

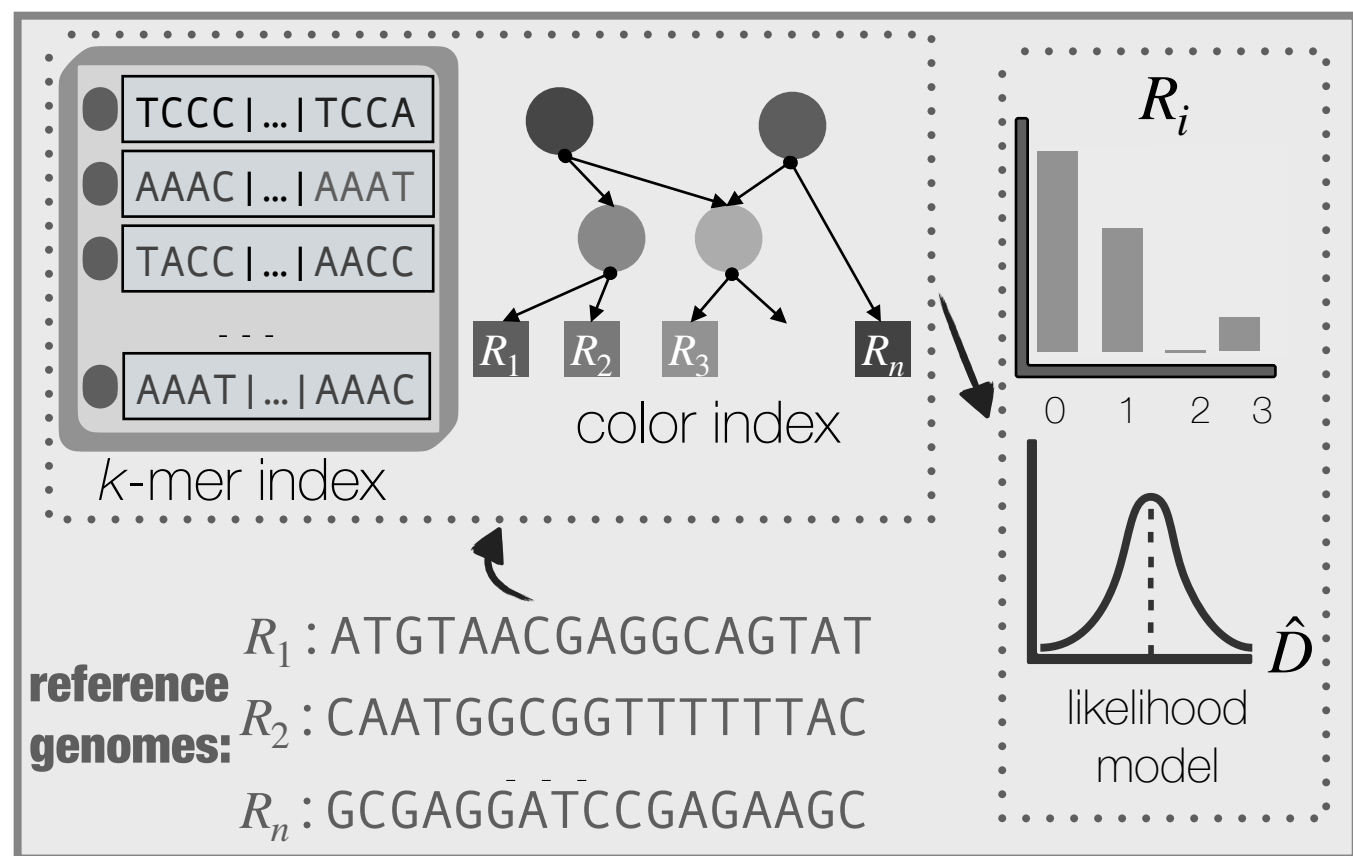
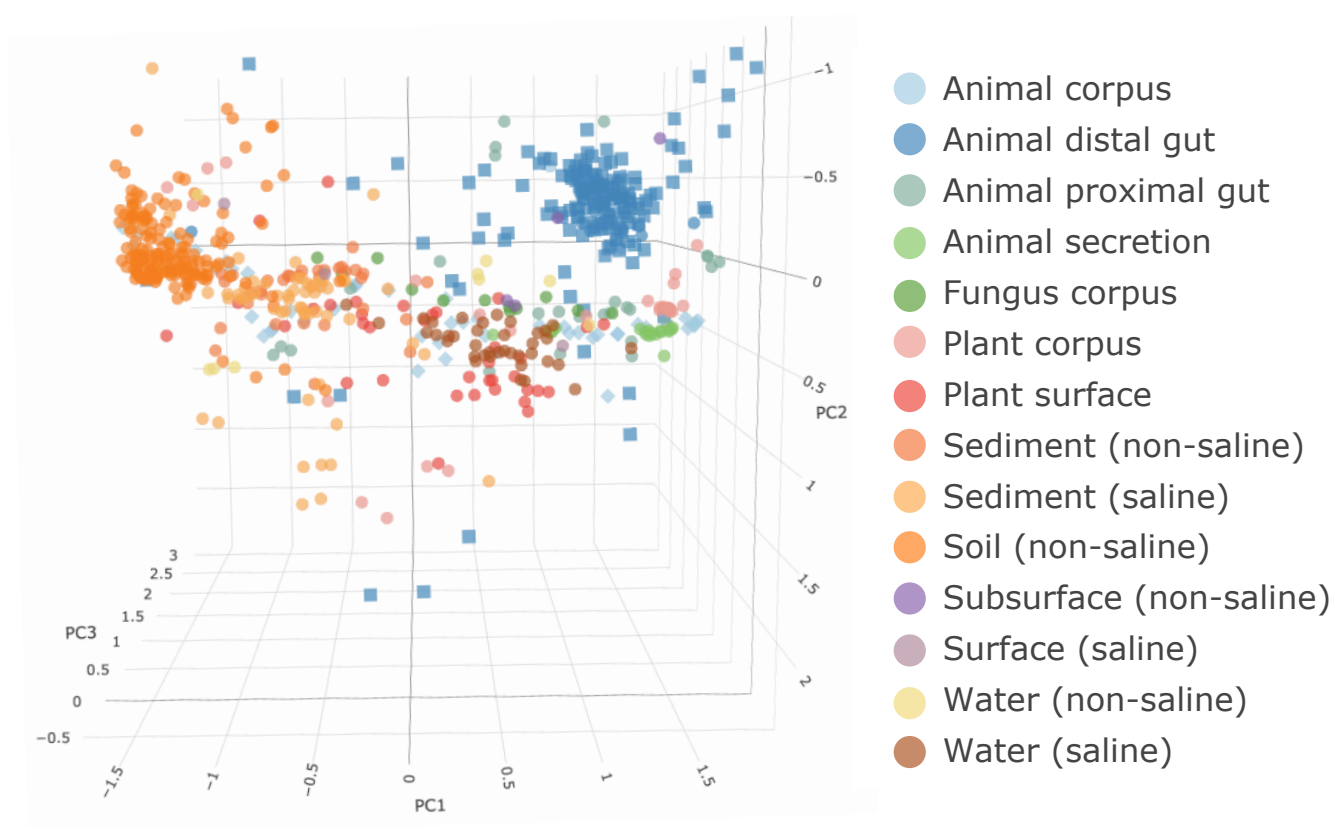
Read-genome distance calculation & phylogenetic placement using **krepp**

Ali Osman Berk Şapcı
UC San Diego
July 12, 2026 — ISMB

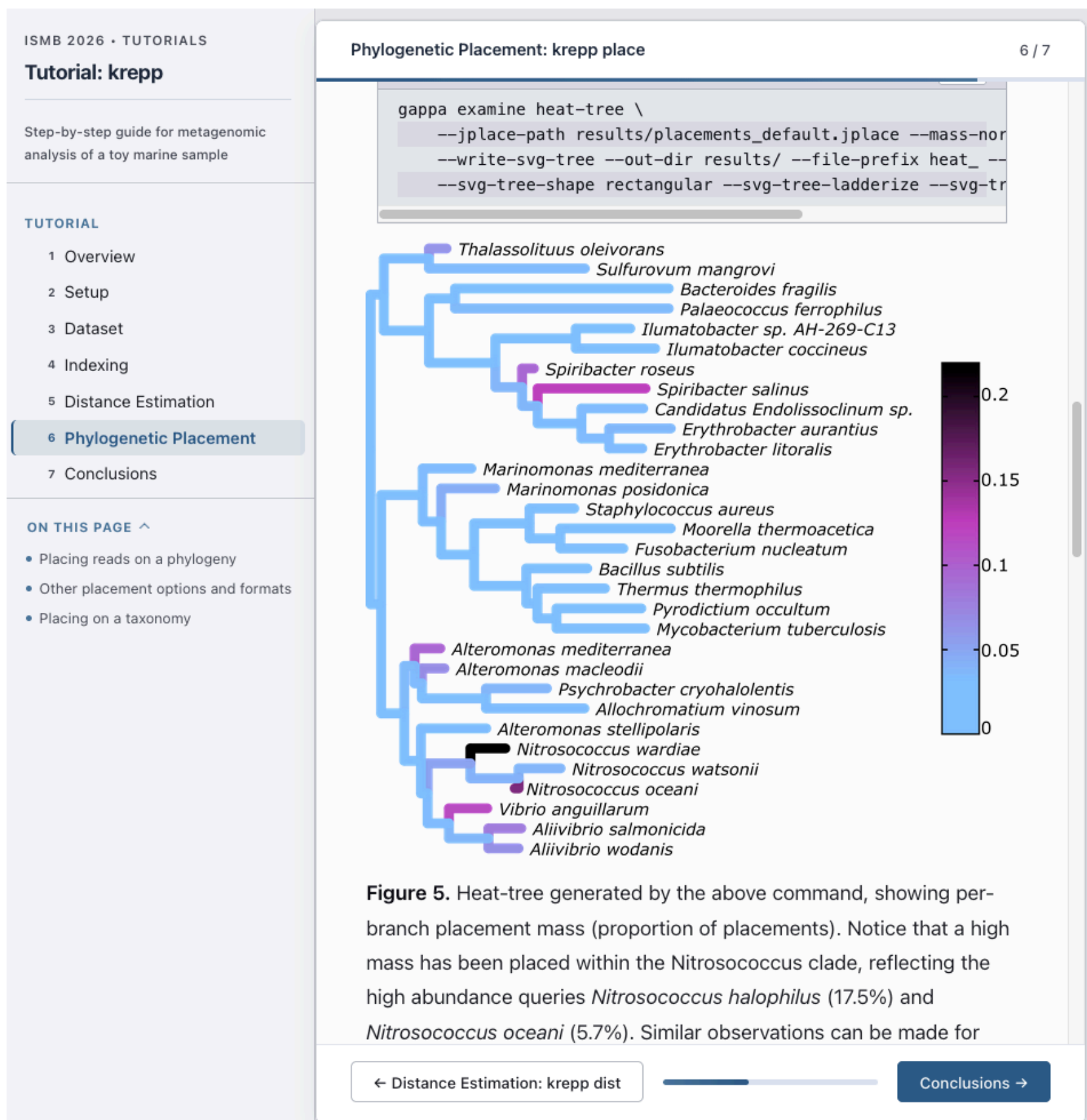


Logistics and outline

~25 minutes:
overview of the method & key results

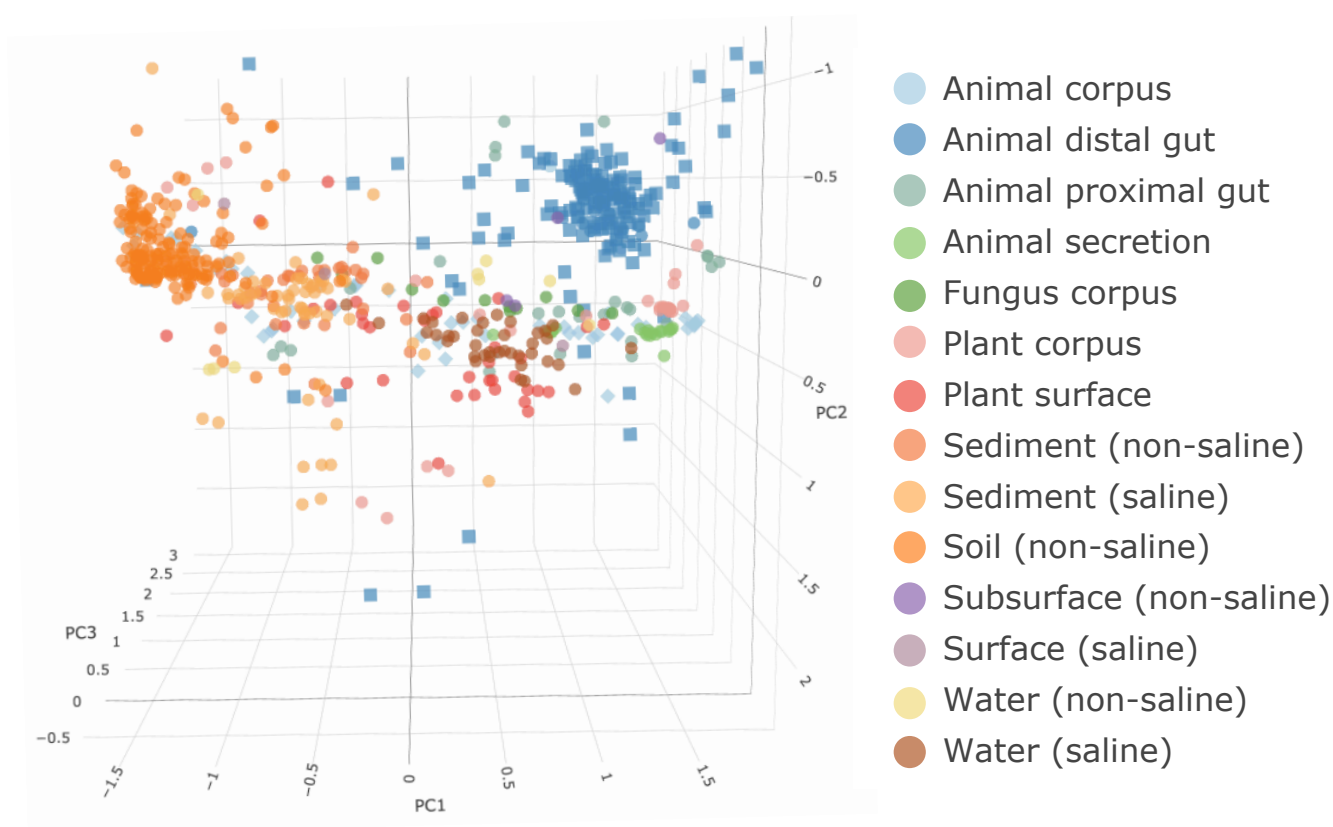


~35 minutes:
overview of the tool & hands-on exercises



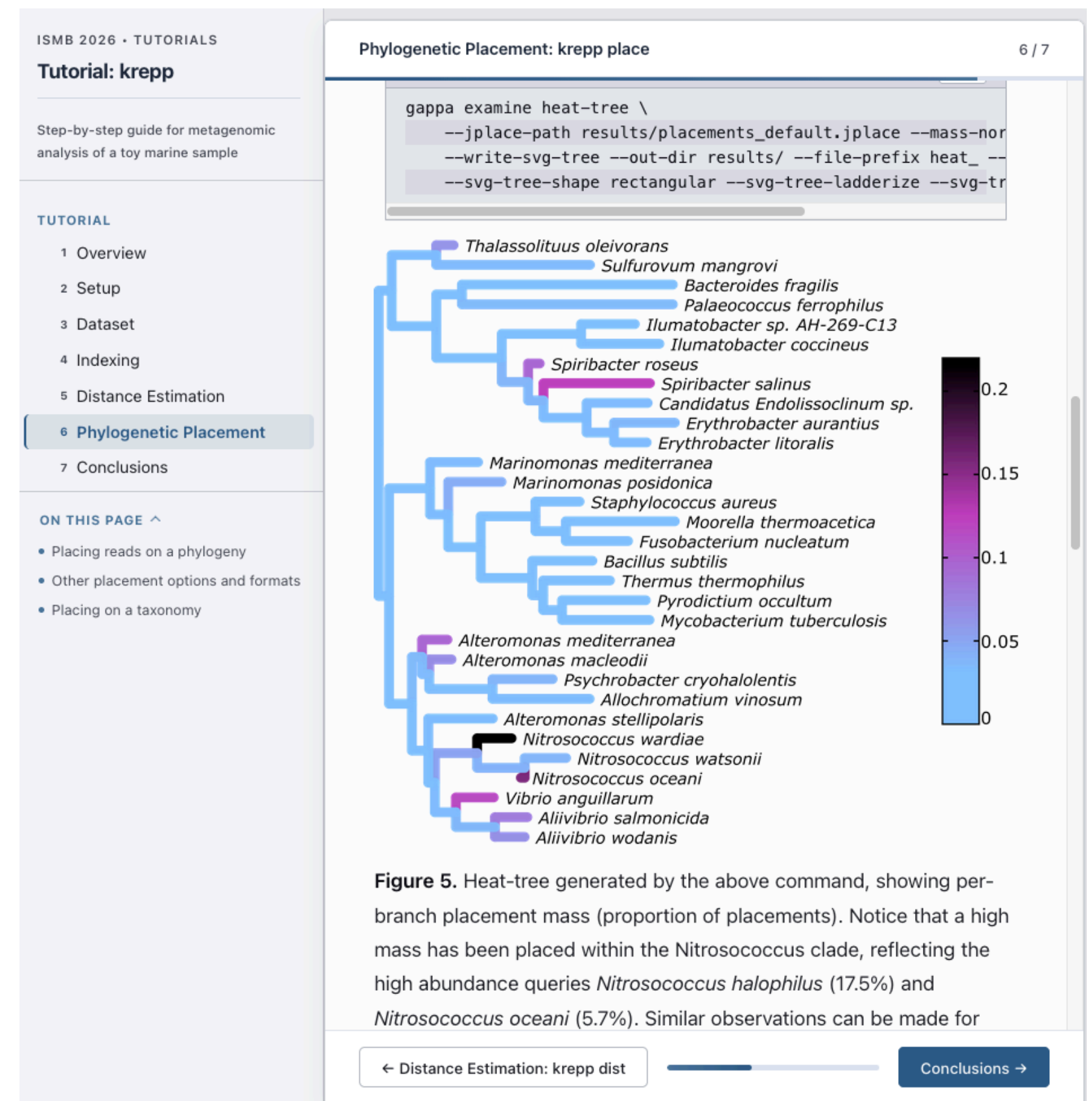
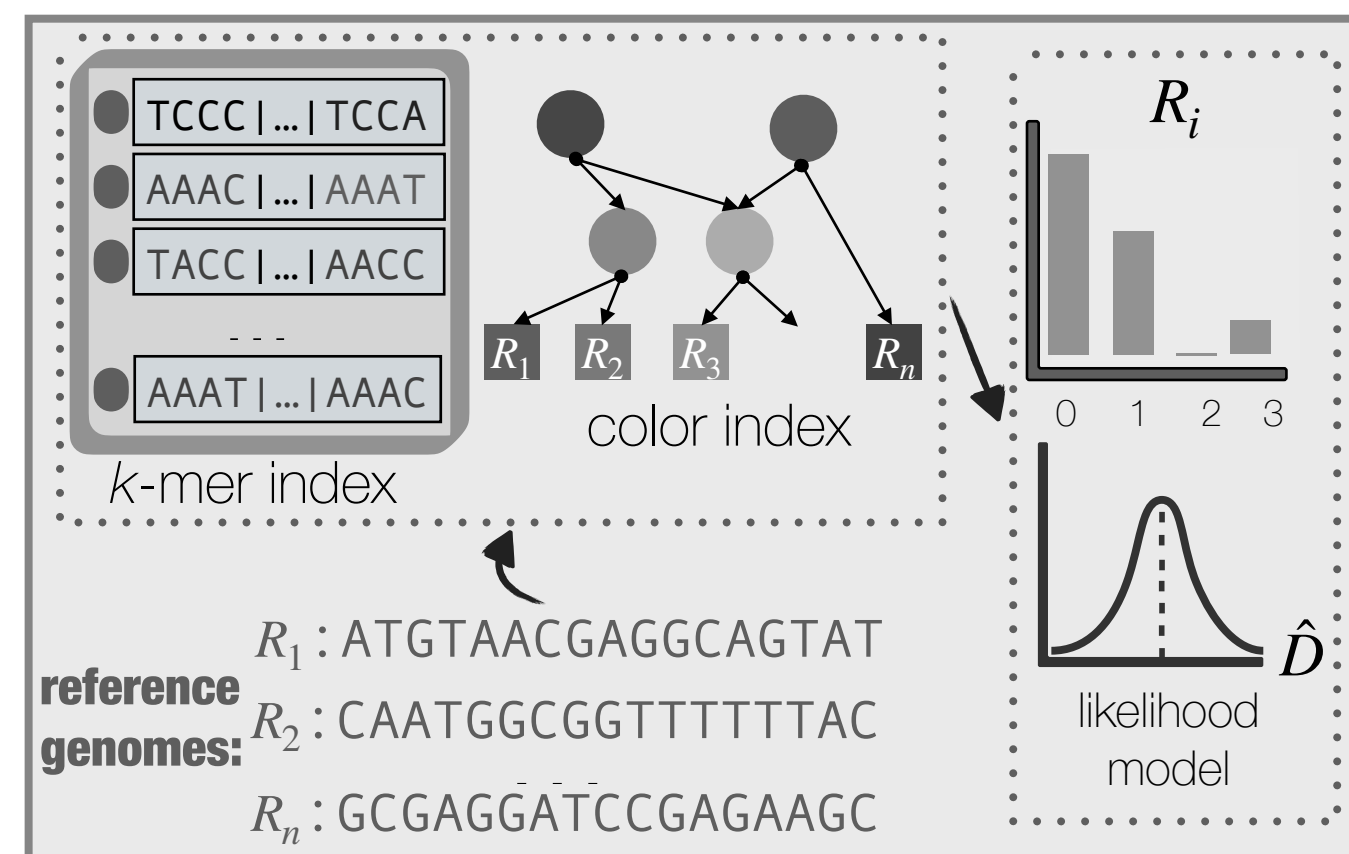
Logistics and outline

~25 minutes:
overview of the method & key results



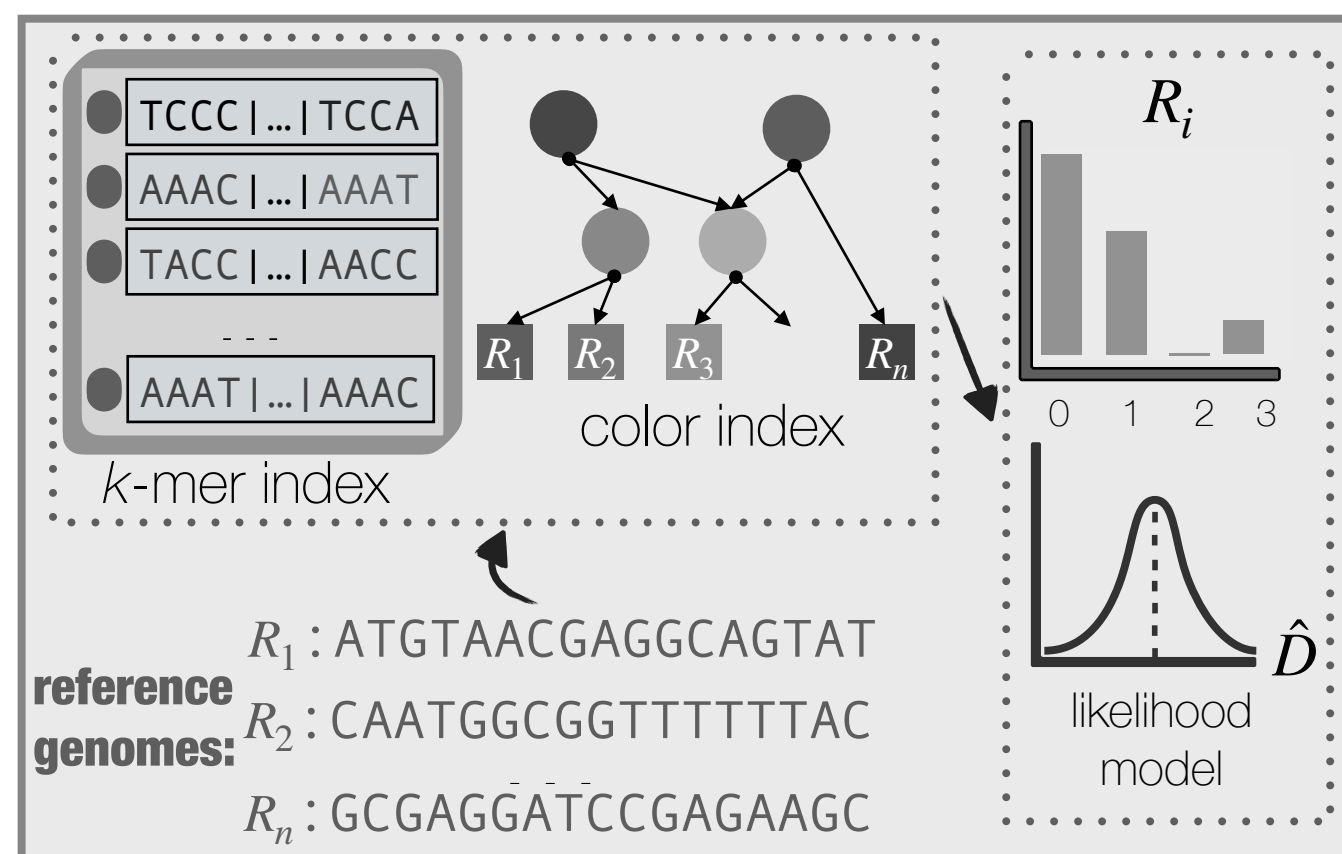
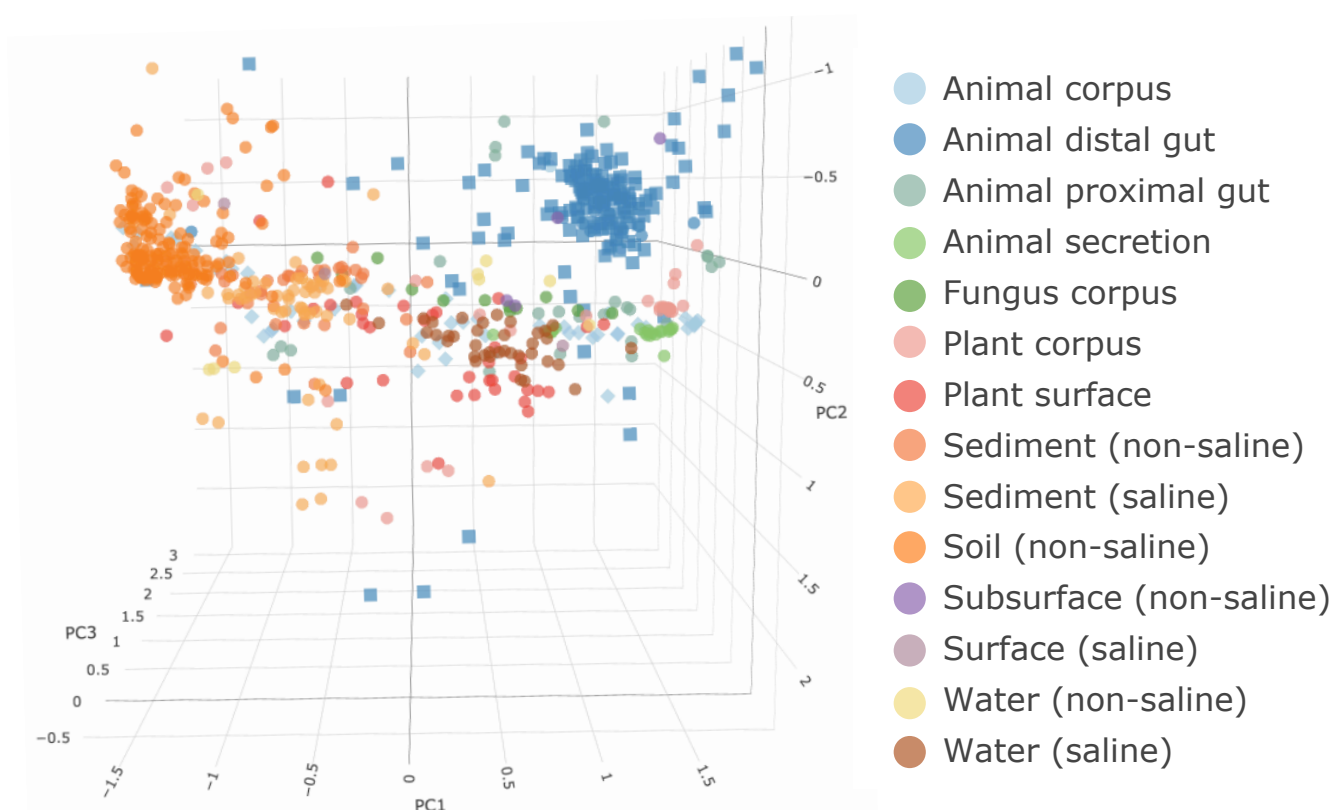
based on slides,
mostly *passive* —
except for questions!
(feel free to interrupt)

~35 minutes:
overview of the tool & hands-on exercises



Logistics and outline

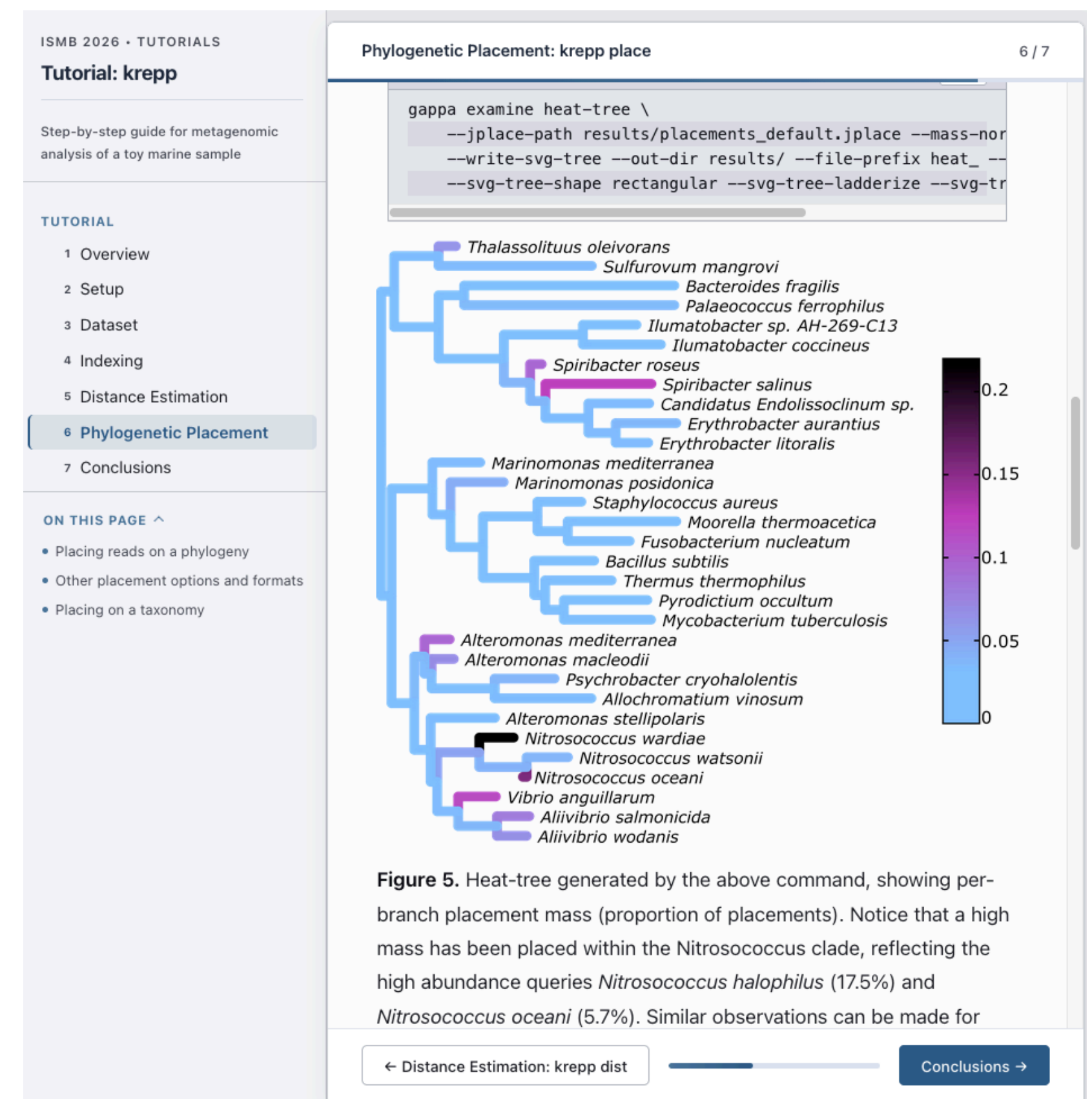
~25 minutes:
overview of the method & key results



based on slides,
mostly *passive* —
except for questions!
(feel free to interrupt)

following a website
(and a few slides) —
actively analyzing a toy dataset!

~35 minutes:
overview of the tool & hands-on exercises



If you have not done already:

`bo1929.github.io/documents/krepp-tutorial-ismb2026`

TUTORIAL

1	Overview
2	Setup
3	Dataset
4	Indexing
5	Distance Estimation
6	Phylogenetic Placement
7	Conclusions

Create an environment and install krepp

Set up an environment and install `krepp` (version 0.8.2) via Bioconda, then verify the installation:

```
bash
micromamba create -n krepp-tutorial -c conda-forge -c bioconda krepp=0.8.2 -y
micromamba activate krepp-tutorial
krepp --help
```

COPY

? Expected output:

Install helper tools into the same environment, and verify:

```
bash
micromamba install -c conda-forge -c bioconda gappa=0.9.0 wget -y
gappa --help
```

COPY

? Expected output:

Download and prepare the tutorial data

```
bash
git clone https://github.com/bo1929/ismb2026-krepp-tutorial-materials.git
cd ismb2026-krepp-tutorial-materials
bash scripts/setup.sh
```

COPY

Background, overview & key results

Tool overview

- Easily scales to large reference sets
- High throughput, flexible memory
- Querying reads, contigs, genomes...

reference genomes

R1: TCCCTGCTCA...

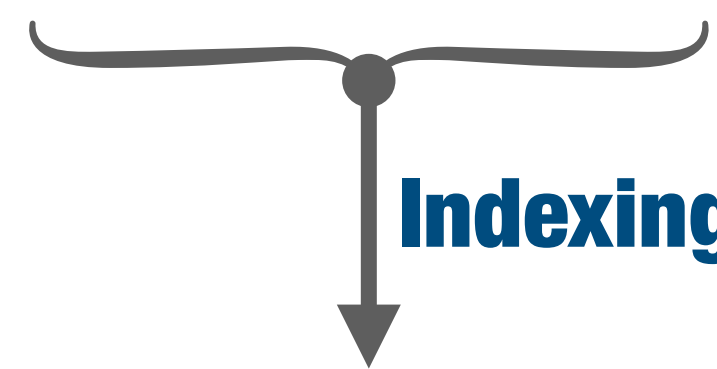
R2: TCCCTGCTCA...

R3: CAATGTGCGG...

R4: CCCCAAACGA...

...

RN: ATTATCTGAT...



e.g., a short read:

@QSEQ
ATACCTAGGAG...



Tool overview

- Easily scales to large reference sets
- High throughput, flexible memory
- Querying reads, contigs, genomes...

reference genomes

R1: TCCCTGCTCA...
R2: TCCCTGCTCA...
R3: CAATGTGCGG...
R4: CCCCAAACGA...
...
RN: ATTATCTGAT...

Indexing

e.g., a short read:

@QSEQ
ATACCTAGGAG...

Mapping and distances:

QSEQ	R1	0.0112
QSEQ	R2	NA
QSEQ	R3	0.23
...		
QSEQ	RN	0.0016

per query

Tool overview

- Easily scales to large reference sets
- High throughput, flexible memory
- Querying reads, contigs, genomes...

reference genomes

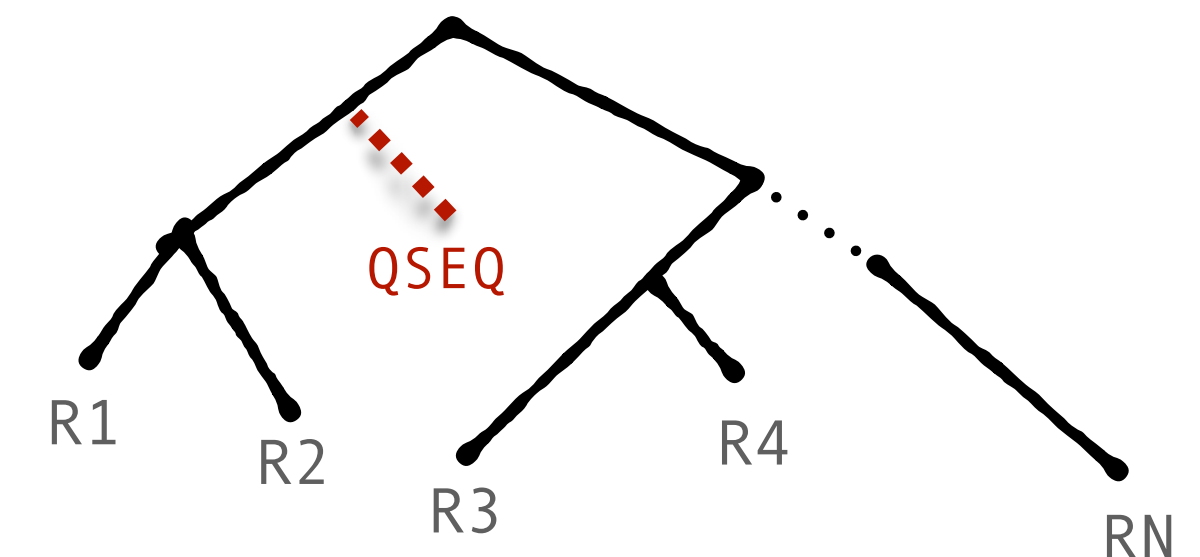
R1: TCCCTGCTCA...
R2: TCCCTGCTCA...
R3: CAATGTGCGG...
R4: CCCCAAACGA...
...
RN: ATTATCTGAT...

Indexing

e.g., a short read:

@QSEQ
ATACCTAGGAG...

Phylogenetic placement:



Mapping and distances:

QSEQ	R1	0.0112
QSEQ	R2	NA
QSEQ	R3	0.23
...		
QSEQ	RN	0.0016

per query

Tool overview

- Easily scales to large reference sets
- High throughput, flexible memory
- Querying reads, contigs, genomes...

e.g., a short read:

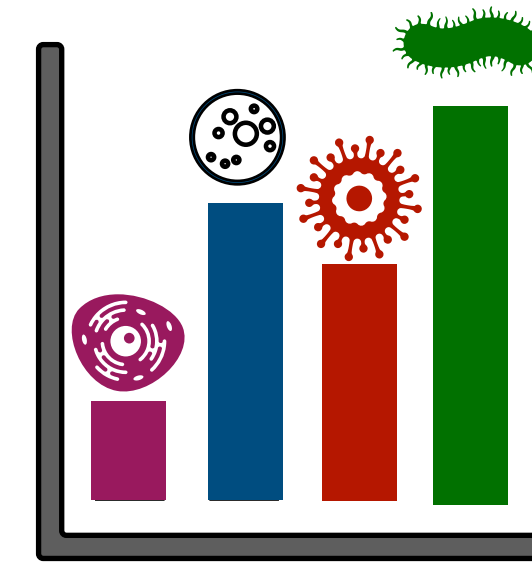
@QSEQ
ATACCTAGGAG...

reference genomes

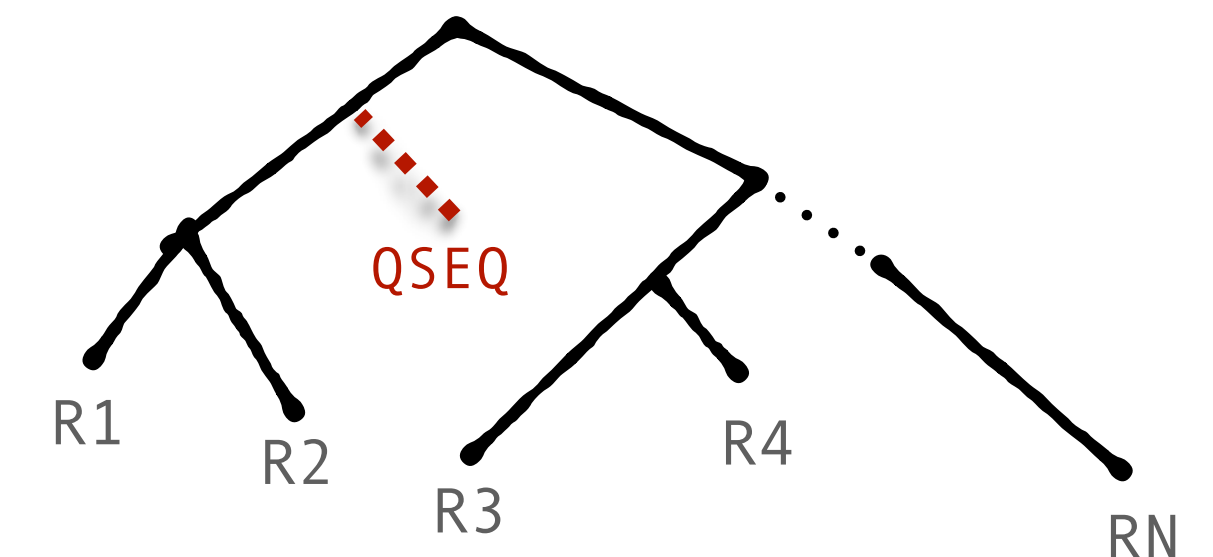
R1: TCCCTGCTCA...
R2: TCCCTGCTCA...
R3: CAATGTGCGG...
R4: CCCCAAACGA...
...
RN: ATTATCTGAT...

Indexing

Taxonomic classification:



Phylogenetic placement:

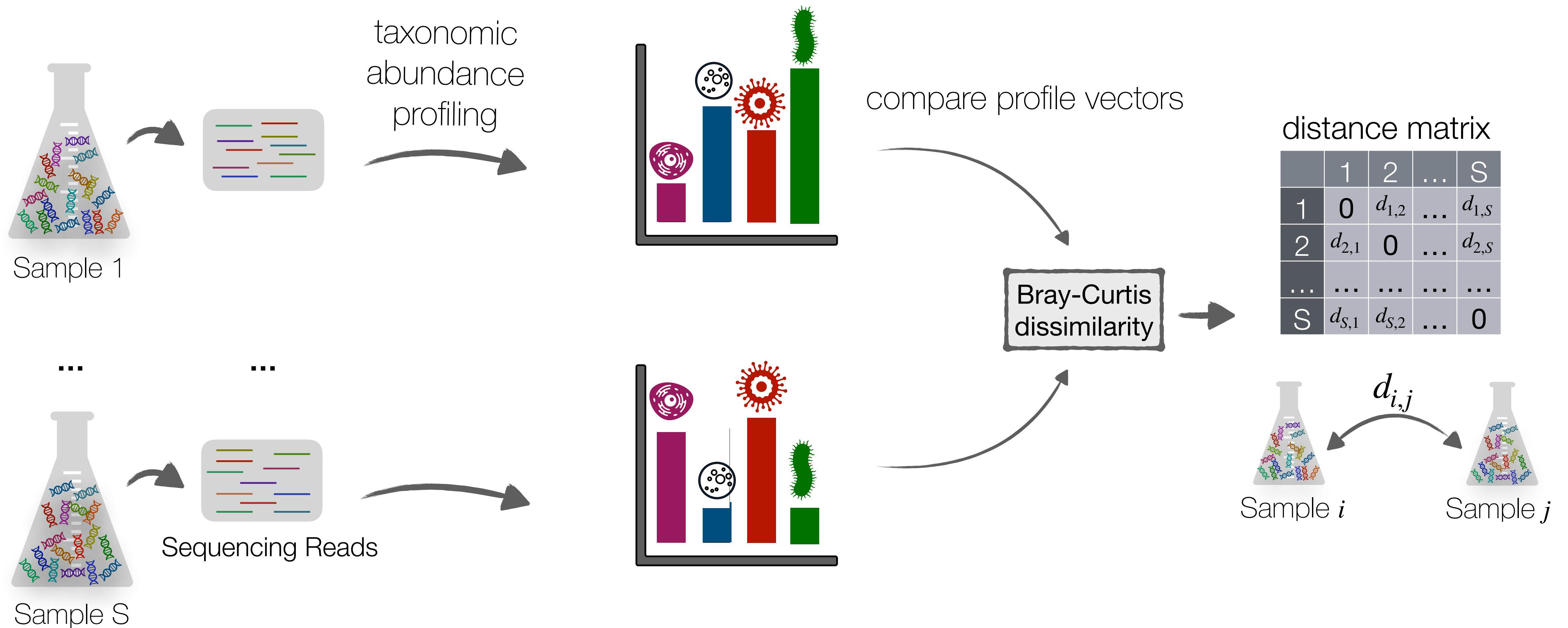


Mapping and distances:

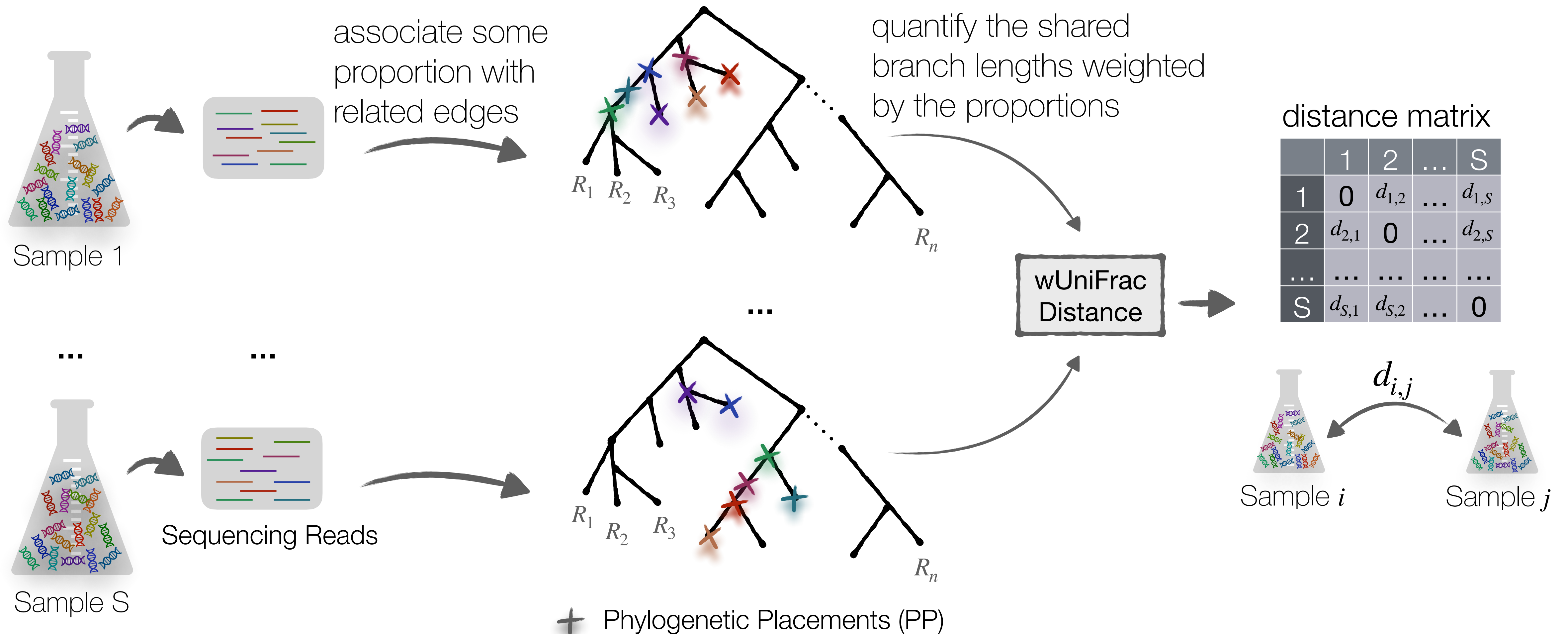
QSEQ	R1	0.0112
QSEQ	R2	NA
QSEQ	R3	0.23
...		
QSEQ	RN	0.0016

per query

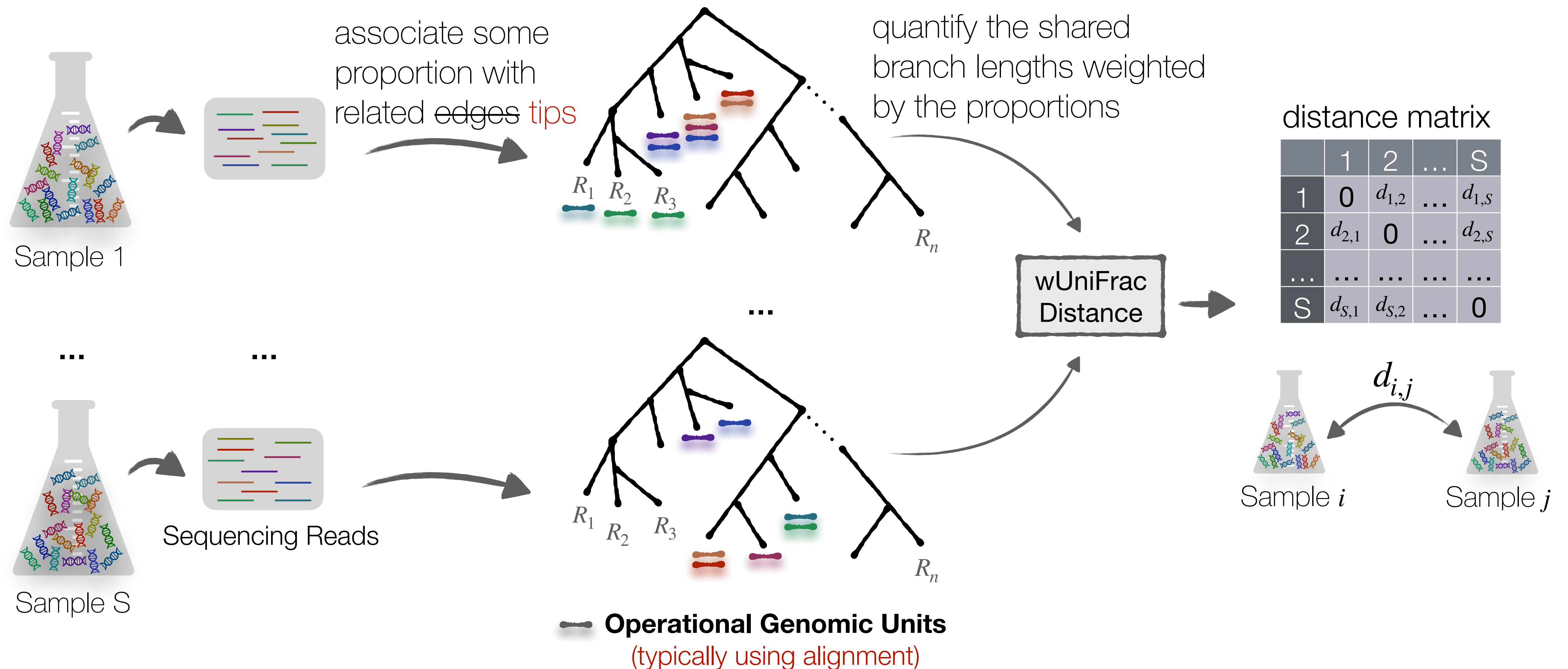
Comparing samples in downstream analysis



Comparing samples in downstream analysis

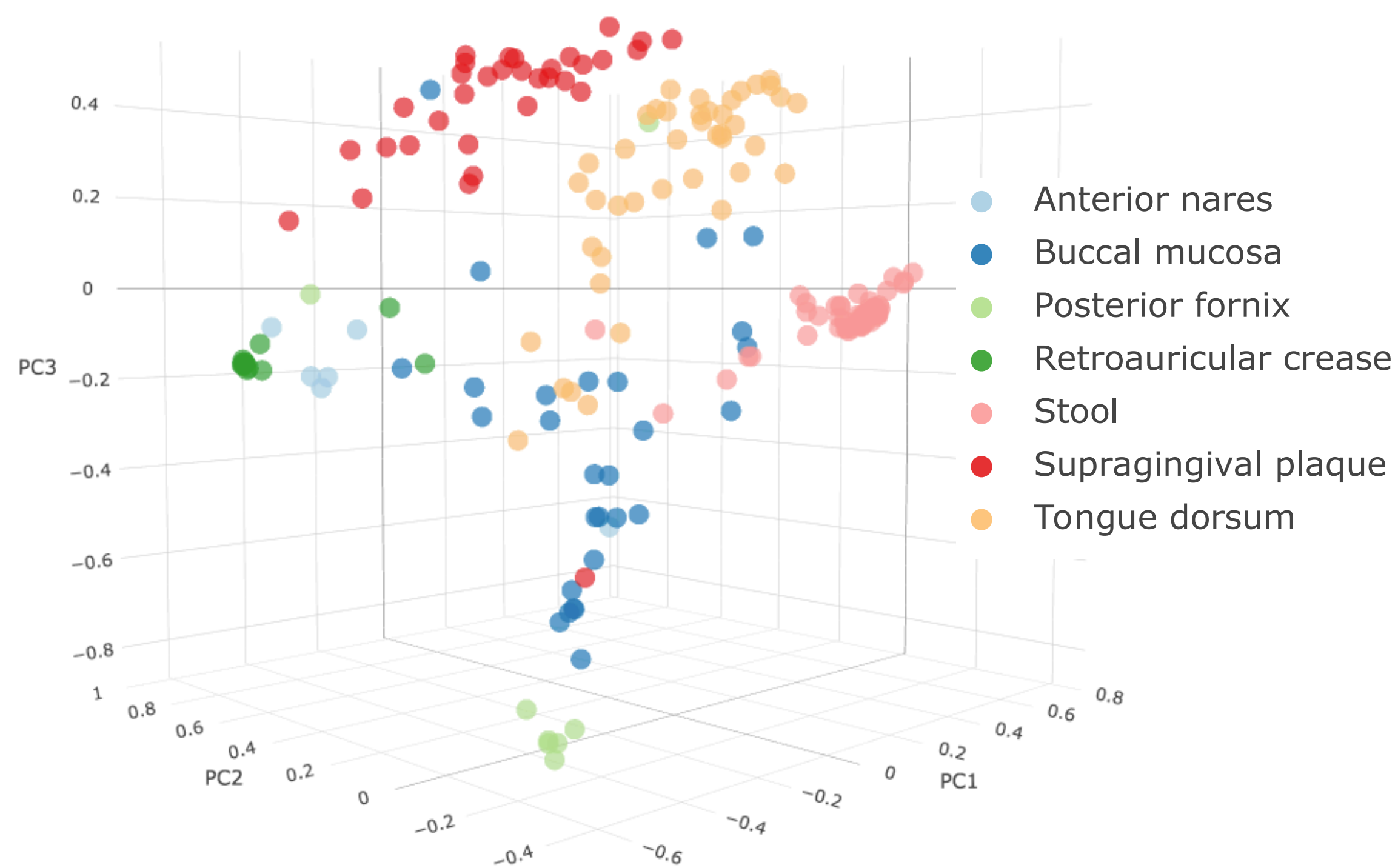


A simpler approach: mapping reads to leaves of a phylogeny

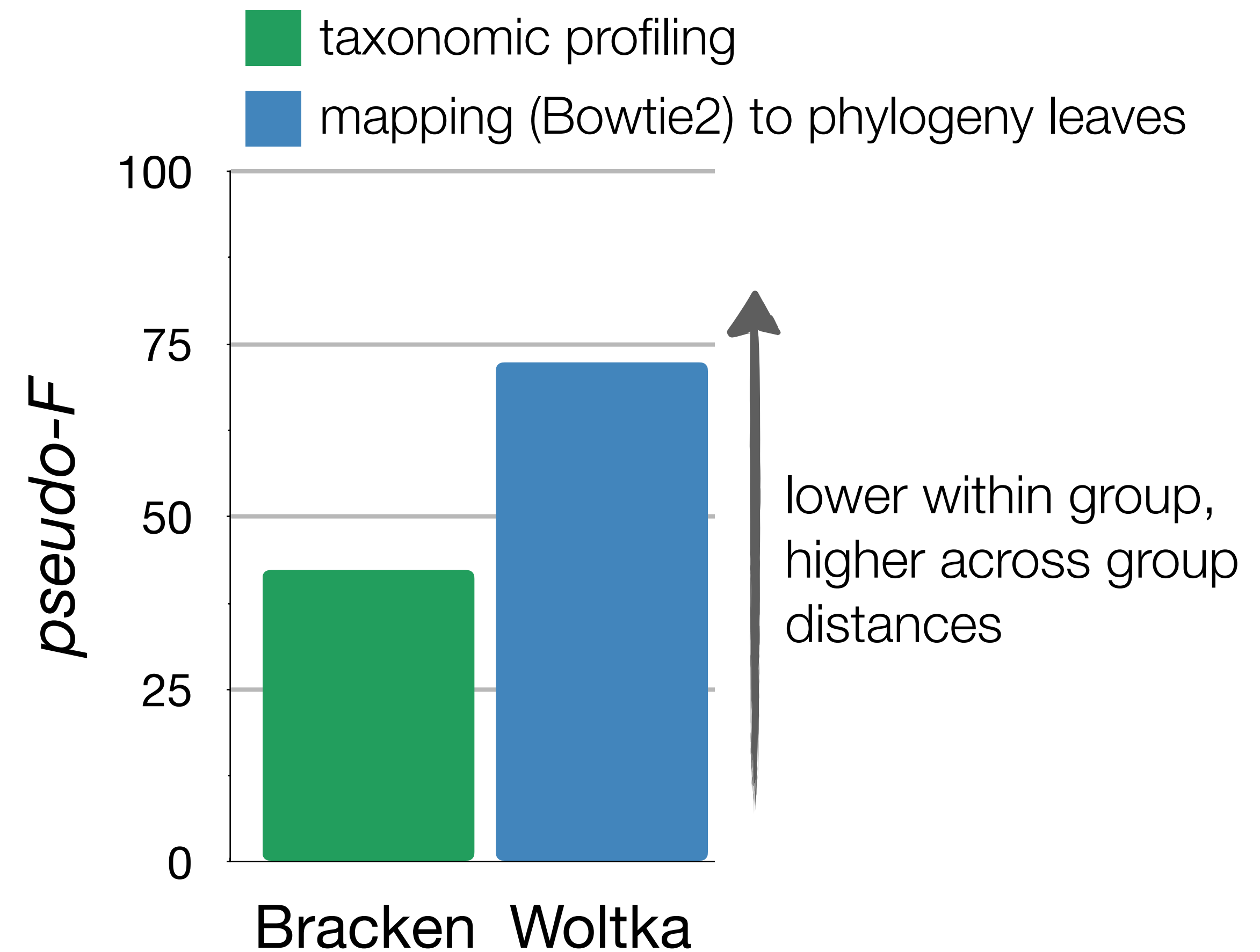


How can we better characterize metagenomic samples?

