 **10**:R25

https://doi.org/10.1186/gb-2009-10-3-r25

Received: 21 October 2008 | **Accepted:** 4 March 2009 | **Published:** 4 March 2009


G1
G2
G3
G4



Treangen and Salzberg, Nature Reviews Genetics, 2015

Nature Reviews | Genetics

Choose your own adventure: how shall we map reads to the recruited genomes?

1. All → map all reads to every equally good mapping location

2. Random → randomly assign reads amongst equally good mapping location

3. Depth → genome with highest depth of coverage takes all of the reads that map to it

4. Breadth → genome with highest breadth of coverage takes all the reads that map to it

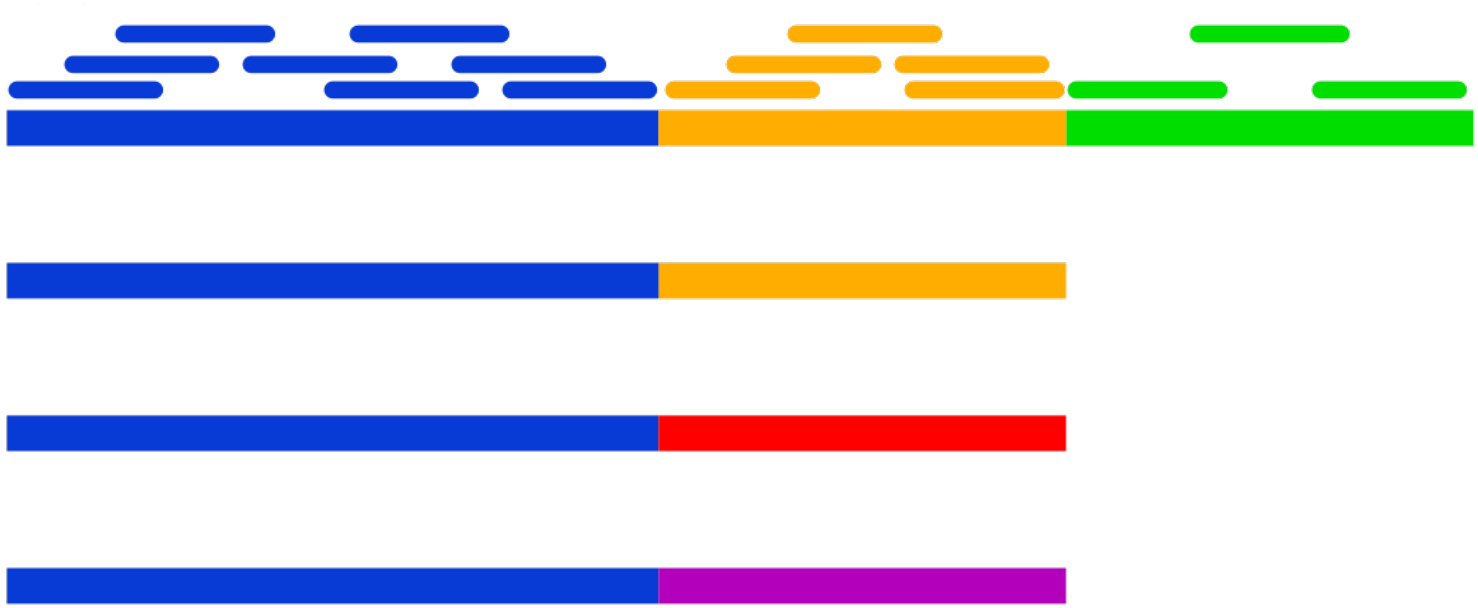
- http://tiny.cc/stamps19

READ MAPPING: How to place reads?

Random assignment “coin flip”

Rank by depth of coverage

Rank by minimum set cover



Survey results:

- https://ql.tc/axwUgB

Minimum set cover?

- Given a set of elements U { 1 , 2 , . . . , n } and a collection S of m sets whose union equals the universe, the set cover problem is to identify the smallest sub-collection of S whose union equals the universe.