Using a **phylogeny** and UniFrac achieves **better characterization!**

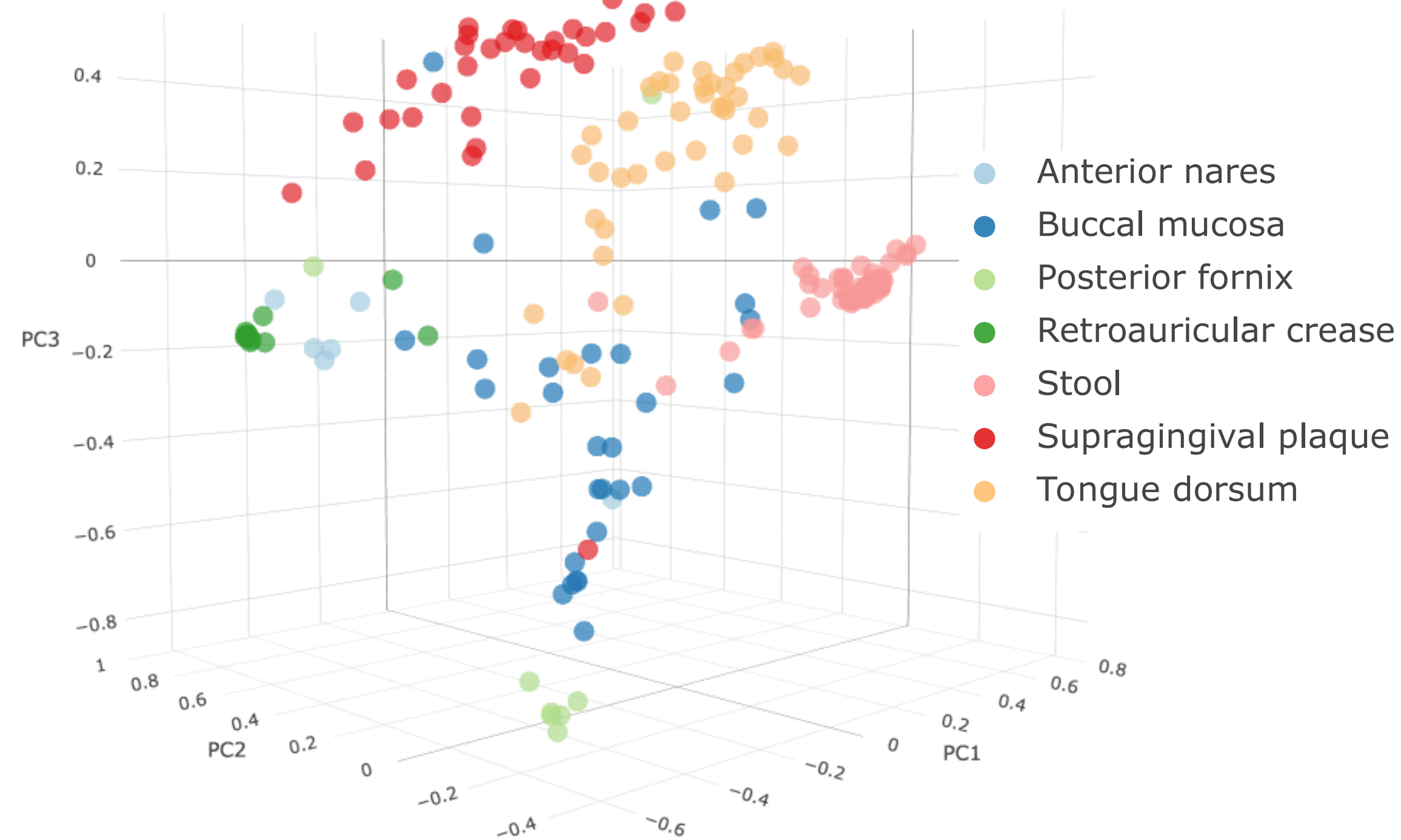


PCoA for 7 body sites, 210 samples



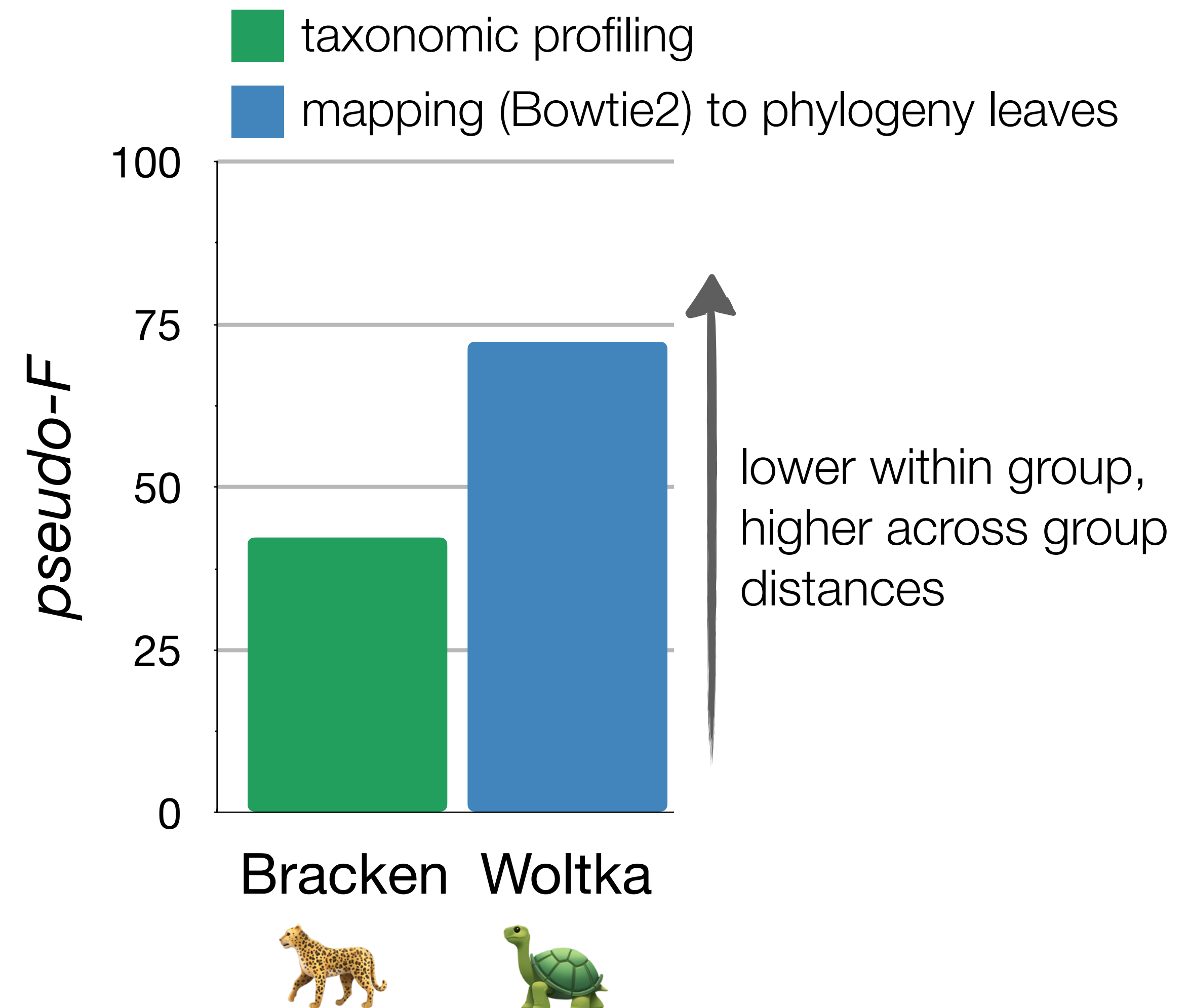
How can we better characterize metagenomic samples?

Using a **phylogeny** and UniFrac achieves **better characterization!**



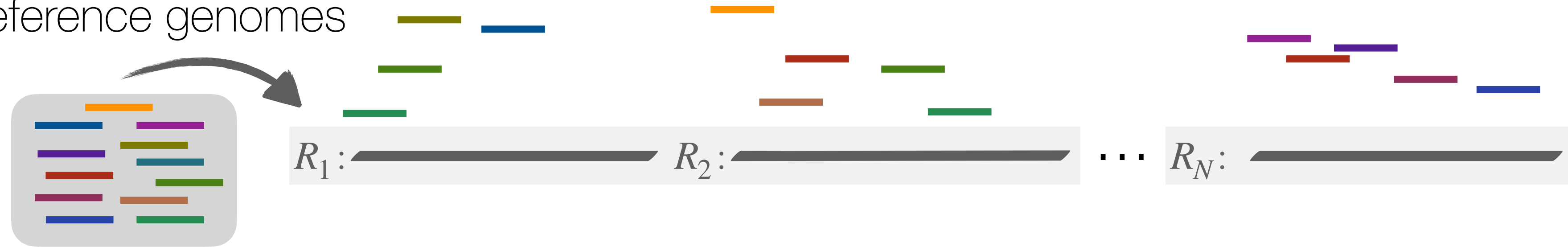
PCoA for 7 body sites, 210 samples

Alignment offers better resolution but slow!



OGUs & limitations of alignment-based mapping

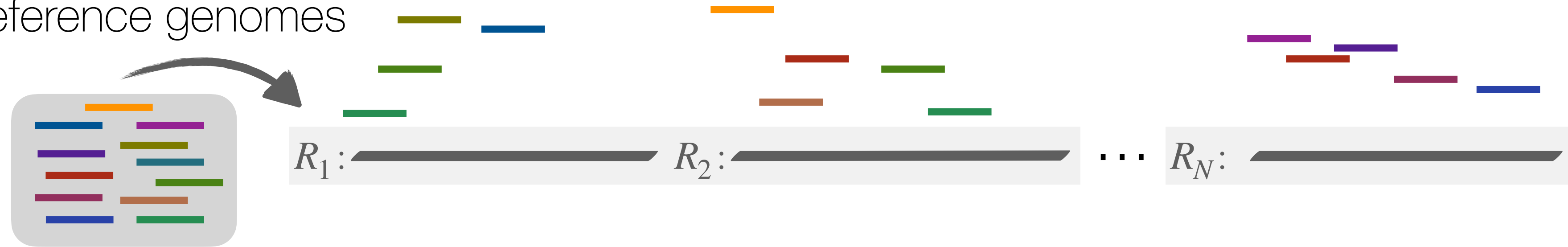
align reads to
reference genomes



(+) measuring similarity — as detailed as it gets

OGUs & limitations of alignment-based mapping

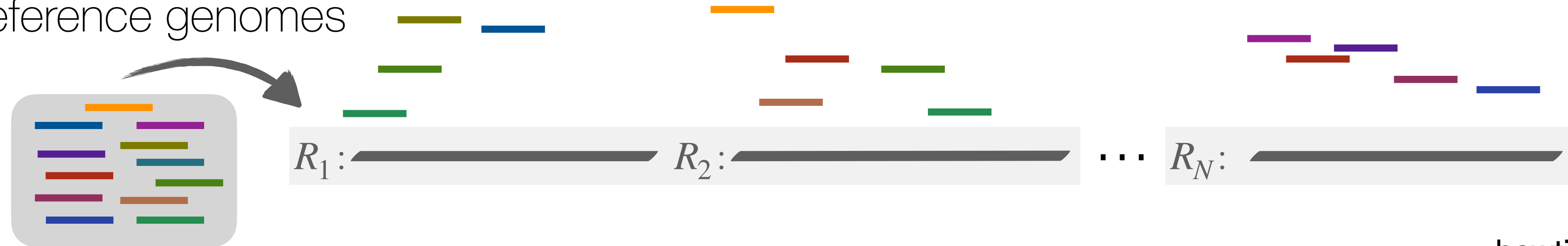
align reads to
reference genomes



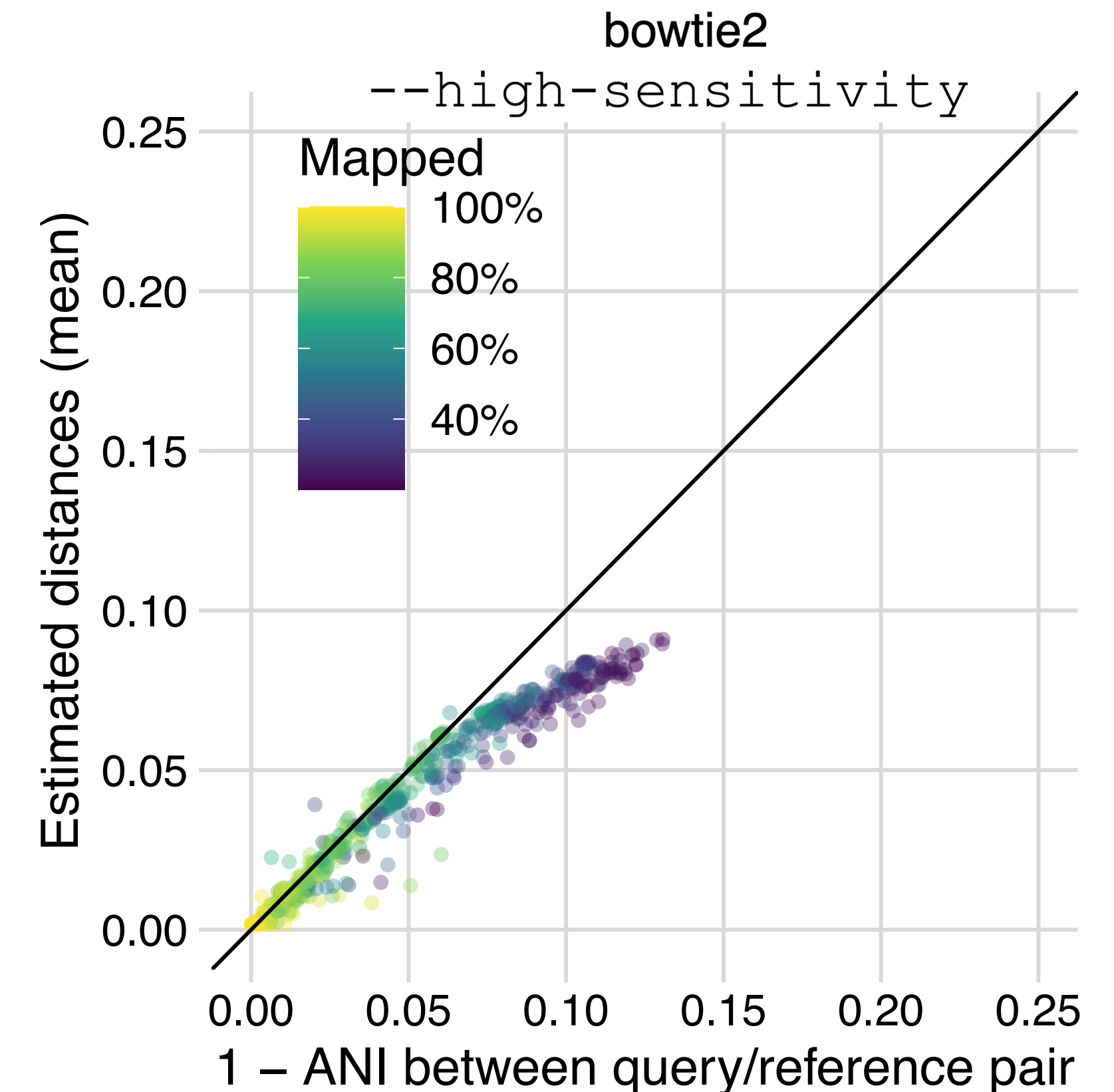
- (+) measuring similarity — as detailed as it gets
- (-) not scalable for large N , even with efficient indexes

OGUs & limitations of alignment-based mapping

align reads to
reference genomes

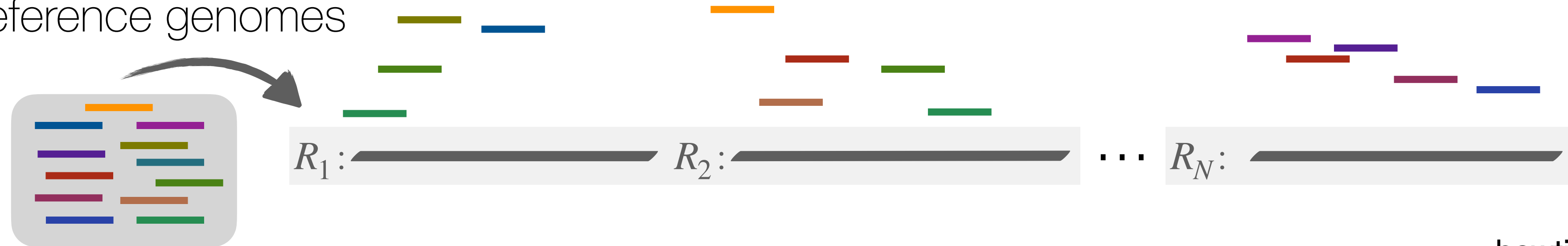


- (+) measuring similarity — as detailed as it gets
- (-) not scalable for large N , even with efficient indexes
- (-) inefficient (+slow) for divergent & novel sequences (>10%)

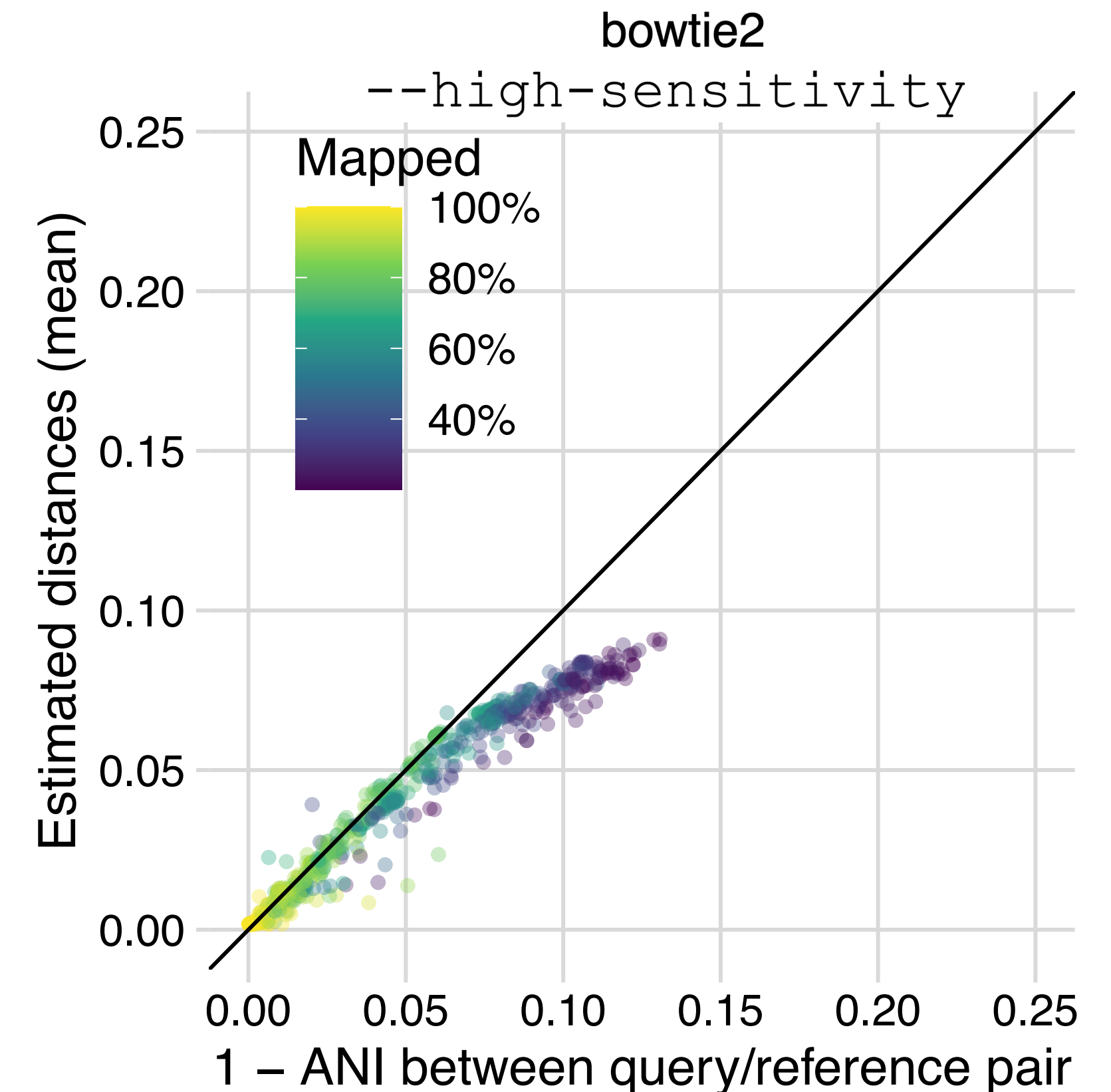


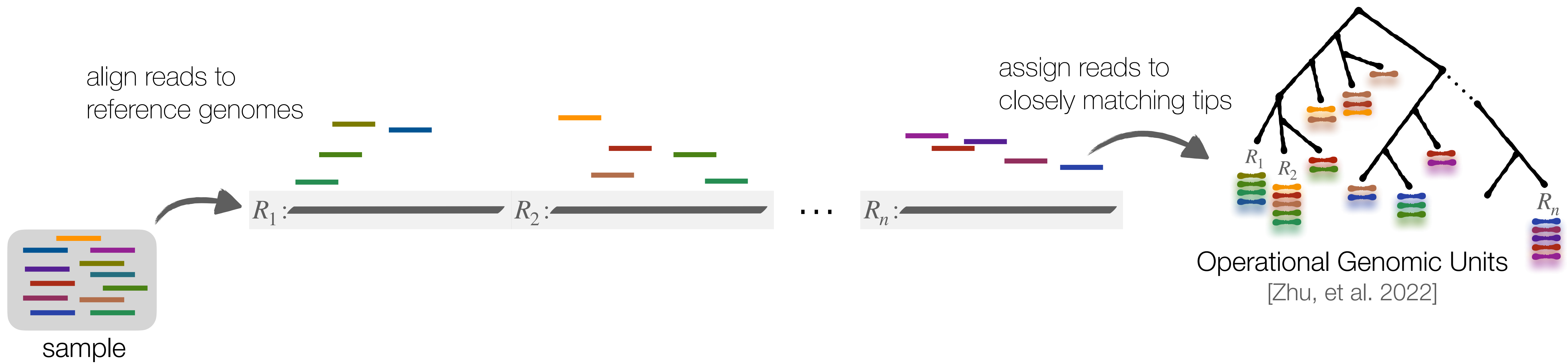
OGUs & limitations of alignment-based mapping

align reads to
reference genomes

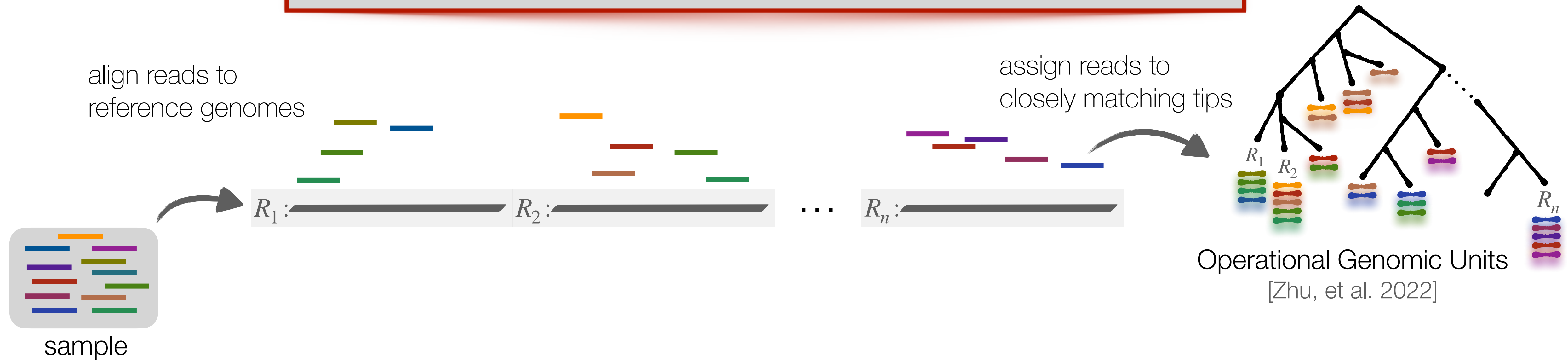


- (+) measuring similarity — as detailed as it gets
- (-) not scalable for large N , even with efficient indexes
- (-) inefficient (+slow) for divergent & novel sequences (>10%)
- (?) excess information: coordinates and exact alignment

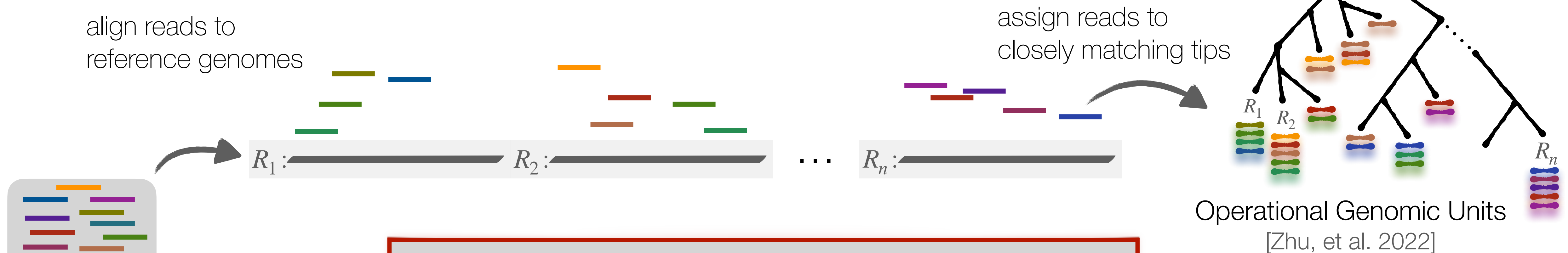




Can we estimate accurate read to genome distances without alignment?

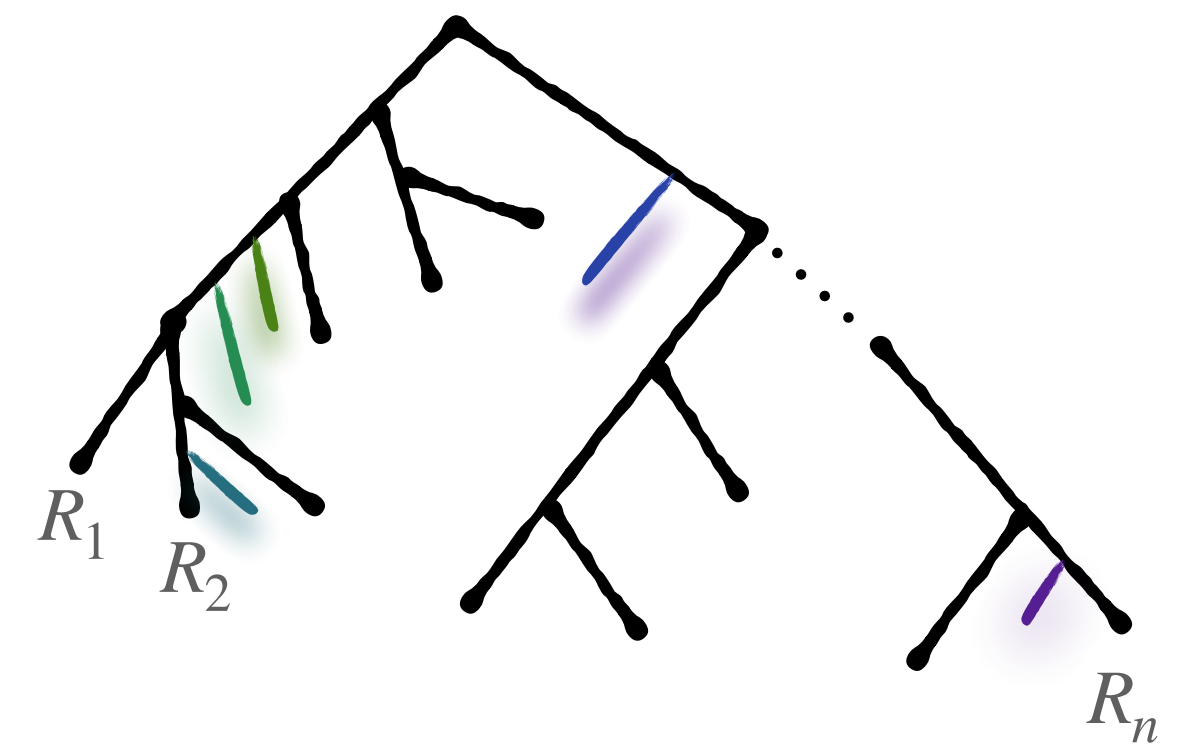


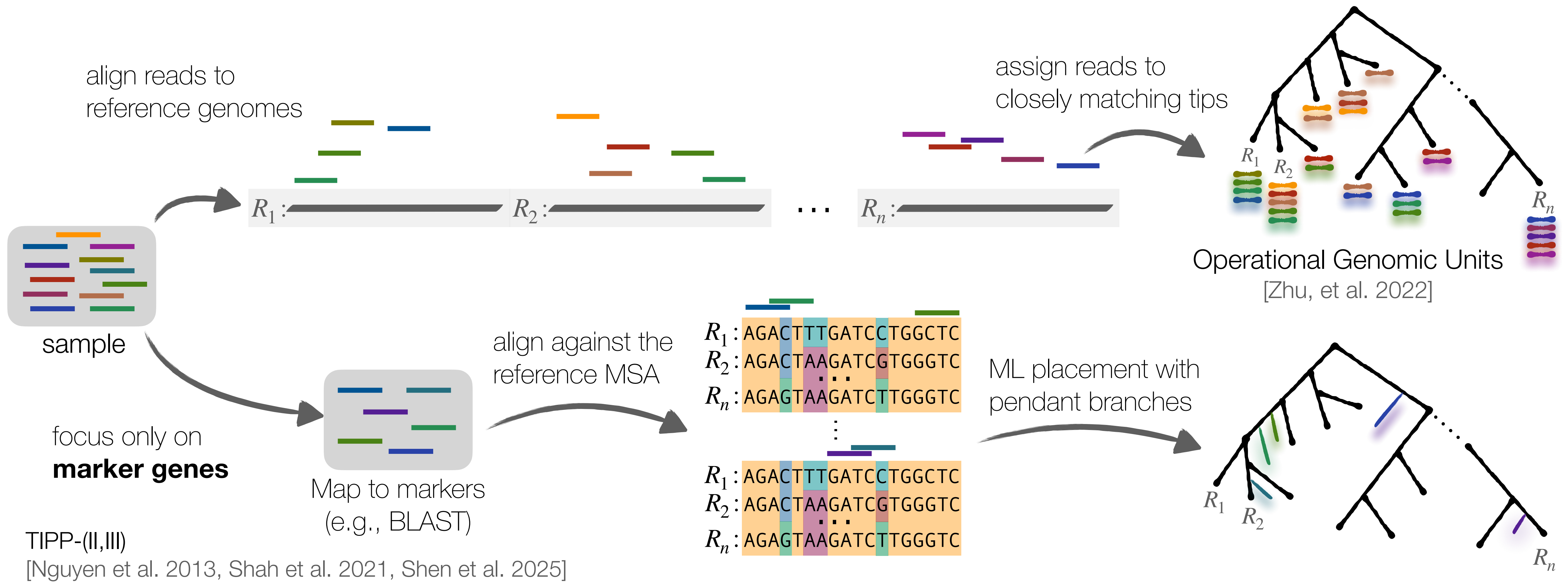
Can we estimate accurate read to genome distances without alignment?



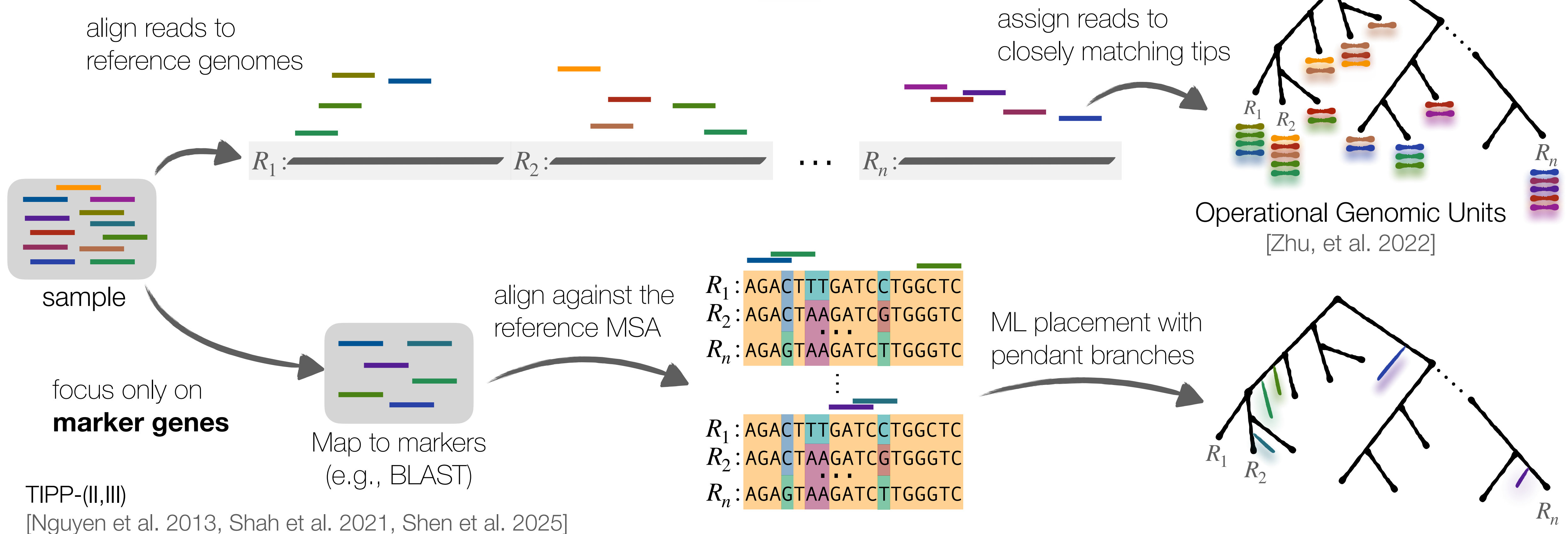
What about a placement in the phylogeny instead of leaf assignments?

Phylogenetic placement has been limited to markers or small phylogenies





Can we place **genome-wide** reads on **ultra-large** phylogenies?



Problem statement and our goals

Given:

- query sequence q (e.g., a short read)
- set of references $\mathcal{R} = \{R_1, \dots, R_N\}$
- a backbone phylogeny T

reference genomes

$>q$
CCTGCTA...



```
R1: TCCCTGCTCA...  
R2: TCCCTGCTAA...  
R3: CCCCTGGCAG...  
R4: ATTATCTGAT...  
...  
RN: CCCCAAACAA...
```

Problem statement and our goals

Given:

- query sequence q (e.g., a short read)
- set of references $\mathcal{R} = \{R_1, \dots, R_N\}$
- a backbone phylogeny T

Interpretation of the distances:

$$d(q, R) = \frac{\text{\# of mismatches}}{\text{length of } q}$$

...AGTTATCCCTGCTC_xA...
 CCTGCTA...

reference genomes

R_1 : TCCCTGCTCA...
 R_2 : TCCCTGCTAA...
 R_3 : CCCCTGGCAG...
 R_4 : ATTATCTGAT...
...
 R_N : CCCCAAACAA...

distance estimates

$d(q, R_1) = 0.141$
 $d(q, R_2) = 0.001$
 $d(q, R_3) = 0.195$
 $d(q, R_4) = \text{NA}$
...
 $d(q, R_N) = 0.244$

$>q$
CCTGCTA...

- from each read to all sufficiently similar genomes

Problem statement and our goals

Given:

- query sequence q (e.g., a short read)
- set of references $\mathcal{R} = \{R_1, \dots, R_N\}$
- a backbone phylogeny T

Interpretation of the distances:

$$d(q, R) = \frac{\text{\# of mismatches}}{\text{length of } q}$$

...AGTTATCCCTGCTCA...
CCTGCTA...
x

reference genomes

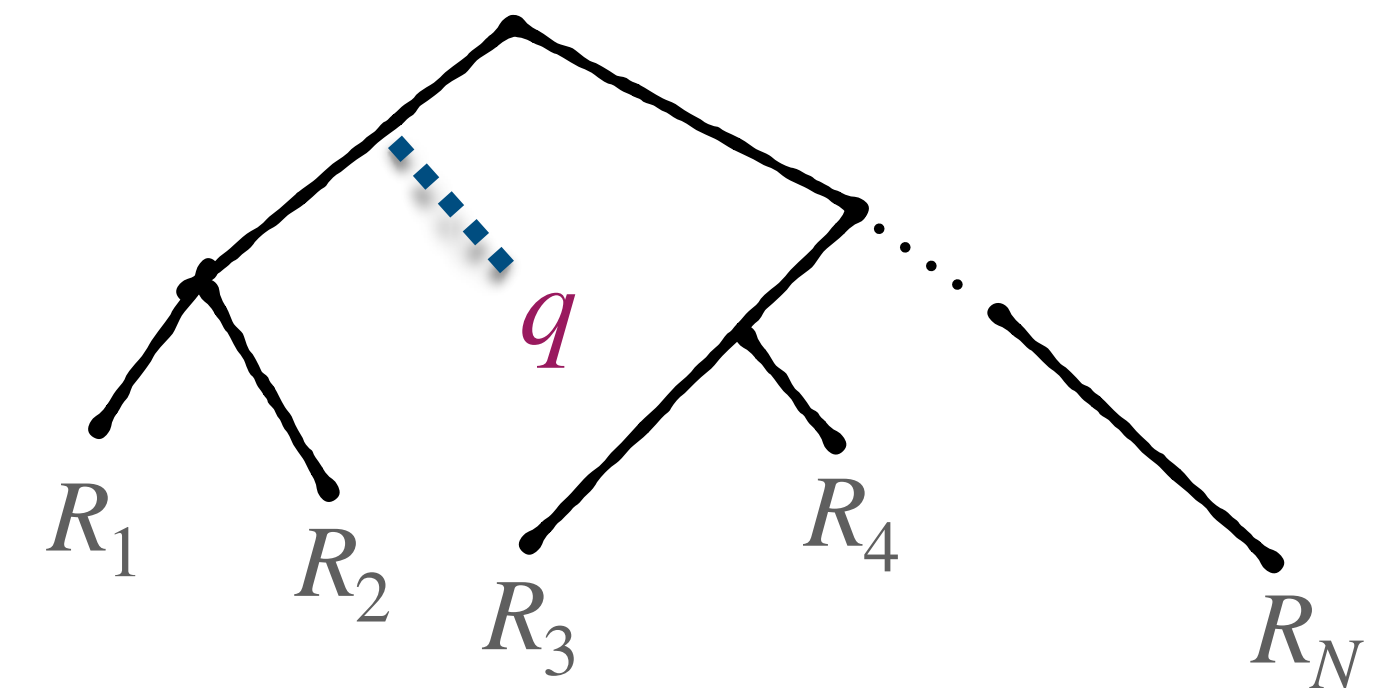
$$\begin{aligned} R_1 &: \text{TCCCTGCTCA...} \\ R_2 &: \text{TCCCTGCTAA...} \\ R_3 &: \text{CCCCTGGCAG...} \\ R_4 &: \text{ATTATCTGAT...} \\ &\dots \\ R_N &: \text{CCCCAAACAA...} \end{aligned}$$

distance estimates

$d(q, R_1)$	$= 0.141$
$d(q, R_2)$	$= 0.001$
$d(q, R_3)$	$= 0.195$
$d(q, R_4)$	$= \text{NA}$
$\dots$	
$d(q, R_N)$	$= 0.244$

- ▶ from each read to all sufficiently similar genomes

phylogenetic placement:



- ▶ go beyond markers
- ▶ ultra-large phylogenies

Problem statement and our goals

Given:

- query sequence q (e.g., a short read)
- set of references $\mathcal{R} = \{R_1, \dots, R_N\}$
- a backbone phylogeny T

krepp solves both of these problems for genome-wide queries (distances & PP)

reference genomes

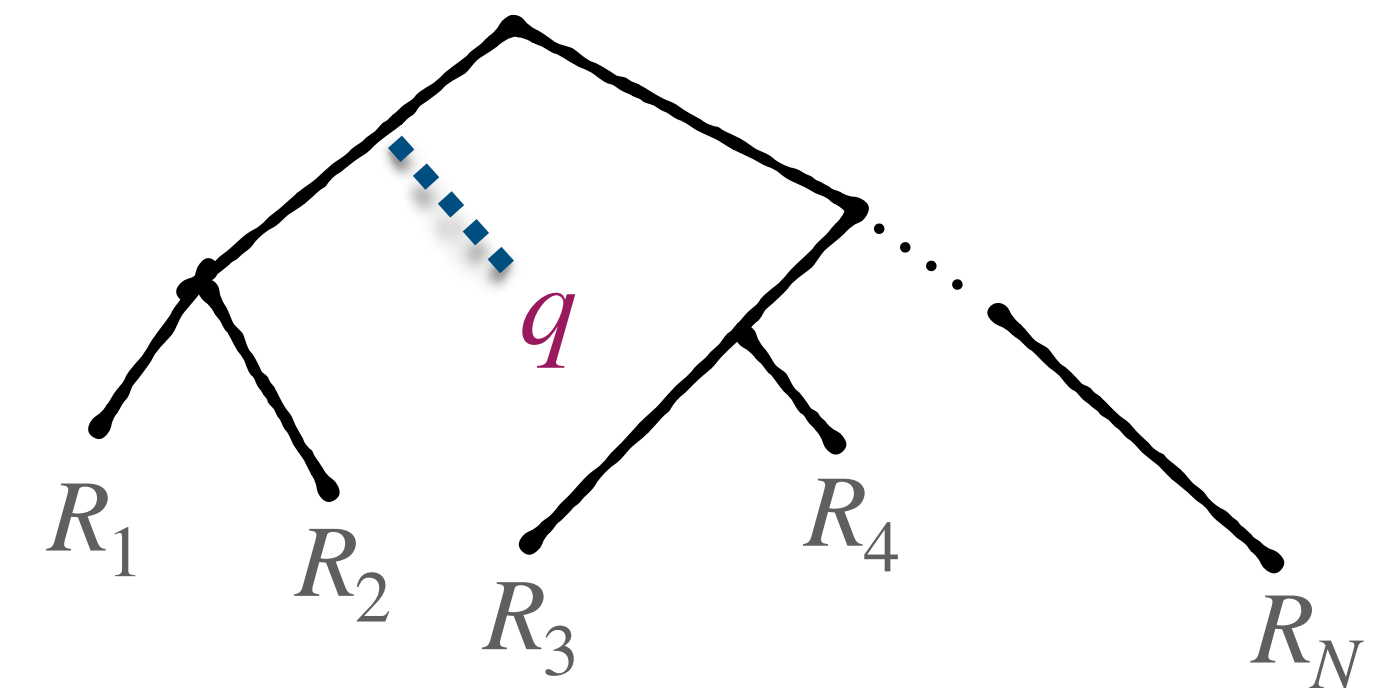
R_1 : TCCCTGCTCA...
 R_2 : TCCCTGCTAA...
 R_3 : CCCCTGGCAG...
 R_4 : ATTATCTGAT...
...
 R_N : CCCCAAACAA...

distance estimates

$d(q, R_1) = 0.141$
 $d(q, R_2) = 0.001$
 $d(q, R_3) = 0.195$
 $d(q, R_4) = \text{NA}$
...
 $d(q, R_N) = 0.244$

- ▶ from each read to all sufficiently similar genomes

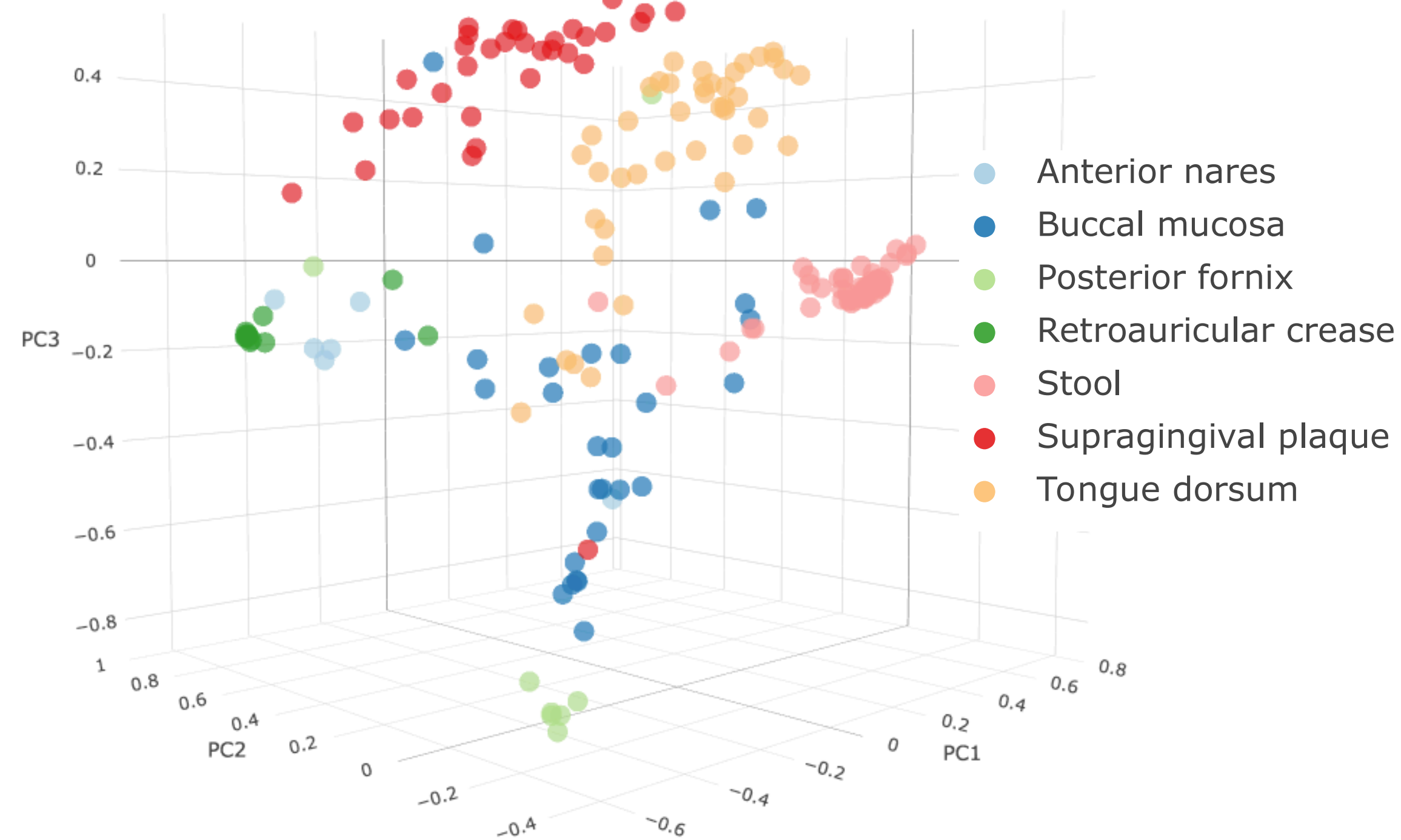
phylogenetic placement:



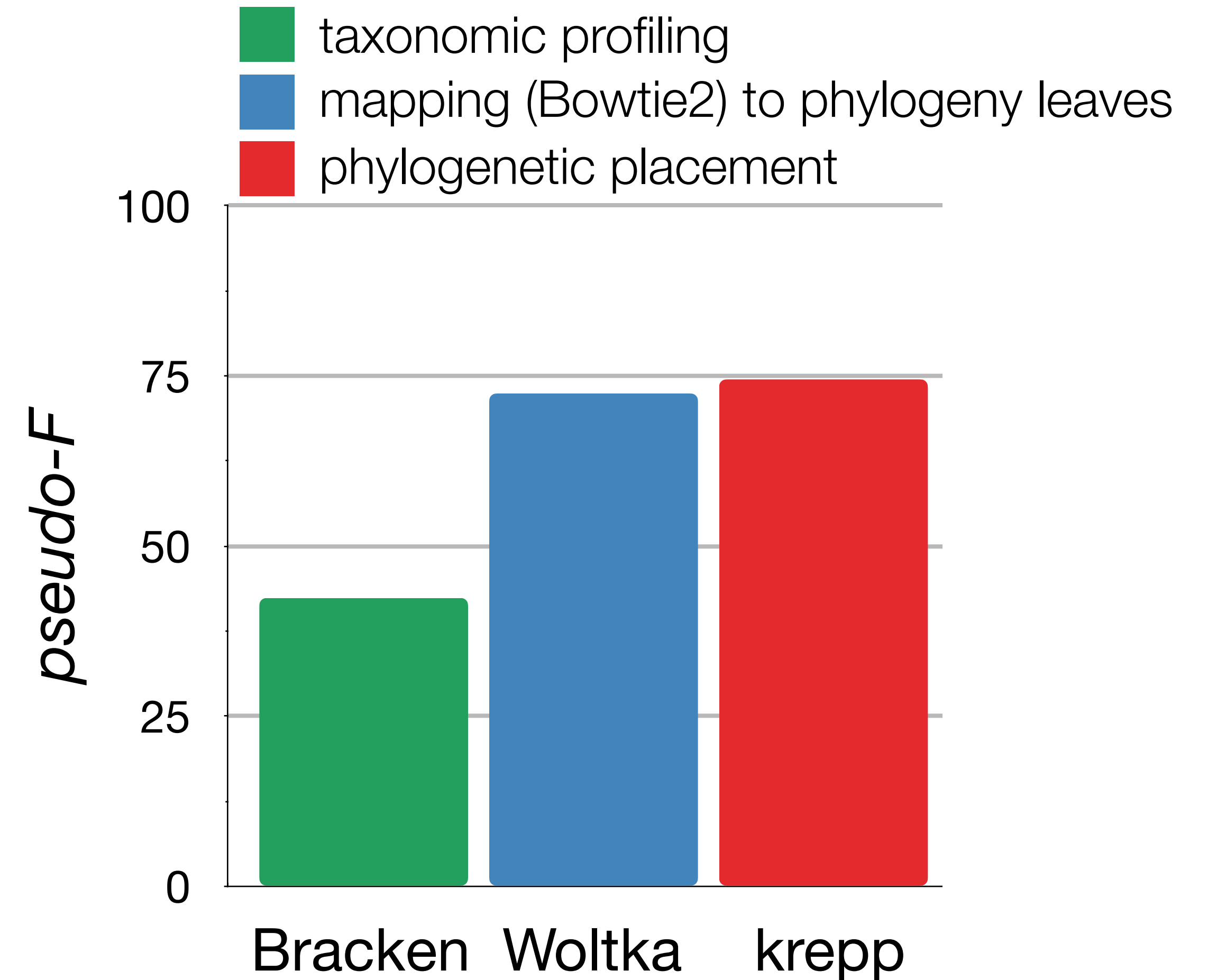
- ▶ go beyond markers
- ▶ ultra-large phylogenies

$>q$
CCTGCTA...

krepp can estimate distances & phylogenetically place reads

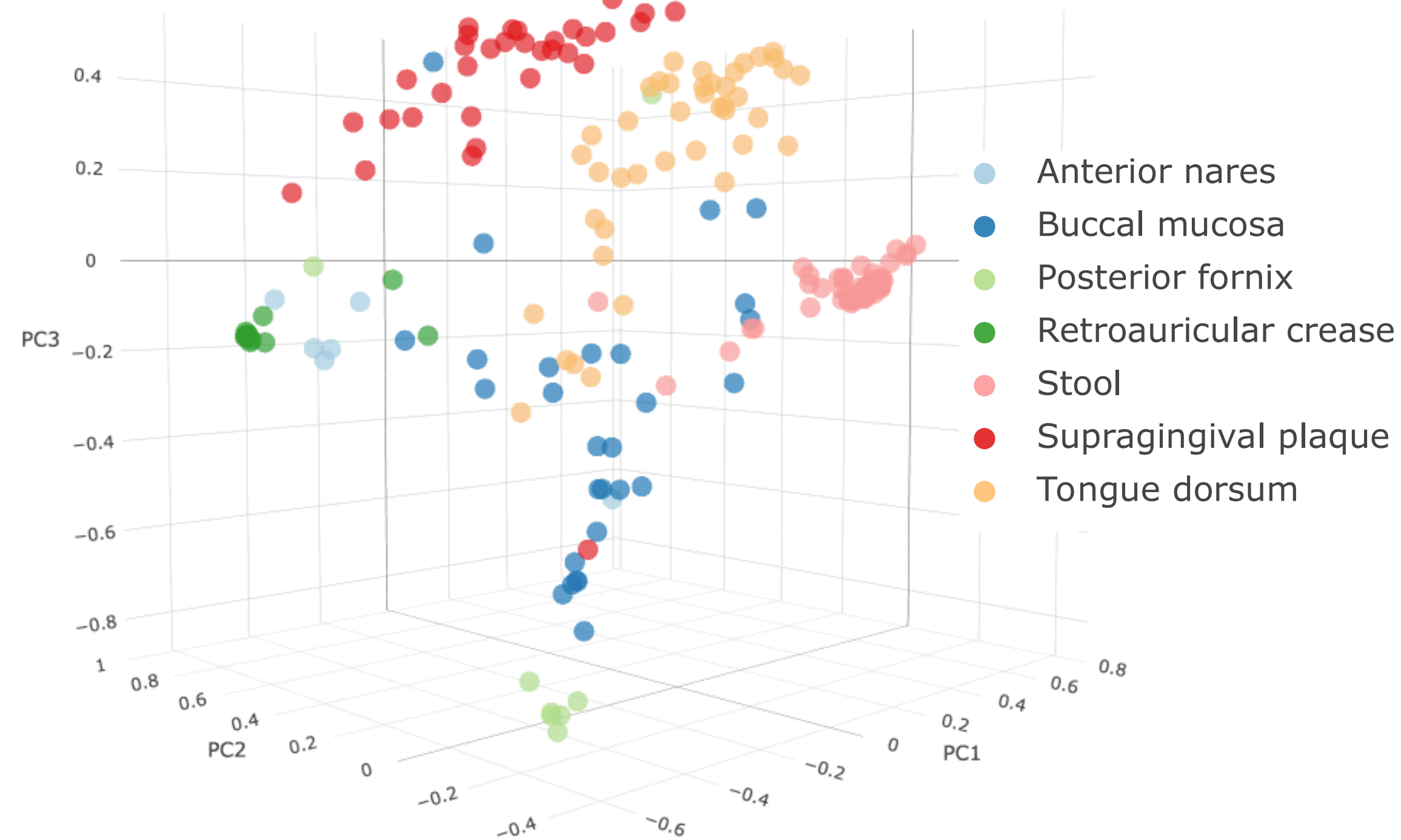


PCoA for 7 body sites, 210 samples



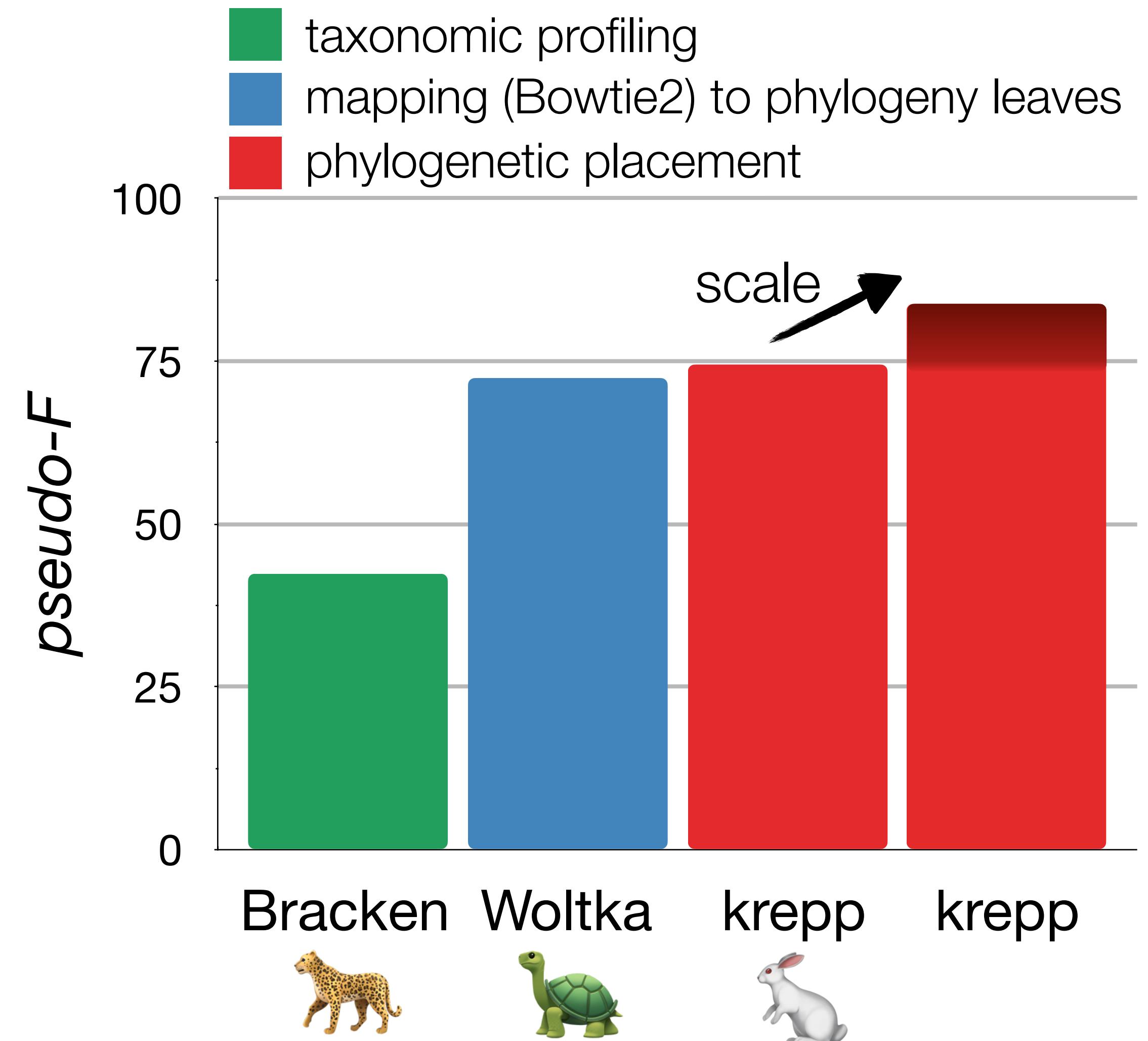
krepp can estimate distances & phylogenetically place reads

Scaling to large references further improves separation of body sites



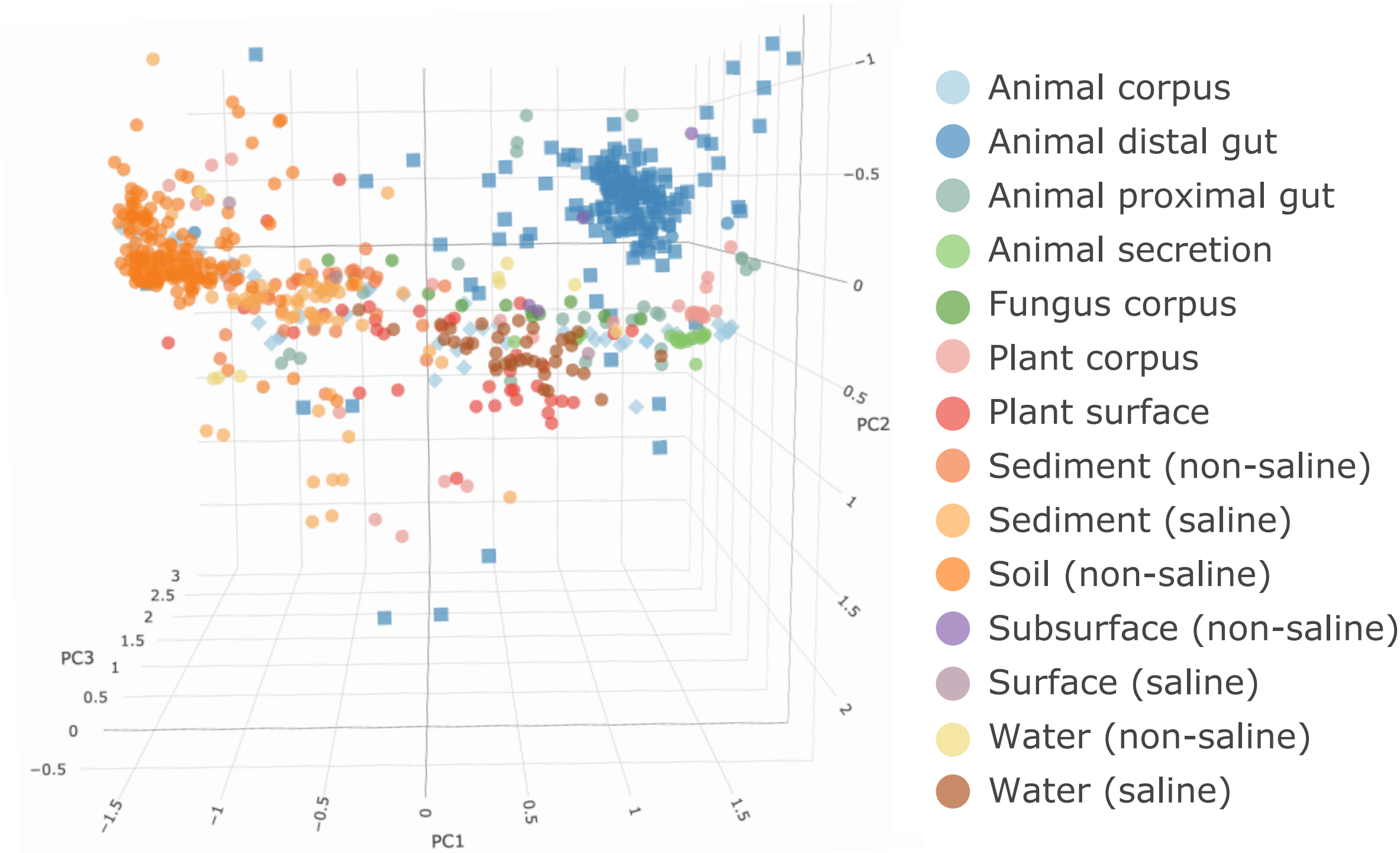
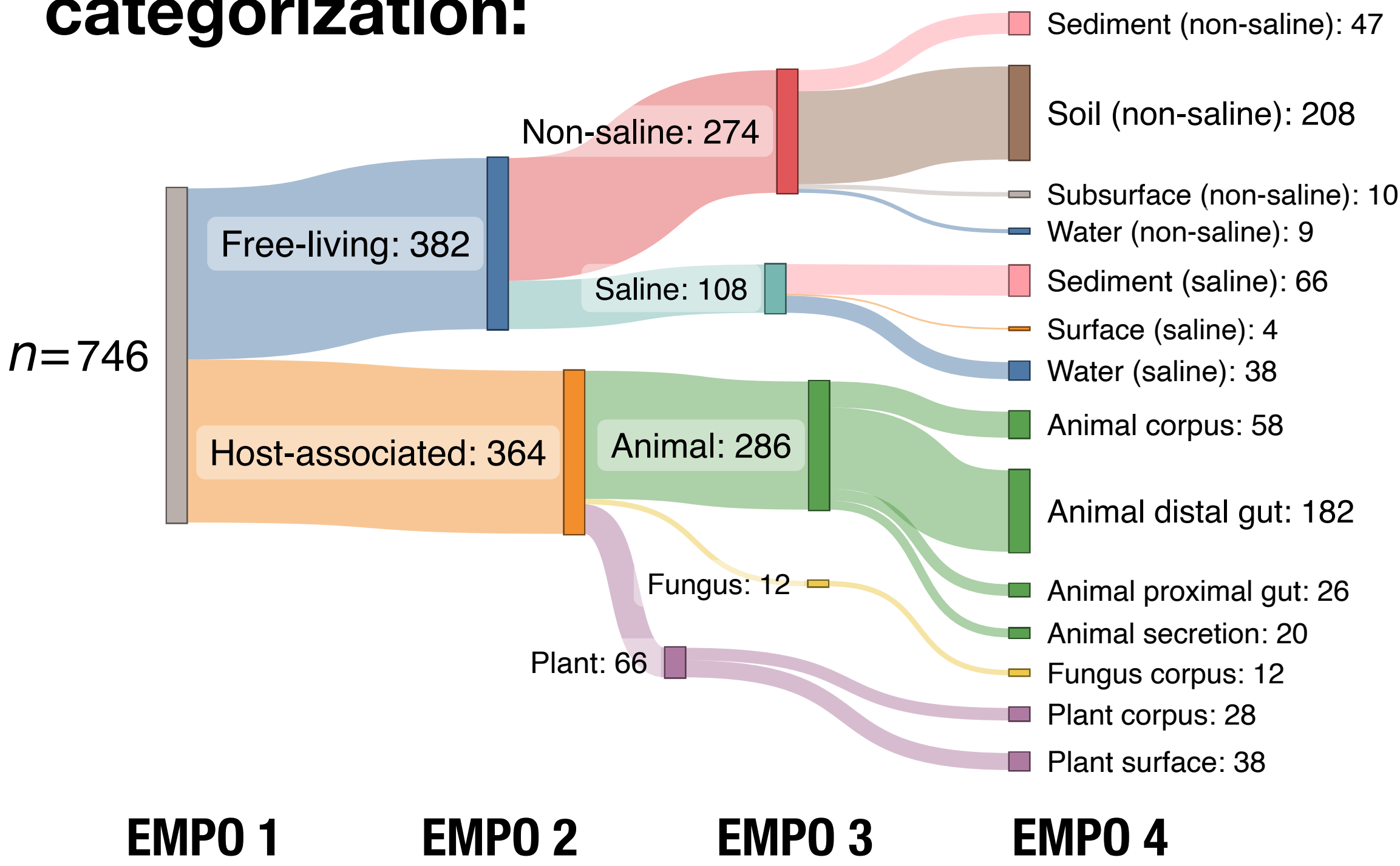
PCoA for 7 body sites, 210 samples

up to 10x faster → use a phylogeny 3x larger

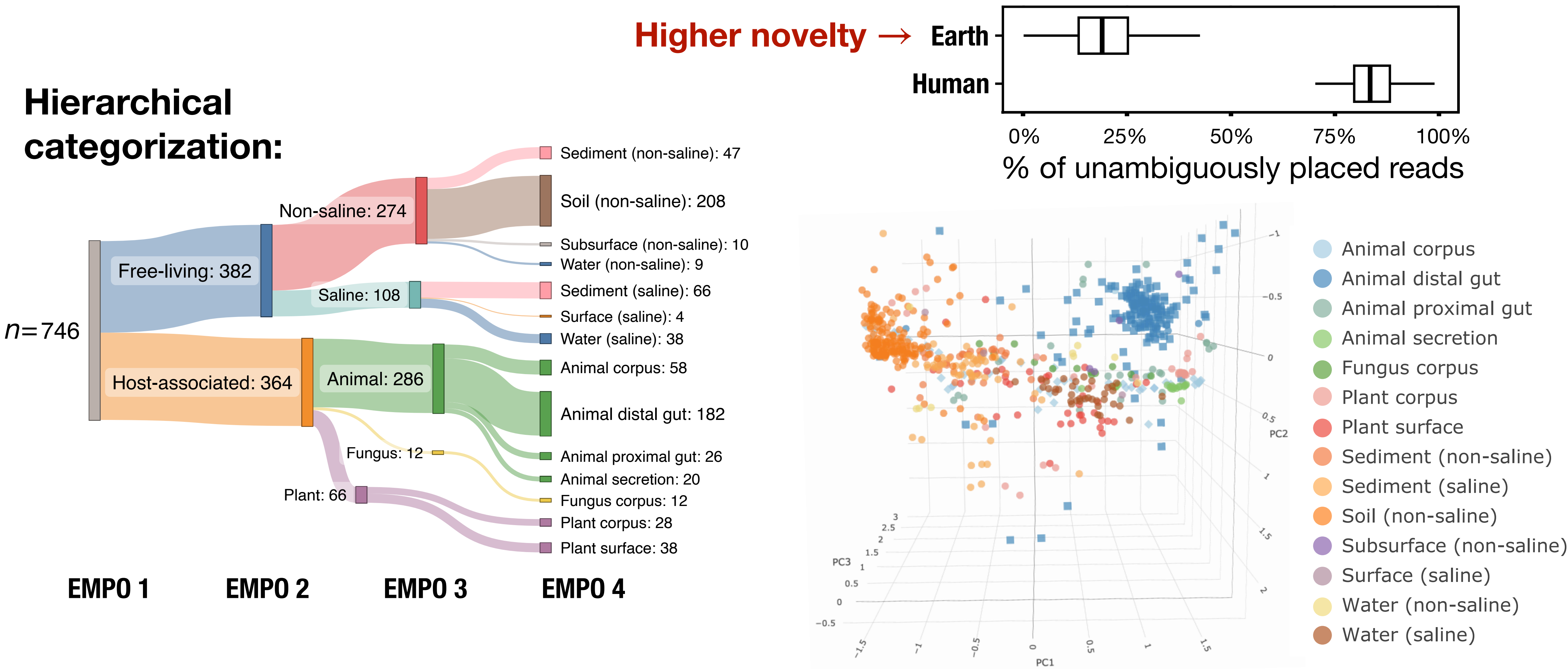


Better characterization of less-studied microbiome of earth

Hierarchical categorization:

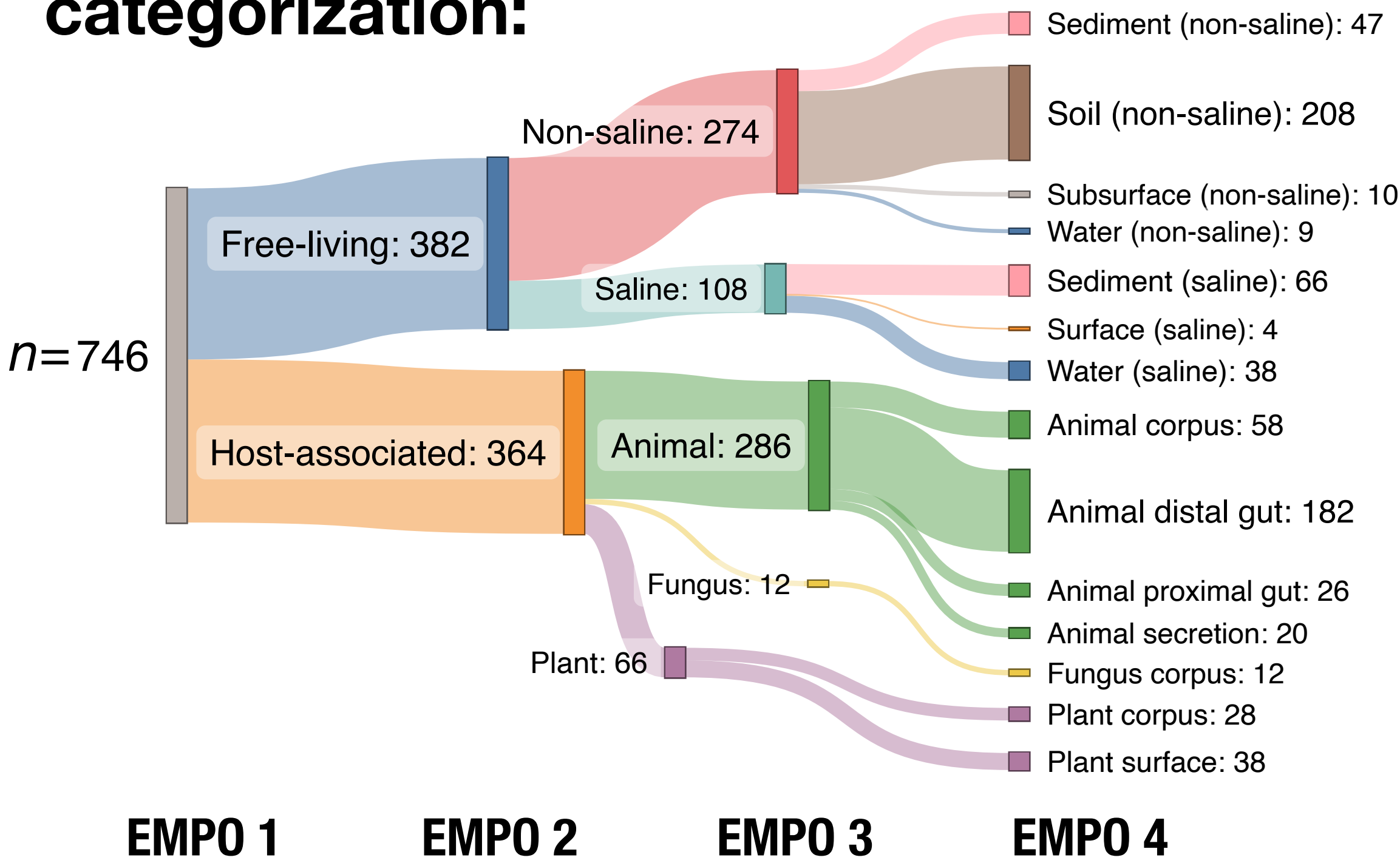


Better characterization of less-studied microbiome of earth

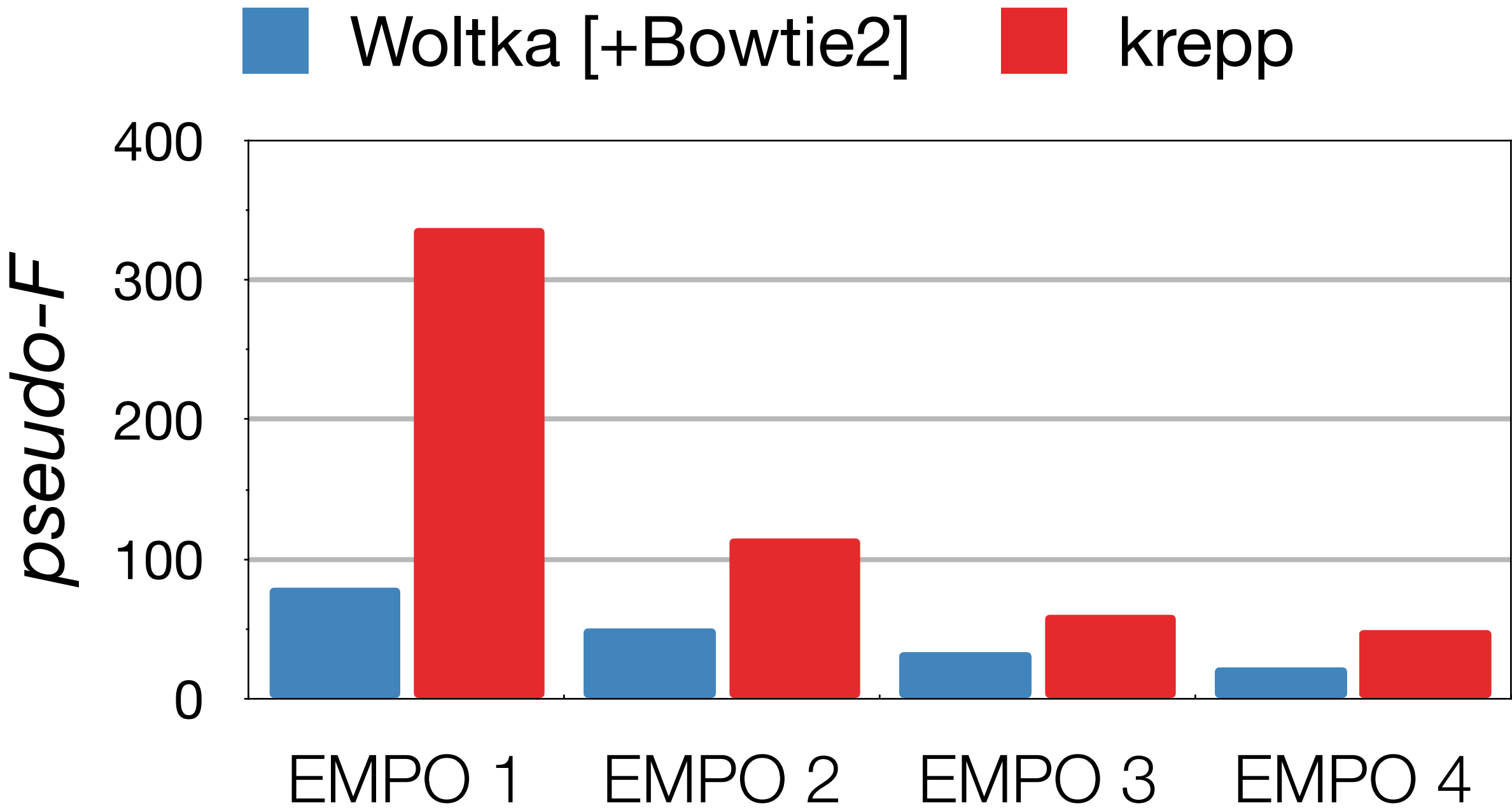


Better characterization of less-studied microbiome of earth

Hierarchical categorization:



krepp: scalable distance-based read mapping with high sensitivity

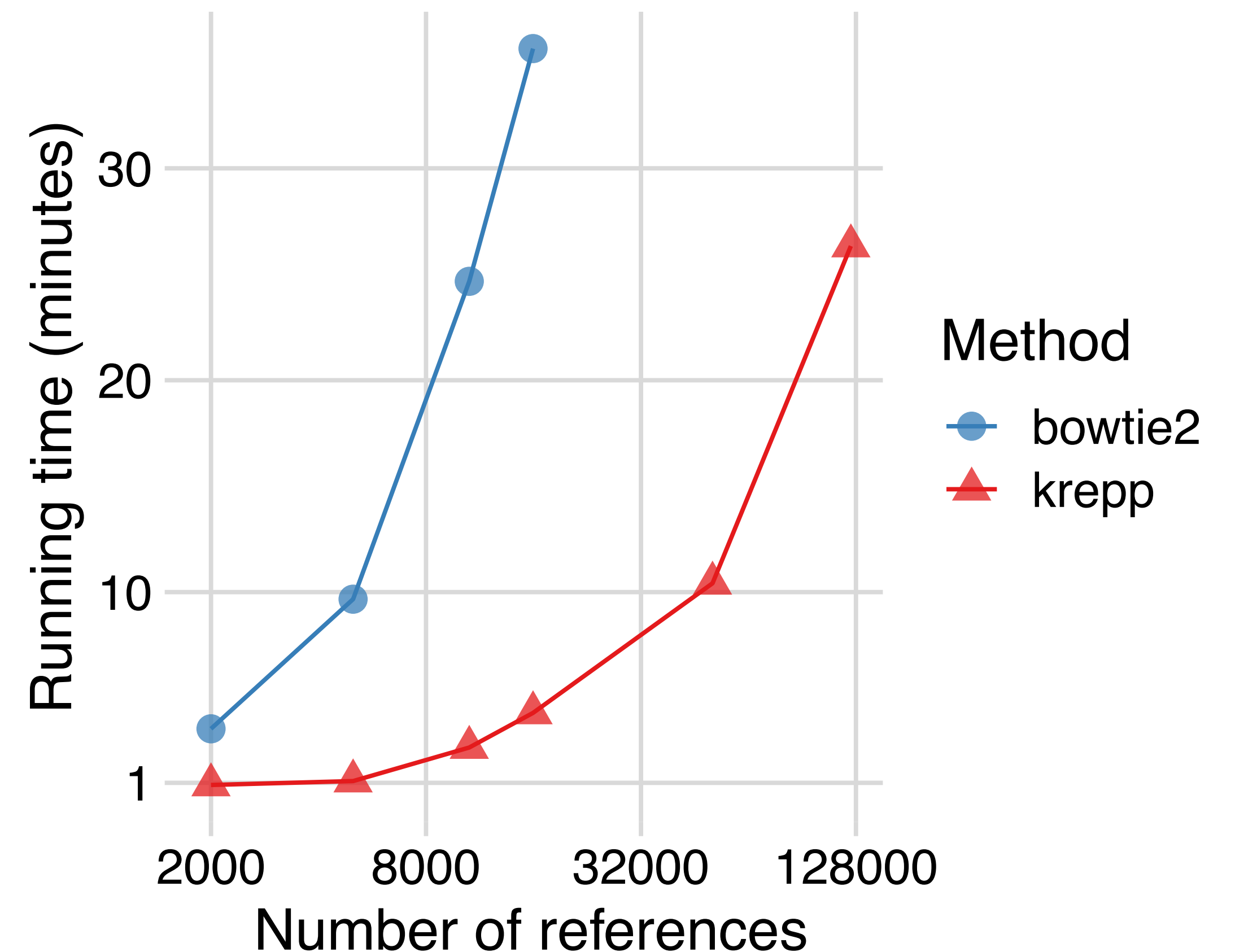


Scalable mapping: **avoiding the costly alignment step**

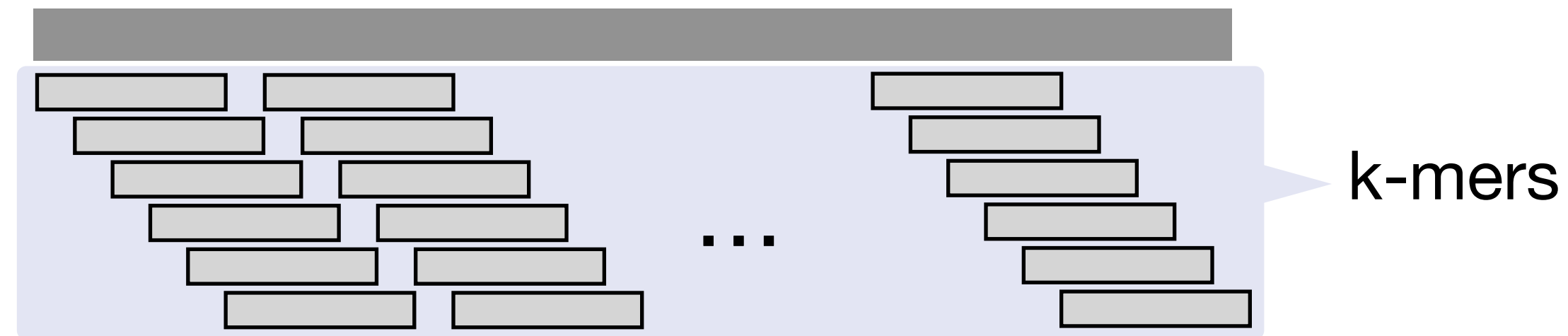
~150 bp Illumina short reads

Mapping 10M reads (16 threads):

- ▶ krepp maps reads **~10x** faster
(compared to `--high-sensitivity`)
- ▶ Indexing is **>3-5x** faster than Bowtie2



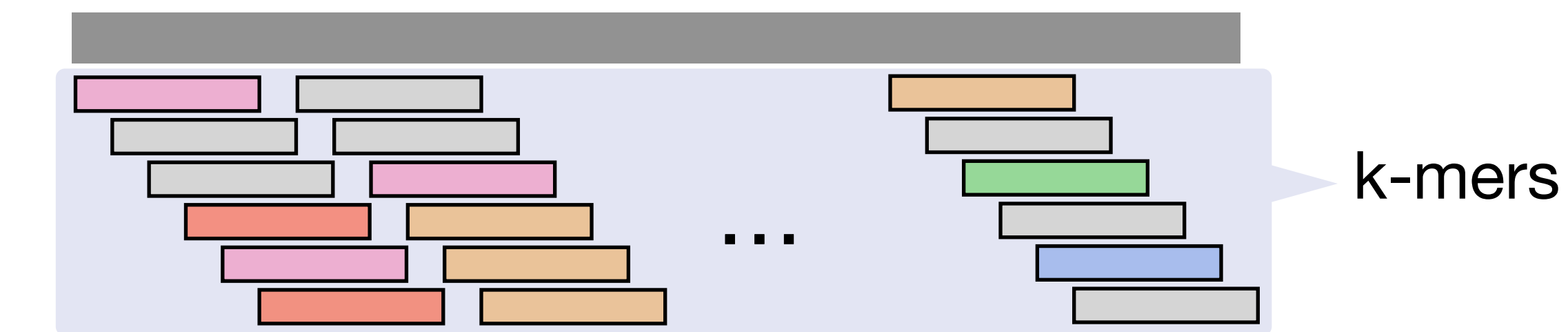
How? From counting matches to estimating distances



Kraken2:

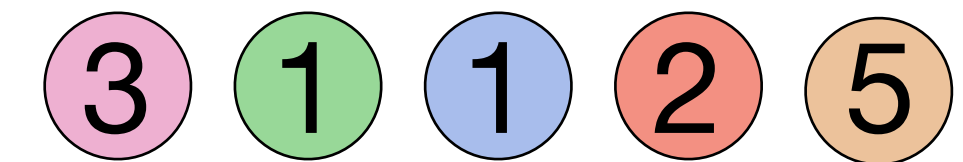
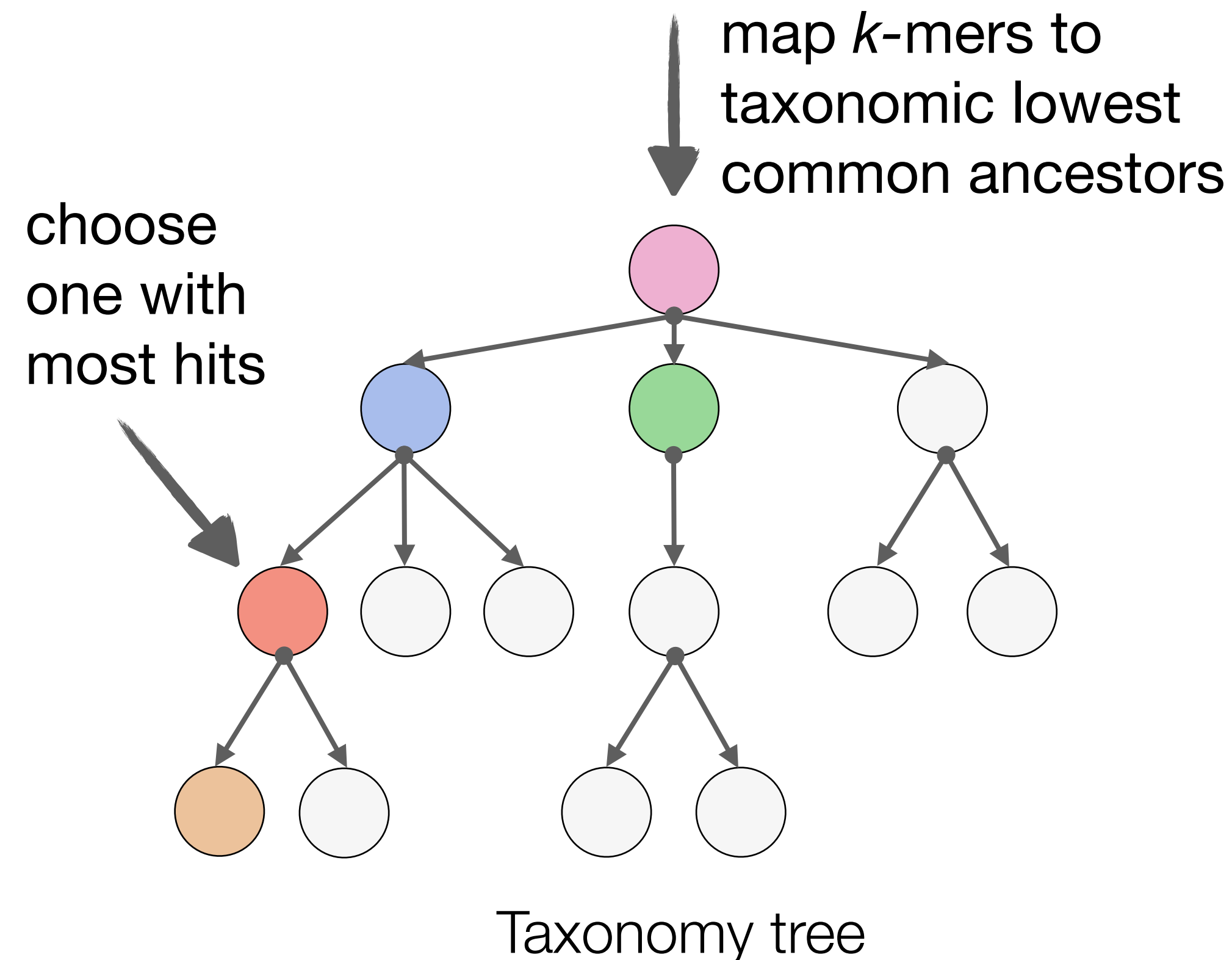
- Map k -mers to **taxonomic LCAs**
- Find the best path based on match counts

How? From counting matches to estimating distances

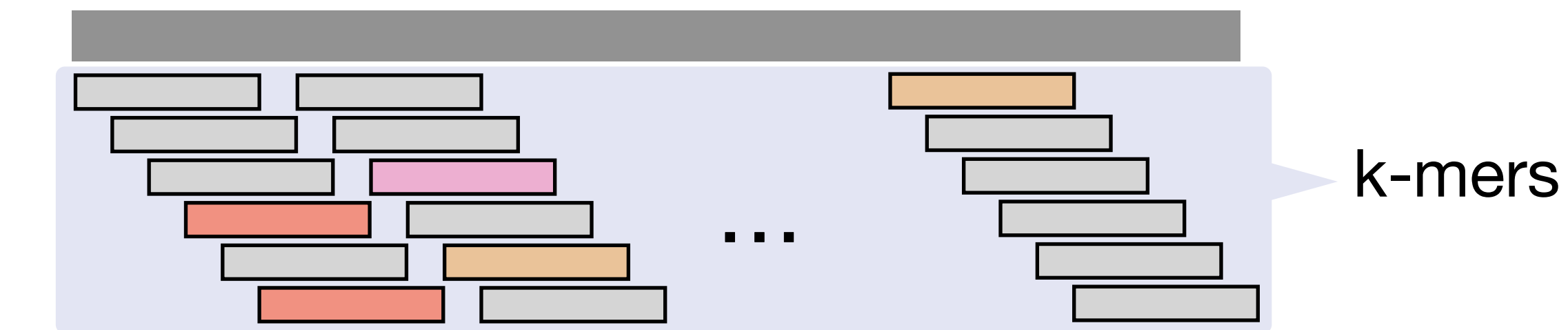


Kraken2:

- Map *k*-mers to **taxonomic LCAs**
- Find the best path based on match counts

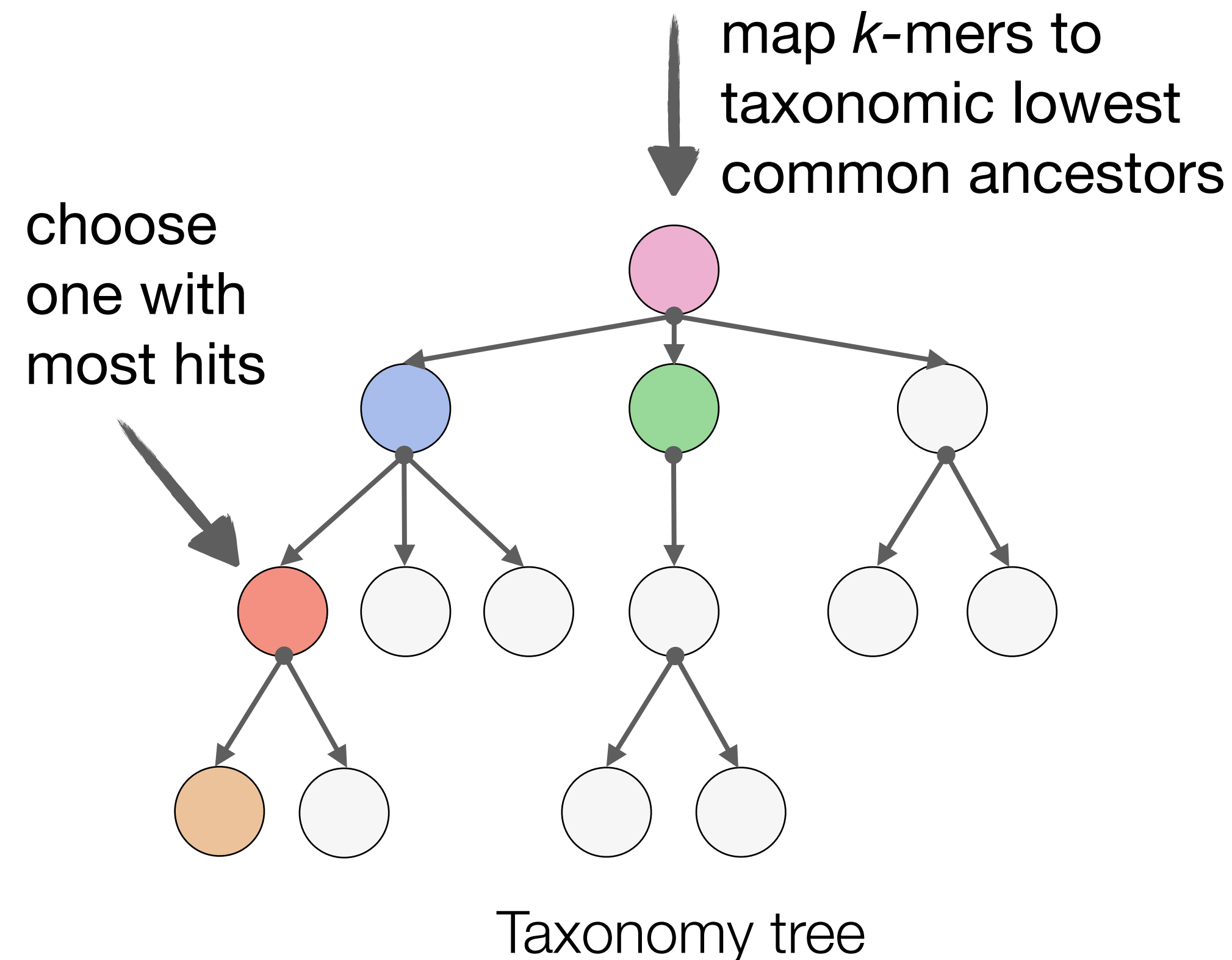


How? From counting matches to estimating distances



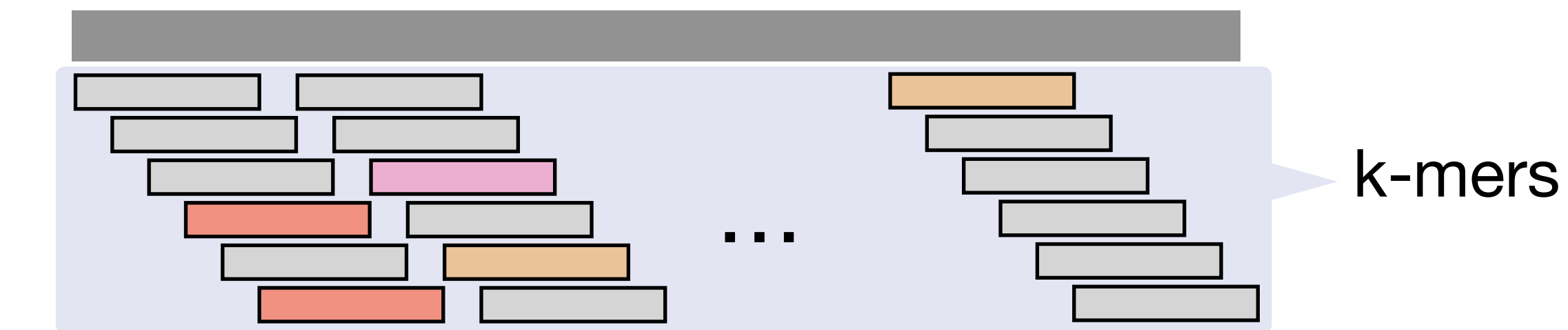
Kraken2:

- Map k -mers to **taxonomic LCAs**
- Find the best path based on match counts



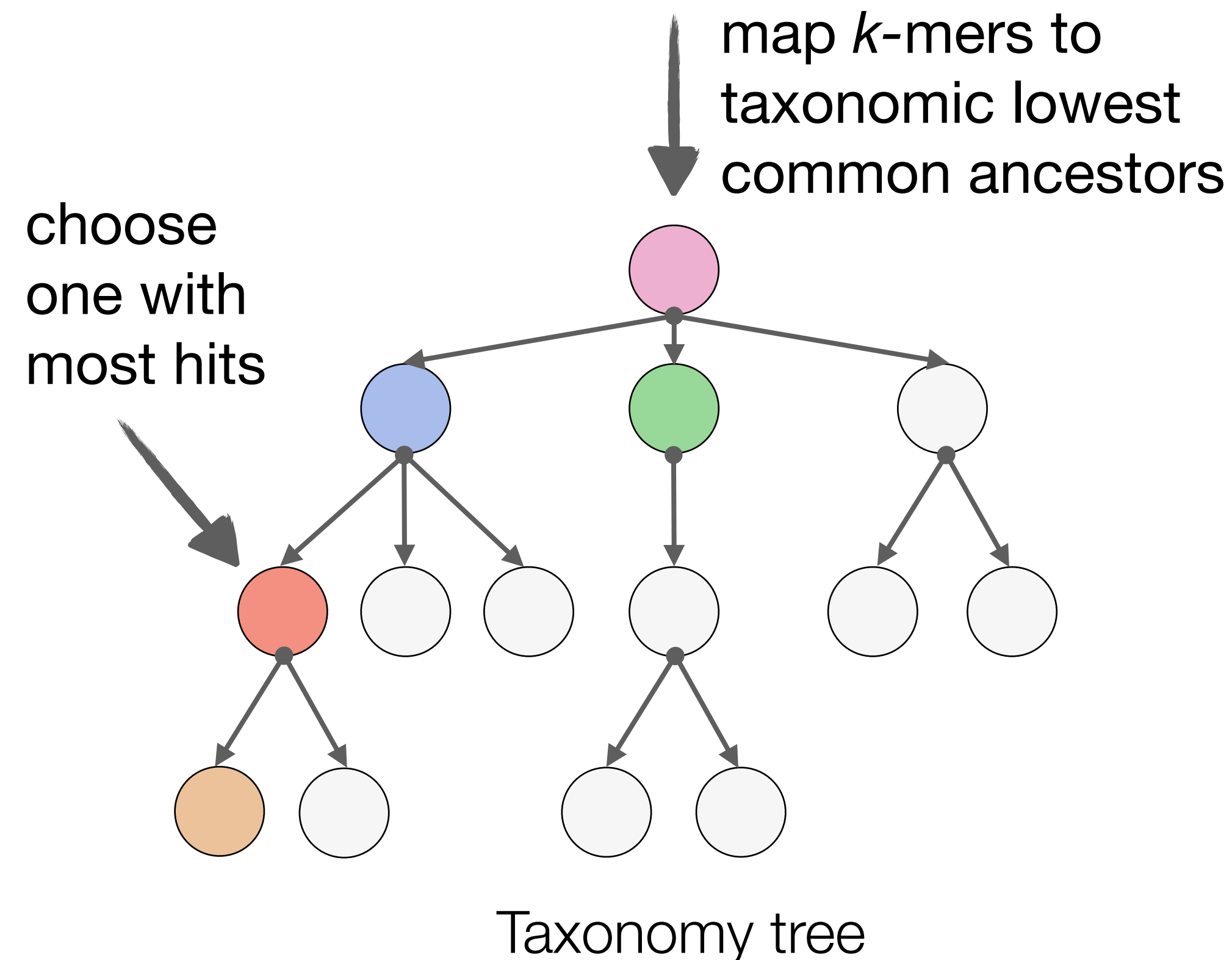
3	1	1	2	5
1	0	0	2	2

How? From counting matches to estimating distances



Kraken2:

- Map k -mers to **taxonomic LCAs**
- Find the best path based on match counts

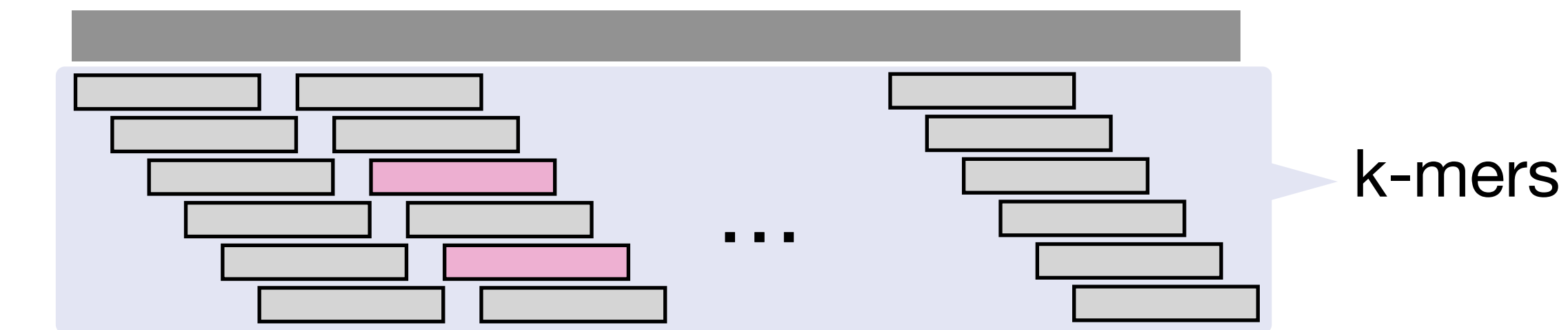


The **number of matching k -mers** tells something about the **similarity!**

quite similar 3 1 1 2 5

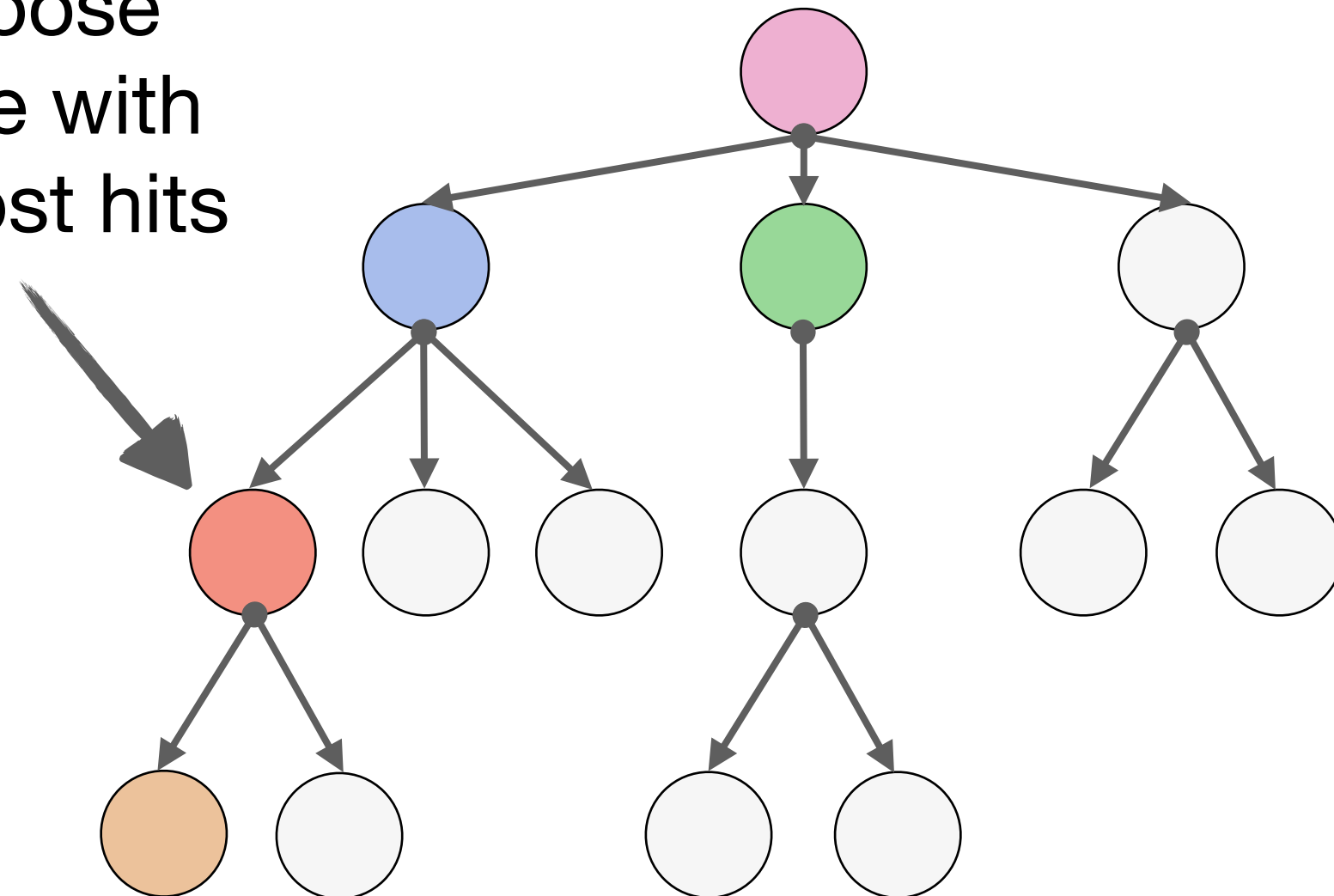
somewhat distant 1 0 0 2 2

How? From counting matches to estimating distances



map k -mers to
taxonomic lowest
common ancestors

choose
one with
most hits



Taxonomy tree

Kraken2:

- Map k -mers to **taxonomic LCAs**
- Find the best path based on match counts

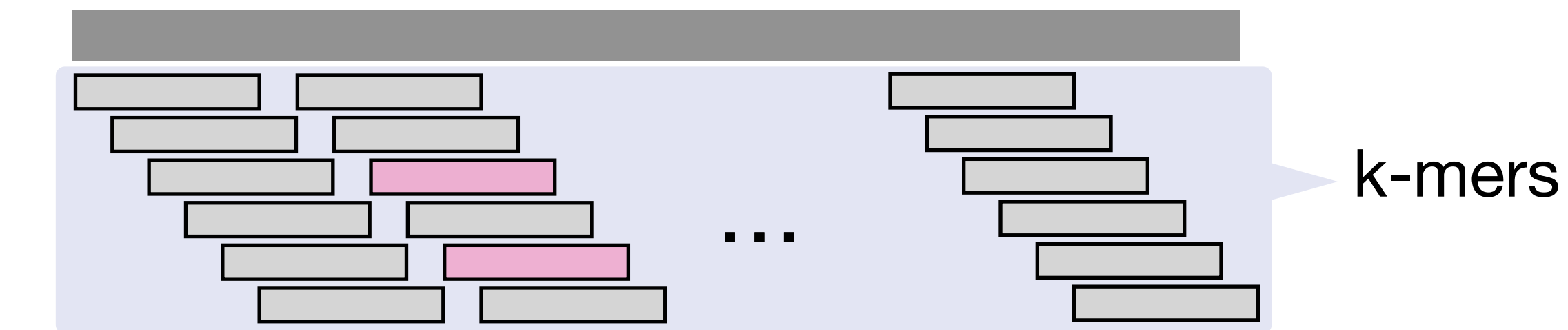
The **number of matching k -mers**
tells something about the **similarity!**

quite similar 3 1 1 2 5

somewhat distant 1 0 0 2 2

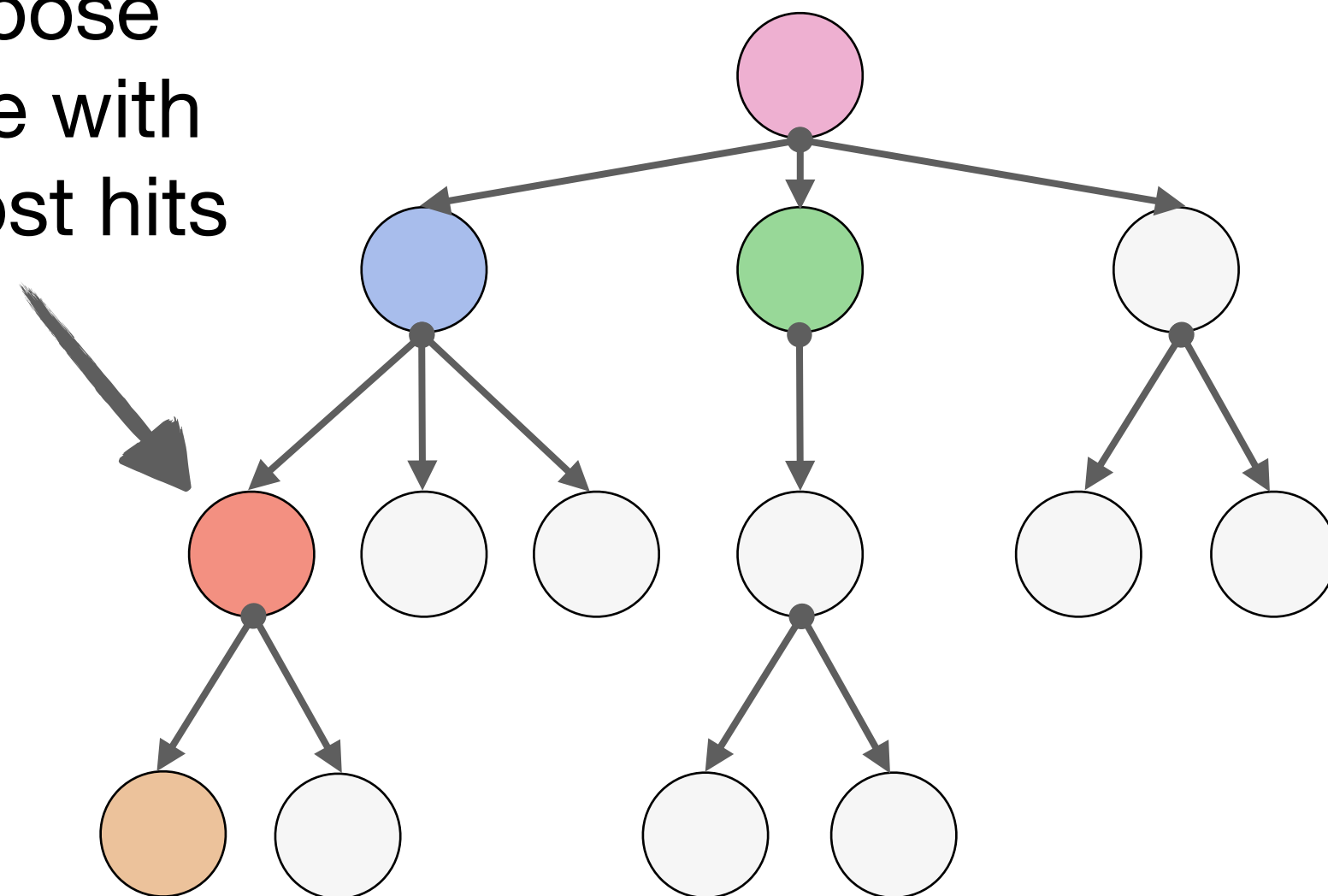
highly novel 2 0 0 0 0

How? From counting matches to estimating distances



map k -mers to
taxonomic lowest
common ancestors

choose
one with
most hits



Taxonomy tree

Kraken2:

- Map k -mers to **taxonomic LCAs**
- Find the best path based on match counts
- Search spaced seeds, masked k -mers...

The **number of matching k -mers**
tells something about the **similarity!**

quite similar 3 1 1 2 5

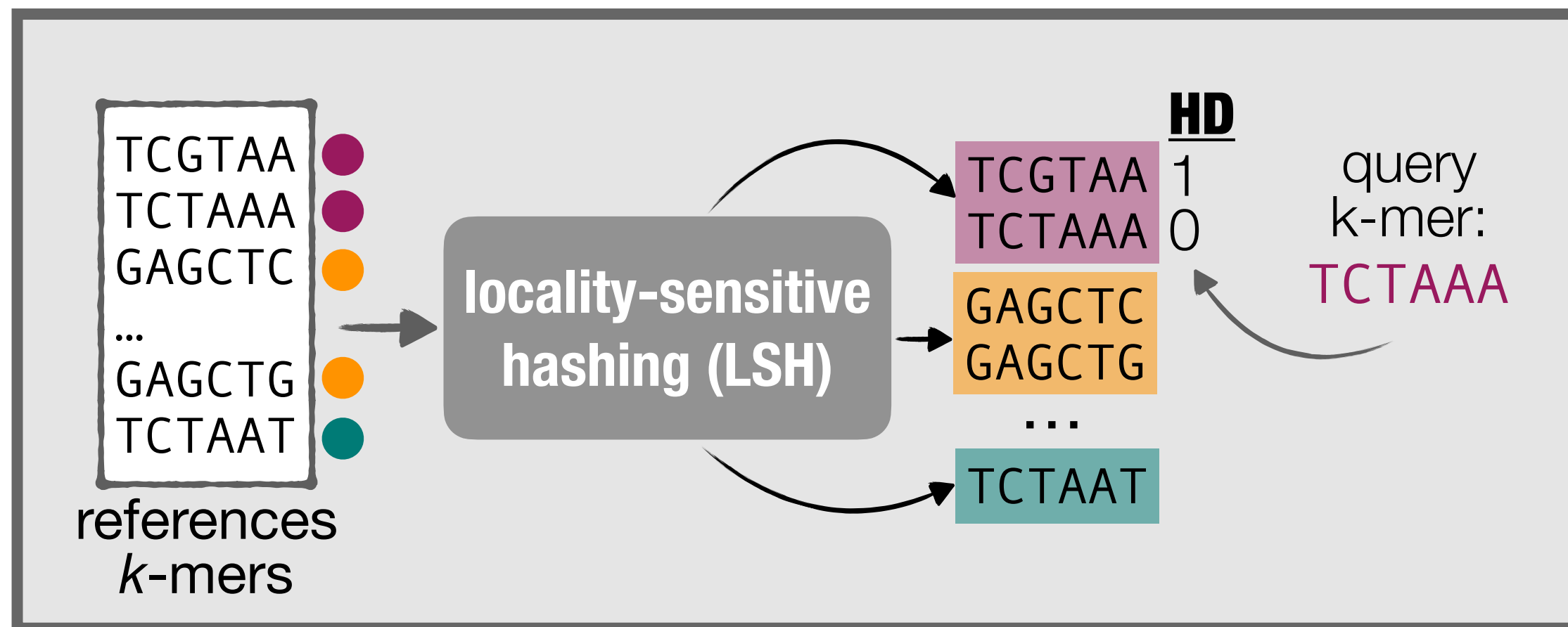
somewhat distant 1 0 0 2 2

highly novel 2 0 0 0 0

A few match may not provide
enough signal to estimate similarity...

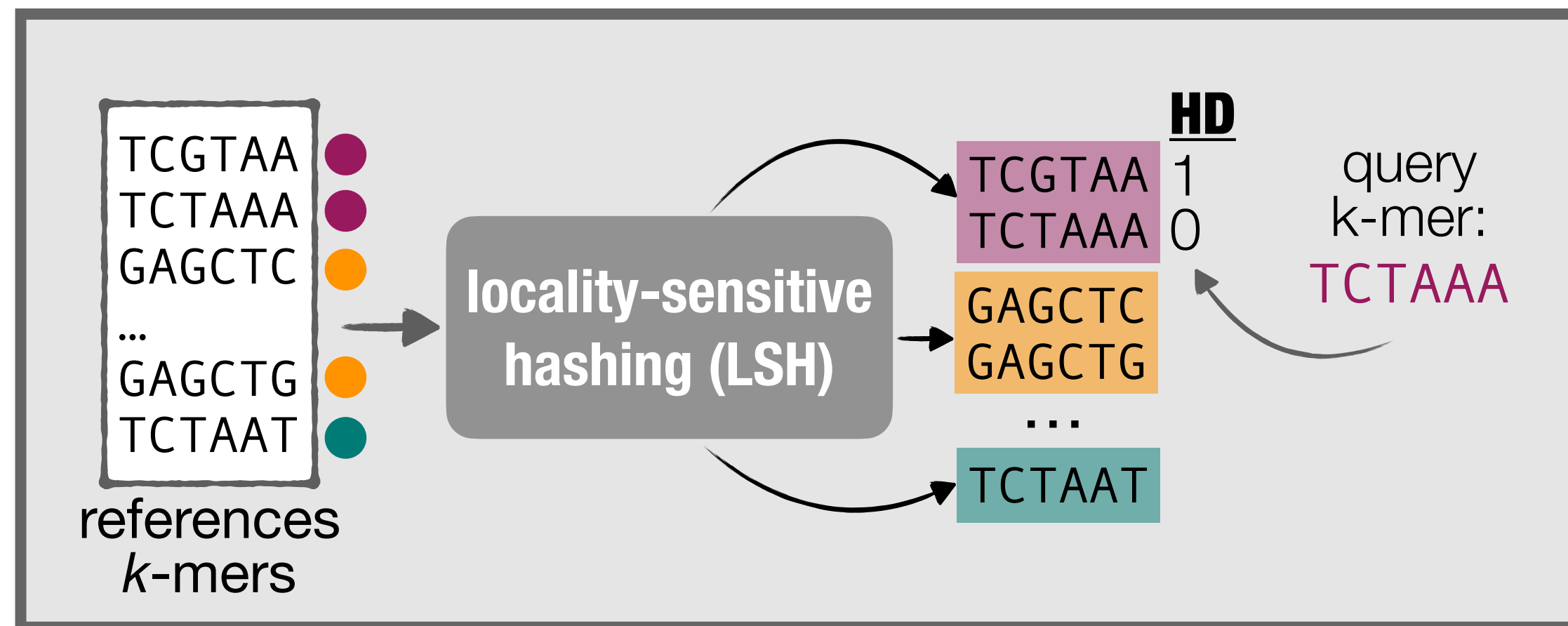
We can do even better: **locality-sensitive hashing**

Finding homologous k-mers using LSH



We can do even better: **locality-sensitive hashing**

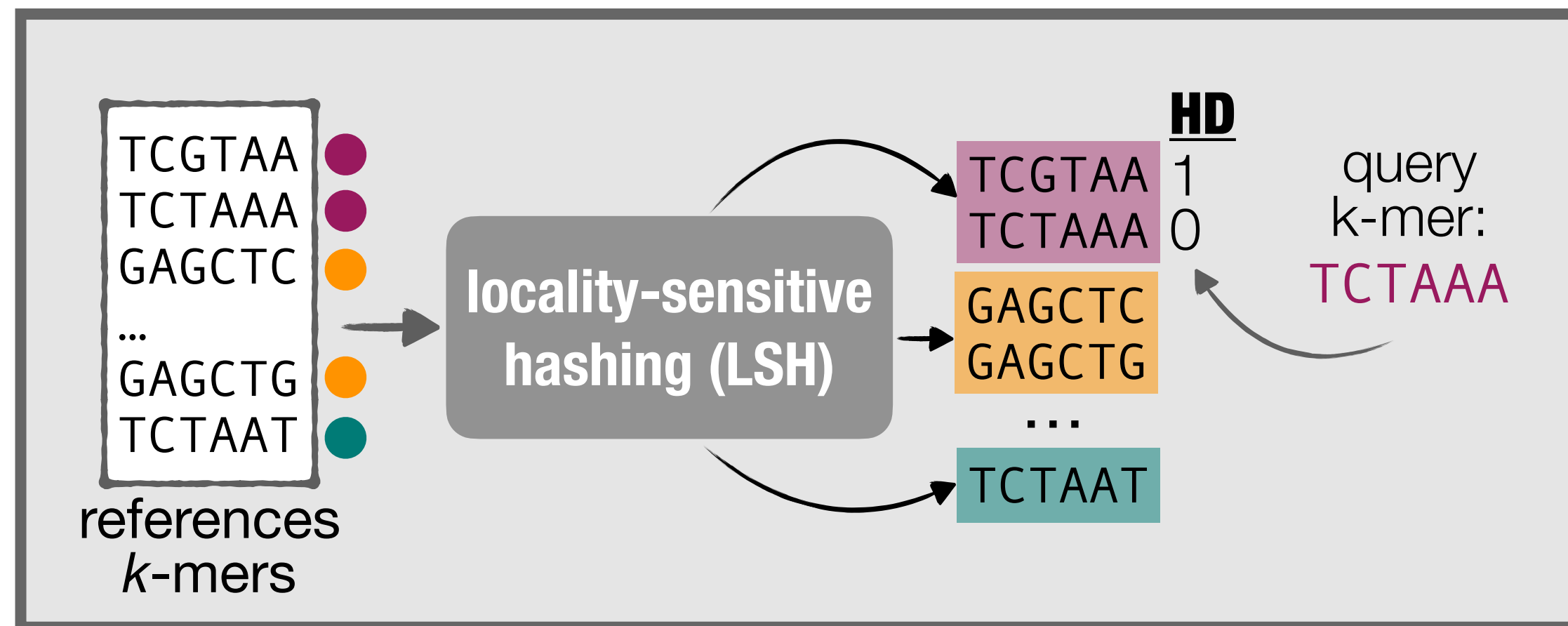
Finding homologous k-mers using LSH



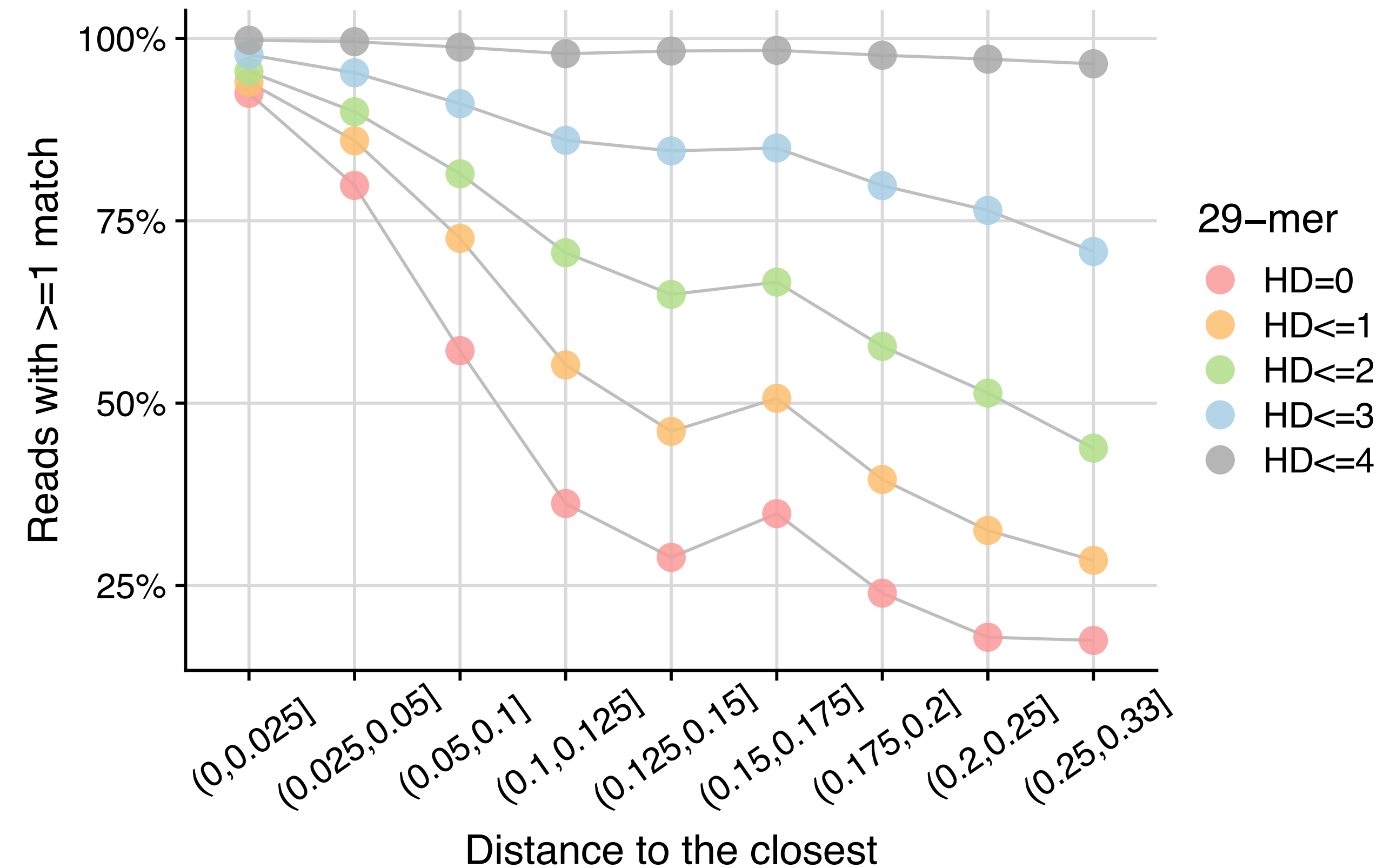
- ▶ **Extra signal:** calculating the Hamming distance
- ▶ **Fast:** limited number of HD calculations
- ▶ Probabilistic guarantees for false negatives rate

We can do even better: **locality-sensitive hashing**

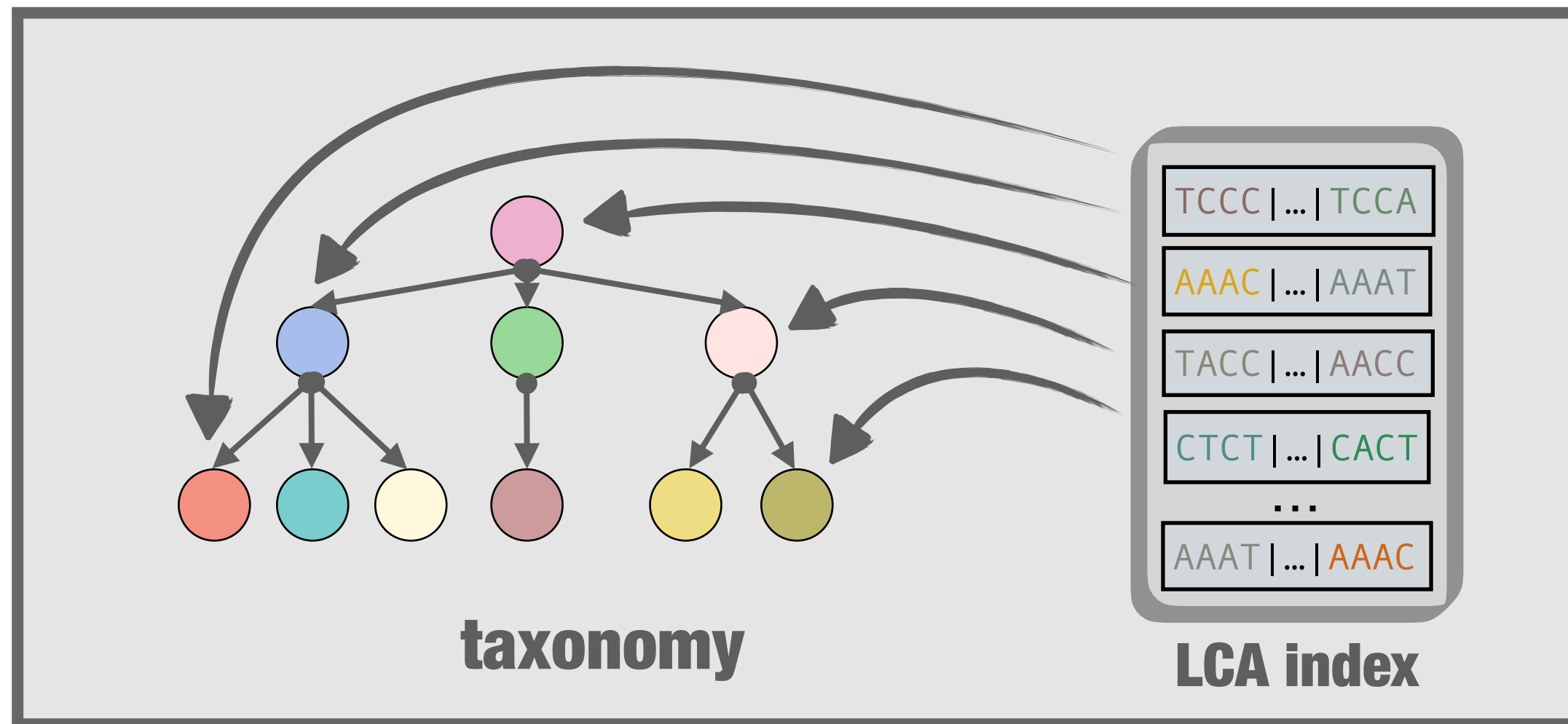
Finding homologous k-mers using LSH



- ▶ **Extra signal:** calculating the Hamming distance
- ▶ **Fast:** limited number of HD calculations
- ▶ Probabilistic guarantees for false negatives rate
- ▶ **More sensitive** than exact *k*-mer search

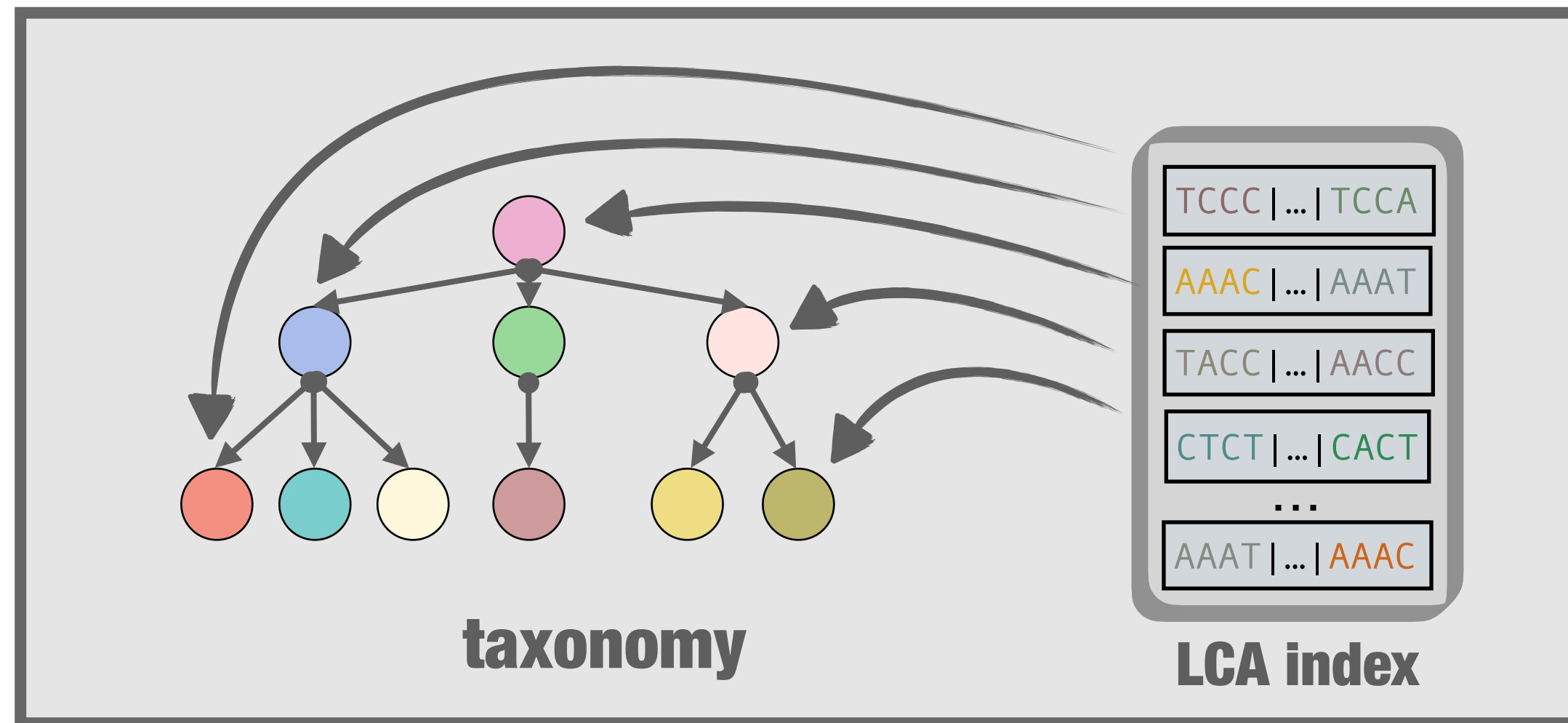


Mapping resolution: reference genomes instead of taxonomic LCAs



- (+) memory efficient; one label per k -mer
- (-) loss of information
- (-) sensitive to outliers, contamination
- (-) too coarse for some tasks

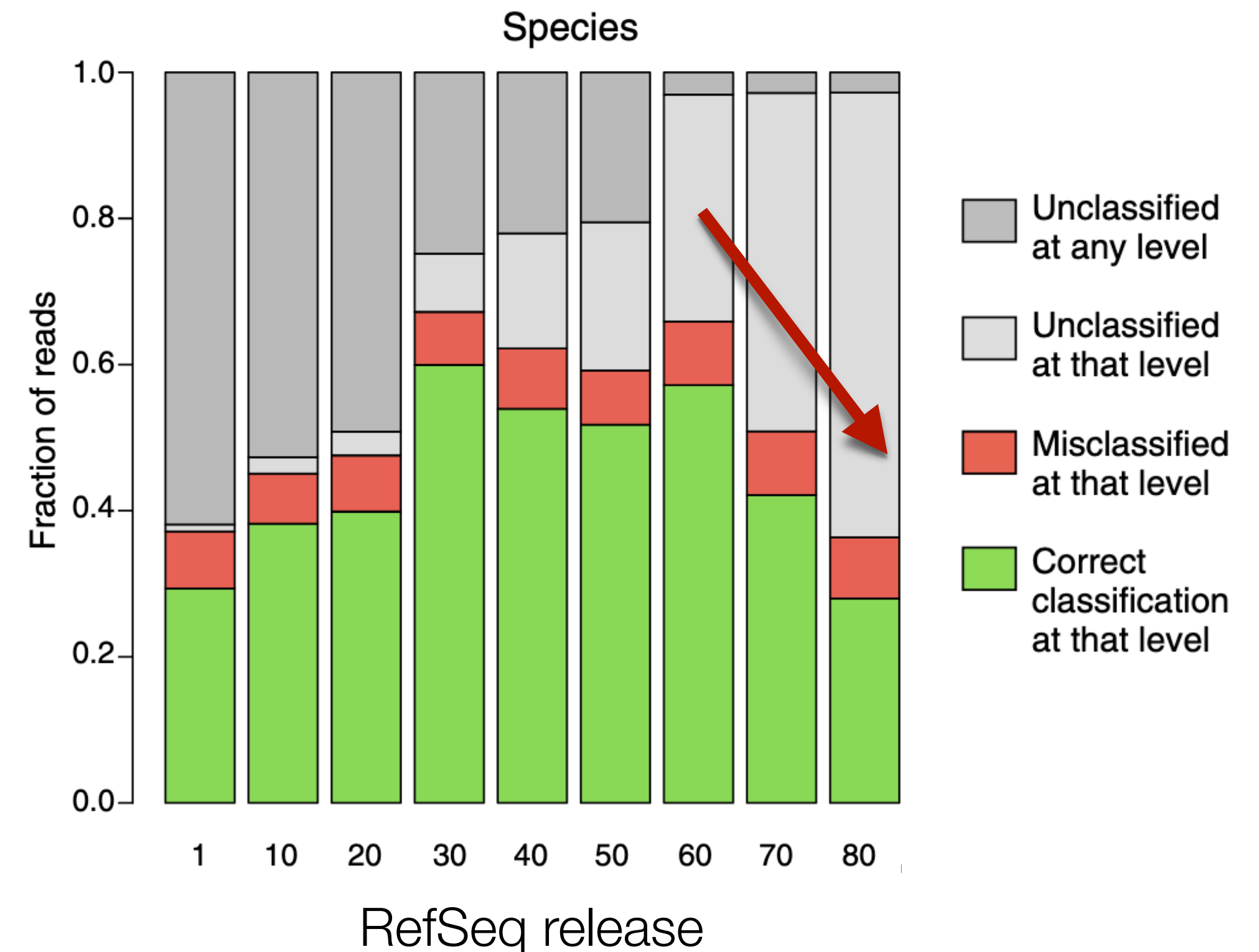
Mapping resolution: reference genomes instead of taxonomic LCAs



- (+) memory efficient; one label per k -mer
- (-) loss of information
- (-) sensitive to outliers, contamination
- (-) too coarse for some tasks

RefSeq database growth influences the accuracy of k -mer-based lowest common ancestor species identification

Daniel J. Nasko¹, Sergey Koren², Adam M. Phillippy² and Todd J. Treangen^{3*} 



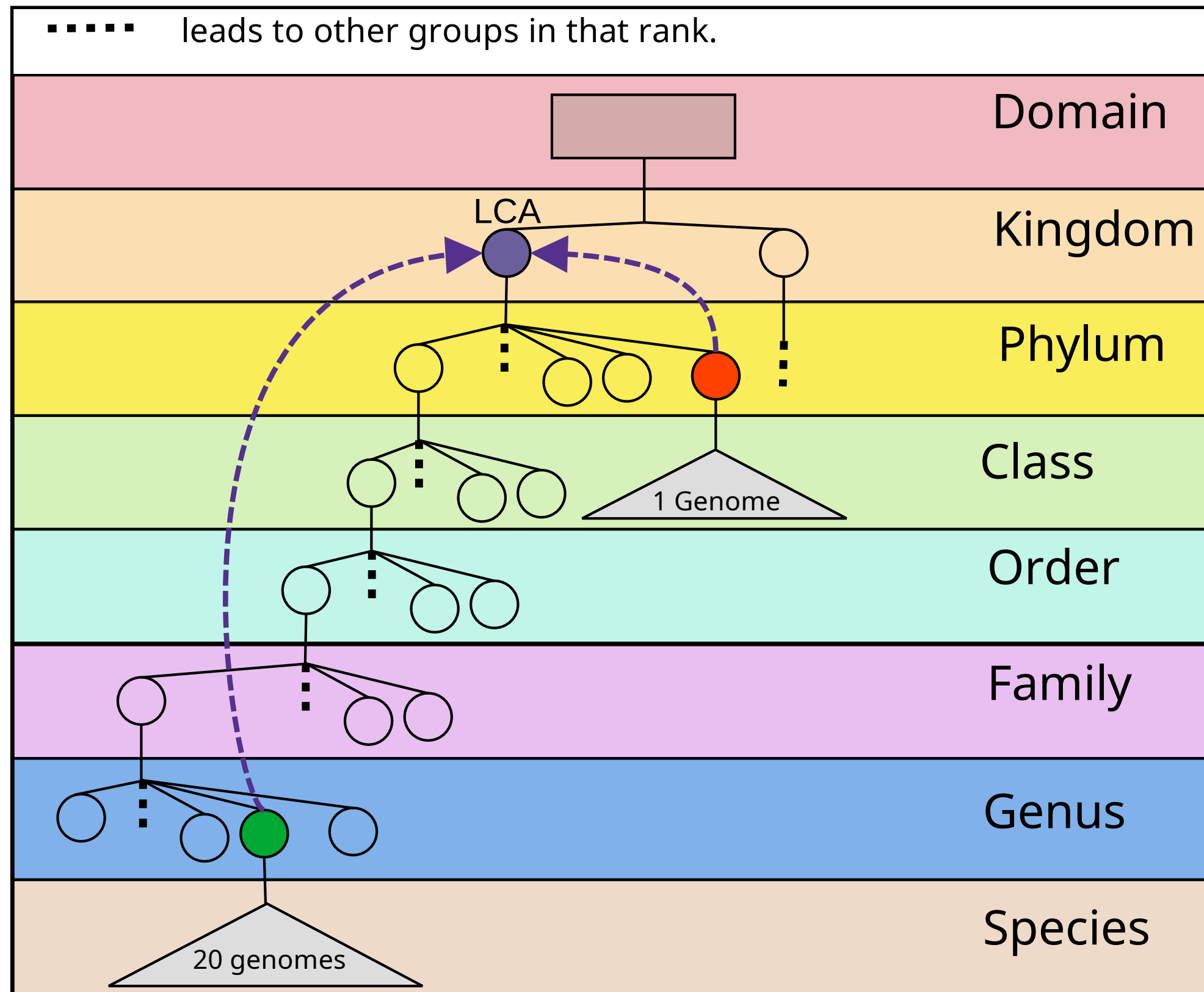
Why not use just the LCA as other methods do?

reference errors, contamination, complex evolutionary processes (e.g., HGT)

Why not use just the LCA as other methods do?

reference errors, contamination, complex evolutionary processes (e.g., HGT)

- ▶ **Correct genus:** 20 genomes
- ▶ **Erroneous phylum:** 1 genome
- Kraken would push the LCA to the **kingdom**



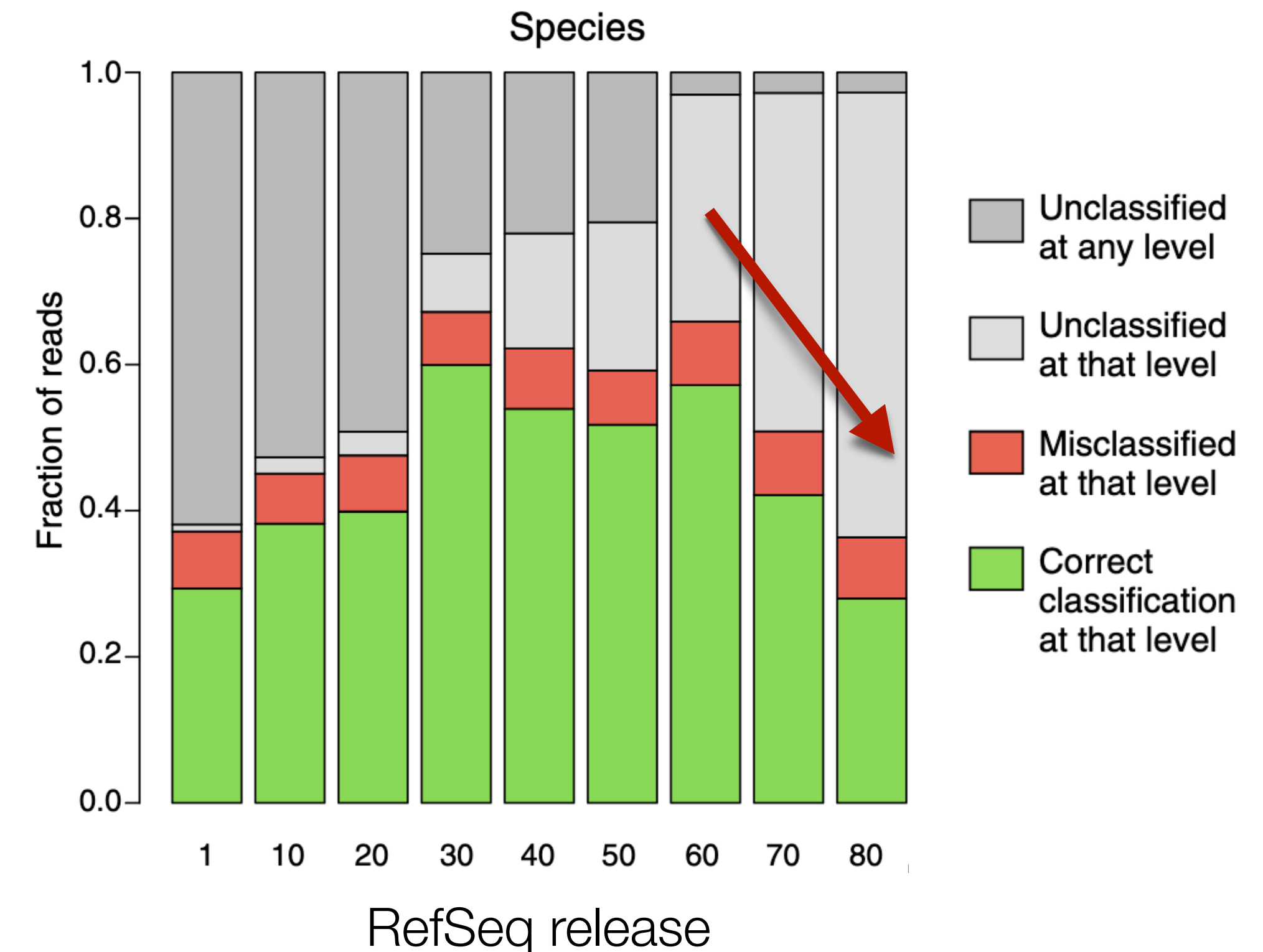
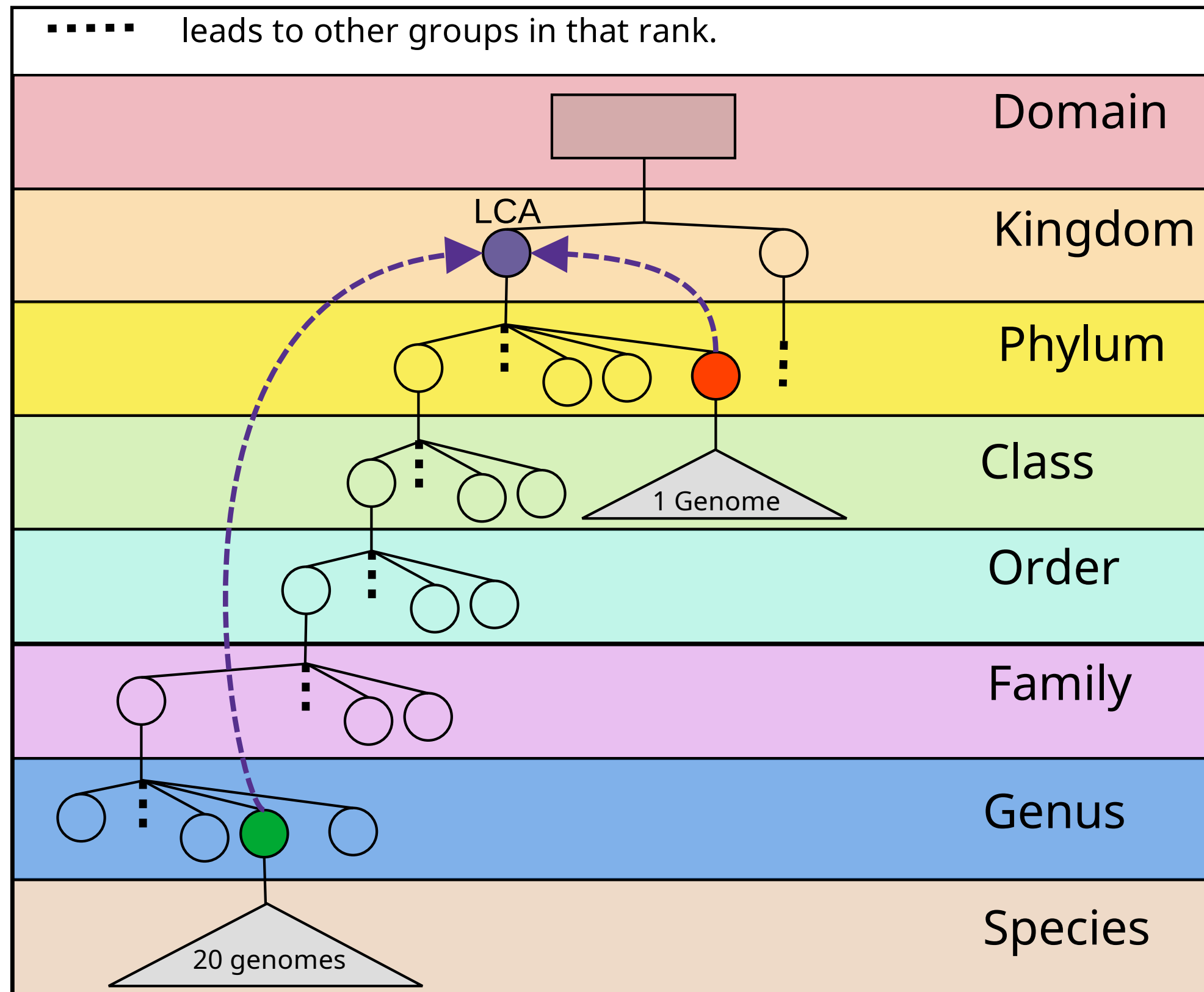
Why not use just the LCA as other methods do?

reference errors, contamination, complex evolutionary processes (e.g., HGT)

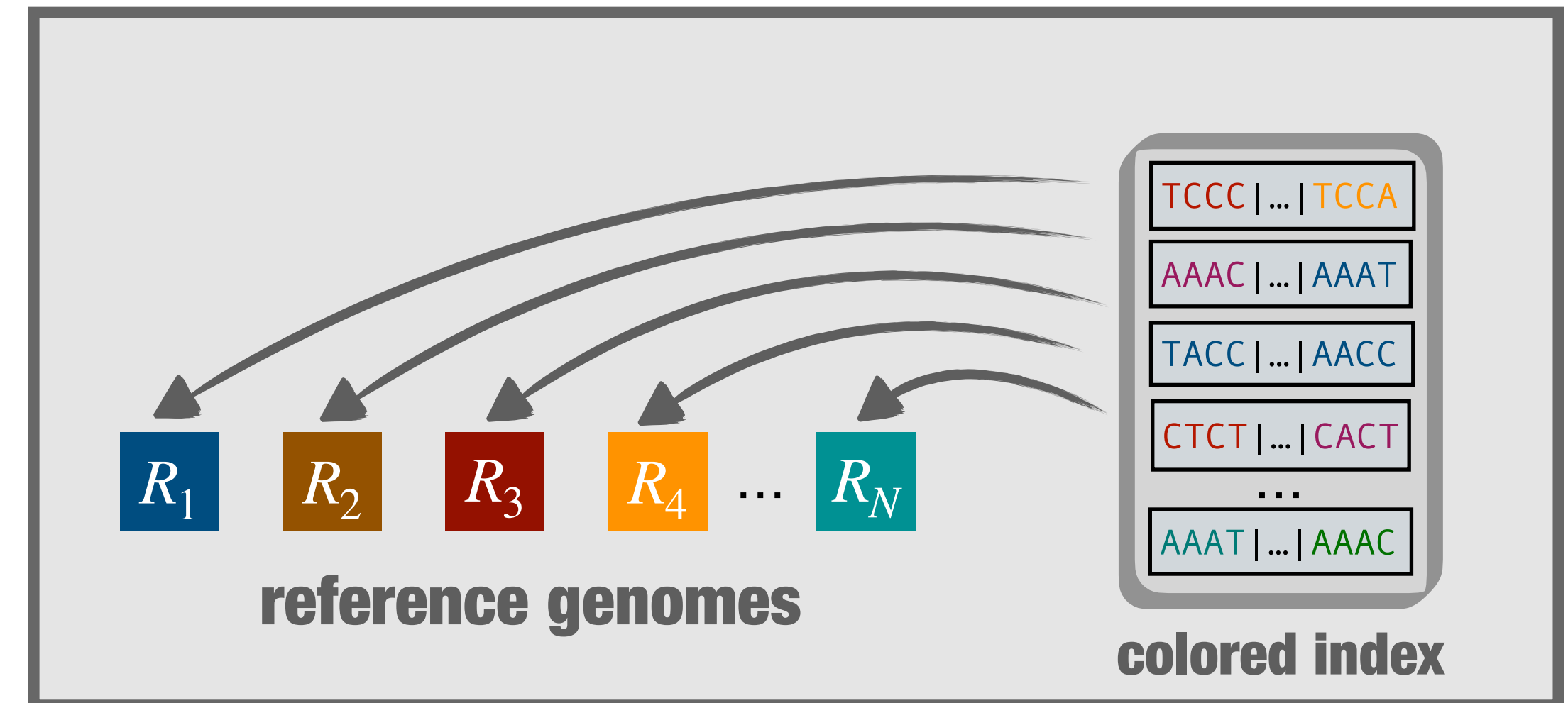
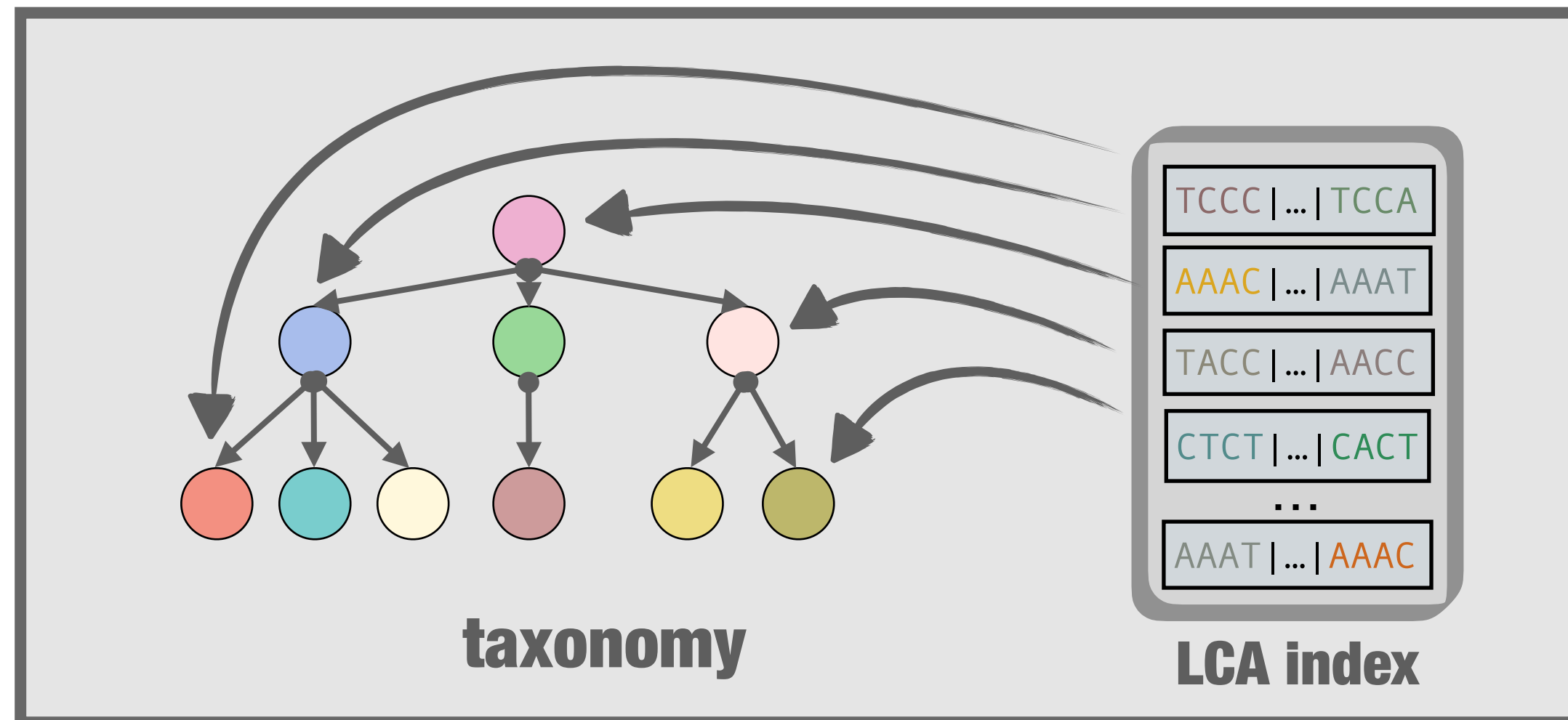
- ▶ **Correct genus:** 20 genomes
- ▶ **Erroneous phylum:** 1 genome
- Kraken would push the LCA to the **kingdom**