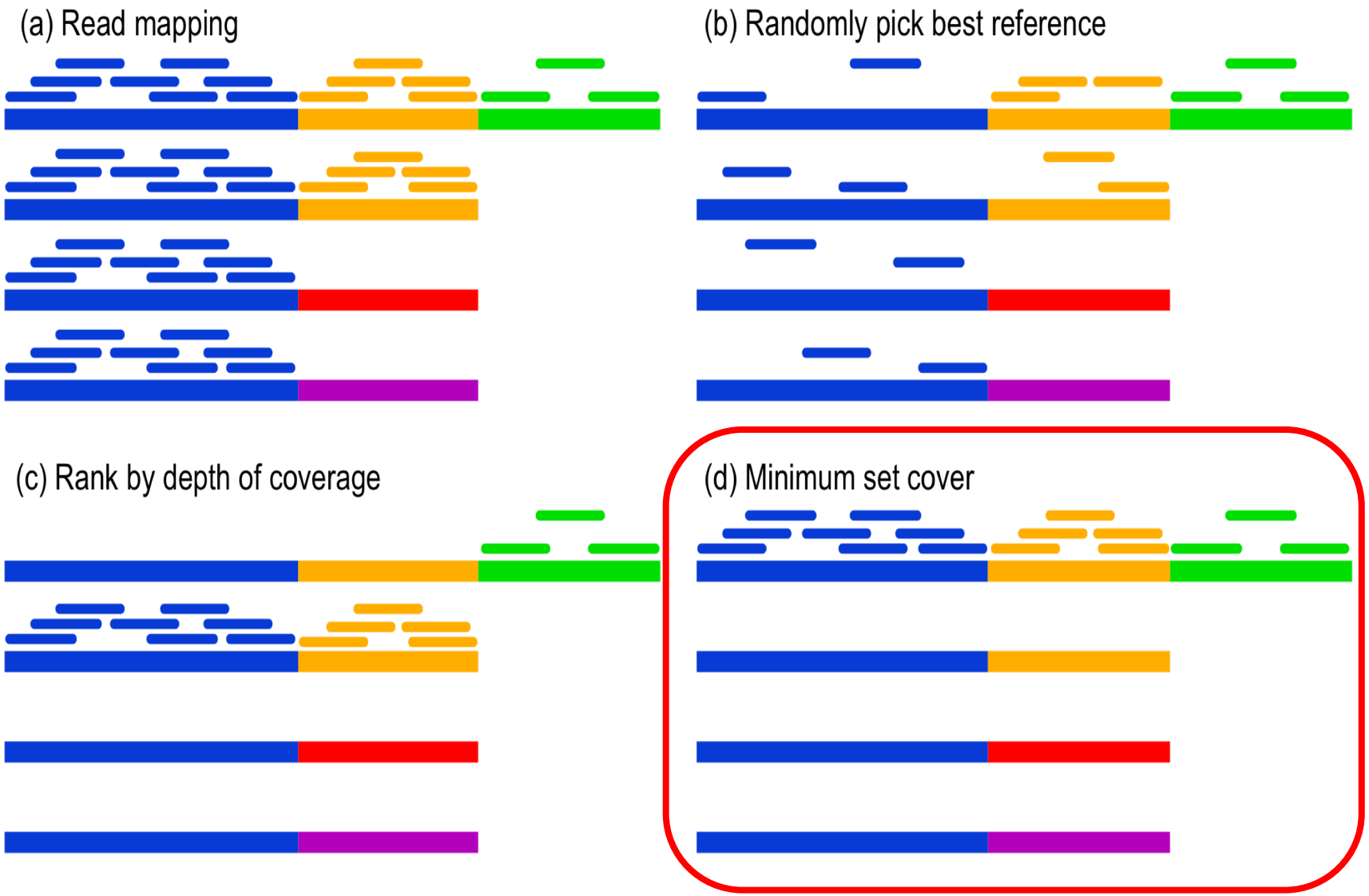
- For example, consider the universe U = { 1 , 2 , 3 , 4 , 5 } and the collection of sets S = { { 1 , 2 , 3 } , { 2 , 4 } , { 3 , 4 } , { 4 , 5 } } the union of S is U.

- The minimum set cover is the smallest number of m sets that cover U:

{ { 1 , 2 , 3 } , { 4 , 5 } }

- For reference selection, it's the smallest number of genomes that cover all of the input reads.

Read mapping selection: Minimum set cover



```
                                  C
Reads      TGCACGGATG   TGCATGCACG
       TTAATGCACG   TG-ATGCATG
  TGGATTAATG   TGGATG-ATG  C
  TGGATTCATGCATGGATGCATGCATGCACG   Reference
  TGGATTAATGCACGGATG-ATGCACTGCACG  Contig
```

Min. depth of coverage:2
Min. length:10

De novo assembly using MEGAHIT

Evaluation Datasets

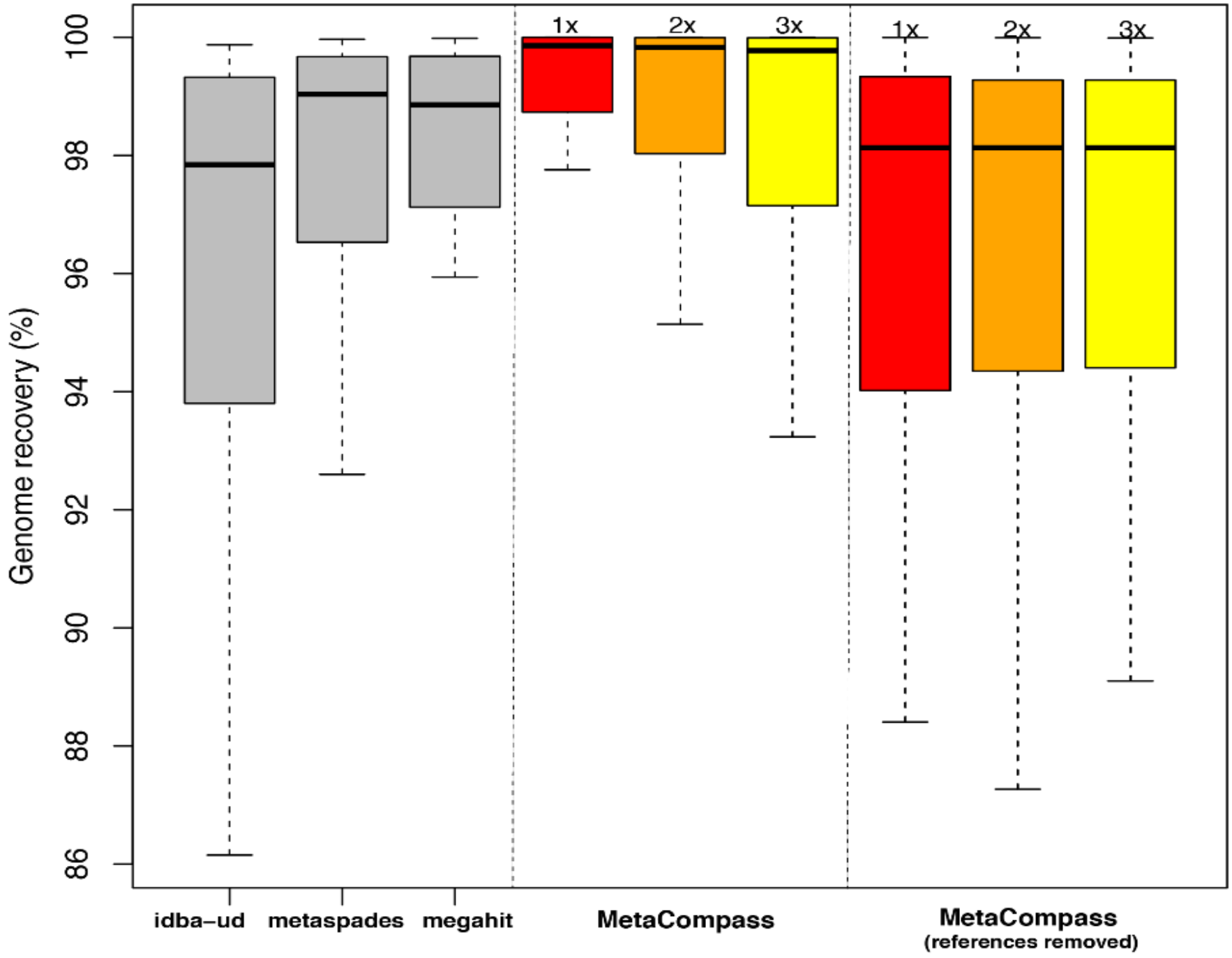
- Dataset 1: Synthetic dataset, Shakya et. al.
- Dataset 2: Down-sampled Dataset 1(low coverage)
- Dataset 3: 2,077 samples from HMP2