RefSeq database growth influences the accuracy of *k*-mer-based lowest common ancestor species identification

Daniel J. Nasko¹, Sergey Koren², Adam M. Phillippy² and Todd J. Treangen^{3*} 



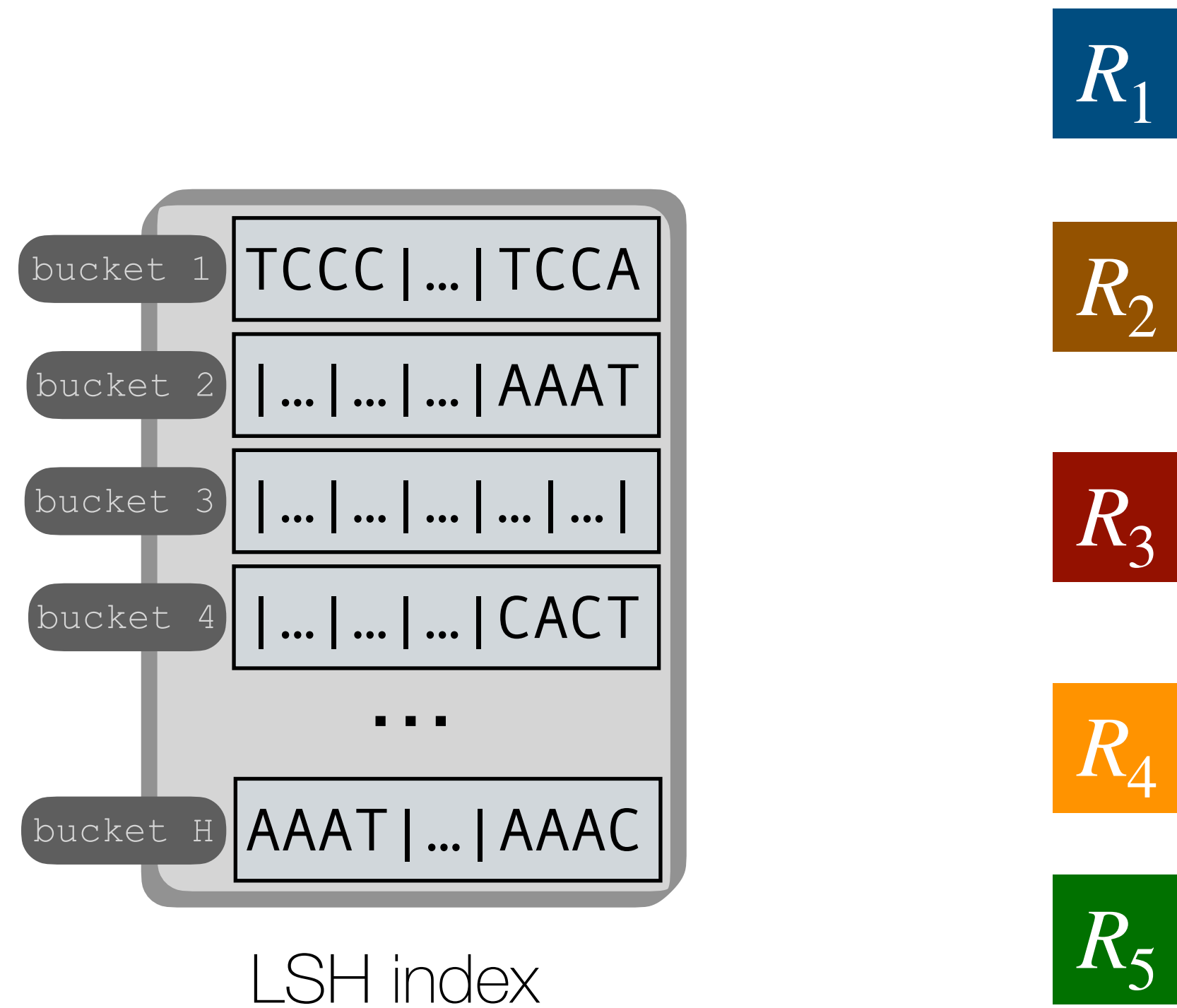
Mapping resolution: reference genomes instead of taxonomic LCAs



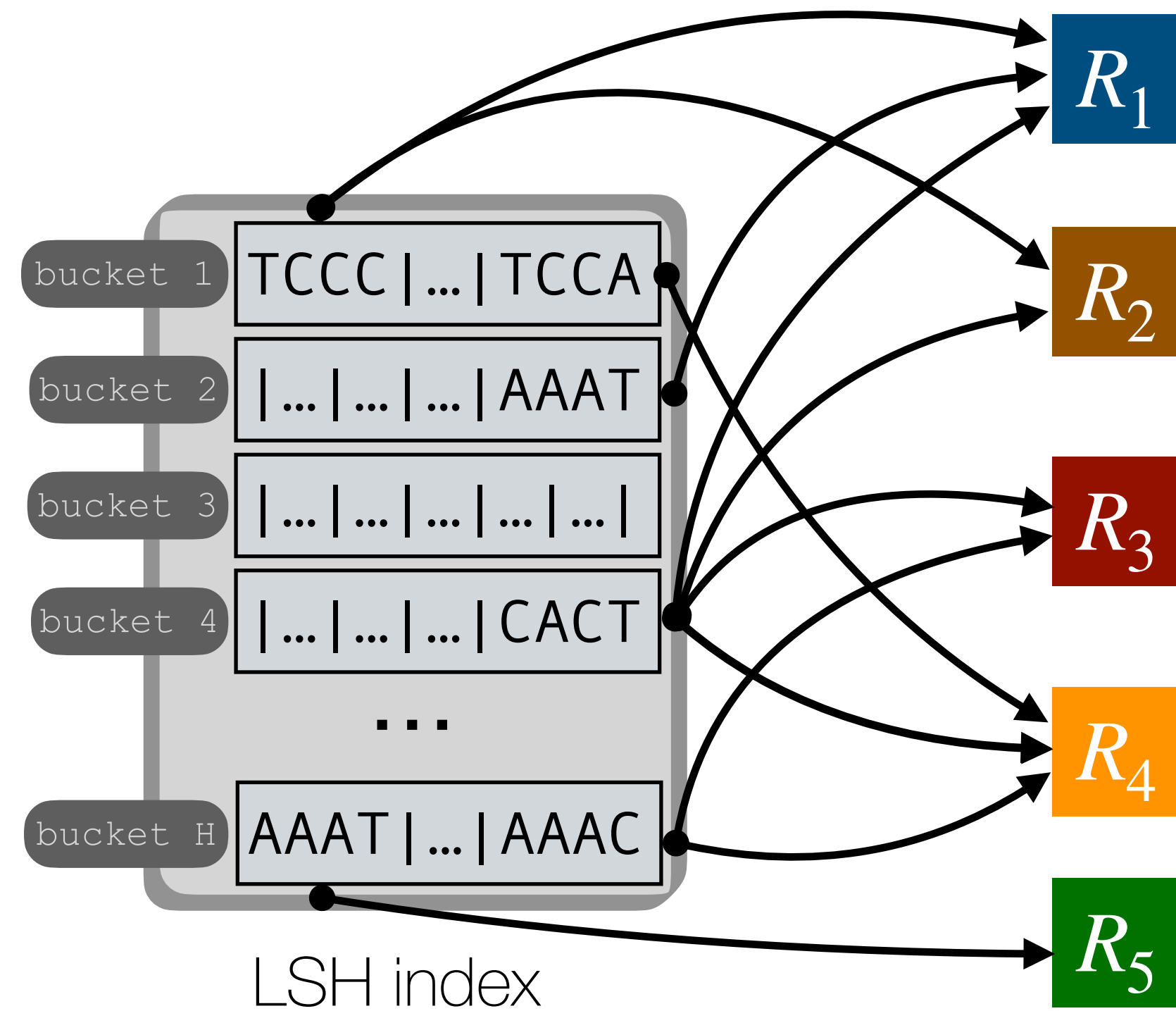
- (+) memory efficient; one label per k -mer
- (-) loss of information
- (-) sensitive to outliers, contamination
- (-) too coarse for some tasks

- (-) memory intensive & slow if done naively
- (+) full resolution & matching information
- (+) robust to outliers, reference errors
- (+) enables PP and per ref. distances

Using the phylogeny as a guide to color k-mers efficiently



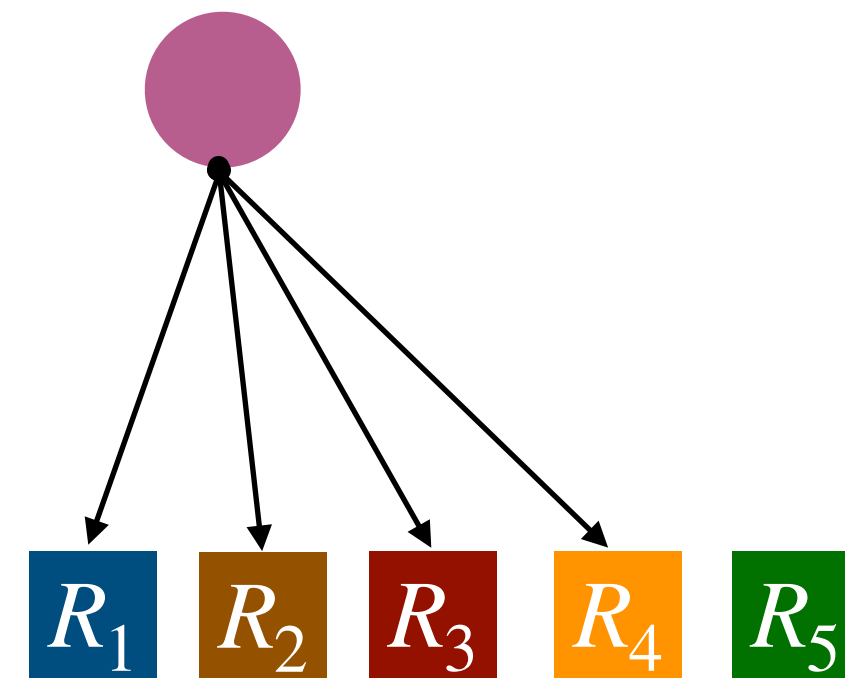
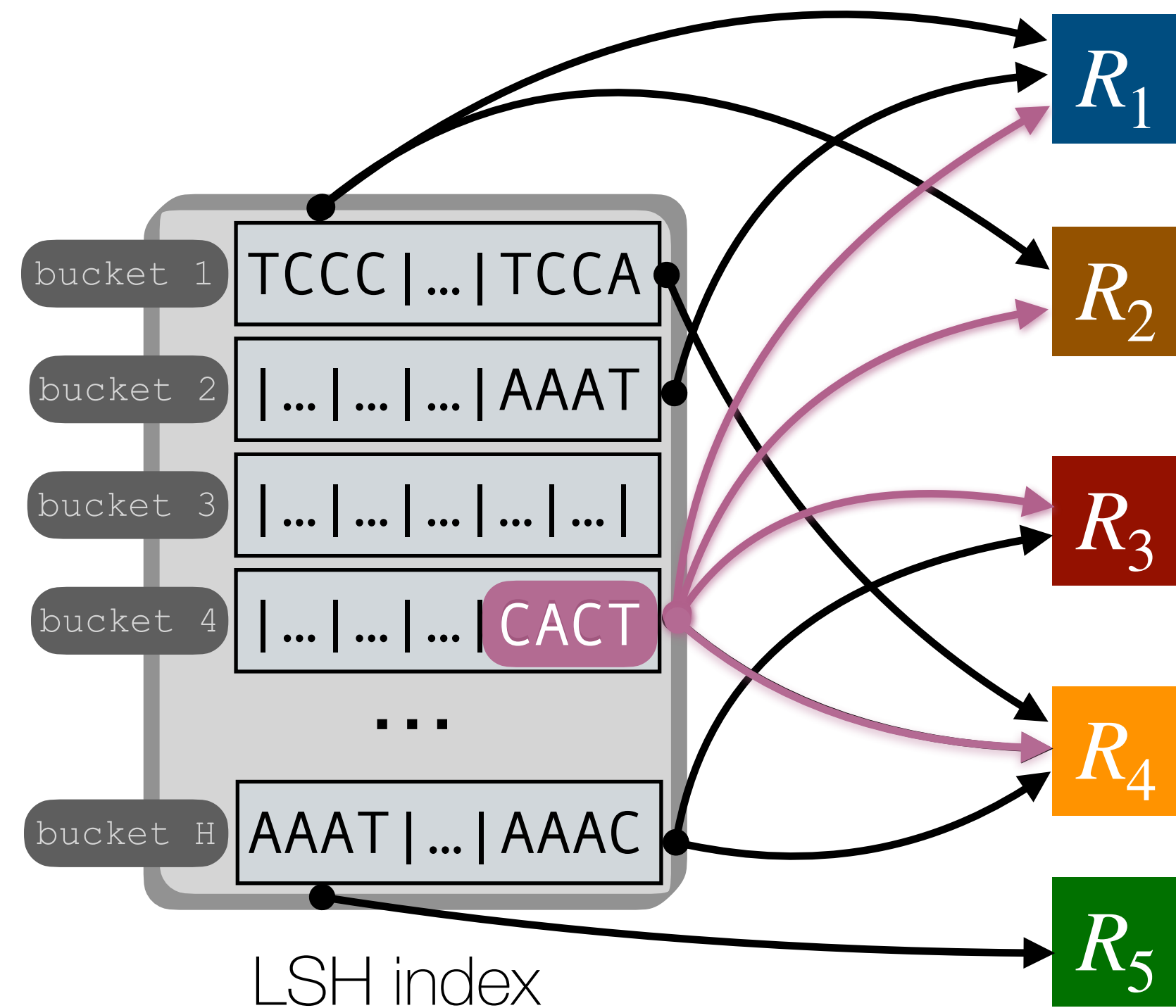
Using the phylogeny as a guide to color k-mers efficiently



well studied **colored k-mer** problem

color: a subset of references
(including singletons)

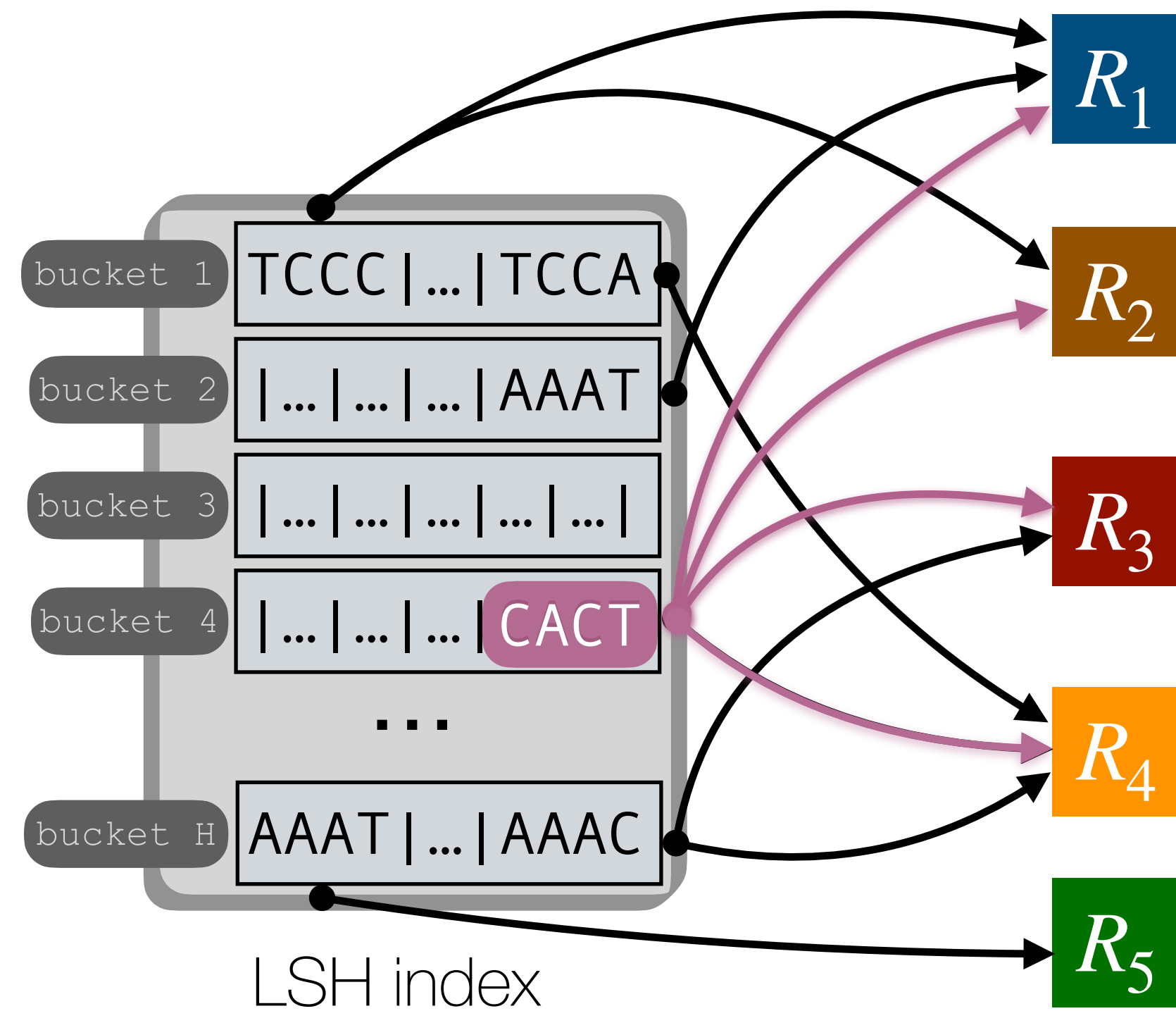
Using the phylogeny as a guide to color k-mers efficiently



well studied **colored k-mer** problem

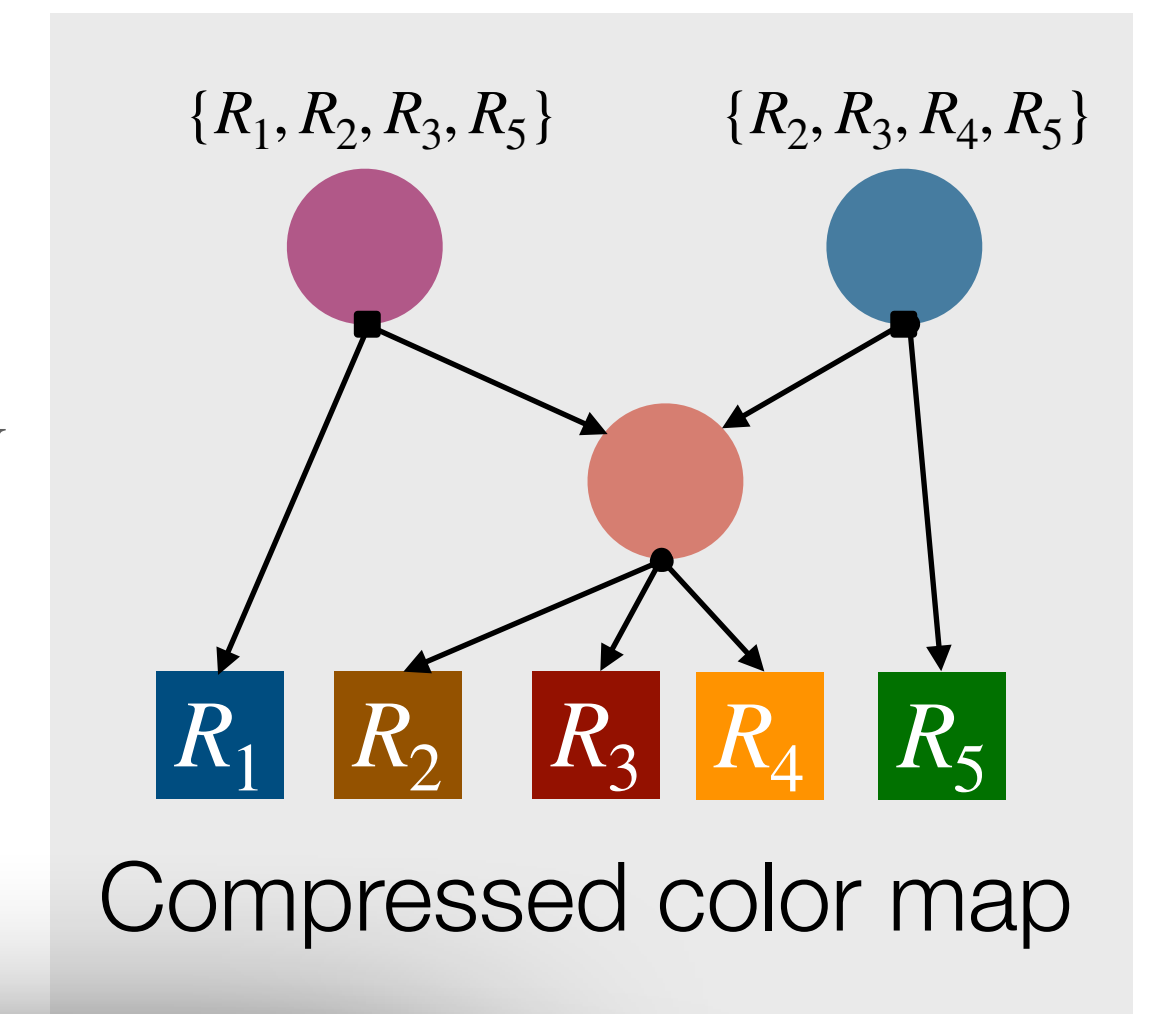
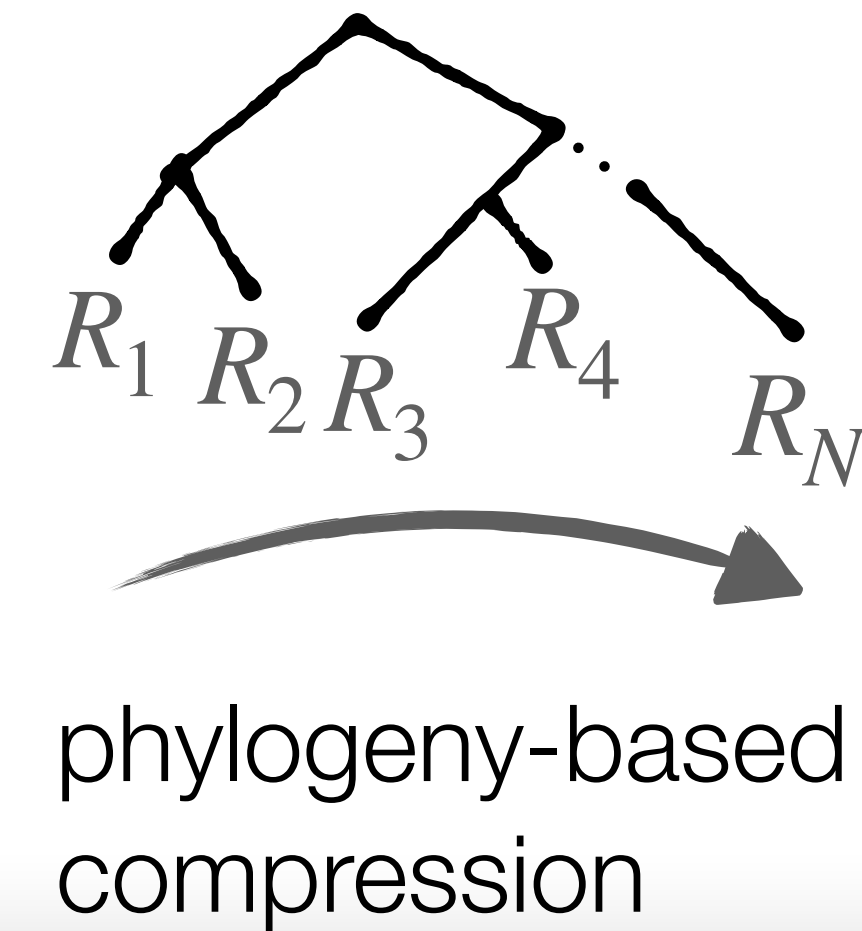
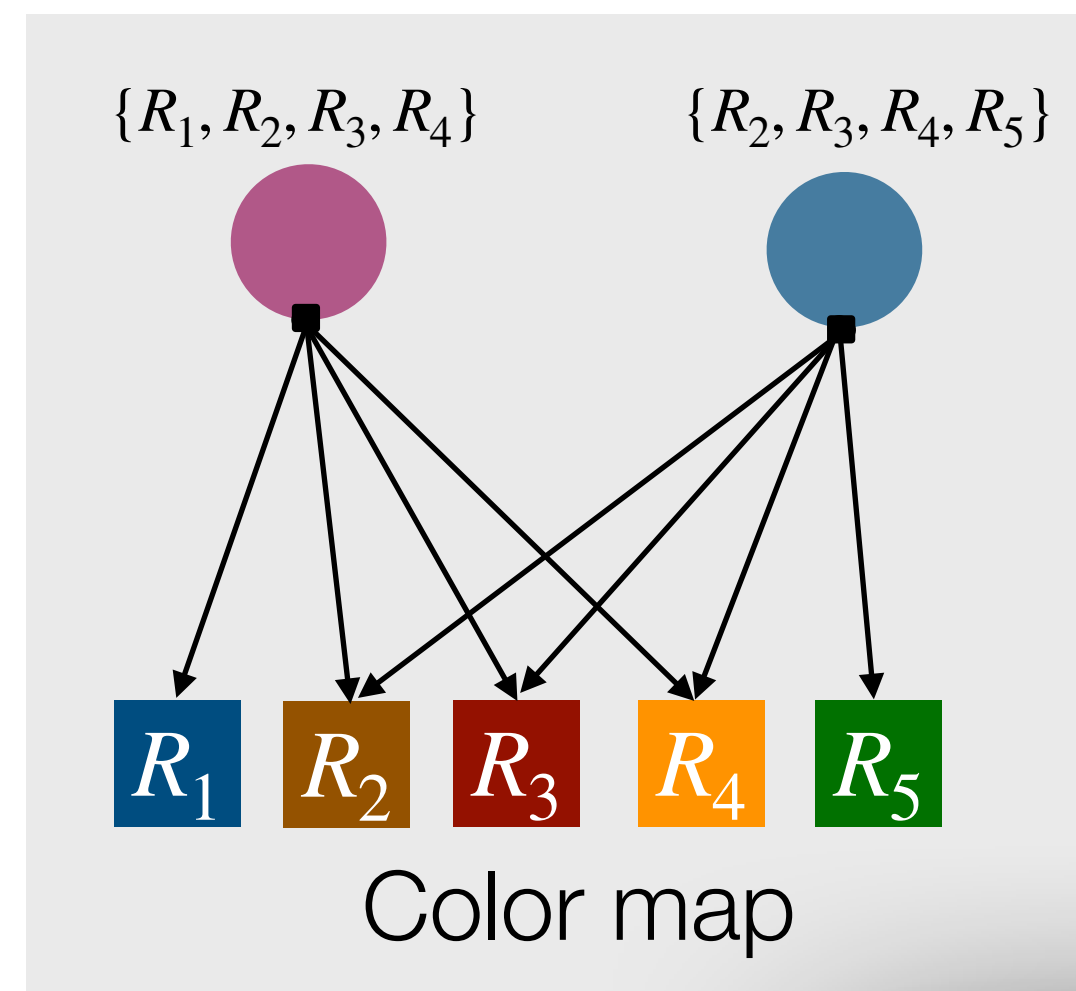
color: a subset of references
(including singletons)

Using the phylogeny as a guide to color k-mers efficiently



well studied **colored k-mer** problem

color: a subset of references
(including singletons)



Skipping...

Finding homologous k-mers in reference genomes

Given a read:

$>q$
ATACCTAGGAGTACGGGAC

1: ATAC
2: TACC
3: ACCT
4: CCTA
5: CTAG

...

L-k+1: GGAC

Finding homologous k-mers in reference genomes

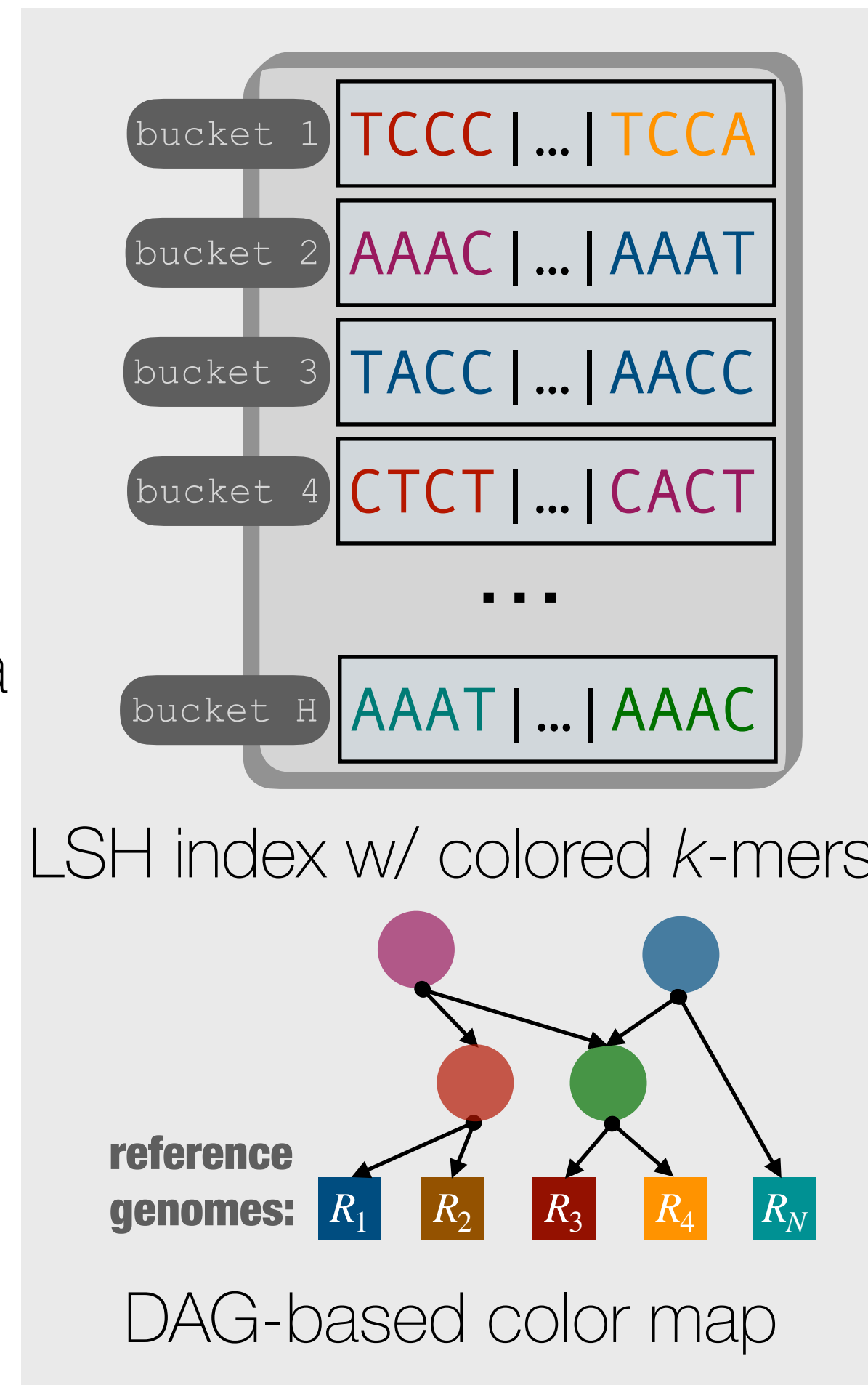
Given a read:

$>q$
ATACCTAGGAGTACGGGAC

1: ATAC
2: TACC
3: ACCT
4: CCTA
5: CTAG
...

L-k+1: GGAC

search k -mer
matches up to a
HD threshold δ



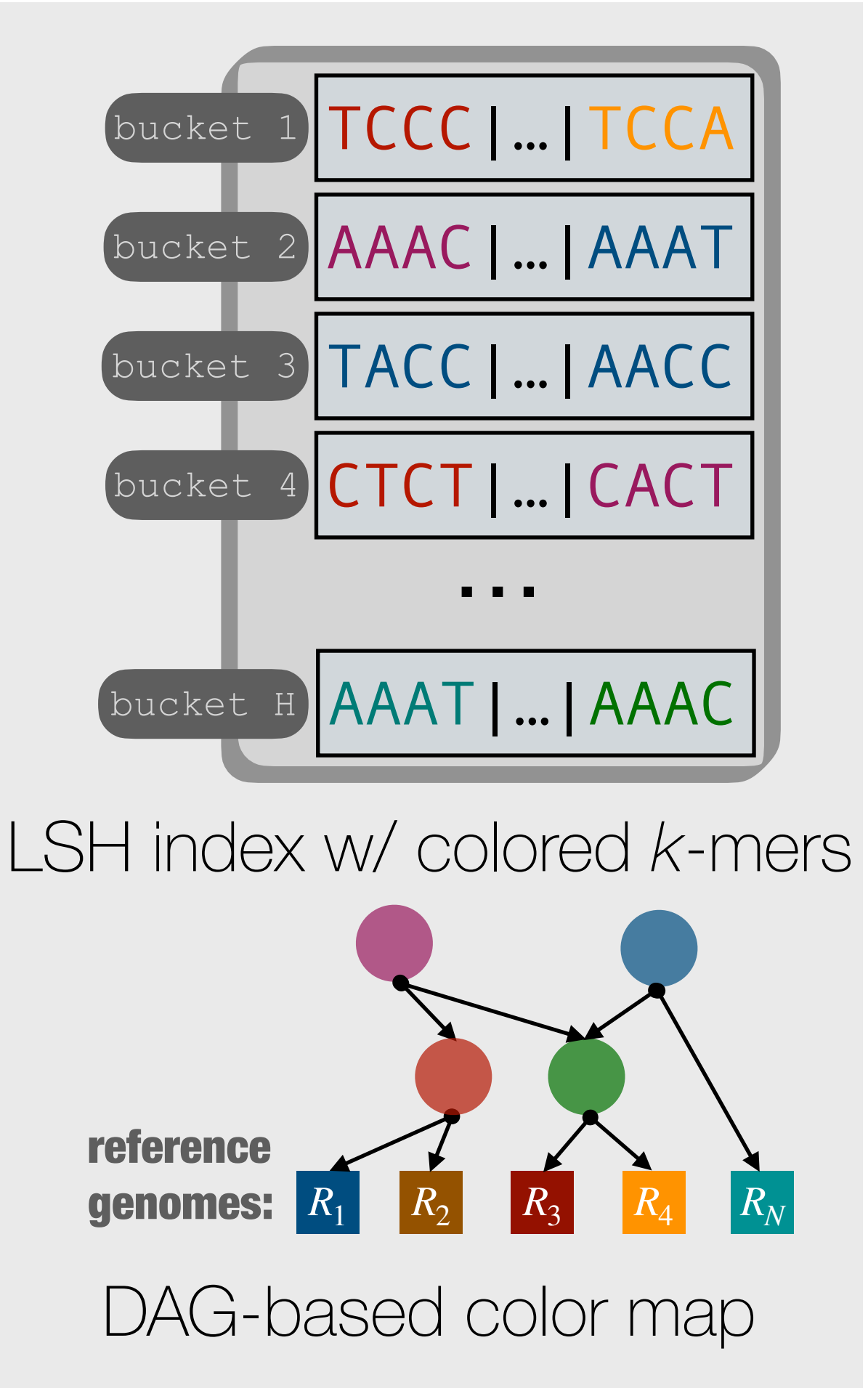
Finding homologous k-mers in reference genomes

Given a read:

$>q$
ATACCTAGGAGTACGGGAC

1: ATAC
2: TACC
3: ACCT
4: CCTA
5: CTAG
...
L-k+1: GGAC

search k -mer
matches up to a
HD threshold δ



keep the
closest match
as homologous

sparse table

	R_1	R_2	...	R_N
<u>1</u>	0	1	...	-
<u>2</u>	1	4	...	4
<u>3</u>	-	-	...	-
<u>4</u>	-	2	...	-
<u>5</u>	0	-	...	3
...	...	...	...	...
<u>L-k+1</u>	3	0	...	4

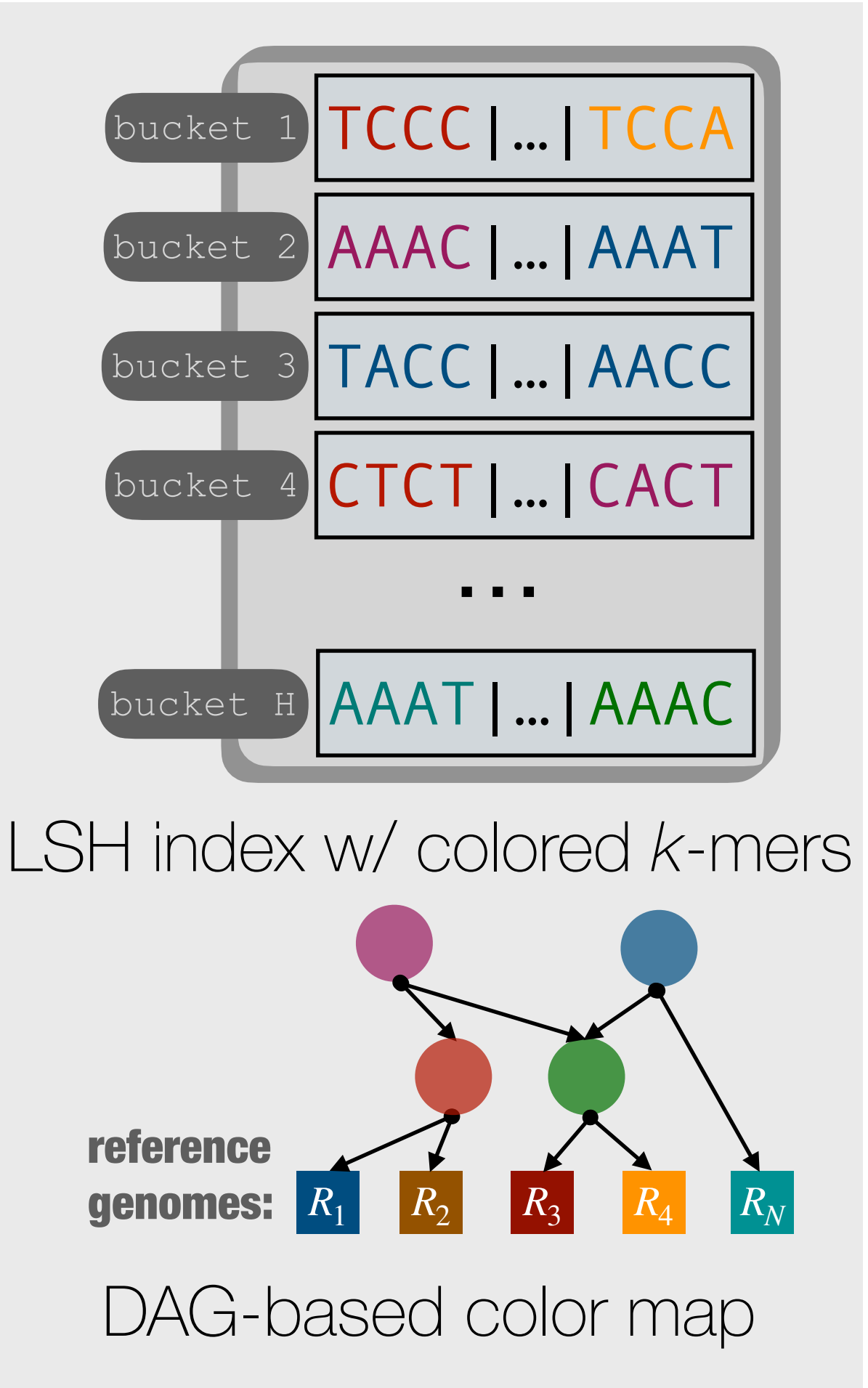
Finding homologous k-mers in reference genomes

Given a read:

$>q$
ATACCTAGGAGTACGGGAC

1: ATAC
2: TACC
3: ACCT
4: CCTA
5: CTAG
...
L-k+1: GGAC

search k -mer
matches up to a
HD threshold δ



keep the
closest match
as homologous

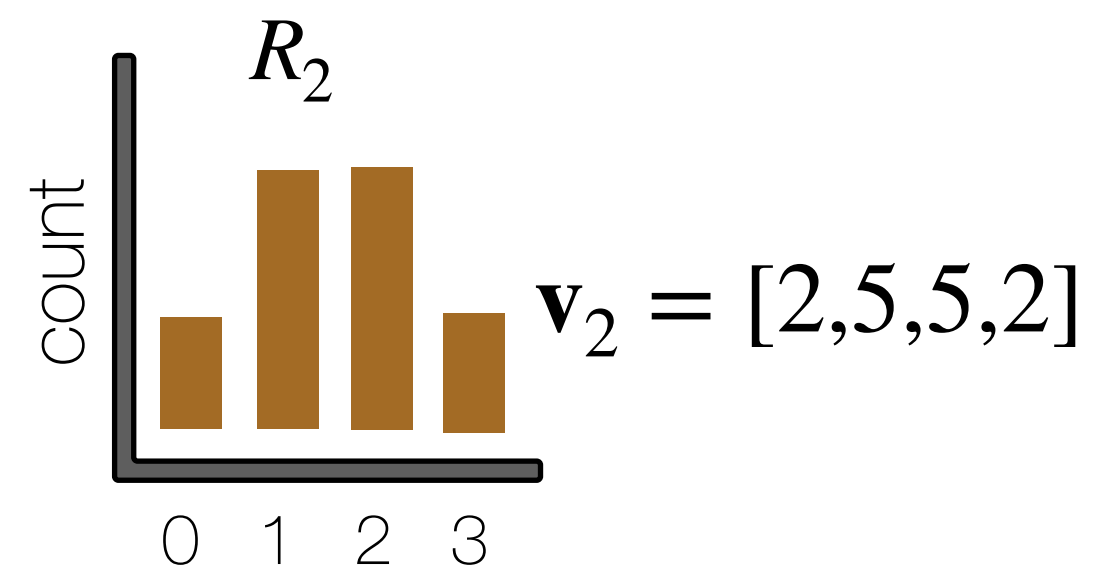
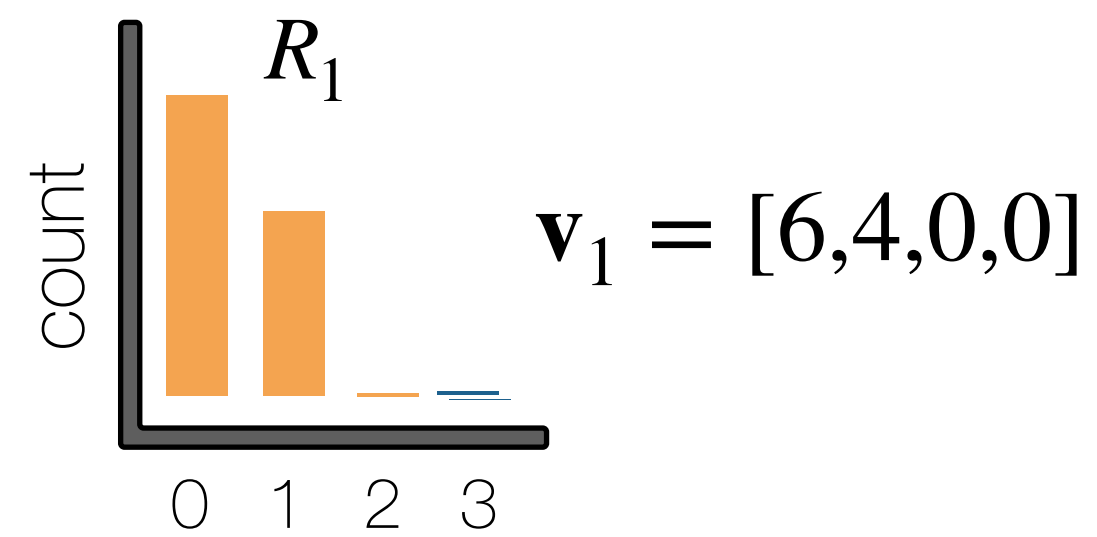
sparse table

	R_1	R_2	...	R_N
<u>1</u>	0	1	...	-
<u>2</u>	1	4	...	4
<u>3</u>	-	-	...	-
<u>4</u>	-	2	...	-
<u>5</u>	0	-	...	3
...	...	...	...	...
<u>L-k+1</u>	3	0	...	4

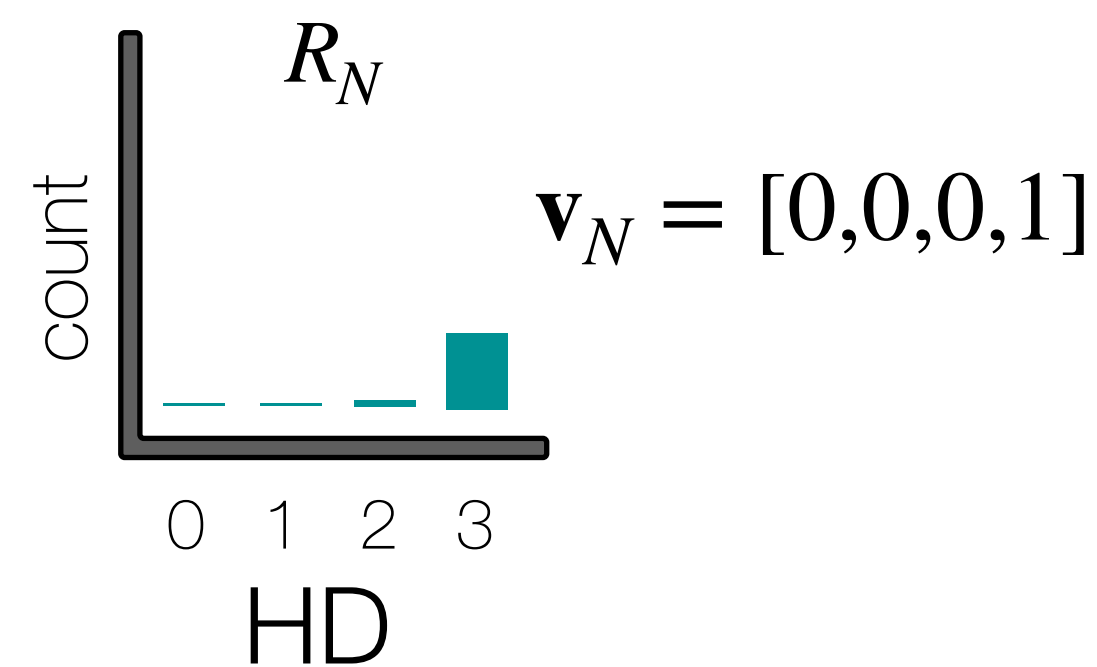
summarize each
column as a
histogram

Optimizing the likelihood of distances using “homologous” k-mers

Hamming distance histograms

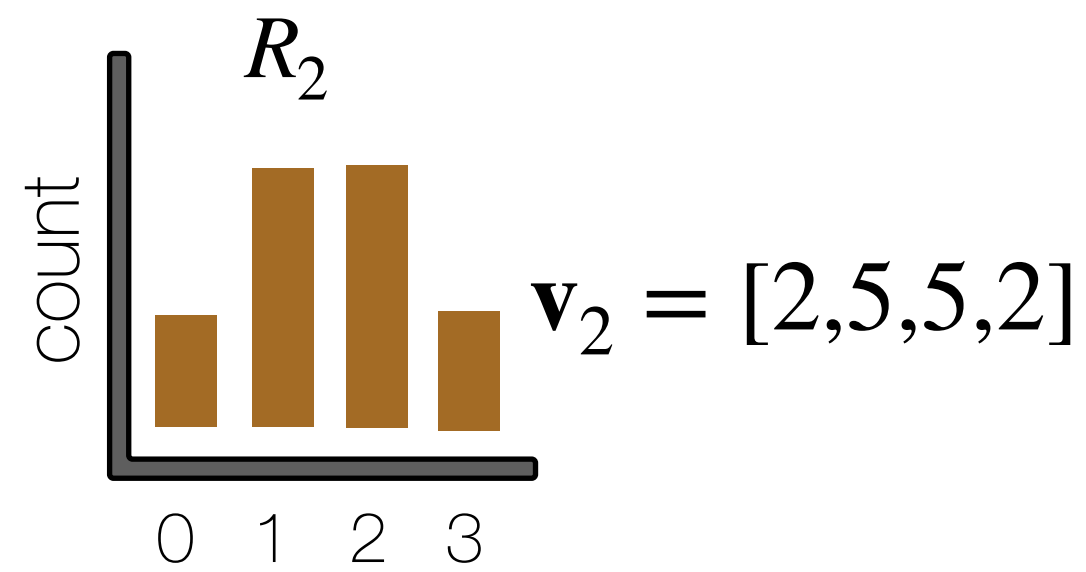
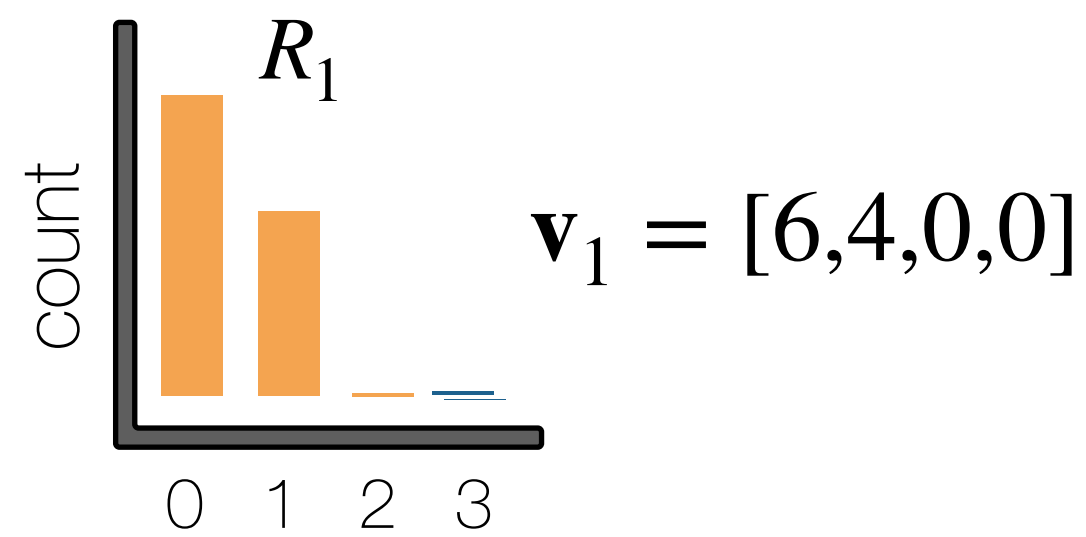


...