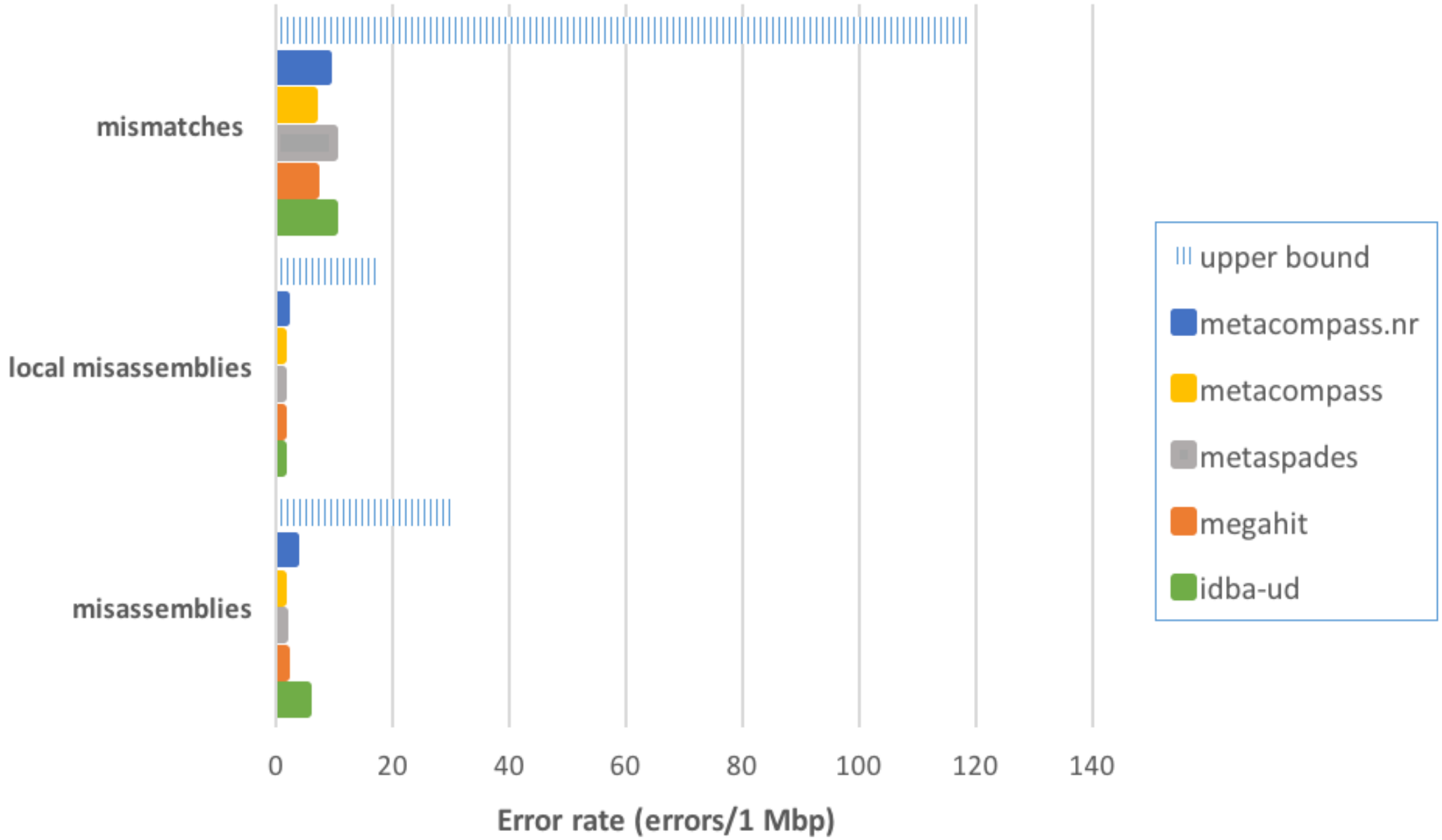
Results - Dataset 1

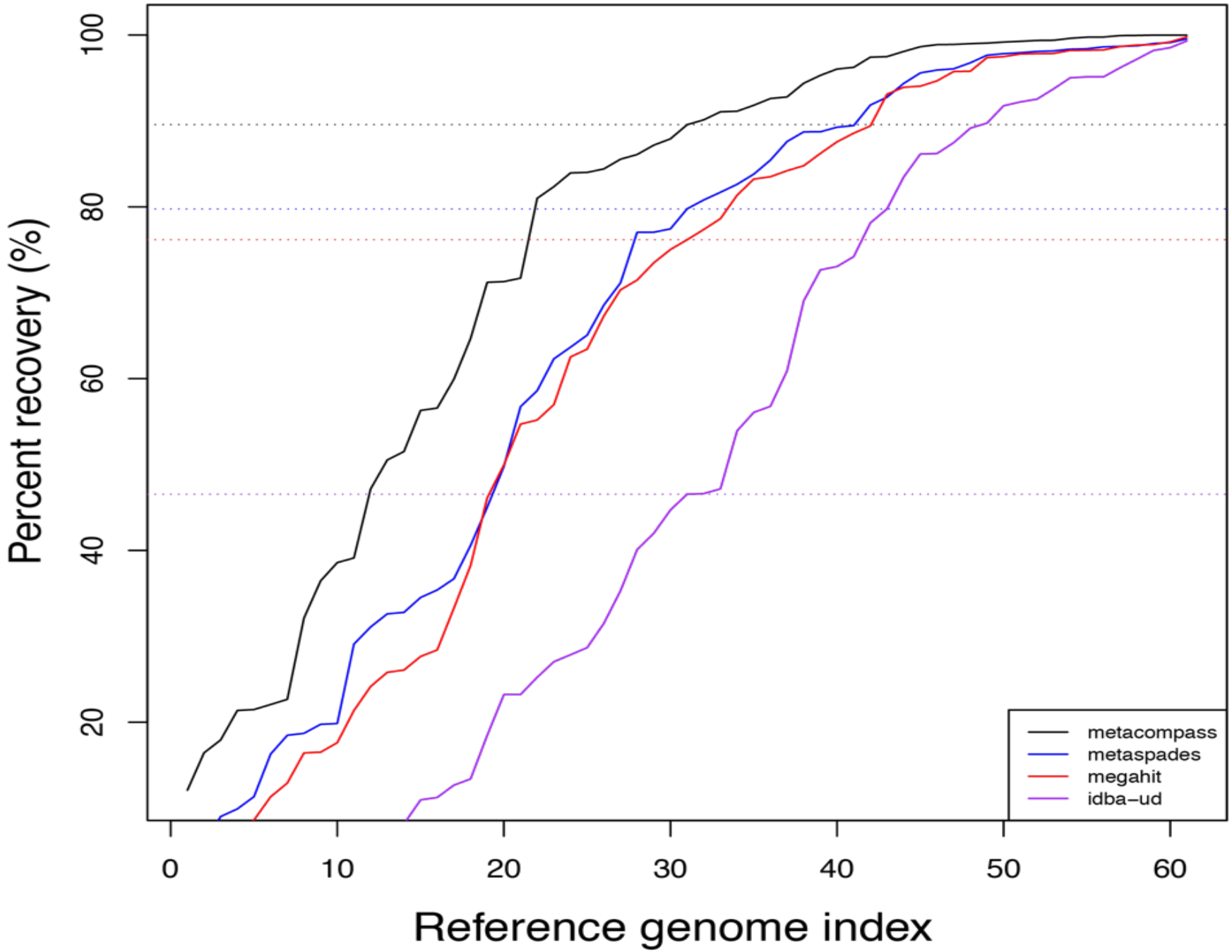
- Mixture of 64 bacterial and archaeal species (Shakya et al., 2013)
- 109 million reads with mean insert size 206 bp and 100 bp read length
- Easier to evaluate assembly since the truth is known

Results – Assembly **Statistics**

Method	No. Contigs	Longest Contig (bp)	Median genome recovery	Mismatches (Per 1Mbp)	Misassemblies (Per 1Mbp)
MetaCompass	**18,766**	**7,057,109**	**100%**	**61.9**	**1.9**
IDBA-UD	22,355	991,792	98%	98.6	6.3
MEGAHIT	35,351	1,151,857	99%	66.5	2.5
metaSPAdes	21,424	1,438,235	99%	97.1	2.3


Genome recovery (%)
100
98
96
94
92
90
88
86
1x
2x
3x
1x
2x
3x
idba−ud
metaspades
megahit
MetaCompass
MetaCompass
(references removed)


mismatches
local misassemblies
misassemblies
upper bound
metacompass.nr
metacompass
metaspades
megahit
idba-ud
0
20
40
60
80
100
120
140
Error rate (errors/1 Mbp)


Percent recovery (%)
100
80
60
40
20
0
10
20
30
40
50
60
Reference genome index
metacompass
metaspades
megahit
idba−ud

MetaCompass tutorial

-- I have a set of metagenomic reads and want to perform reference-guided assembly.

```
python go_metacompass.py -P [read1.fq,read2.fq] -l [max read length]-o [output_folder] -t [ncpu]
```

-- I know the reference genomes and have a set of metagenomic reads

```
python go_metacompass.py -r [references.fasta] -P [read1.fq,read2.fq] -o [output_folder] -t [ncpu]
```

Output directory:

```
 Assembled contigs:
            metacompass_output/metacompass.final.ctf.fa

  Selected Reference genomes:
            metacompass_output/metacompass.recruited.fa
```

Docker details

- Where do I save my output?

/output

- How do I exit the container?

exit

- How do I access my output after exiting the container?

cd /home/stamps19/chiron/metacompass

Your turn!

- https://gitlab.com/treangen/stamps2019/README.md#part2