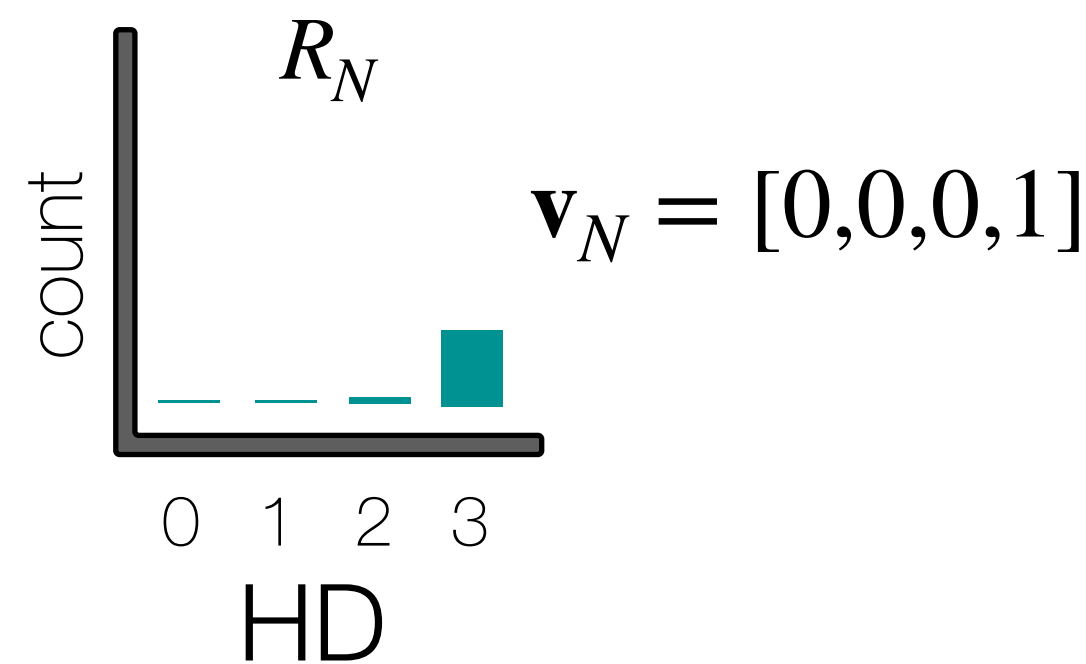


Optimizing the likelihood of distances using “homologous” k-mers

Hamming distance histograms



...

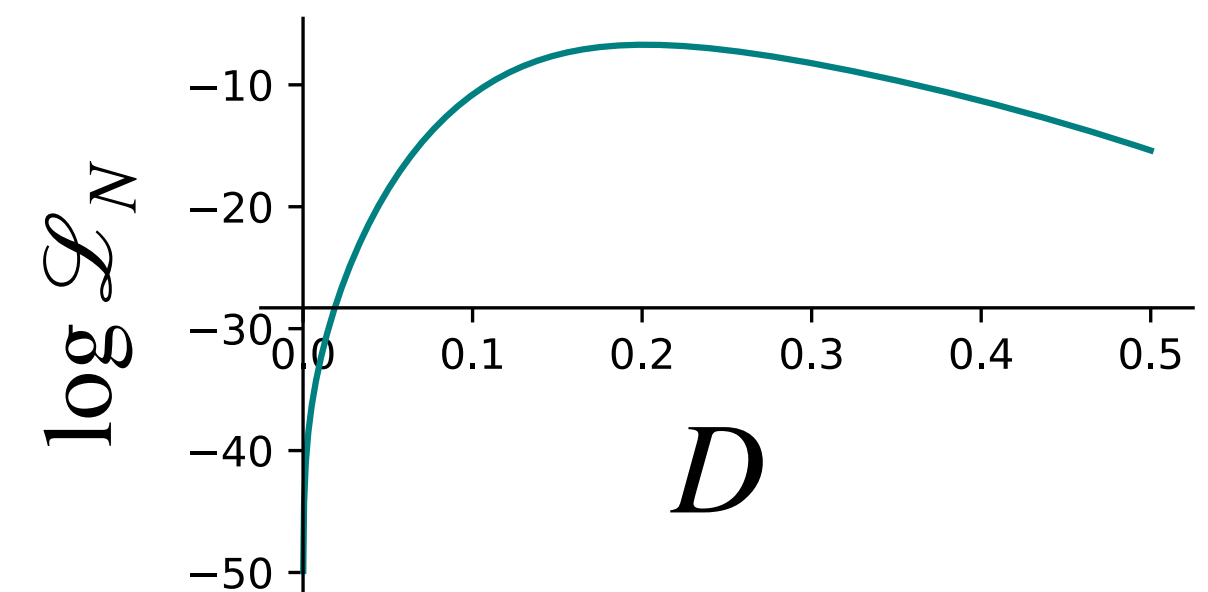
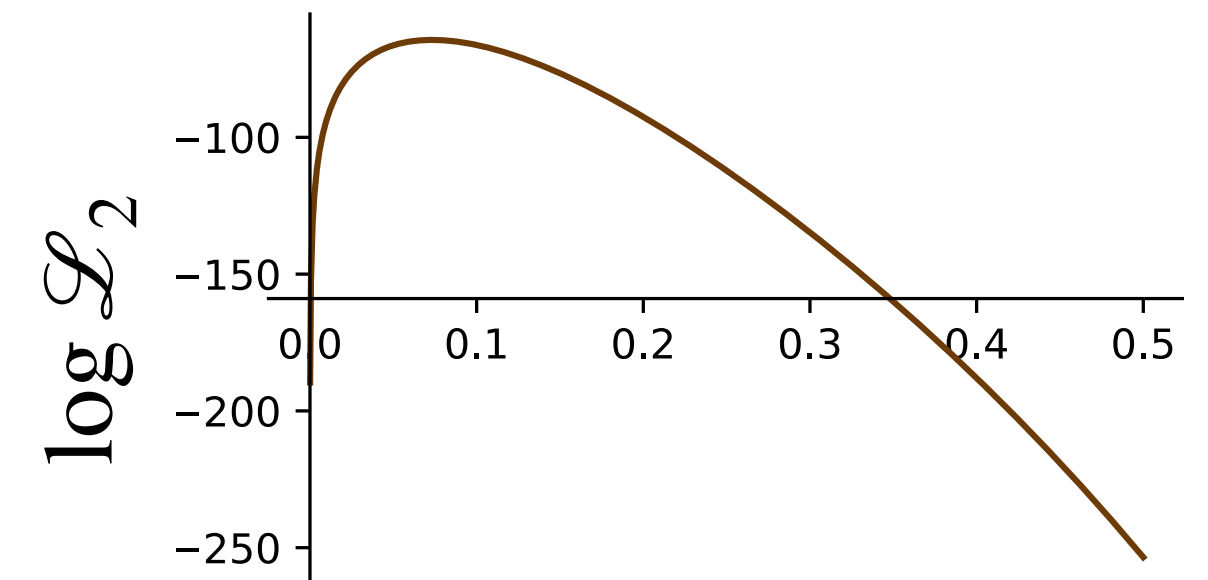
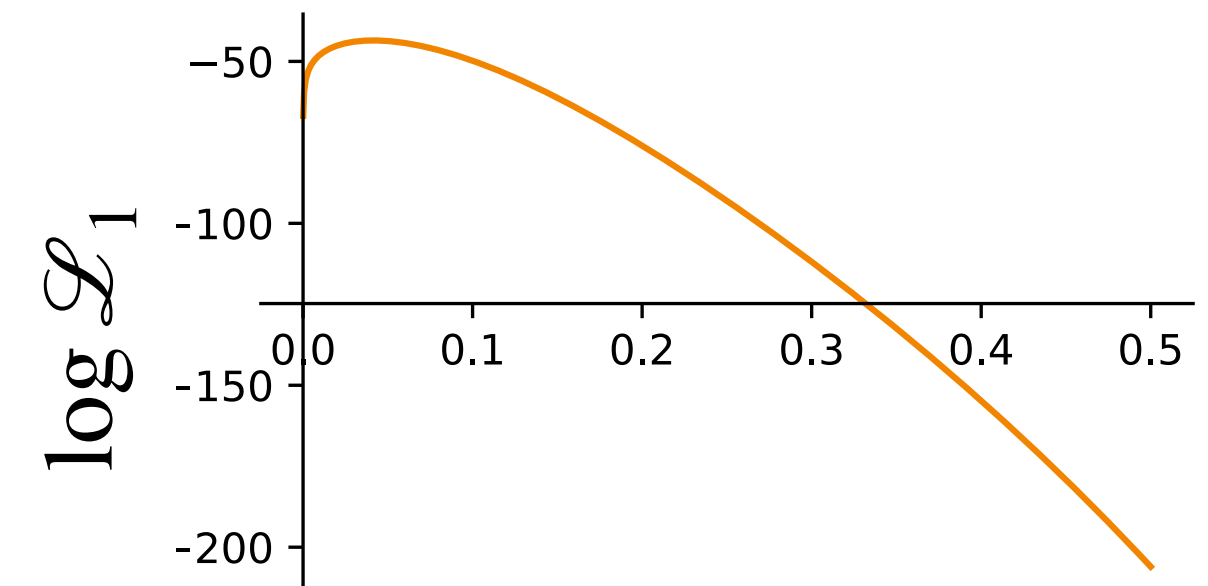


Likelihood of q having distance D to R :

$$P_{miss}(D; k, h, \delta)^u \prod_{d=0}^{\delta} P_{match}(D; d, k, h)^{v_d}$$

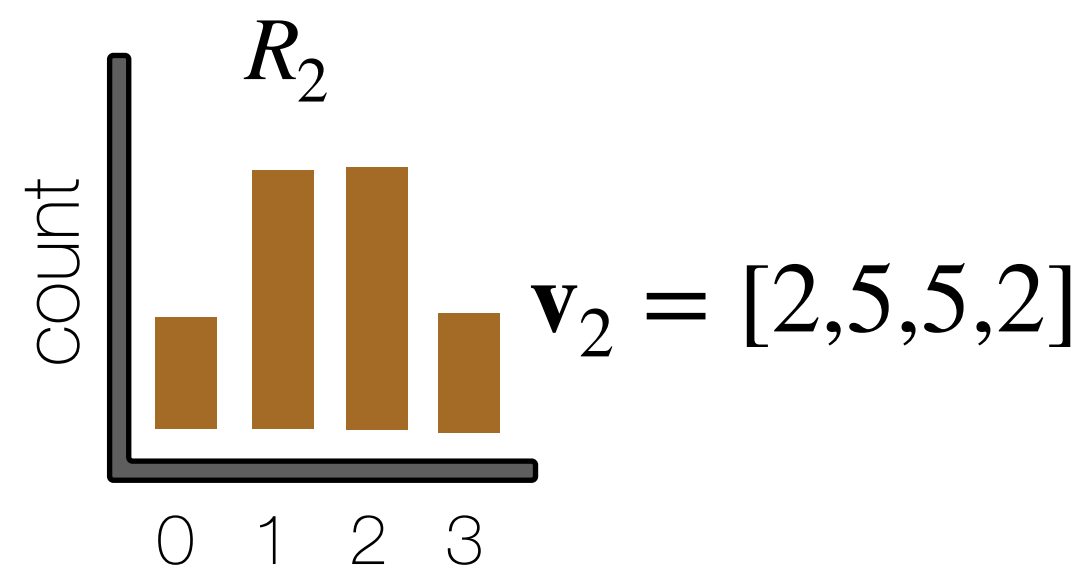
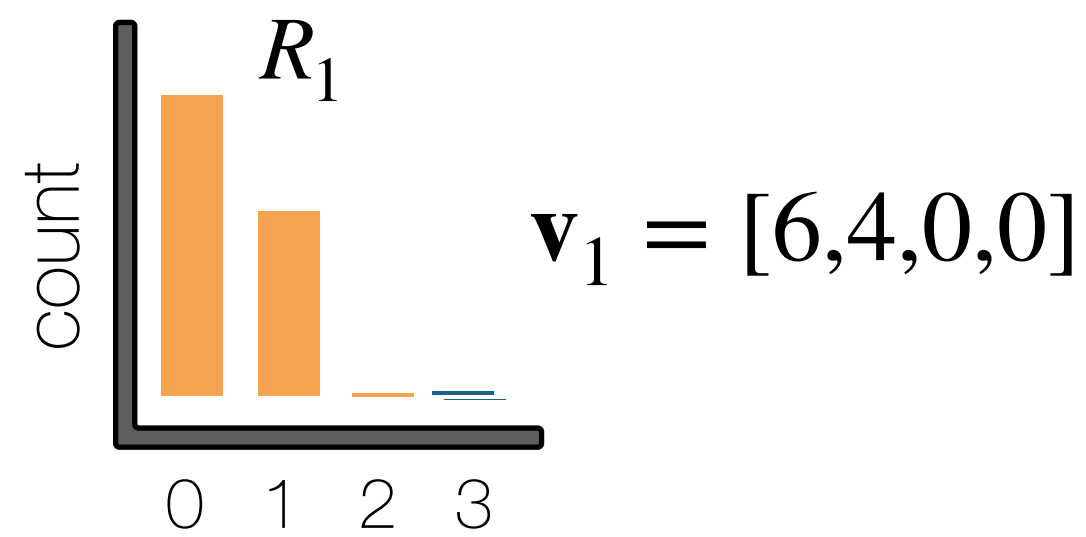
Probability of having u misses in total

Probability of having v_d matches at HD = d

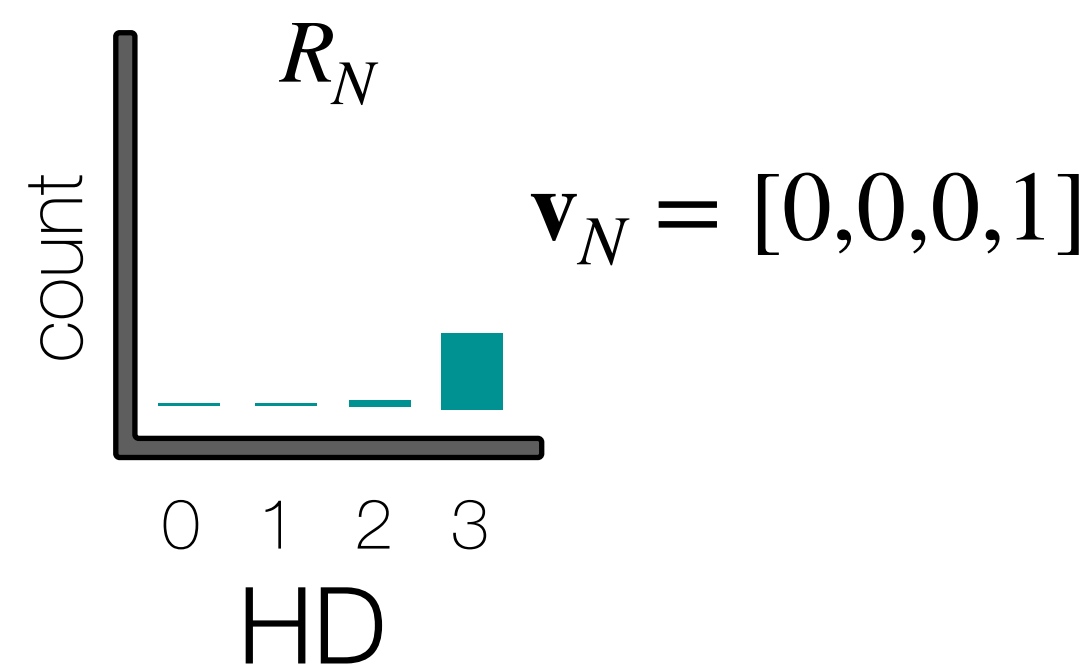


Optimizing the likelihood of distances using “homologous” k-mers

Hamming distance histograms



...

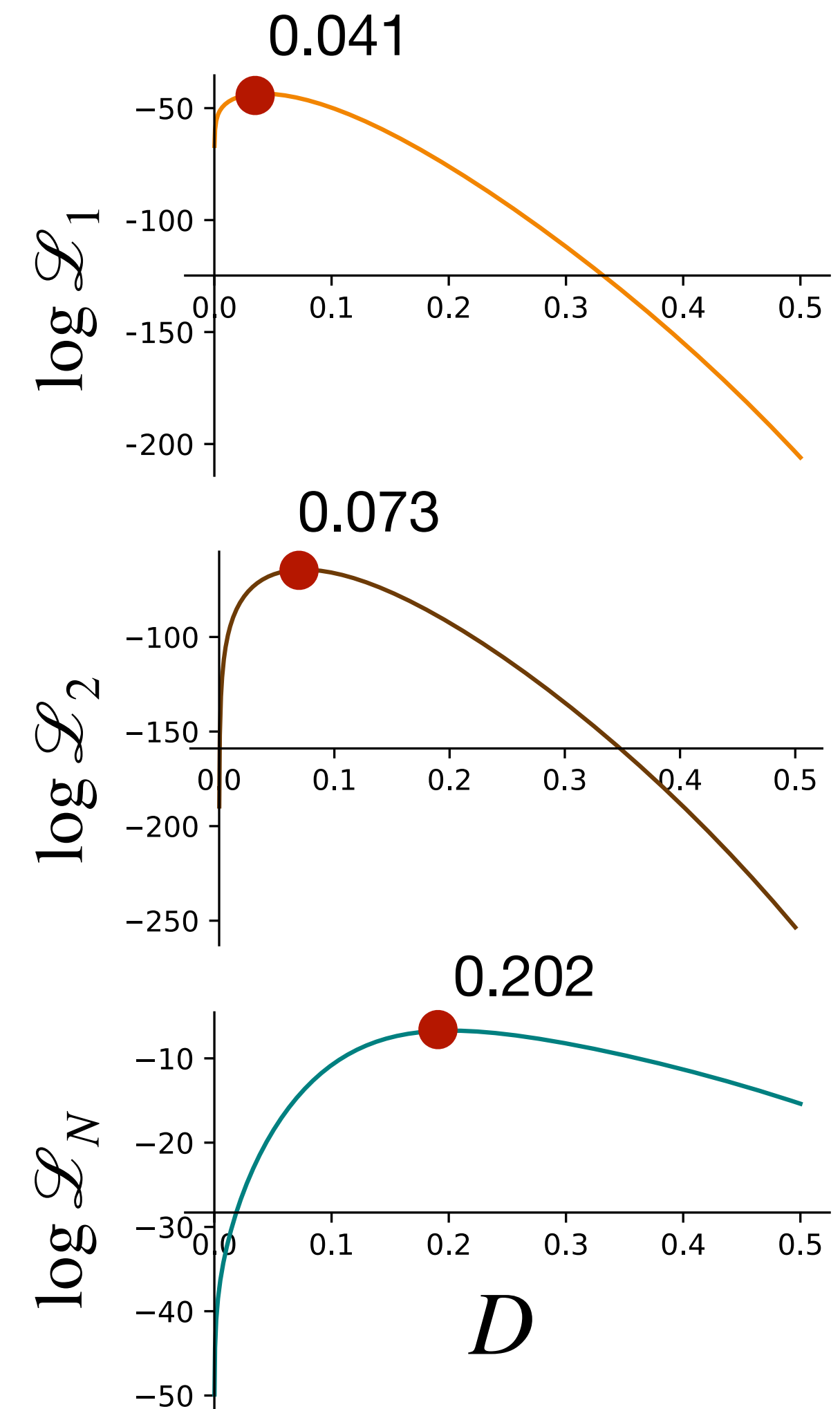


Likelihood of q having distance D to R :

$$\arg \max_D P_{\text{miss}}(D; k, h, \delta)^u \prod_{d=0}^{\delta} P_{\text{match}}(D; d, k, h)^{v_d}$$

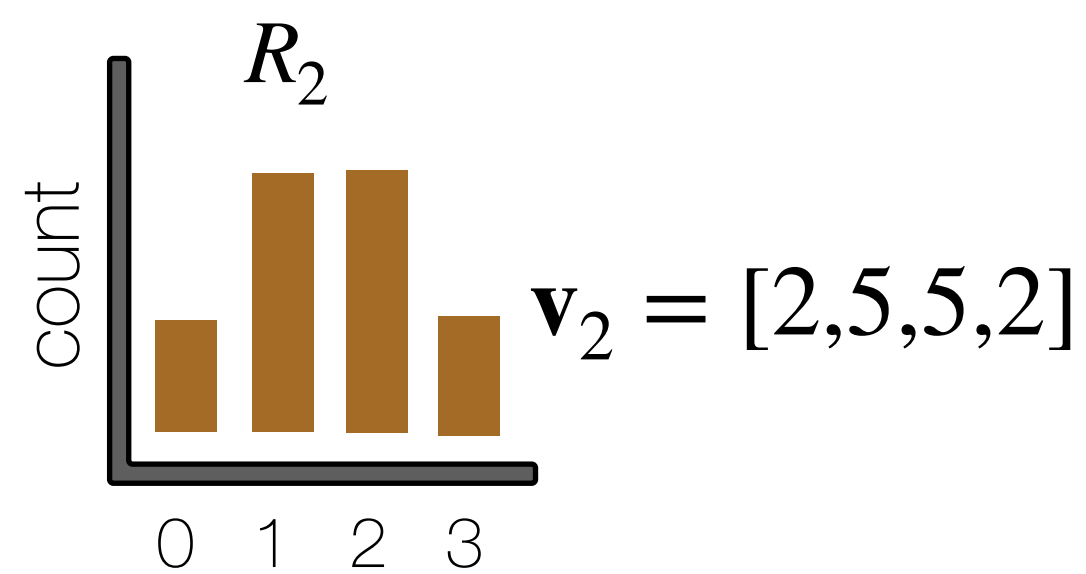
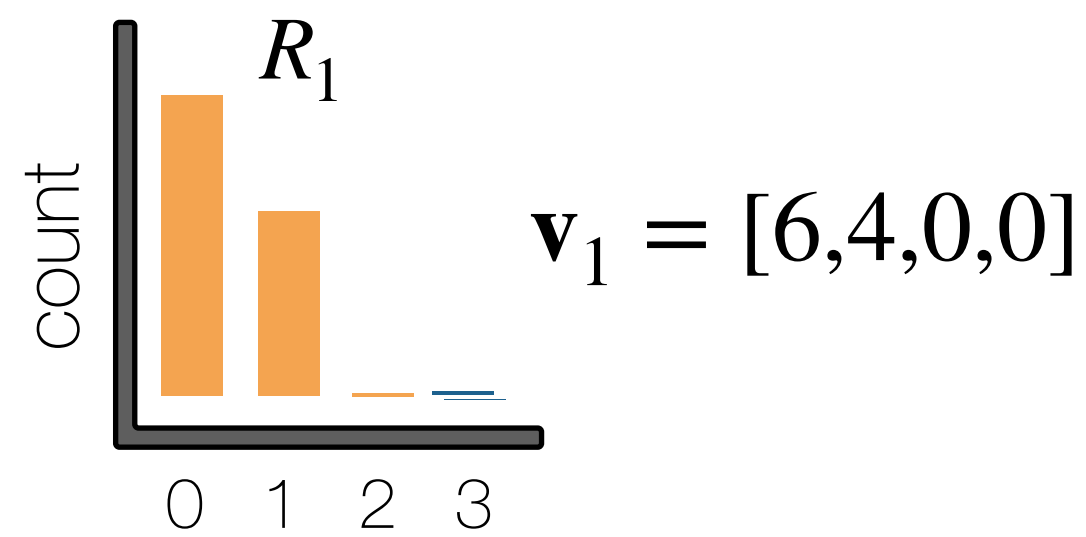
Probability of having u misses in total

Probability of having v_d matches at HD = d

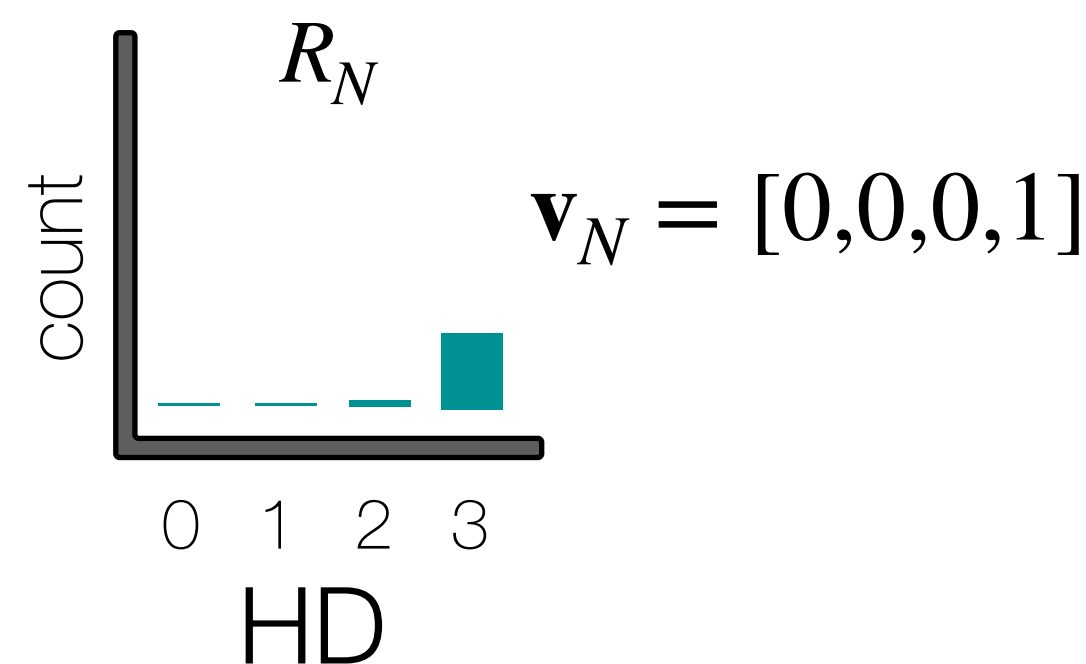


Optimizing the likelihood of distances using “homologous” k-mers

Hamming distance histograms



...



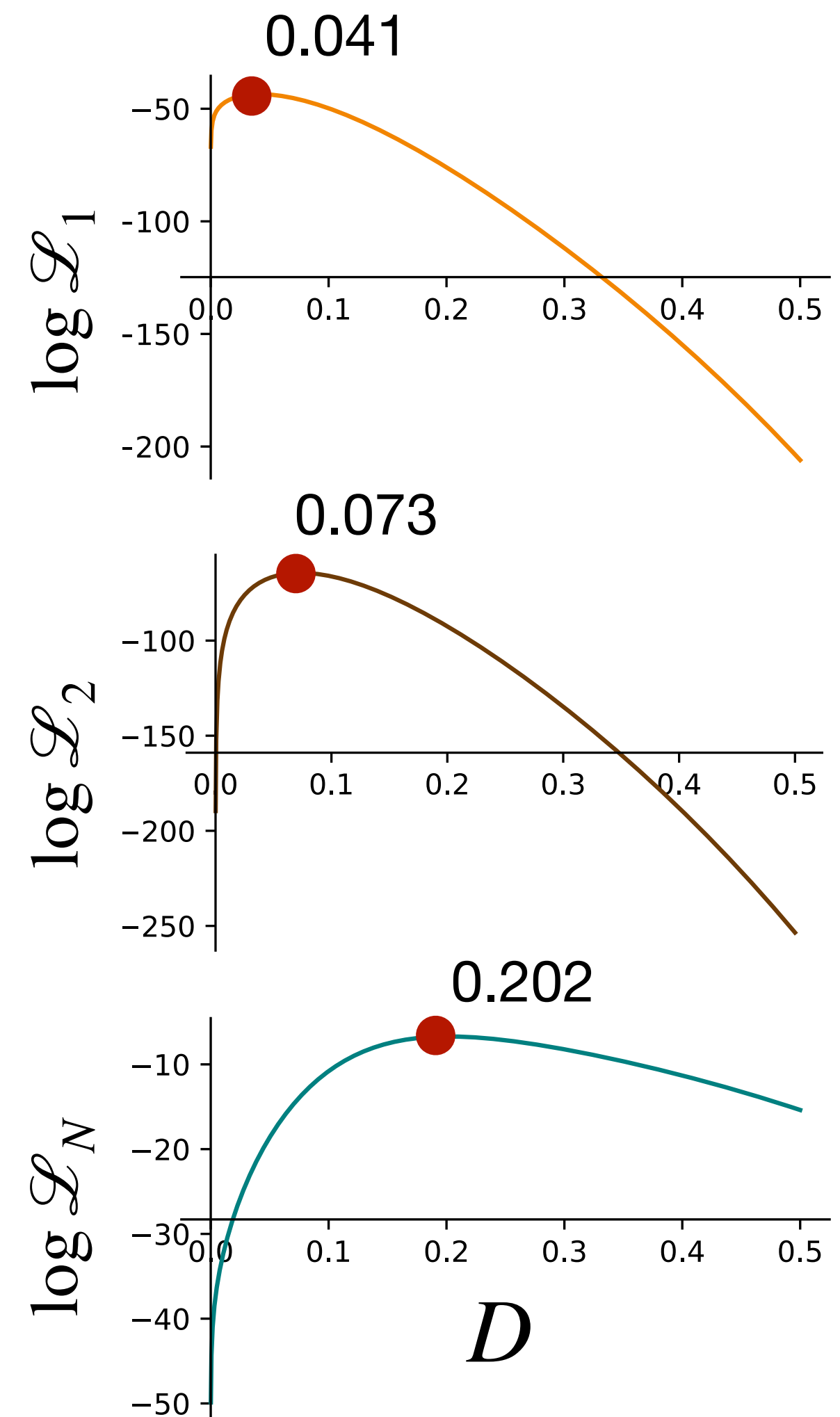
Likelihood of q having distance D to R :

$$\arg \max_D P_{miss}(D; k, h, \delta)^u \prod_{d=0}^{\delta} P_{match}(D; d, k, h)^{v_d}$$

Probability of having u misses in total

Probability of having v_d matches at HD = d

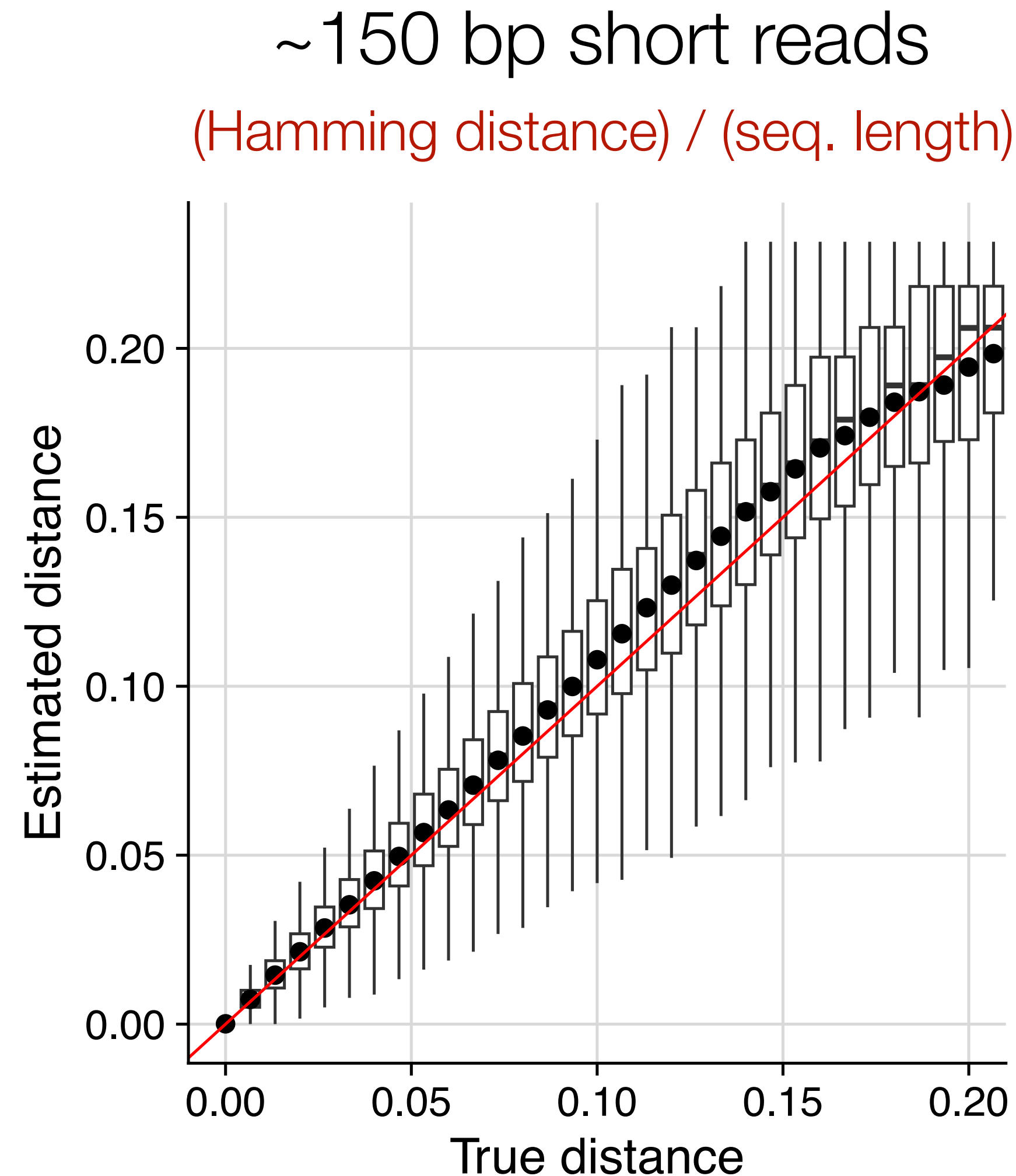
Highly accurate in benchmarkings:
even for short sequences & high distances!



krepp estimates distances accurately at the read-level

default: 29-mer
minimizers of 35-mers

- Simulation experiments
(true read distances)
- **Highly accurate**
(despite some noise)



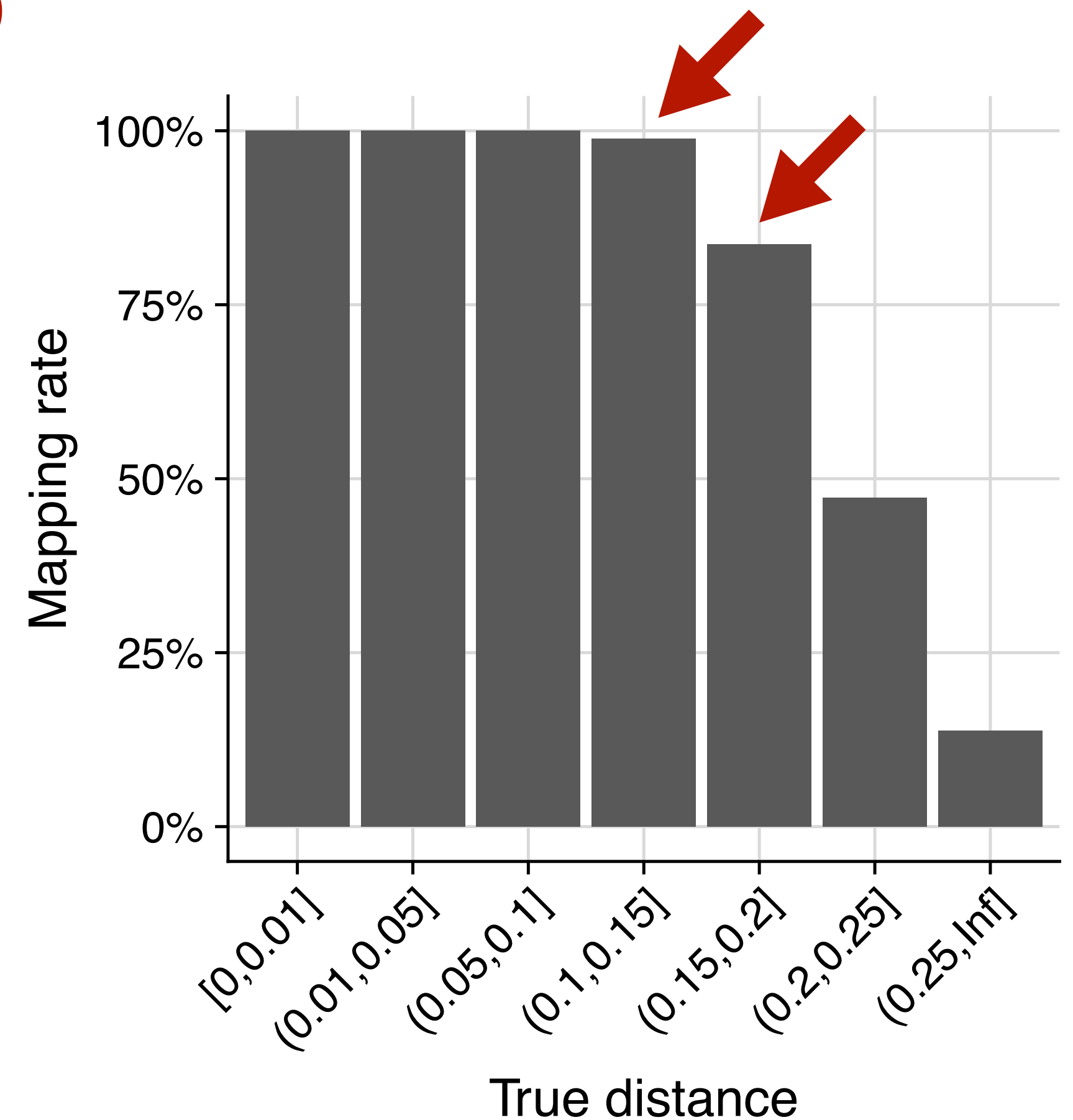
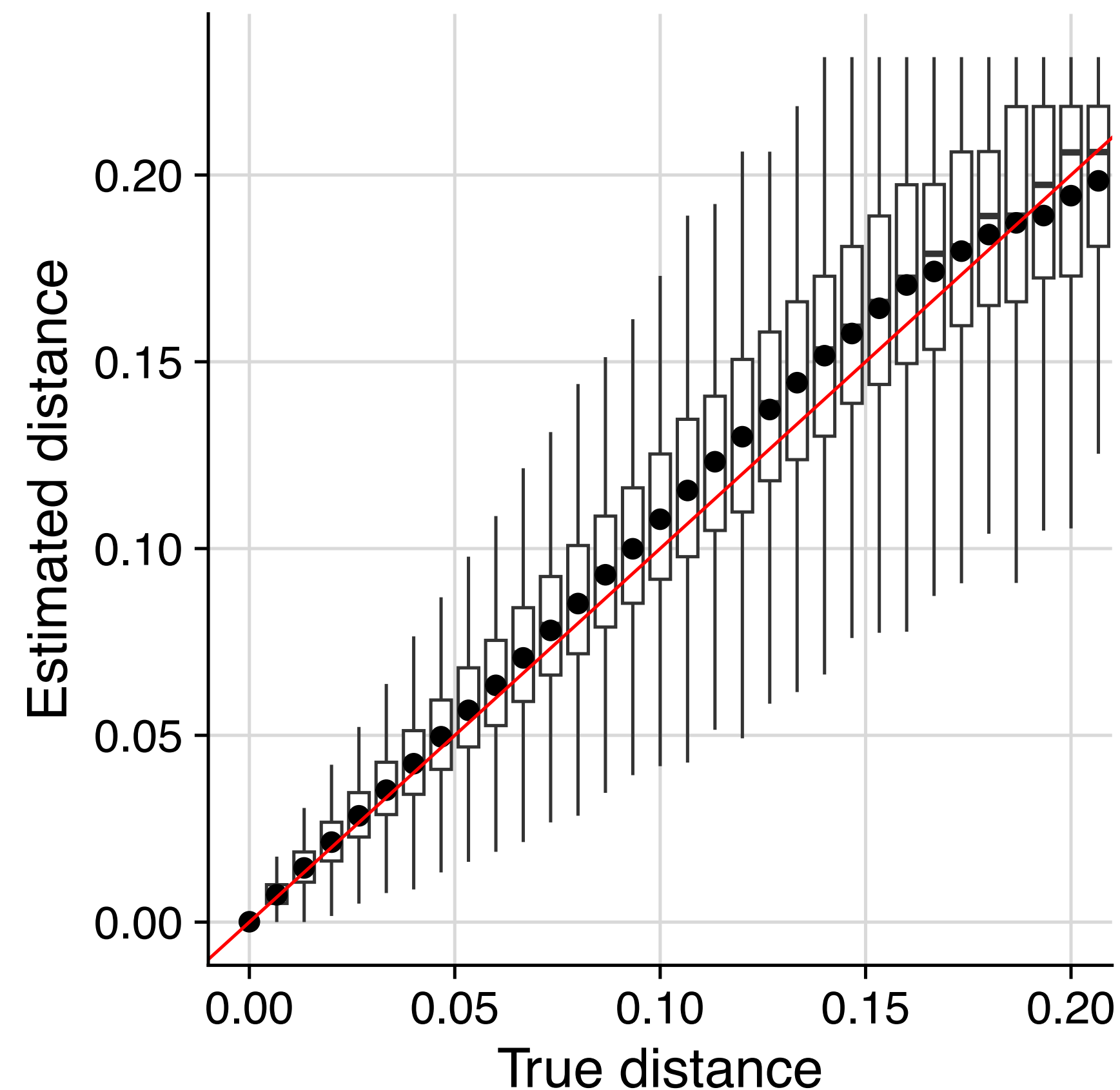
krepp estimates distances accurately at the read-level

default: 29-mer
minimizers of 35-mers

- Simulation experiments (true read distances)
- **Highly accurate** (despite some noise)
- **High mapping rate** even for novel reads >15%

~150 bp short reads

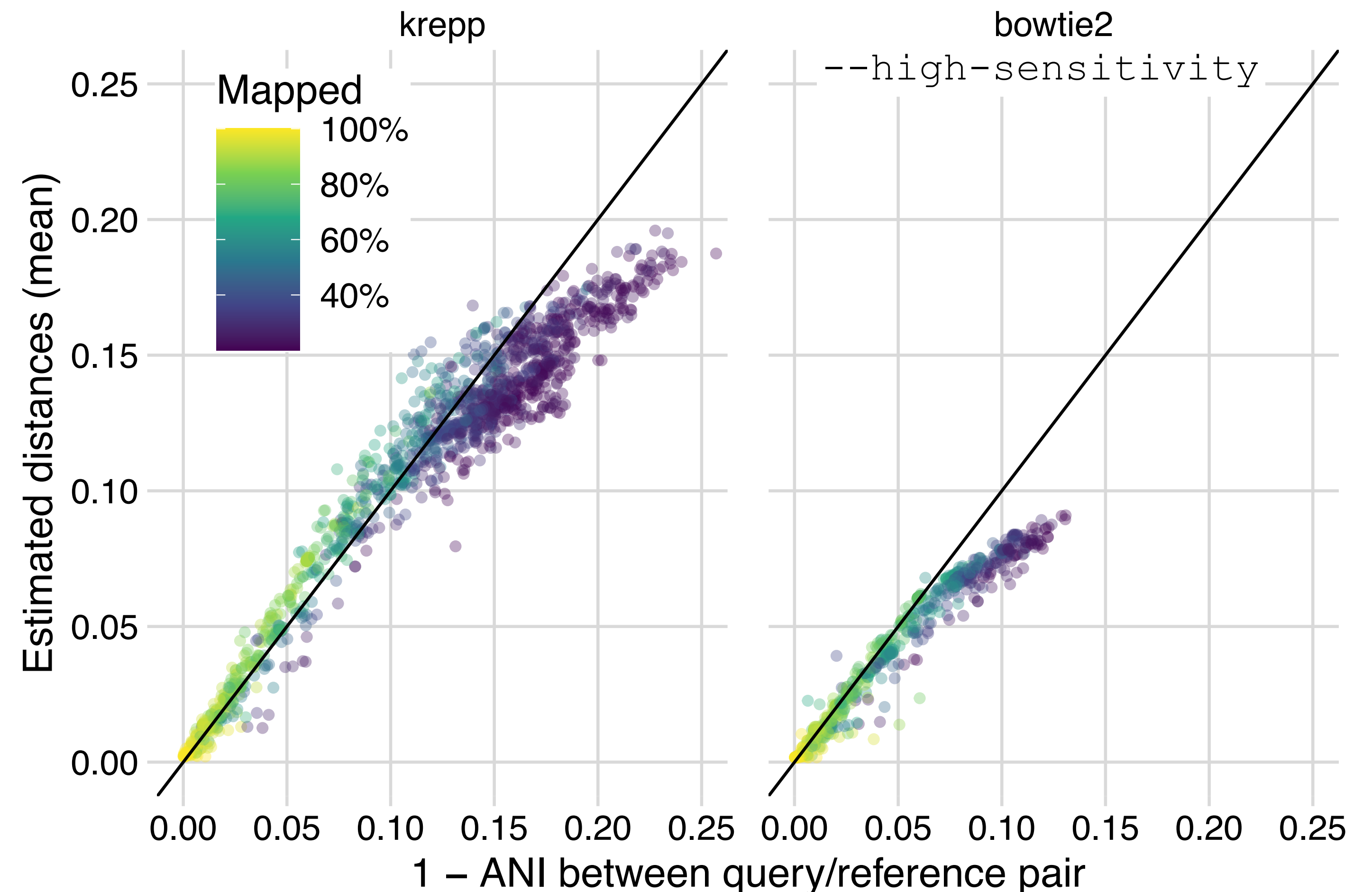
(Hamming distance) / (seq. length)



krepp matches nucleotide identity on average for real genomes

~150 bp Illumina short reads

- Real query/reference genomes (pairs with >20% mapping rate)
- krepp extends to distant (>10%) reference genomes accurately
- Increasing the sensitivity by relaxing the alignment is costly

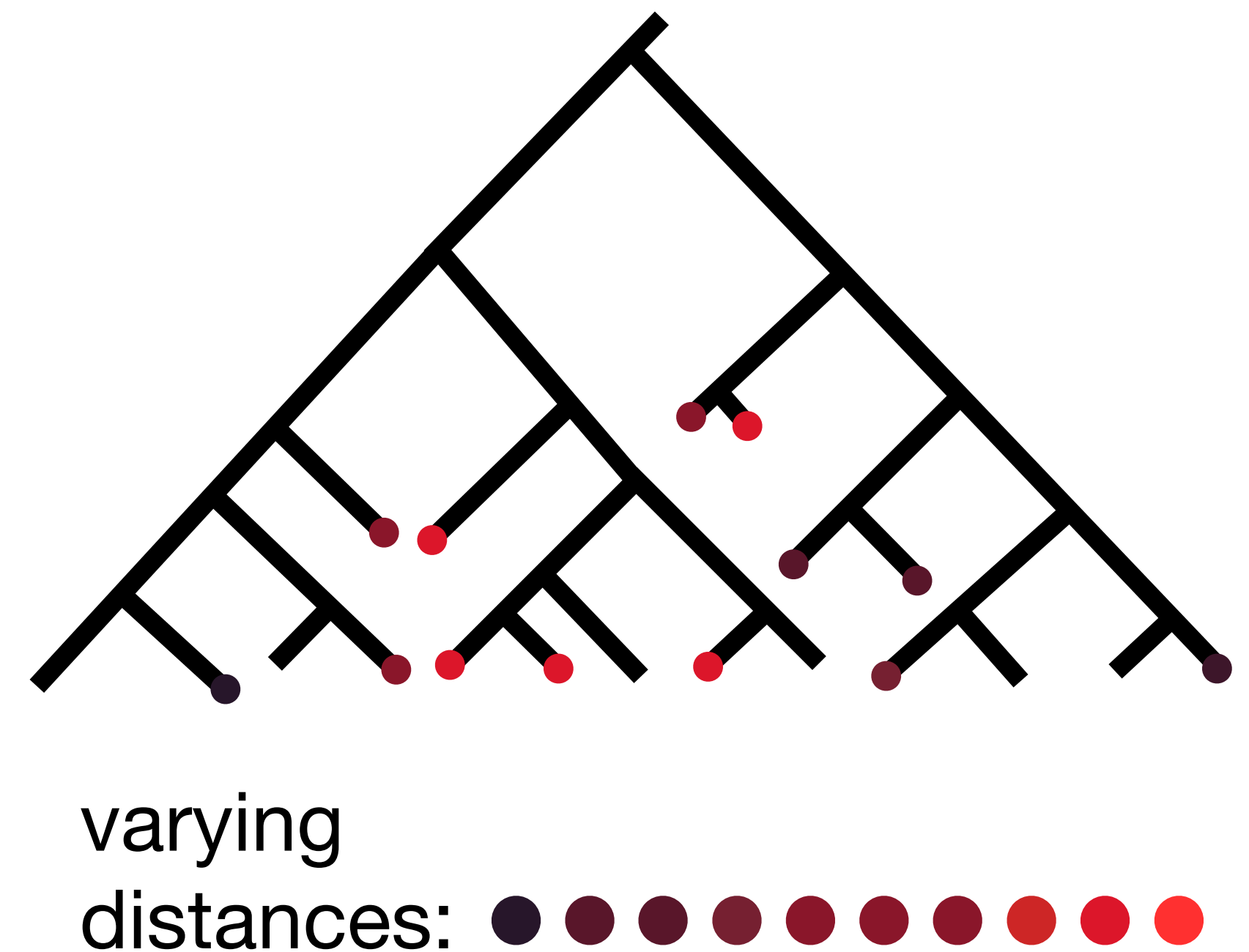


Distance-based phylogenetic placement

Given $d(q, R_i)$ for many R_i s, find the “best” placement of q on T

Challenges:

- Short reads — **low signal**
 - ▶ Small differences in distances may not be meaningful (**statistical distinguishability**)
- Tree branch lengths and distances may not match



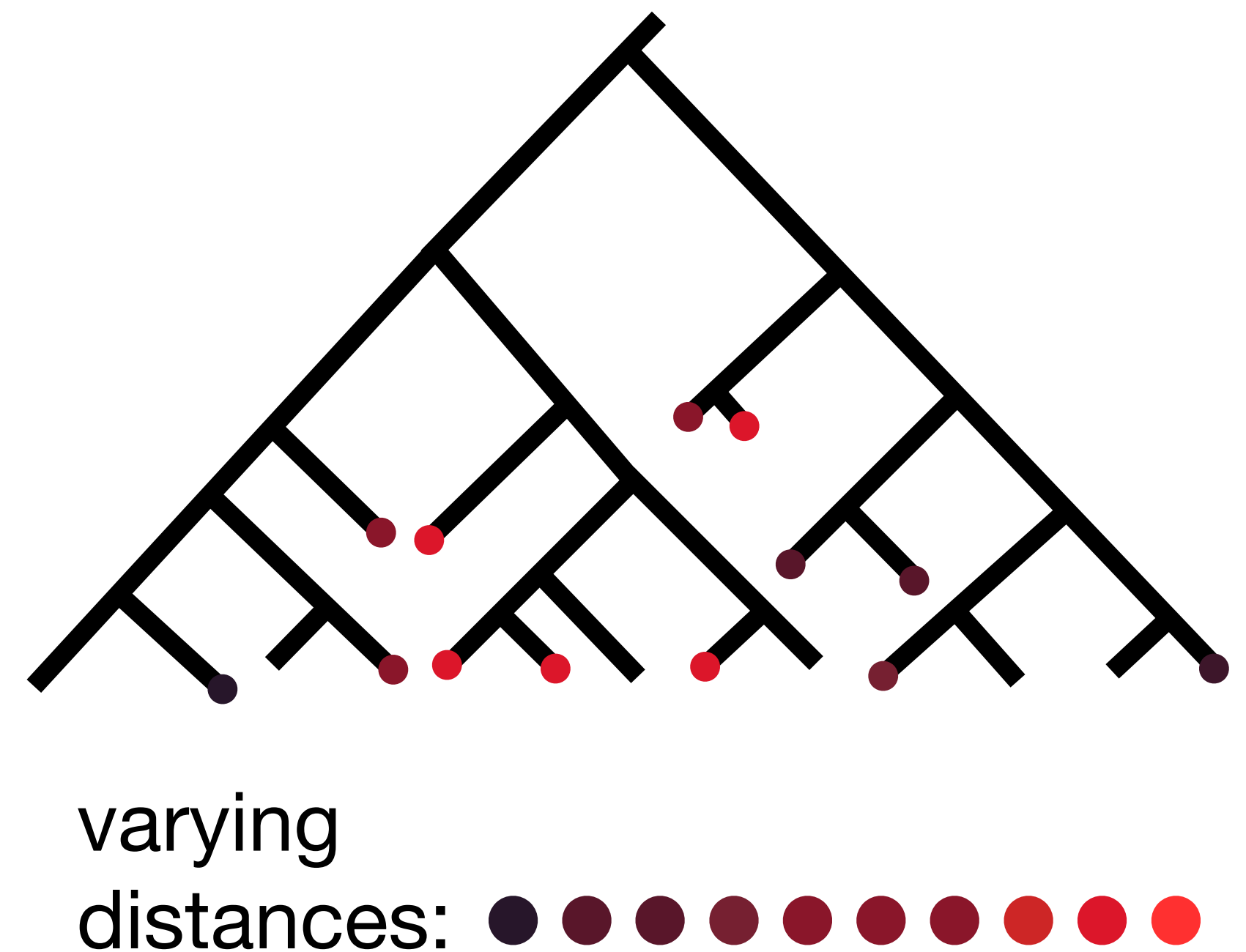
Distance-based phylogenetic placement

Given $d(q, R_i)$ for many R_i s, find the “best” placement of q on T

Challenges:

- Short reads — **low signal**
 - ▶ Small differences in distances may not be meaningful (**statistical distinguishability**)
- Tree branch lengths and distances may not match

Branch length agnostic algorithm
using **likelihood-ratio test**
(compatible with taxonomies)



Distance-based phylogenetic placement

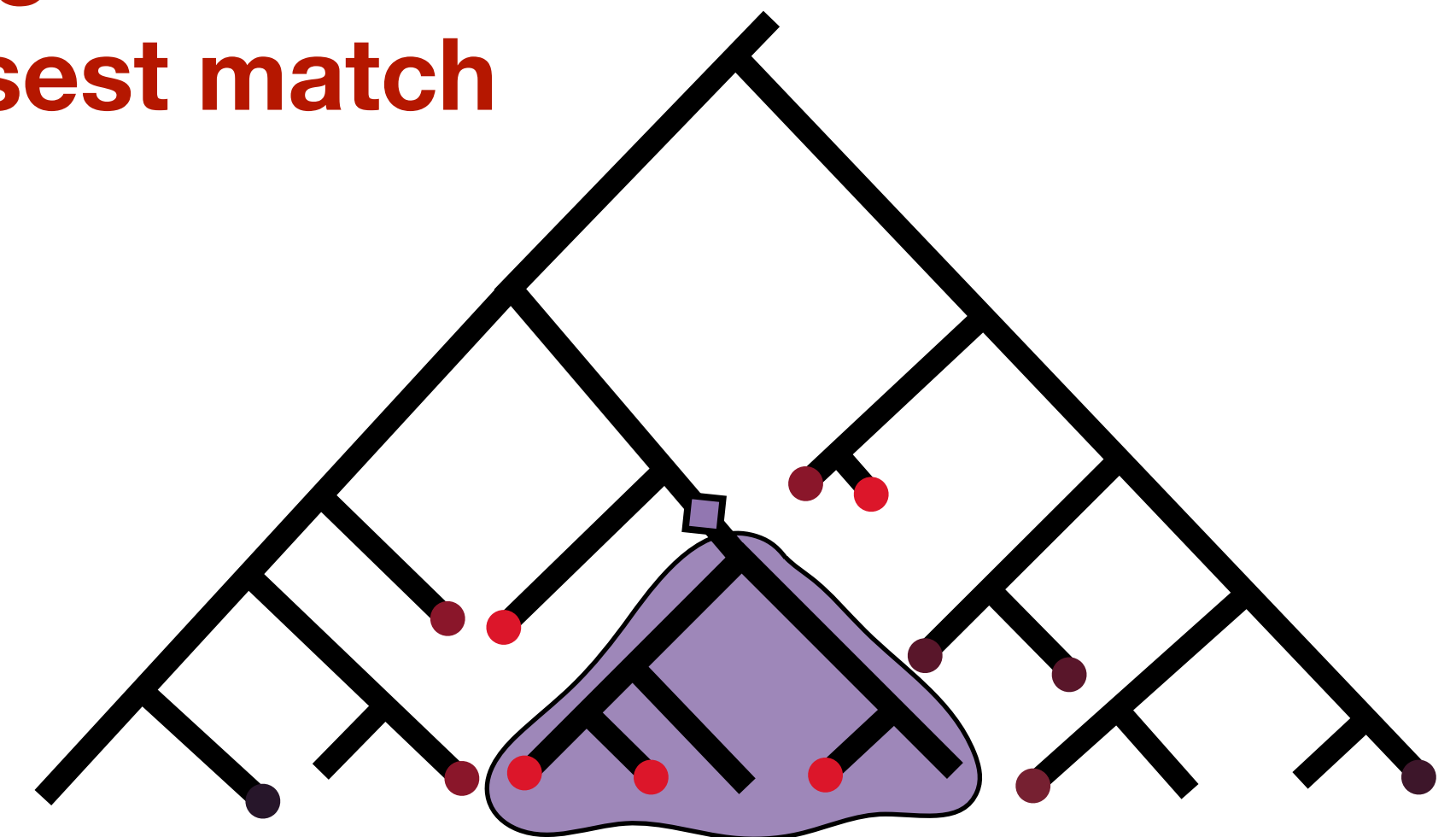
Given $d(q, R_i)$ for many R_i s, find the “best” placement of q on T

Challenges:

- Short reads — **low signal**
 - ▶ Small differences in distances may not be meaningful (**statistical distinguishability**)
- Tree branch lengths and distances may not match

Branch length agnostic algorithm
using **likelihood-ratio test**
(compatible with taxonomies)

place on the **largest clade**
indistinguishable from
the closest match



varying
distances: ●●●●●●●●●●

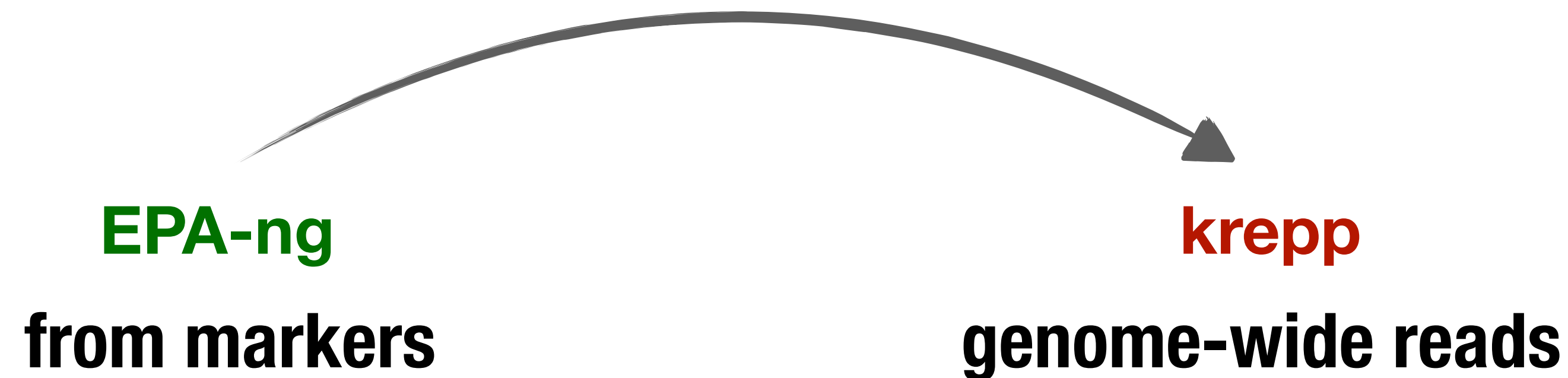
Comparing krepp & marker-based phylogenetic placement

from markers

genome-wide reads

Comparing krepp & marker-based phylogenetic placement

several orders of magnitude more reads placed
& highly accurate placements on large phylogenies



- **EPA-ng:** needs a MSA; only markers
- **krepp** can place all sequences at high rate

Working with krepp

Tool overview

- Easily scales to large reference sets
- High throughput, flexible memory
- Querying reads, contigs, genomes...

reference genomes

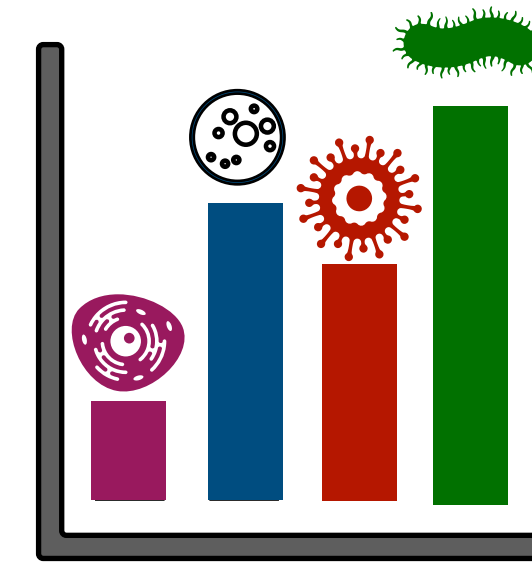
R1: TCCCTGCTCA...
R2: TCCCTGCTCA...
R3: CAATGTGCGG...
R4: CCCCAAACGA...
...
RN: ATTATCTGAT...

Indexing

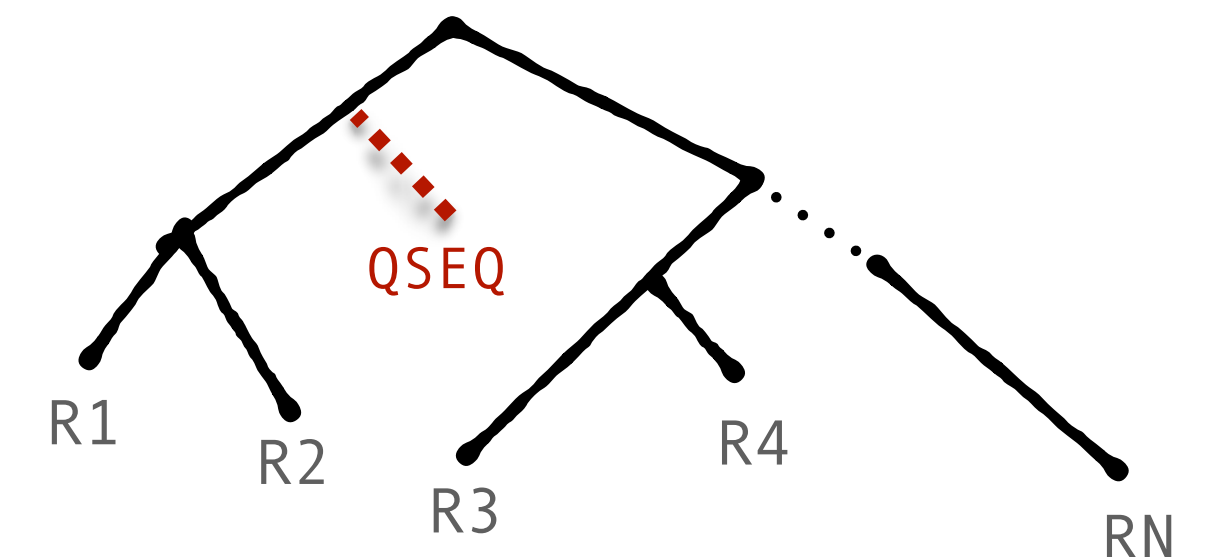
A query of any length:

@QSEQ
ATACCTAGGAG...

Taxonomic classification:



Phylogenetic placement:



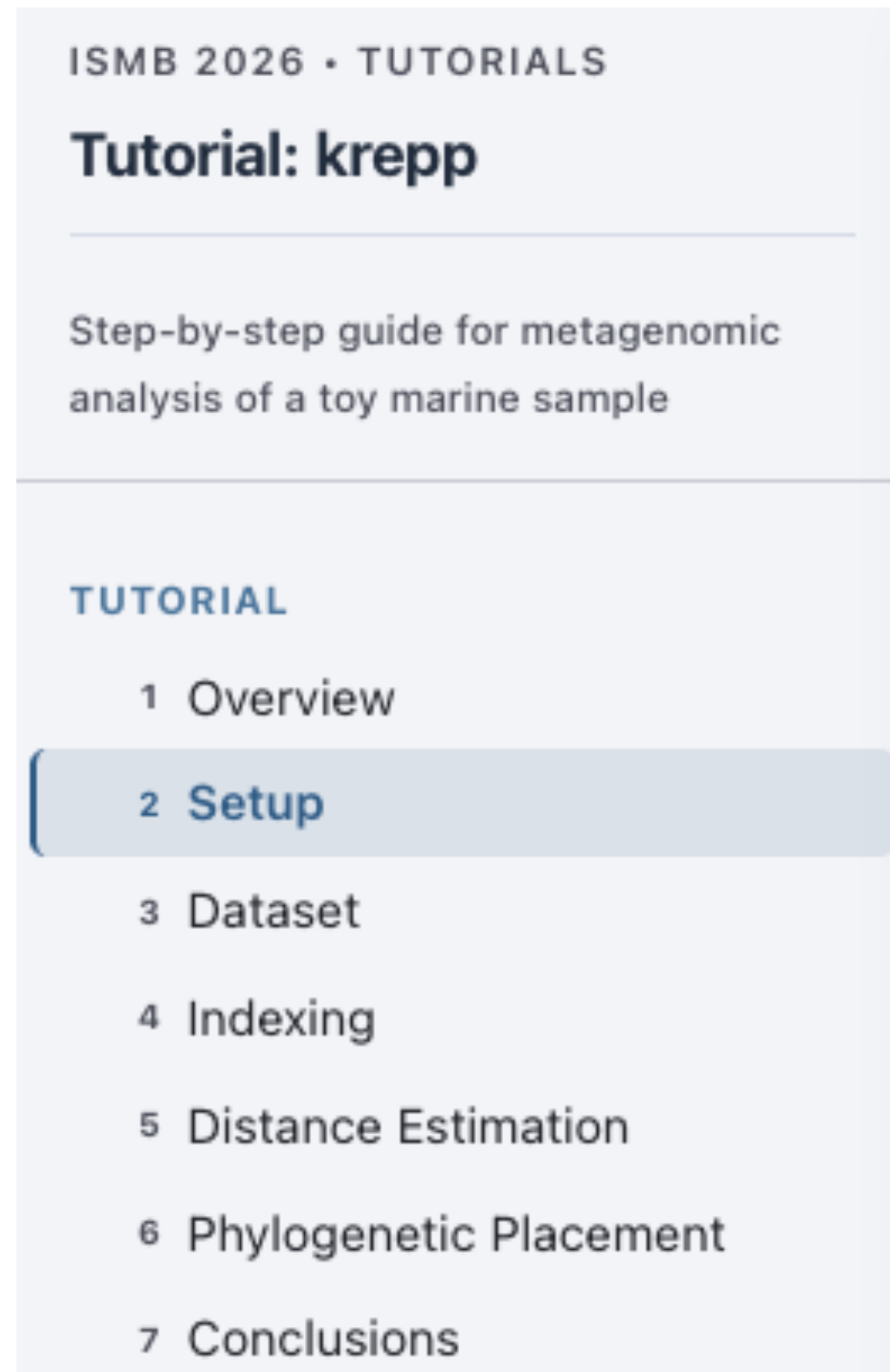
Mapping and distances:

QSEQ	R1	0.0112
QSEQ	R2	NA
QSEQ	R3	0.23
...		
QSEQ	RN	0.0016

per query

Step 1: Overview and logistics

`bo1929.github.io/documents/krepp-tutorial-ismb2026`



- Multiple sections — starting with the setup!
- **Copy & paste commands and run in your terminal!**
- You can refer to this page after the tutorial, too!

For the rest of the talk;

- Show a couple of slides before each section
- Switch to the website (please keep it open)
- Follow the step together, and running commands

Step 2: Setup

Creating an environment and installing krepp:

```
micromamba create -n krepp-tutorial -c conda-forge -c bioconda krepp=0.8.2 -y  
micromamba activate krepp-tutorial  
krepp --help
```

Install helper tools in the same environment, and verify

```
micromamba install -c conda-forge -c bioconda gappa=0.9.0 wget -y  
gappa --help
```

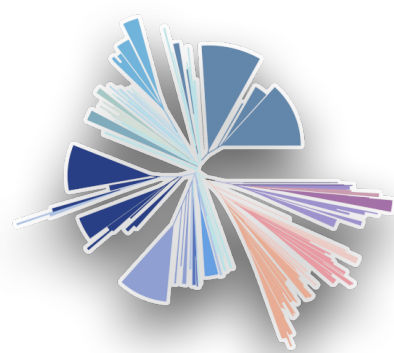
Download and prepare the tutorial dataset

```
git clone https://github.com/bo1929/ismb2026-krepp-tutorial-materials.git  
cd ismb2026-krepp-tutorial-materials  
bash scripts/setup.sh
```

Step 3: Dataset

A mock marine sample (100K reads):

- References:
- i) Toy reference set
 - ii) Web of Life



How query reads relate to the reference set



species (33%) family (2%)
genus (53%) kingdom (13%)

Kingdom composition



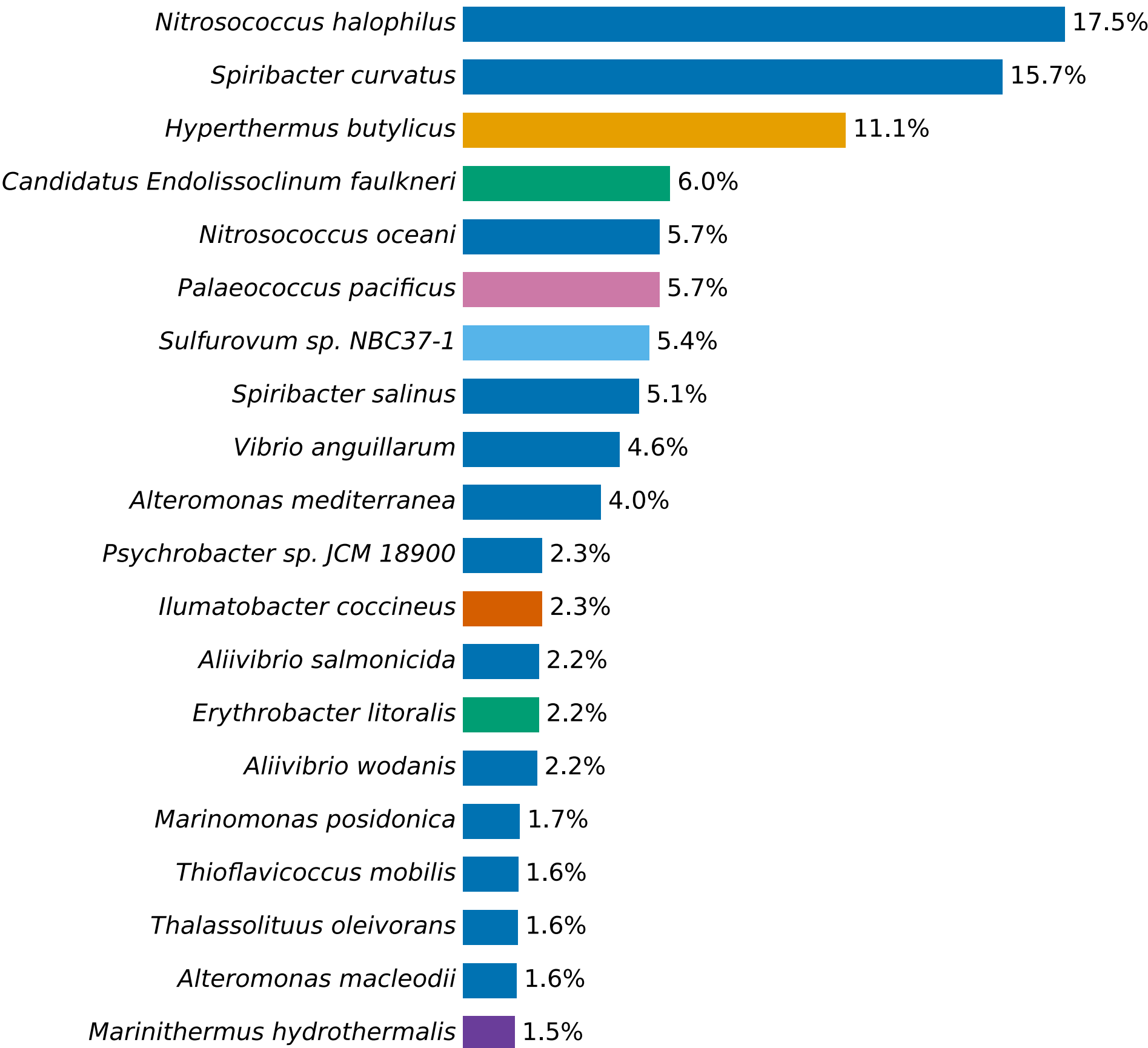
Bacteria (83%) Archaea (17%)

Class composition



Gammaproteobacteria (66%) Epsilonproteobacteria (5%)
Thermoprotei (11%) Acidimicrobiia (2%)
Alphaproteobacteria (8%) Deinococci (2%)
Thermococci (6%)

Query species (coloured by class)



0.0 2.5 5.0 7.5 10.0 12.5 15.0 17.5 20.0
Percent of query mixture

Step 4: Indexing reference genomes

reference genomes

r1: TCCCTGCTCA...

r2: TCCCTGCTCA...

r3: CAATGTGCGG...

r4: CCCCAAACGA...

...

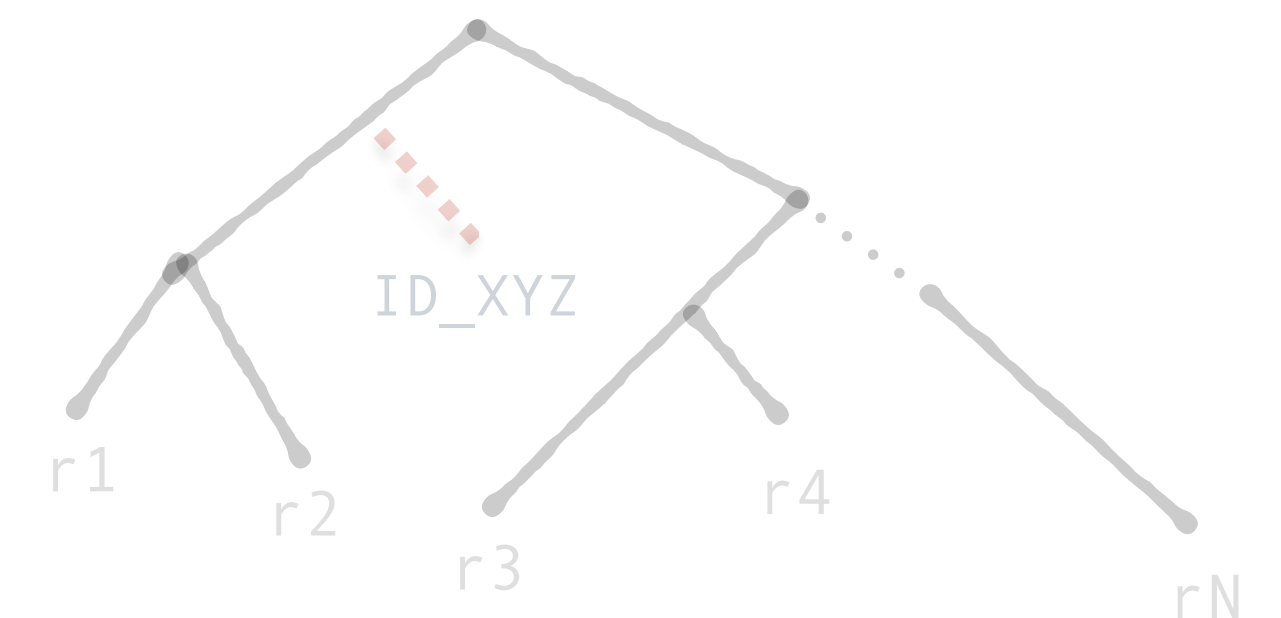
rN: ATTATCTGAT...

Indexing

A query sequence of any length:
(short reads, long reads, contigs)

>ID_XYZ
ATACCTAGGAG...

Phylogenetic placement:



Distance estimates:

ID_XYZ	r1	0.0112
ID_XYZ	r2	0.1445
ID_XYZ	r4	0.23
...		
ID_XYZ	rN	0.0016

Step 4: Indexing reference genomes

```
krepp index -i $INPUT_FILE -o $INDEX_DIR
```

-i \$INPUT_FILE: a TSV file <path> mapping reference IDs to paths/URLs

REF-ID	<TAB>	PATH
REF_XYA		/path/to/XYB.fa.gz
REF_XYB		https://url/XYB.fa.gz
...		
REF_XYZ		/path/to/XYZ.fasta

PATH:  FASTA/FASTQ
 URL/FTP/FILEPATH
 COMPRESSED

-o \$INDEX_DIR: directory <path> to store the index

output: a directory with a bunch of binary files
required for distance estimation & placement

Parameters and configuration of the index

Algorithm parameters:

- `-k, --kmer-len [19 - 31]`: Length of k-mers [29]
- `-w, --win-len`: Length of minimizer window ($w > k$) [$k+6$]
- `-h, --num-positions`: Number of positions for the LSH [$k-16$]

Symmetric DUST to filter low-complexity regions:

- `--sdust-t`: SDUST threshold
- `--sdust-w`: SDUST window

Parameters and configuration of the index

Algorithm parameters:

- `-k, --kmer-len [19 - 31]`: Length of k-mers [29]

Defaults just work, don't worry about these!

(unless you are doing something peculiar like aDNA analysis)

- `--sdust-t`: SDUST threshold
- `--sdust-w`: SDUST window

Parameters and configuration of the index

Algorithm parameters:

- `-k, --kmer-len [19 - 31]`: Length of k-mers [29]

Defaults just work, don't worry about these!

(unless you are doing something peculiar like aDNA analysis)

- `--sdust-t`: SDUST threshold
- `--sdust-w`: SDUST window

What if we don't have enough memory?

Partial indexing & partial loading existing full indices:

- `-m, --modulo-lsh`: Modulo value to partition LSH space
- `-r, --residue-lsh`: include a k -mer only if $r = \text{LSH}(x) \bmod m$

subsampling
rate:
 $(r + 1)/m$

Parameters and configuration of the index

Algorithm parameters:

- `-k, --kmer-len [19 - 31]`: Length of k-mers [29]

Defaults just work, don't worry about these!

(unless you are doing something peculiar like aDNA analysis)

- `--sdust-t`: SDUST threshold
- `--sdust-w`: SDUST window

What if we don't have enough memory?

Limit the memory with little expense in accuracy!

(more details in the documentation)

subsampling
rate:
 $(r + 1)/m$

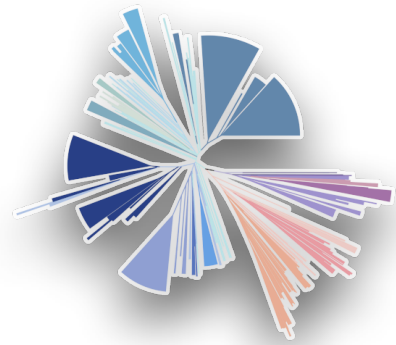
Our index catalogue

- Indexing is relatively a one time but expensive operation
- We have a catalogue: **available to download** from our FTP server and AWS!

Our index catalogue

- Indexing is relatively a one time but expensive operation
- We have a catalogue: **available to download** from our FTP server and AWS!

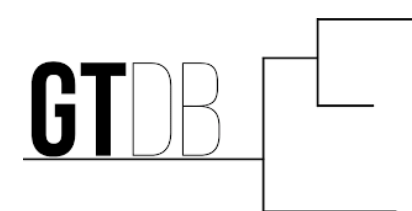
Microbial genomes (archaeal and bacterial):



Web of Life (WoL): lightweight and the most reliable phylogeny
16,000 refs. (65 GB / 35 GB / ...)



RefSeq-microbial: denser sampling and a high quality phylogeny
120,000 refs. (180 GB / 95 GB / ...)



GTDB: the latest release, an up-to-date reference
145,000 refs. (230 GB / 120 GB / ...)

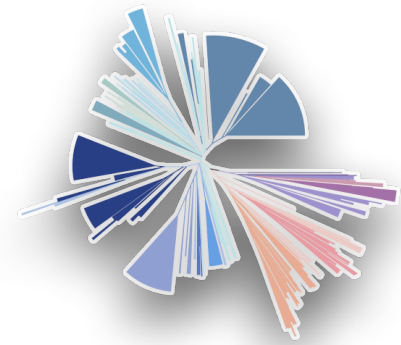
Indexes for eukaryotic genomes — not available yet!

Links are on GitHub!

Our index catalogue

- Indexing is relatively a one time but expensive operation
- We have a catalogue: **available to download** from our FTP server and AWS!

Microbial genomes (archaeal and bacterial):



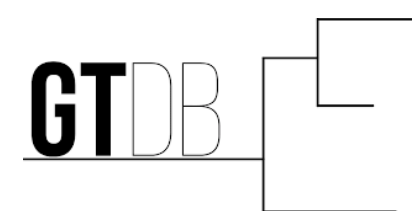
Web of Life (WoL): lightweight and the most reliable phylogeny
16,000 refs. (65 GB / 35 GB / ...)



**we will use a 1.5 GB
version: WoL-tiny**



RefSeq-microbial: denser sampling and a high quality phylogeny
120,000 refs. (180 GB / 95 GB / ...)



GTDB: the latest release, an up-to-date reference
145,000 refs. (230 GB / 120 GB / ...)

Indexes for eukaryotic genomes — not available yet!

Links are on GitHub!

Tutorial: krepp

Step-by-step guide for metagenomic analysis of a toy marine sample

TUTORIAL

- 1 Overview
- 2 Setup
- 3 Dataset
- 4 **Indexing**
- 5 Distance Estimation
- 6 Phylogenetic Placement
- 7 Conclusions

ON THIS PAGE ^

- Indexing genomes from scratch
- Installing pre-built indexes

Step 5: Estimating read-to-reference distances

Distances of each read to all sufficiently close (~25%) references!

A query sequence of any length:
(short reads, long reads, contigs)

@QSEQ_1
ATACCTAGGAG...

reference genomes

r1: TCCCTGCTCA...

r2: TCCCTGCTCA...

r3: CAATGTGCGG...

r4: CCCCAAACGA...

...

rN: ATTATCTGAT...

Indexing

**Estimates are akin to alignment HD,
yet more sensitive and faster!**

Distance estimates:

QSEQ_1	r1	0.0112
QSEQ_1	r2	0.1445
QSEQ_1	r4	0.23
...		
QSEQ_1	rN	0.0016

Estimating read-to-reference distances

```
krepp dist -i $INDEX_DIR -q $QUERY_FILE -o $OUTPUT_FILE
```

- i \$INDEX_DIR:** directory <path> containing the reference index
- q \$QUERY_PATH:** query FASTA/FASTQ file <path> (or URL) (gzip compatible)
- o \$OUTPUT_FILE:** write output (TSV) to a file at <path> (default: stdout)

Estimating read-to-reference distances

```
krepp dist -i $INDEX_DIR -q $QUERY_FILE -o $OUTPUT_FILE
```

- i \$INDEX_DIR:** directory <path> containing the reference index
- q \$QUERY_PATH:** query FASTA/FASTQ file <path> (or URL) (gzip compatible)
- o \$OUTPUT_FILE:** write output (TSV) to a file at <path> (default: stdout)

Query FASTQ

```
@QSEQ_1
GCGGATATCTGGGAAACGCCGGTTGCAAGCAGAG
+
=CCGGGGGGGGGGGGGCJJJJGJJGGGJJJJCC8JJ
@QSEQ_2
TCACCCTCTATTTCCACCACACATACCCGGCAAC
+
=C=GGGCG=CGGGG=JJJJJJ=JJCJJJJJJJJ
...
```

krepp dist



Distance report

SEQ_ID	REFERENCE_NAME	DIST
QSEQ_1	REF_XYA	0.02
QSEQ_1	REF_XYD	0.12
QSEQ_1	REF_XYZ	0.03
QSEQ_2	REF_XYA	0.001
QSEQ_2	REF_XYC	0.22
QSEQ_3	NA	NA
...		

Some relevant options for basic filtering

Default: `--no-filter --multi`

SEQ_ID	REFERENCE_NAME	DIST
QSEQ_1	REF_XYA	0.02
QSEQ_1	REF_XYD	0.12
QSEQ_1	REF_XYZ	0.03
QSEQ_2	REF_XYA	0.001
QSEQ_2	REF_XYC	0.22
QSEQ_3	NA	NA
...		

all reference hits

Some relevant options for basic filtering

`--filter`: if indistinguishable from the closest

Default: `--no-filter --multi`

SEQ_ID	REFERENCE_NAME	DIST
QSEQ_1	REF_XYA	0.02
QSEQ_1	REF_XYD	0.12
QSEQ_1	REF_XYZ	0.03
QSEQ_2	REF_XYA	0.001
QSEQ_2	REF_XYC	0.22
QSEQ_3	NA	NA
...		

all reference hits

SEQ_ID	REFERENCE_NAME	DIST
QSEQ_1	REF_XYA	0.02
QSEQ_1	REF_XYD	0.12
QSEQ_1	REF_XYZ	0.03
QSEQ_2	REF_XYA	0.001
QSEQ_2	REF_XYC	0.22
QSEQ_3	NA	NA
...		

Some relevant options for basic filtering

--filter: if indistinguishable from the closest

Default: --no-filter --multi

SEQ_ID	REFERENCE_NAME	DIST
QSEQ_1	REF_XYA	0.02
QSEQ_1	REF_XYD	0.12
QSEQ_1	REF_XYZ	0.03
QSEQ_2	REF_XYA	0.001
QSEQ_2	REF_XYC	0.22
QSEQ_3	NA	NA
...		

all reference hits

SEQ_ID	REFERENCE_NAME	DIST
QSEQ_1	REF_XYA	0.02
QSEQ_1	REF_XYD	0.12
QSEQ_1	REF_XYZ	0.03
QSEQ_2	REF_XYA	0.001
QSEQ_2	REF_XYC	0.22
QSEQ_3	NA	NA
...		

--no-multi: only the best matching ref.

SEQ_ID	REFERENCE_NAME	DIST
QSEQ_1	REF_XYA	0.02
QSEQ_1	REF_XYD	0.12
QSEQ_1	REF_XYZ	0.03
QSEQ_2	REF_XYA	0.001
QSEQ_2	REF_XYC	0.22
QSEQ_3	NA	NA
...		

Tutorial: krepp

Step-by-step guide for metagenomic analysis of a toy marine sample

TUTORIAL

- 1 Overview
- 2 Setup
- 3 Dataset
- 4 Indexing
- 5 Distance Estimation**
- 6 Phylogenetic Placement
- 7 Conclusions

ON THIS PAGE ^

- Per-read distances (default)
- Other options for distance estimation
- Interpretation of results
- Counting operational genomic units

Step 6: Placing reads on a backbone tree

Placing each query read on a backbone tree to explain its distance estimates!

Query sequences of any length:
(short reads, long reads, contigs)

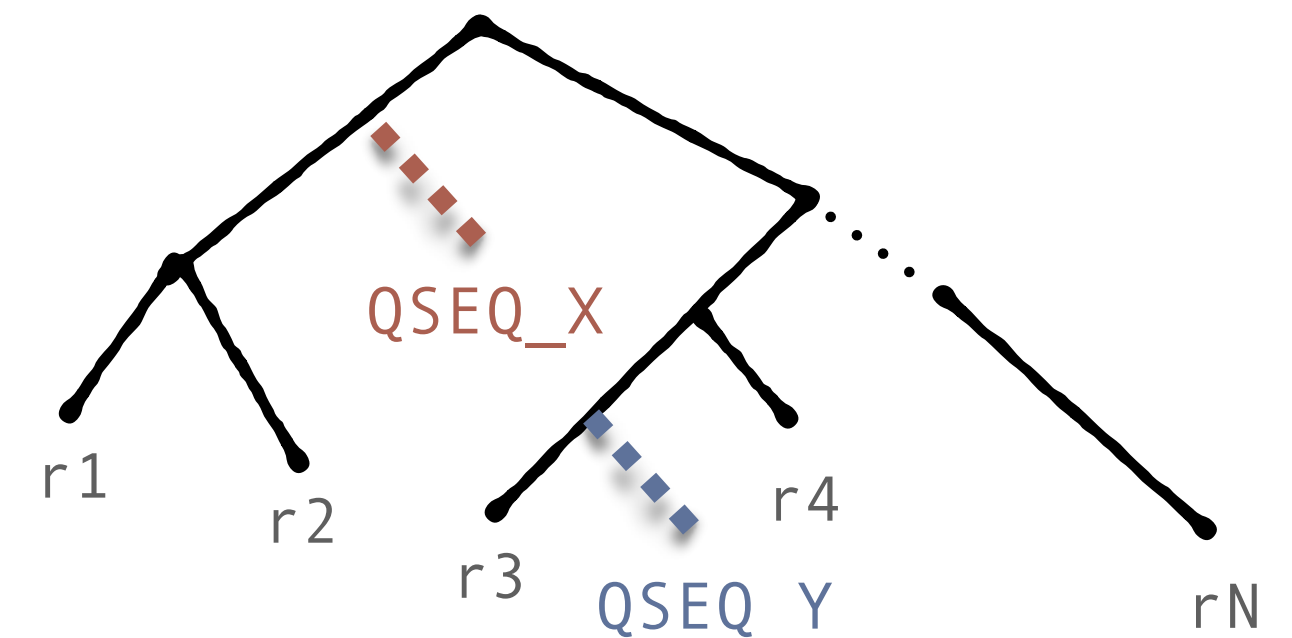
>QSEQ_X
ATACCTAGGAG...
>QSEQ_Y
G TTCCTCTGAG...

reference genomes

r1: TCCCTGCTCA...
r2: TCCCTGCTCA...
r3: CAATGTGCGG...
r4: CCCCAAACGA...
...
rN: ATTATCTGAT...

Indexing

Phylogenetic placement:



Distance estimates:

QSEQ_X	r1	0.0112
QSEQ_X	r9	0.1445
...		
QSEQ_Y	r3	0.0016
QSEQ_Y	rN	0.1253

Placing reads on a backbone tree

```
krepp place -i $INDEX_DIR -t $NWK_PATH -q $QUERY_FILE
```

- i \$INDEX_DIR:** directory <path> containing the reference index
- t \$NWK_PATH:** backbone tree in Newick format — tip labels must match ref. IDs
- q \$QUERY_PATH:** query FASTA/FASTQ file <path> (or URL, gzip compatible)
- o \$OUTPUT_FILE:** write output (jplace—JSON-based) to a file at <path> (default: stdout)

Placing reads on a backbone tree

```
krepp place -i $INDEX_DIR -t $NWK_PATH -q $QUERY_FILE
```

- i \$INDEX_DIR:** directory <path> containing the reference index
- t \$NWK_PATH:** backbone tree in Newick format — tip labels must match ref. IDs
- q \$QUERY_PATH:** query FASTA/FASTQ file <path> (or URL, gzip compatible)
- o \$OUTPUT_FILE:** write output (jplace—JSON-based) to a file at <path> (default: stdout)

Query FASTQ

```
@QSEQ_1
GCGGATATCTGGGAAACGCCGGTTGCAA
+
=CCGGGGGGGGGGGGGGGCJJJJGJJGGGJJJ
@QSEQ_2
TCACCCTCTATTTCACACACATACCC
+
=C=GGGCG=CGGGG=JJJJJJ=JJCJJJ
...
```

krepp place



jplace report

```
{
  "version" : 3,
  "fields" : ["edge_num", ..., "like_weight_ratio"],
  "placements" : [
    { "n" : ["QSEQ_1"], "p" : [
      [39, 0.0010, ..., 1]] },
    { "n" : ["QSEQ_2"], "p" : [
      [35, 0.005, ..., 0.437993]] }
  ]
  "metadata" : {<METADATA>},
  "tree" : <LABELED_NEWICK_STRING>
}
```

More on jplace format

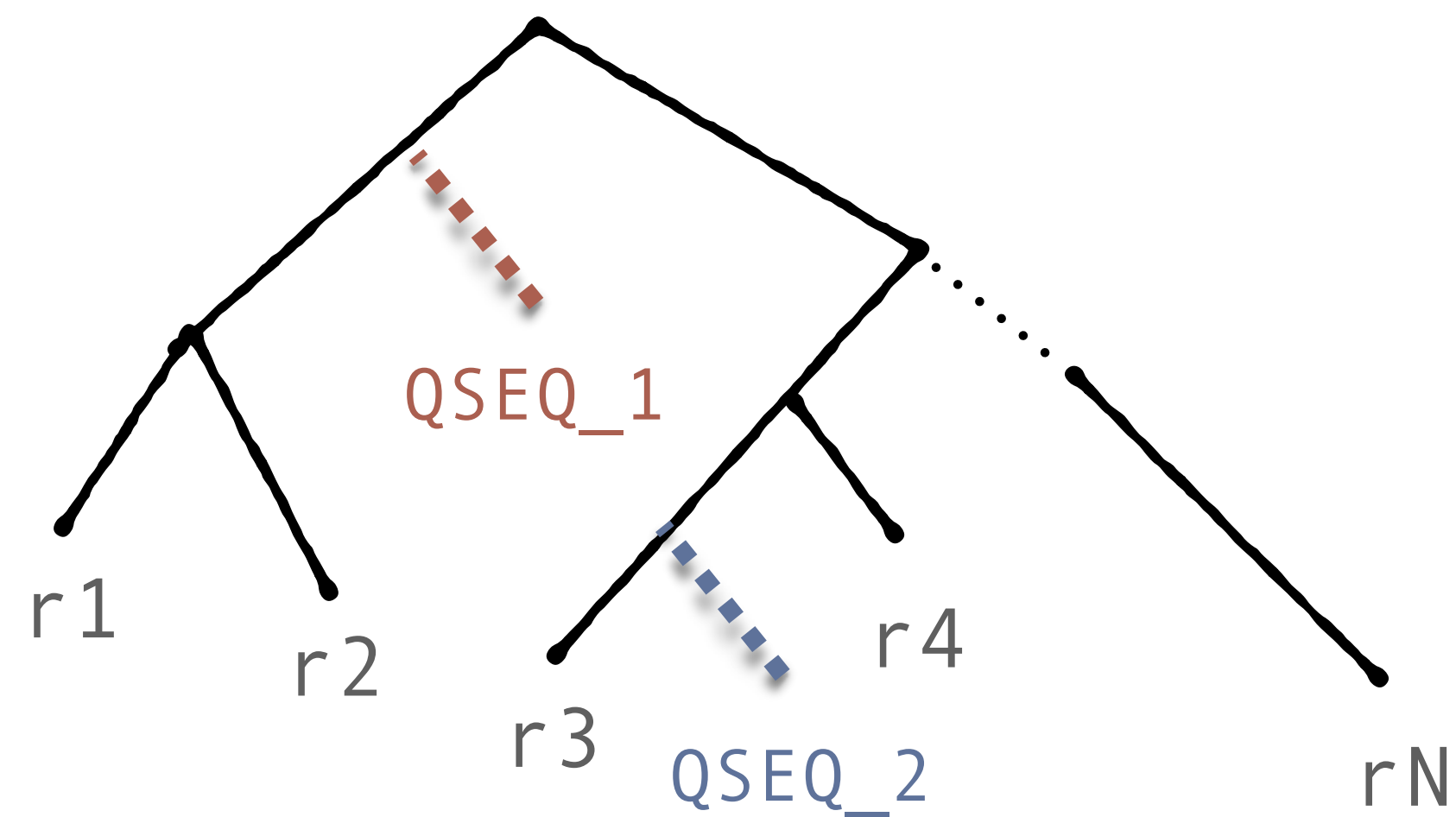
Just a JSON; not meant to be directly inspected; process using suitable tools: **gappa**
github.com/lczech/gappa

 branch numbers from a post-order traversal

```
{
  "version" : 3,
  "fields" : ["edge_num", "pendant_length", "distal_length", "likelihood", "like_weight_ratio"],
  "placements" : [
    {"n" : ["QSEQ_1"], "p" : [
      3, 0, 0.00108799, -11.5816, 1]}},
    {"n" : ["QSEQ_2"], "p" : [
      18, 0, 0.00585016, -16.6413, 0.437993]}]}
  ]
  "metadata" : {
    "software" : "krepp",
    "version" : "v0.4.8",
    "repository" : "https://github.com/bo1929/krepp",
    "num_queries" : "2",
    "invocation" : "krepp place -i index -q query.fq"
  },
  "tree" : "(REF_XYA:0.42{0}, ((REF_XYB:0.03{1}, REF_XYC:0.12{2})N_XY:0.22{3}, ...) )"
}
```

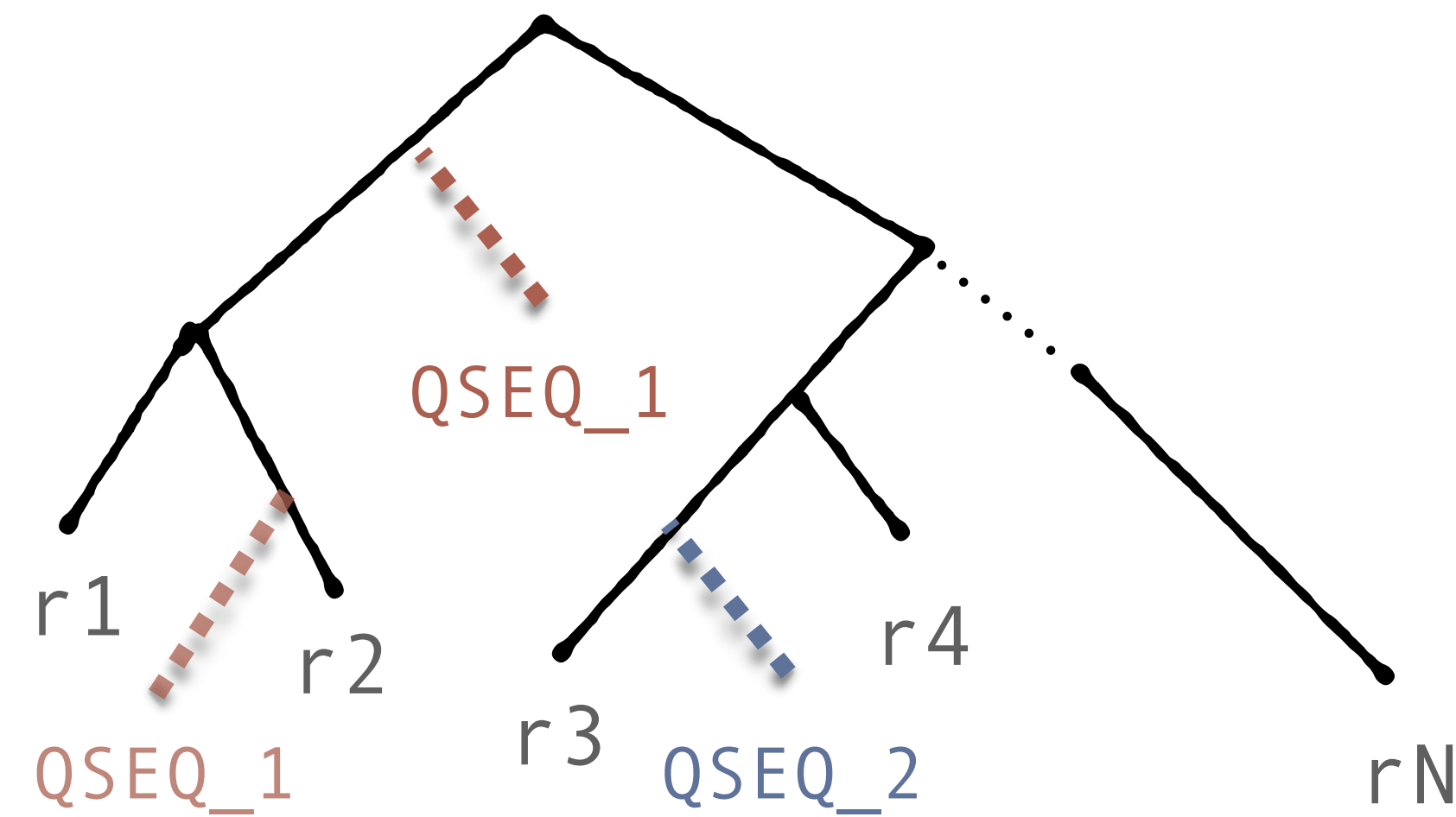

Multiplacement option

Default: --no-multi



Reports only the best placement

--multi



Multiple likely placements w/ weights

Tutorial: krepp

Step-by-step guide for metagenomic analysis of a toy marine sample

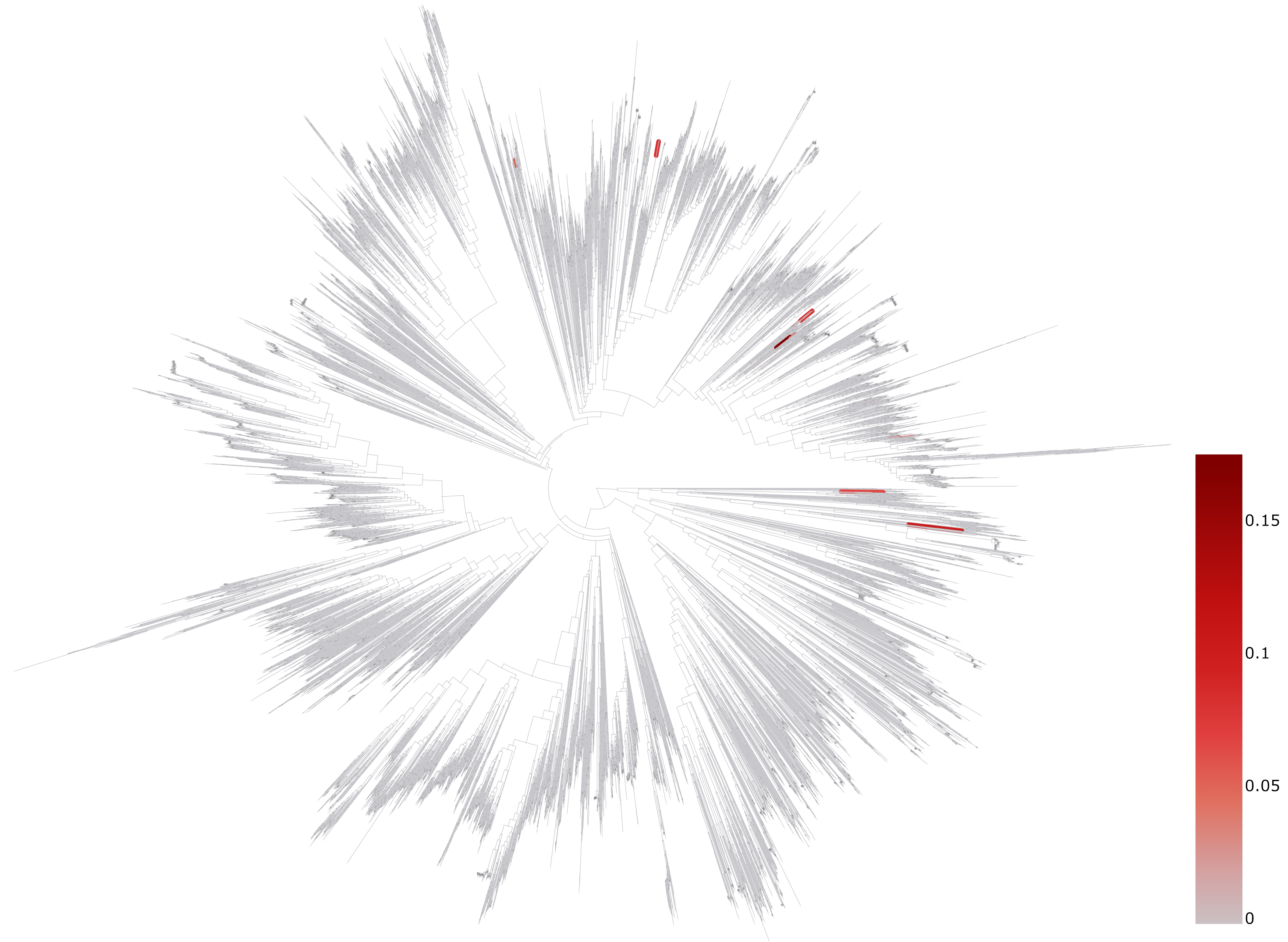
TUTORIAL

- 1 Overview
- 2 Setup
- 3 Dataset
- 4 Indexing
- 5 Distance Estimation
- 6 Phylogenetic Placement**
- 7 Conclusions

ON THIS PAGE ^

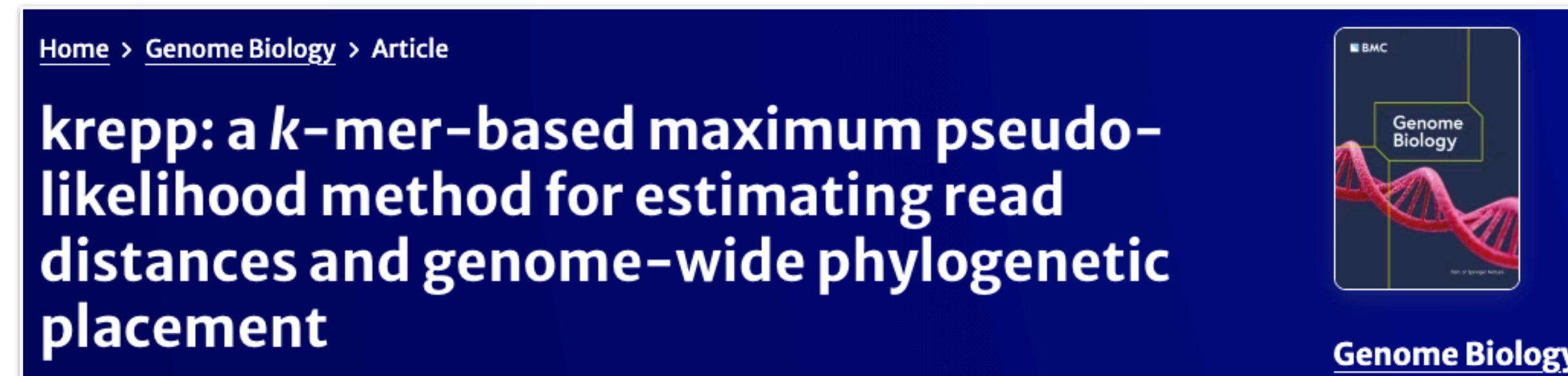
- Placing reads on a phylogeny
- Other placement options and formats
- Placing on a taxonomy

Step 7: Conclusions and Exercises



Thank you!

paper: **Genome Biology**



software: **github.com/bo1929/krepp**

recipe **krepp**

krepp: a k-mer-based maximum pseudo-likelihood method for estimating read distances and genome-wide phylogenetic placement.

Homepage: <https://github.com/bo1929/krepp>

License: MIT / MIT

Recipe: [/krepp/metayaml](#)

Links: doi: [10.1186/s13059-026-03999-y](https://doi.org/10.1186/s13059-026-03999-y), doi: [10.5281/zenodo.15466359](https://doi.org/10.5281/zenodo.15466359)

package **krepp**

BIOCONDA

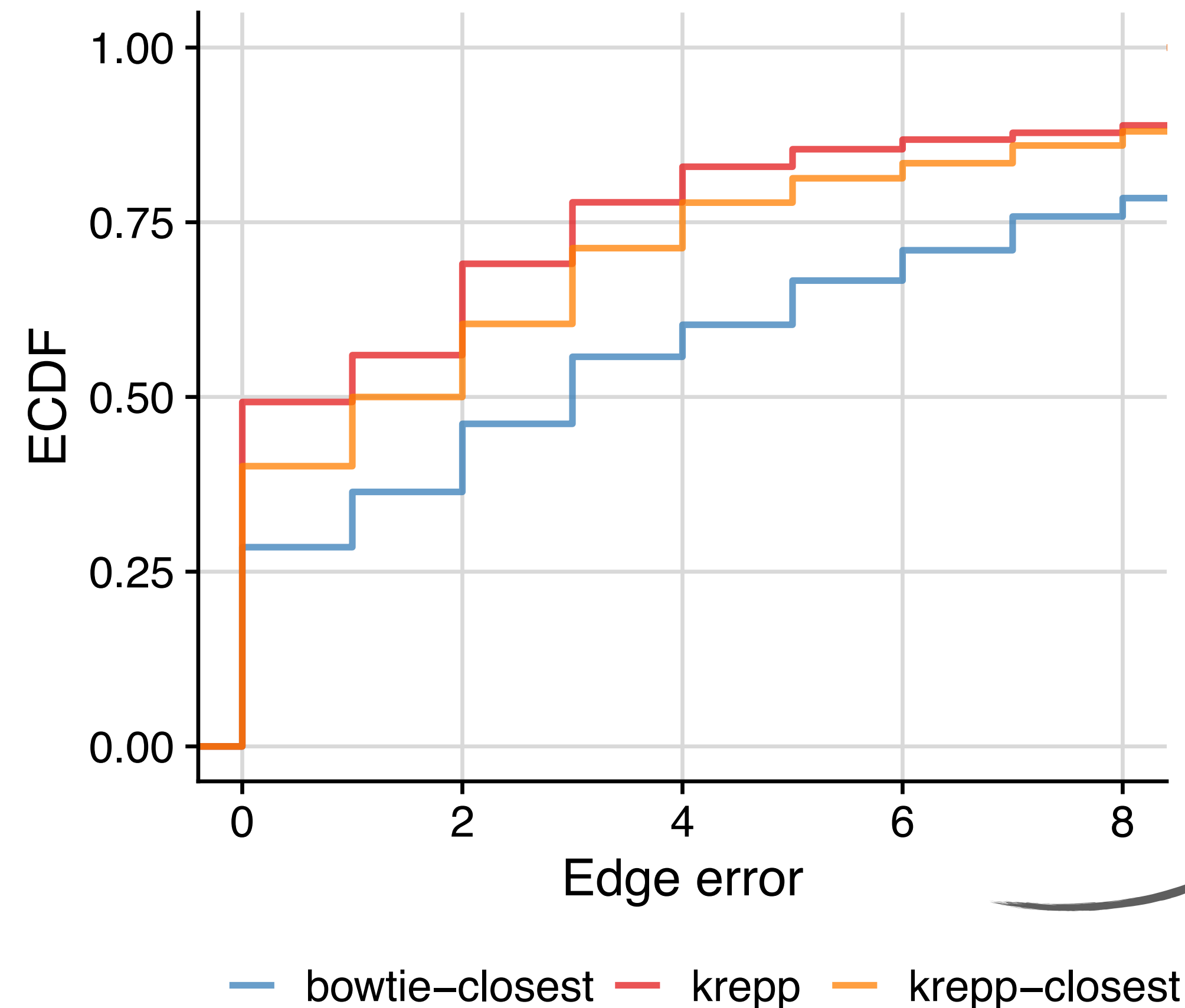
downloads 1.6k container none



Extra Slides

krepp's heuristic improves closest-tip placement

- Leave one out: 100/16,000 (WoLv2)
- Outperforms baselines:
on the closest, on the LCA, etc.
- >80% of all reads within four edges



how many edges
away is the
placement from
the correct edge?

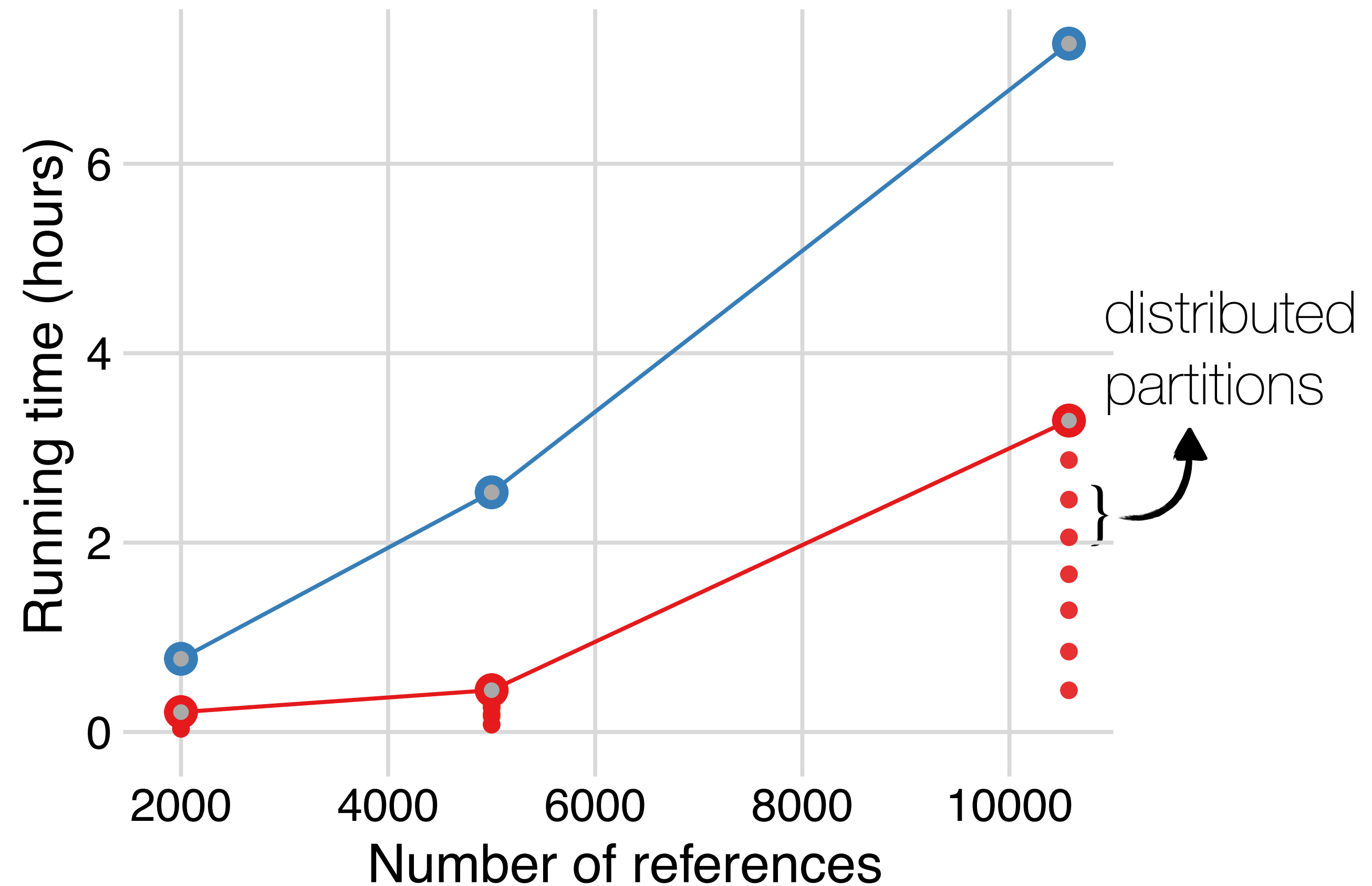
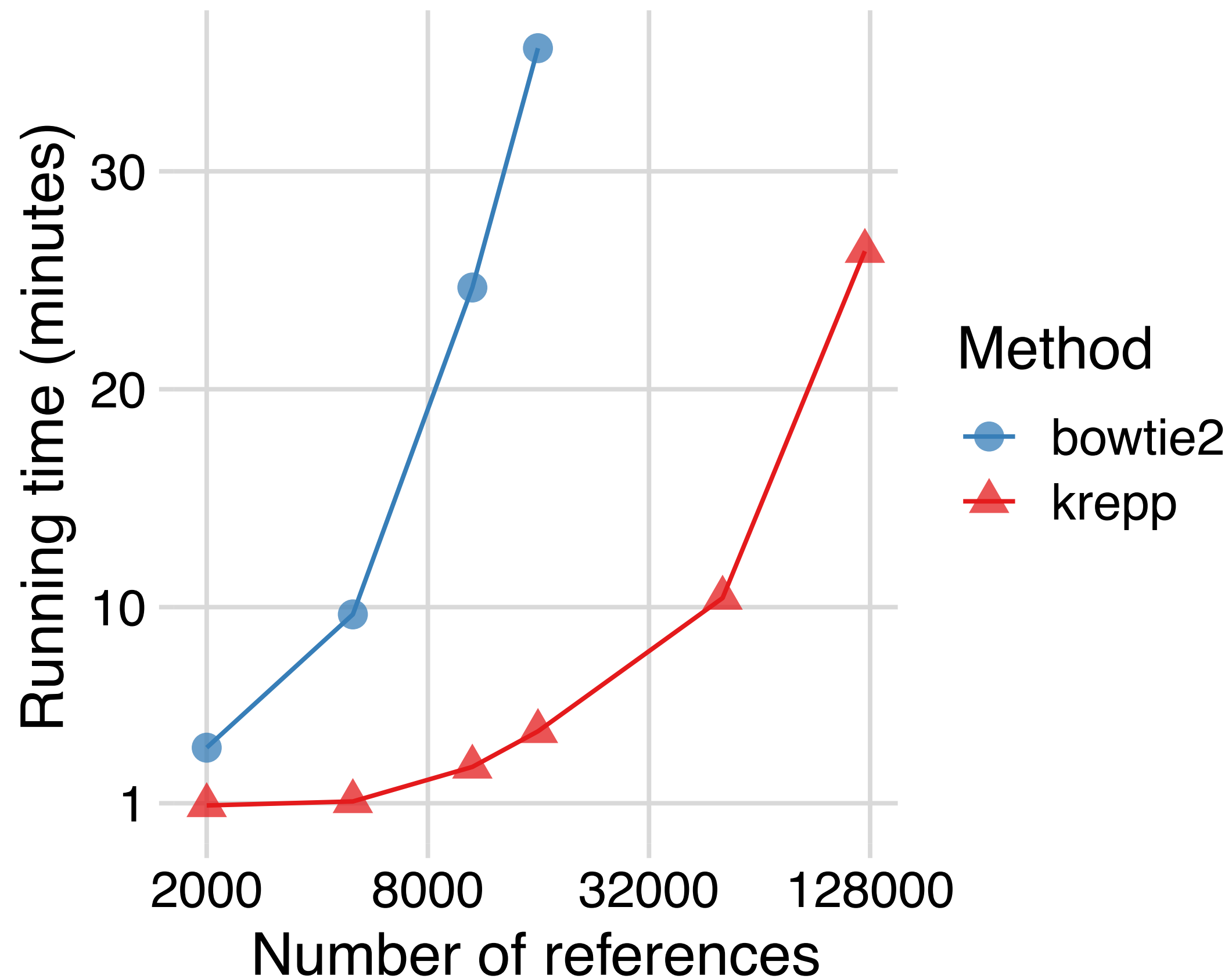
Scalability:

Avoiding the more difficult problem & effective parallelization

120,000 genomes in <1 hour!

Mapping 10M reads (16 threads):

Indexing microbial genomes (32 threads):

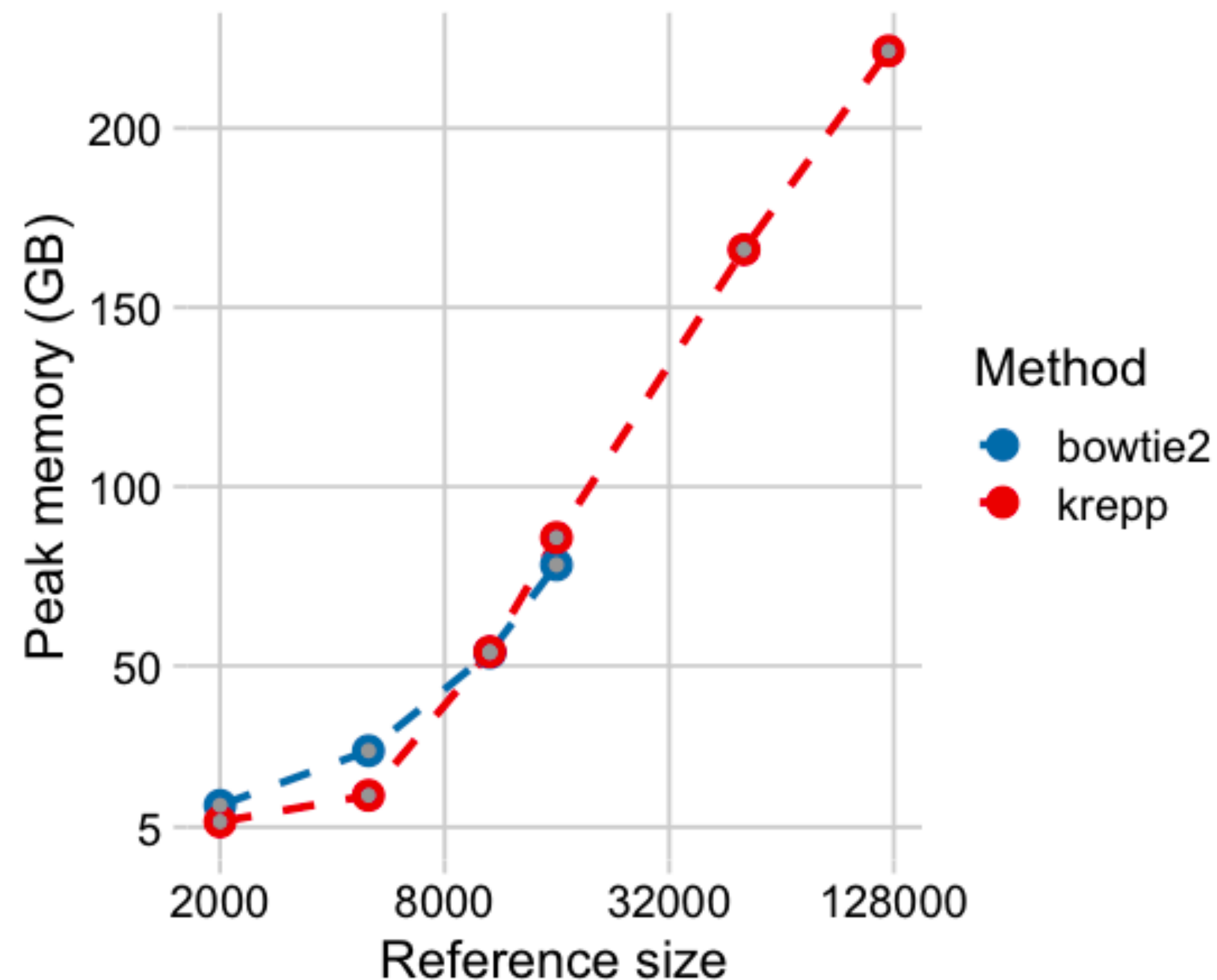


Scalability:

krepp can be distributed and has flexible memory requirements

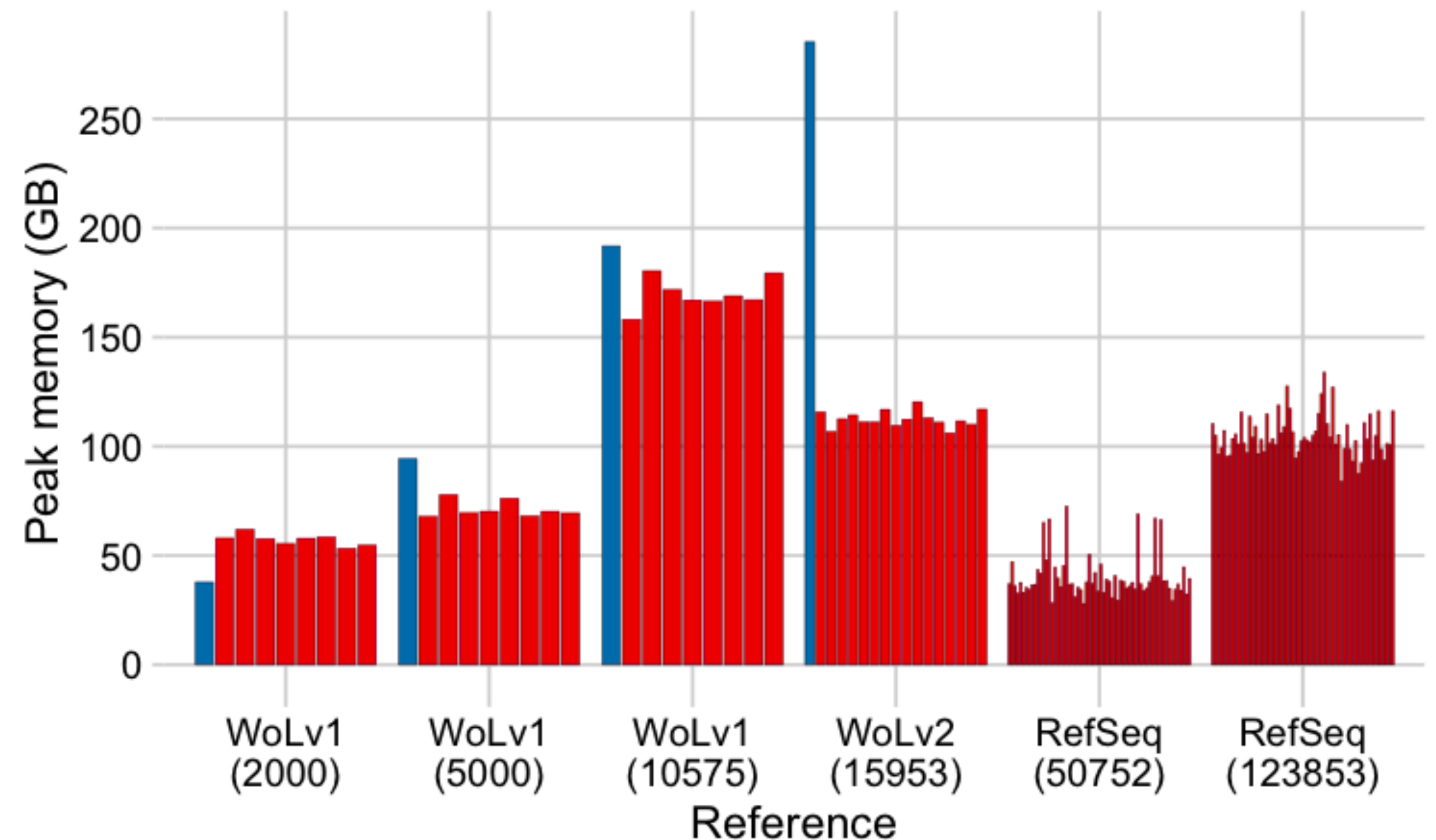
Mapping 10M reads (16 threads):

Reducing memory use
with further subsampling...

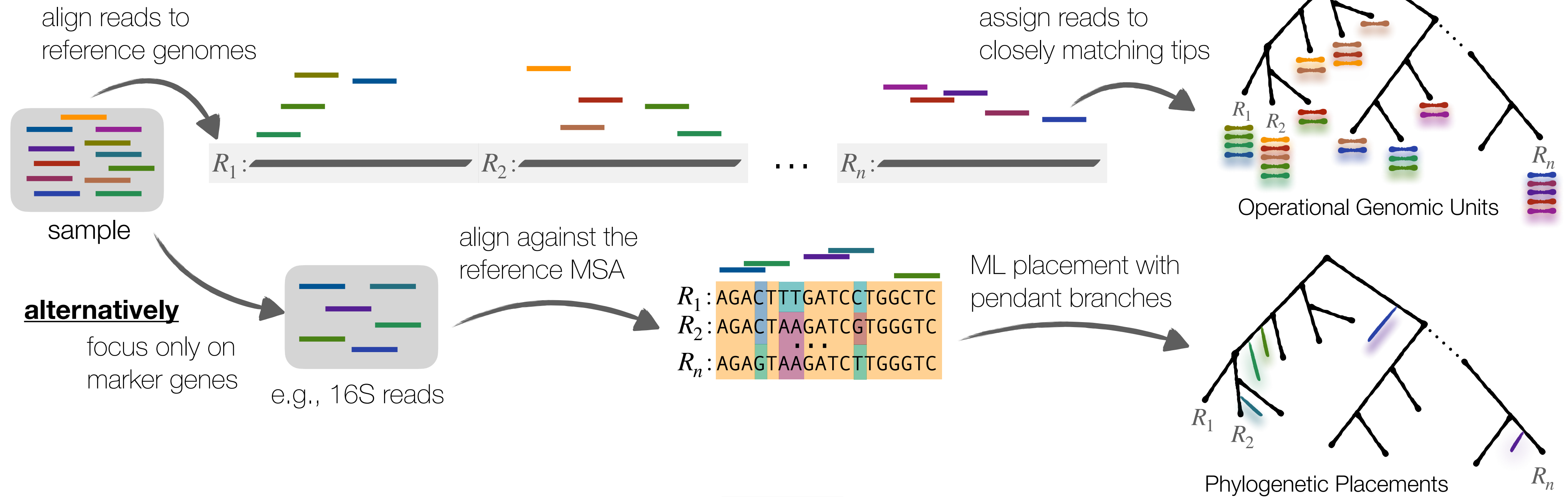


Indexing microbial genomes (32 threads):

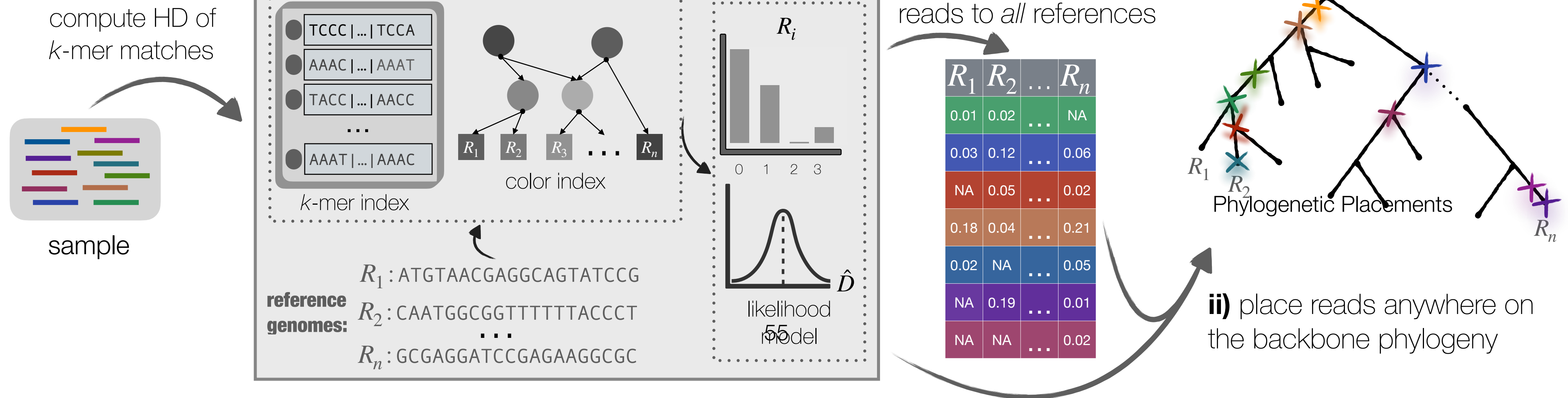
Adjusting partitioning based on the
input size & available memory...



Existing Pipelines



krepp

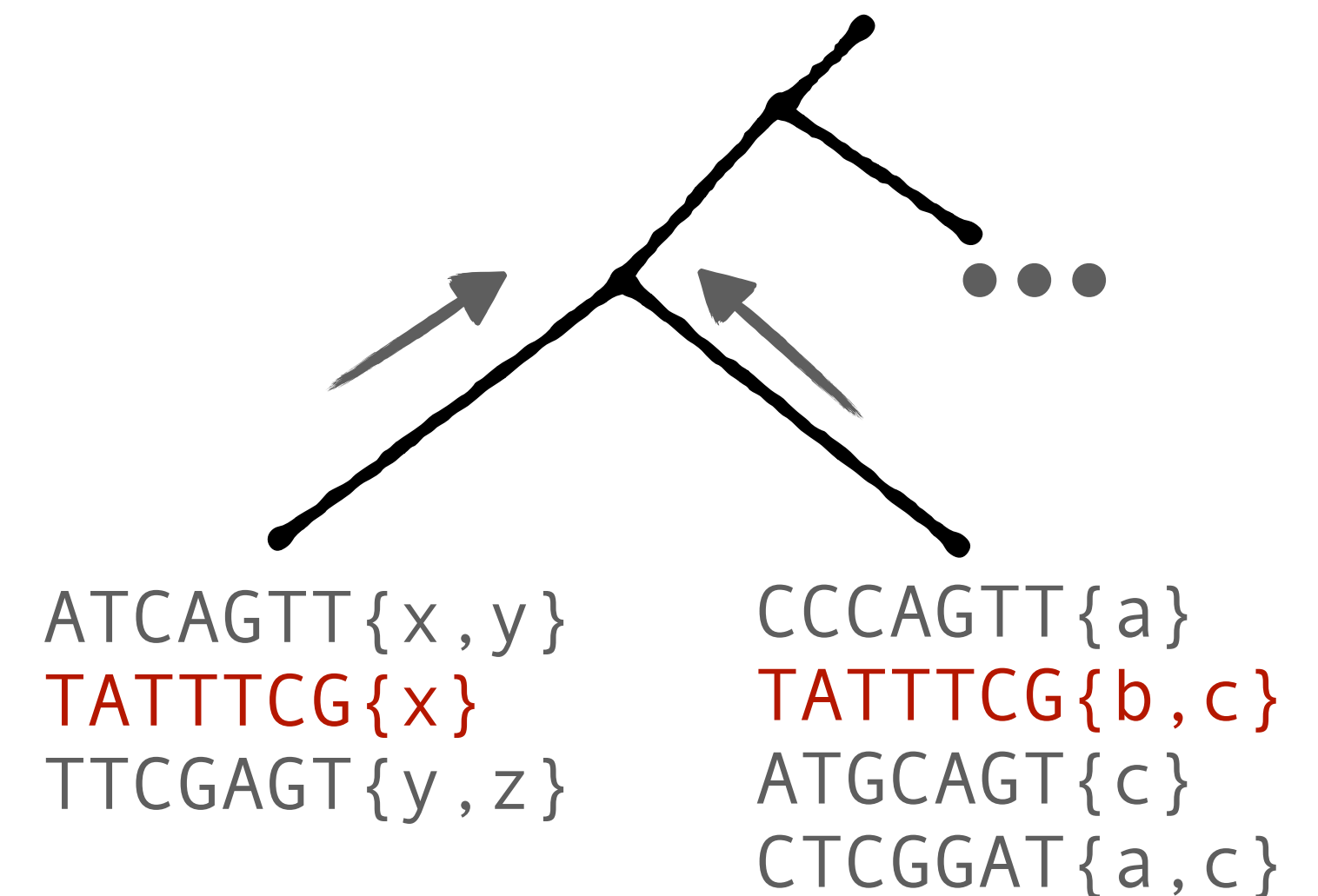


Constructing the color multi-tree

- Start with k -mer sets of all reference
- Initialize the multi-tree as unconnected singletons
- Label k -mers with singletons

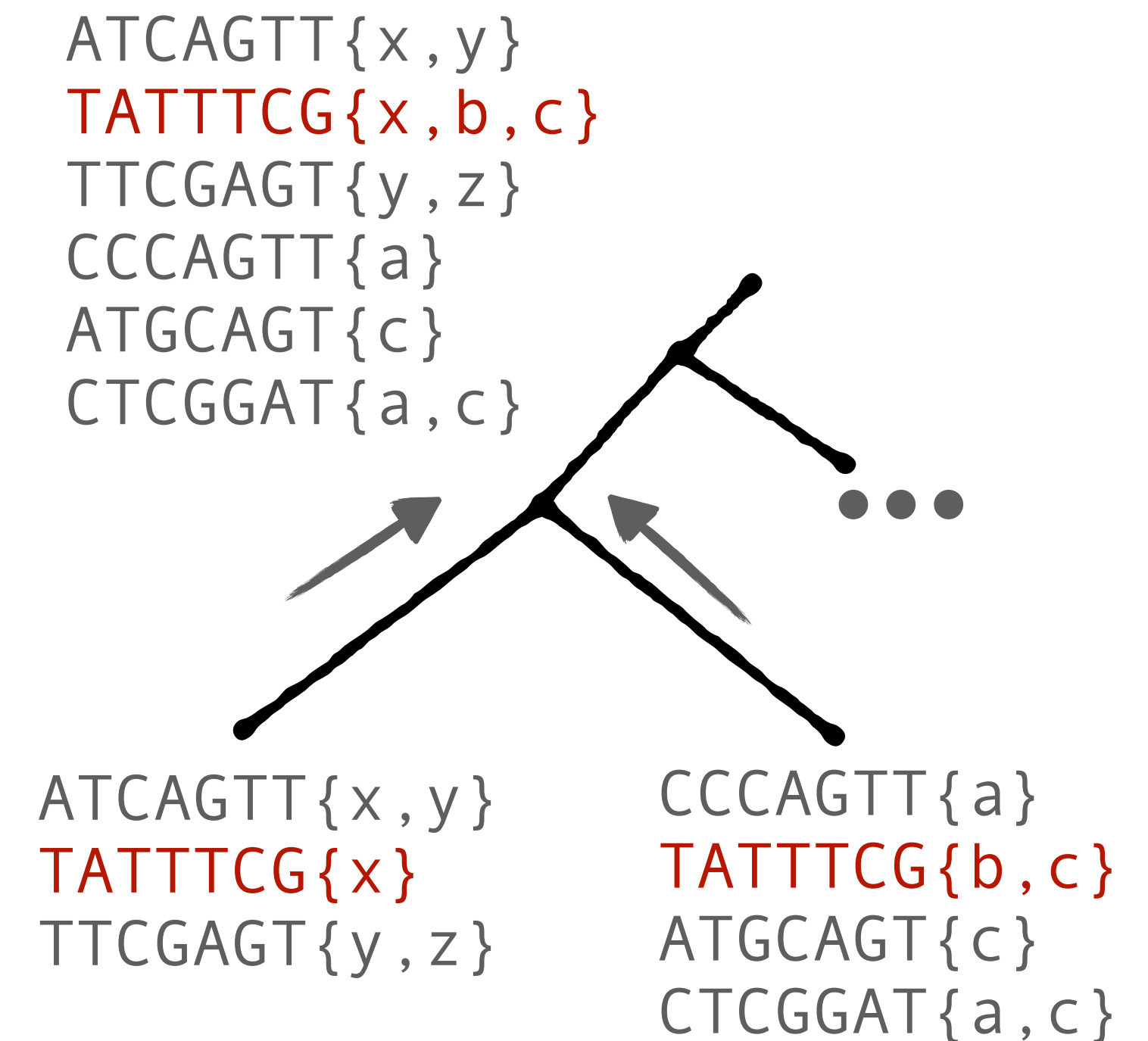
Constructing the color multi-tree

- Start with k -mer sets of all reference
- Initialize the multi-tree as unconnected singletons
- Label k -mers with singletons
- During the traversal, add a color for the union if a k -mer exist in both children and update its label



Constructing the color multi-tree

- Start with k -mer sets of all reference
- Initialize the multi-tree as unconnected singletons
- Label k -mers with singletons
- During the traversal, add a color for the union if a k -mer exist in both children and update its label

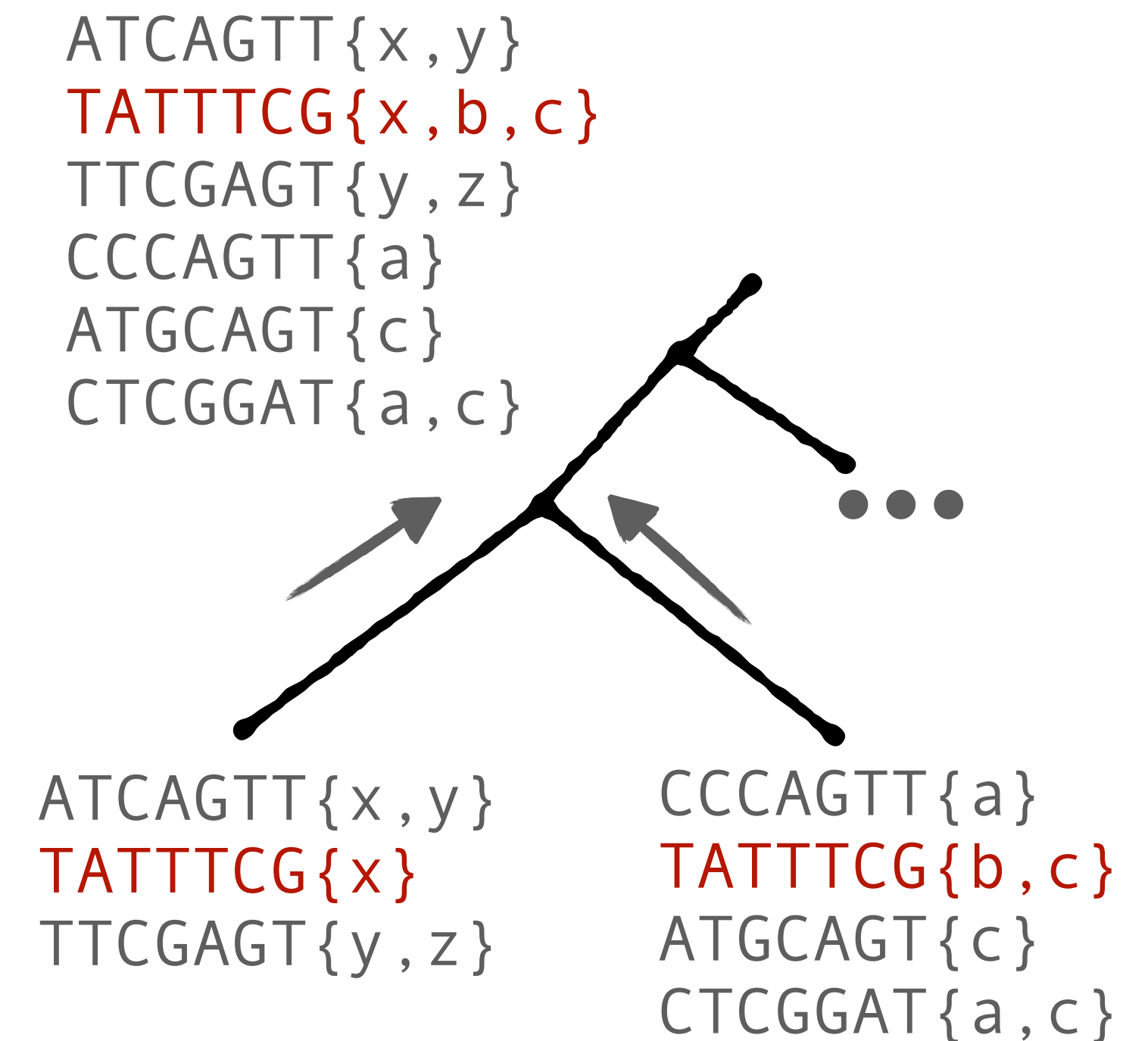


Constructing the color multi-tree

- Start with k -mer sets of all reference
- Initialize the multi-tree as unconnected singletons
- Label k -mers with singletons
- During the traversal, add a color for the union if a k -mer exist in both children and update its label

Challenges:

- How to intersections of large sets? **LSH partitions**
- How to test if a color is already added? **Abelian group hashing**
- How to represent set labels of k -mers? **Abelian group hashing**



Statistical distinguishability tests → placement

- Small differences may not be statistically meaningful
 - ▶ **test distinguishability**

likelihood-ratio test

w.r.t. the closest reference:

D : alternative distance

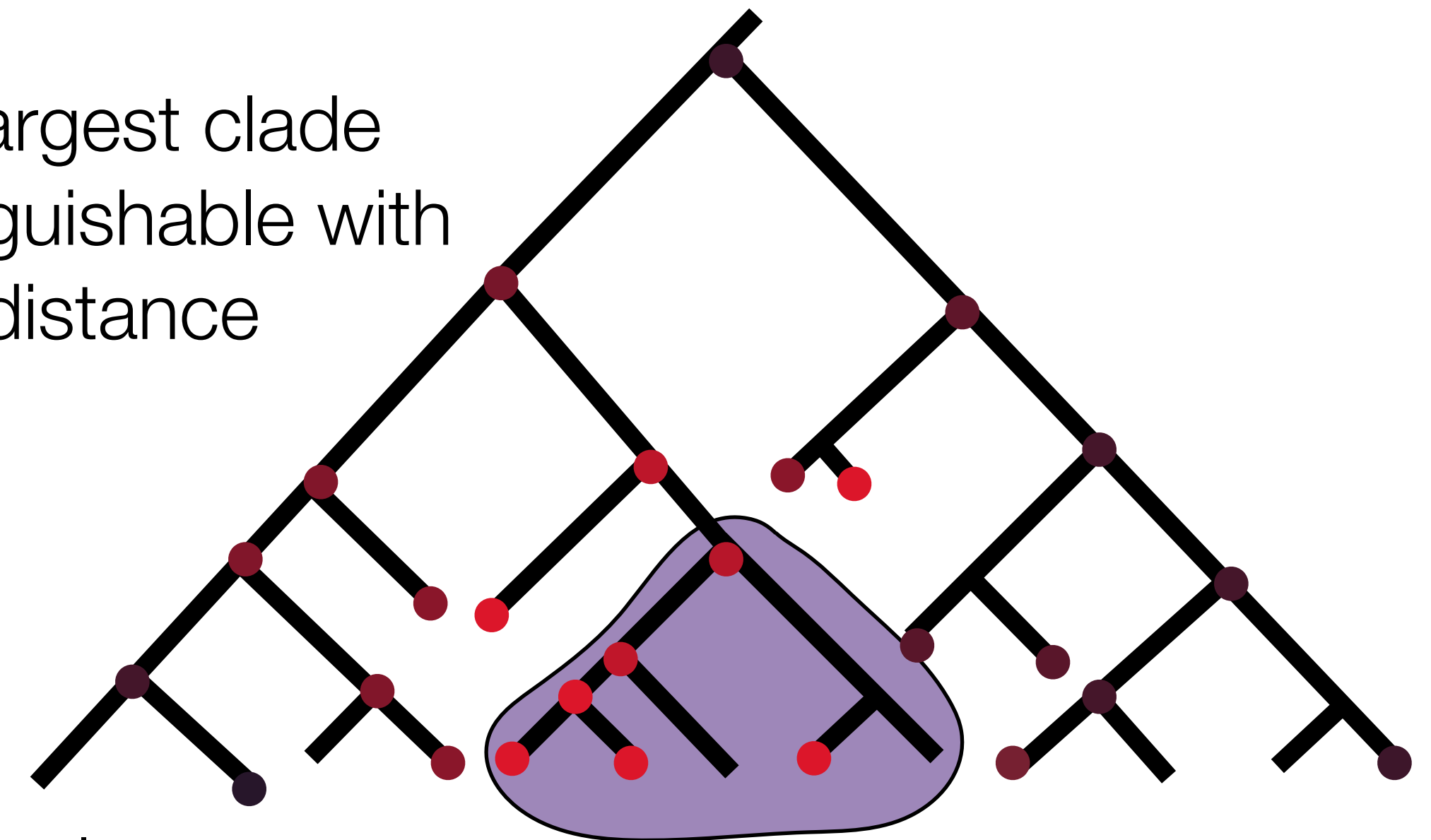
$$\lambda_{LR} = \frac{\mathcal{L}_{i^*}(D; k, h, \delta, u_{i^*}, \mathbf{v}_{i^*})}{\mathcal{L}_{i^*}(D^*; k, h, \delta, u_{i^*}, \mathbf{v}_{i^*})}$$

i^* : closest reference

$$\lambda_{LR} \sim \chi^2$$

- ▶ select a significance level
(default: $\alpha=90\%$)

place on the largest clade
that is indistinguishable with
the minimum distance



indistinguishable w.r.t.
the closest reference

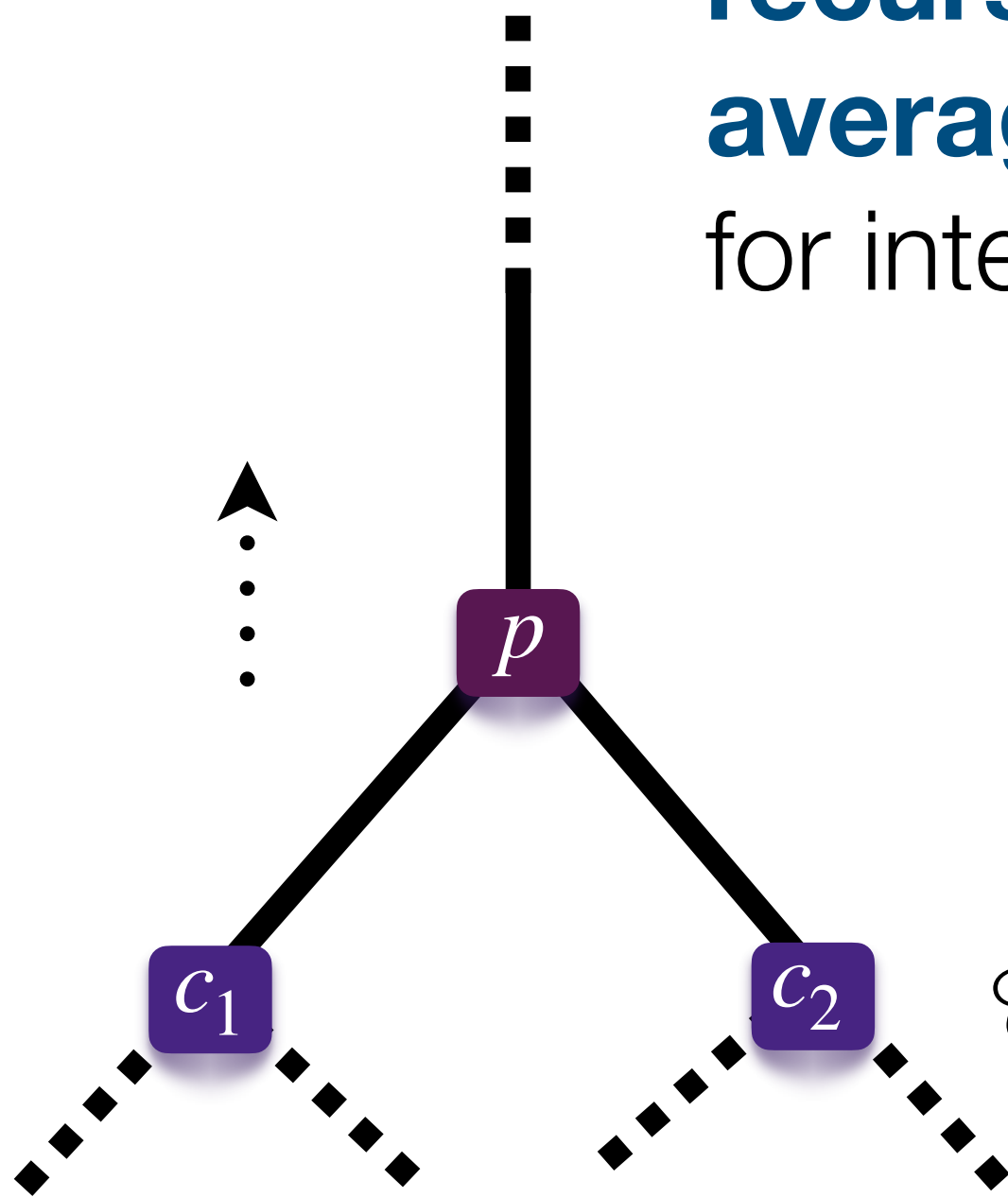
Defining clade distances & branch length agnostic placement

A notion of distance
for internal nodes:

**recursively compute
average** HD histograms
for internal nodes

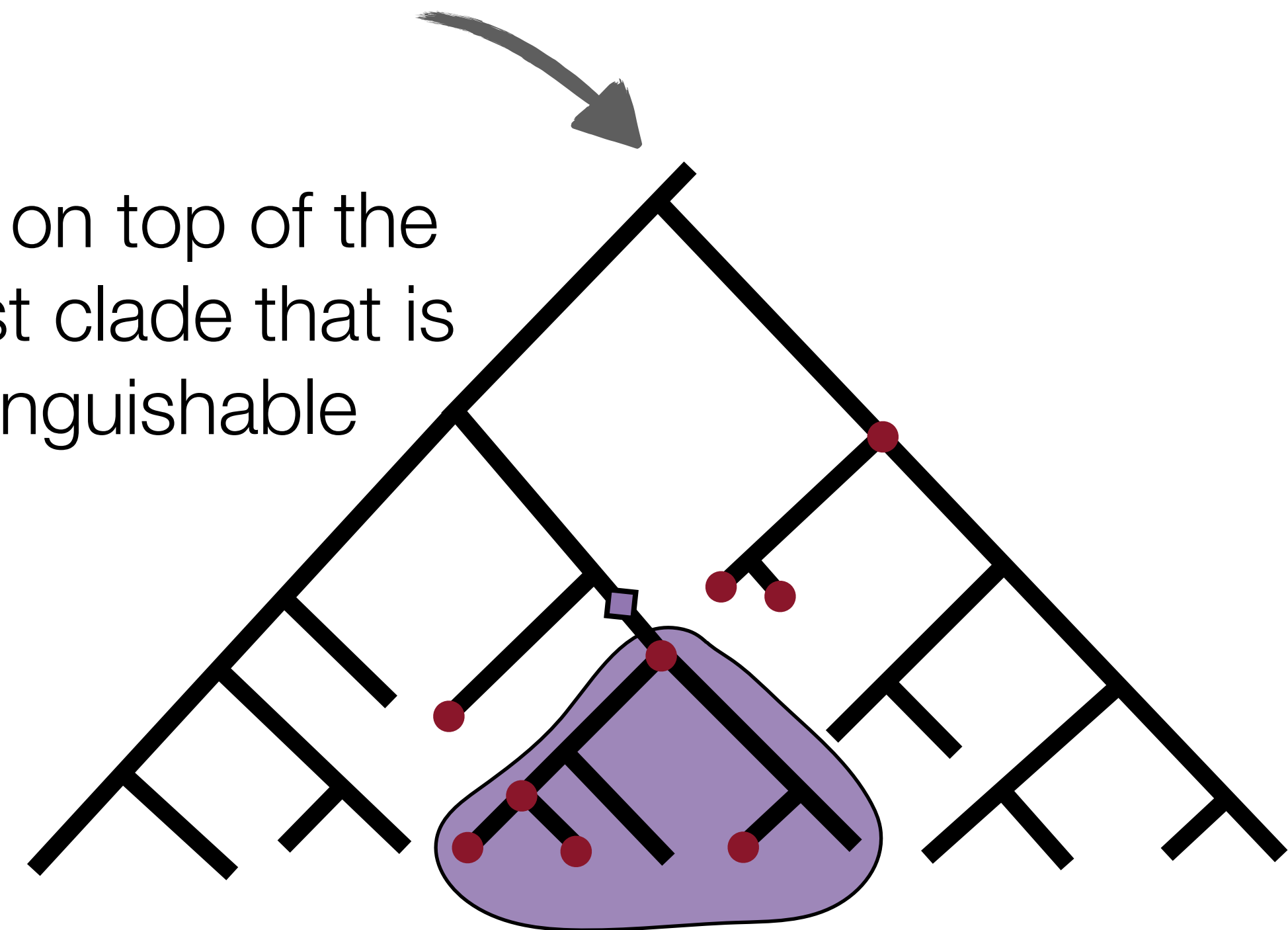
$$\mathbf{v}_p = \frac{\sum_{c \in \mathcal{C}(p)} \mathbf{v}_c}{|\mathcal{C}(p)|}$$

$\mathcal{C}(p)$: set of children of p , $\{c_1, c_2\}$



use the same likelihood model
and log-likelihood ratio test

place on top of the
largest clade that is
indistinguishable



● : indistinguishable w.r.t.
the closest reference

Dealing with uncertainty: statistically distinguishability

- short reads — **low signal**
- high distances — fewer matching k -mers
- small differences may not be statistically meaningful
 - ▶ **test distinguishability**

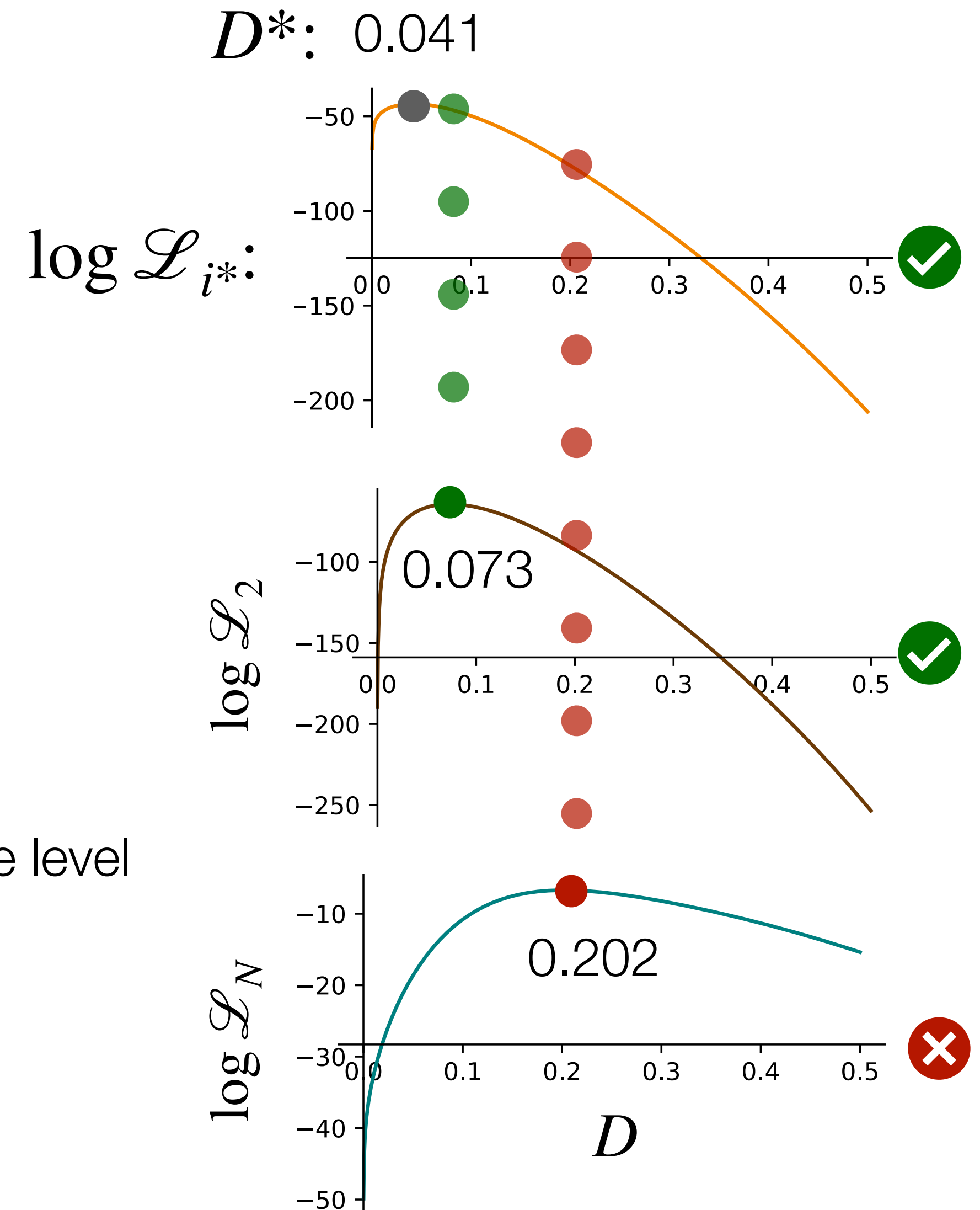
likelihood-ratio test
with the closest reference:

$$\lambda_{LR} = \frac{\mathcal{L}_{i^*}(D; k, h, \delta, u_{i^*}, \mathbf{v}_{i^*})}{\mathcal{L}_{i^*}(D^*; k, h, \delta, u_{i^*}, \mathbf{v}_{i^*})}$$

D : alternative distance
 i^* : closest reference

$$\lambda_{LR} \sim \chi^2$$

- ▶ select a significance level
(default: $\alpha=90\%$)



Indexing GORG-Tropics

12,715 genomes: 9.7GB index size

created in less than 30 minutes using `krepp index`

You can even use it on your personal computer!

Simply install via:

```
curl -k https://ter-trees.ucsd.edu/data/krepp/gorg-tropic-kreppref.tar  
# or from our AWS bucket  
aws s3 cp s3:kreppref/gorg-tropic-kreppref.tar ./  
  
tar -cvf gorg-tropic-kreppref.tar
```

We have a rich catalogue of reference indices with phylogenies: GTDB, Web of Life, RefSeq...

as large as
200K genomes

