Estimating read-genome distances with homologous *k*-mer matches

(and genome-wide phylogenetic placement)

Ali Sapci

8 April 2025

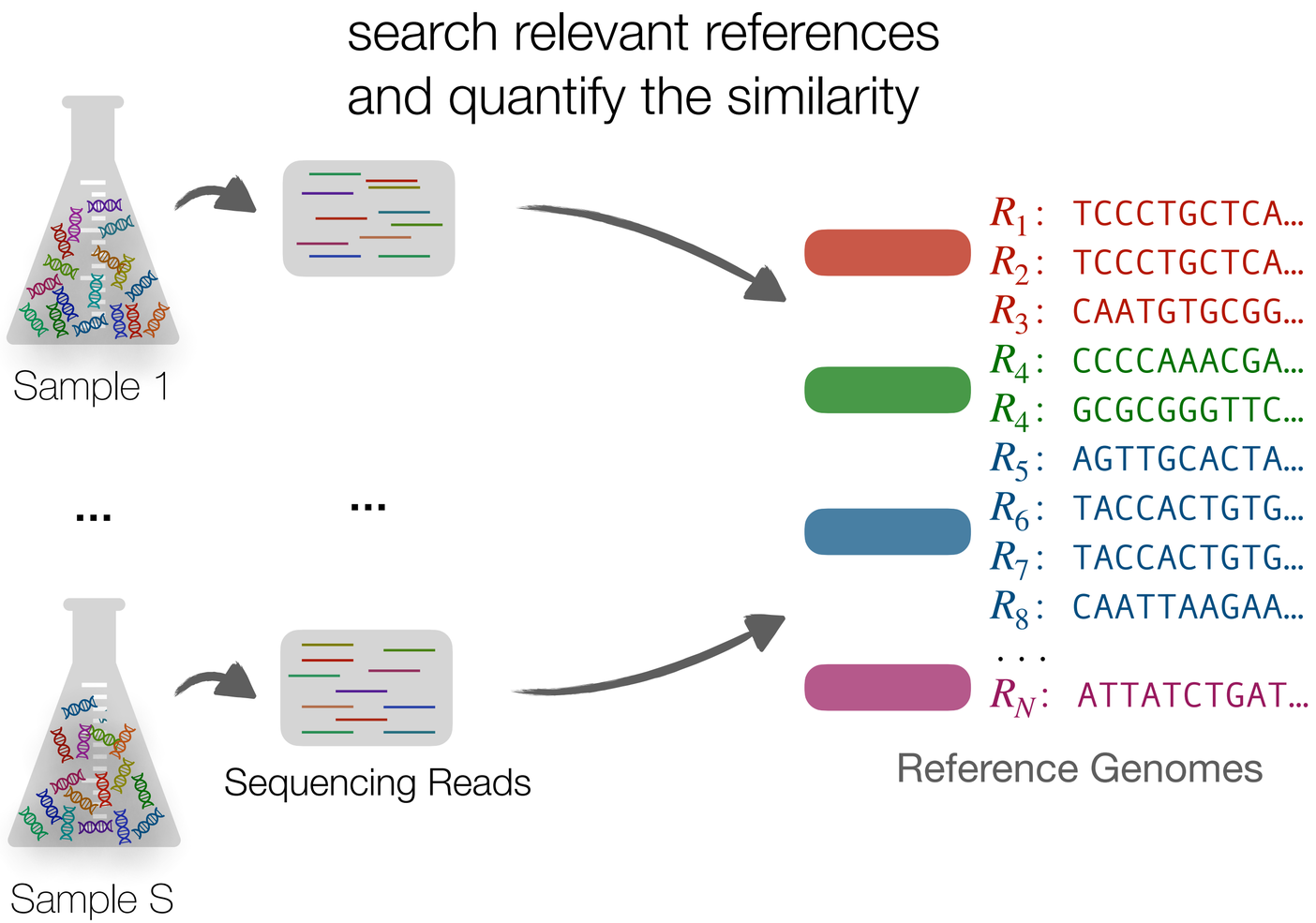
Identifying metagenomic sequences and comparing samples

Identifying metagenomic sequences and comparing samples

High-level goals:

- Analyze & monitor present microbial communities
- Compare microbial composition of samples
- Measure diversity and detect novel sequences

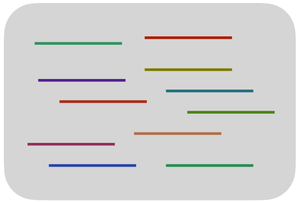
Identifying metagenomic sequences and comparing samples



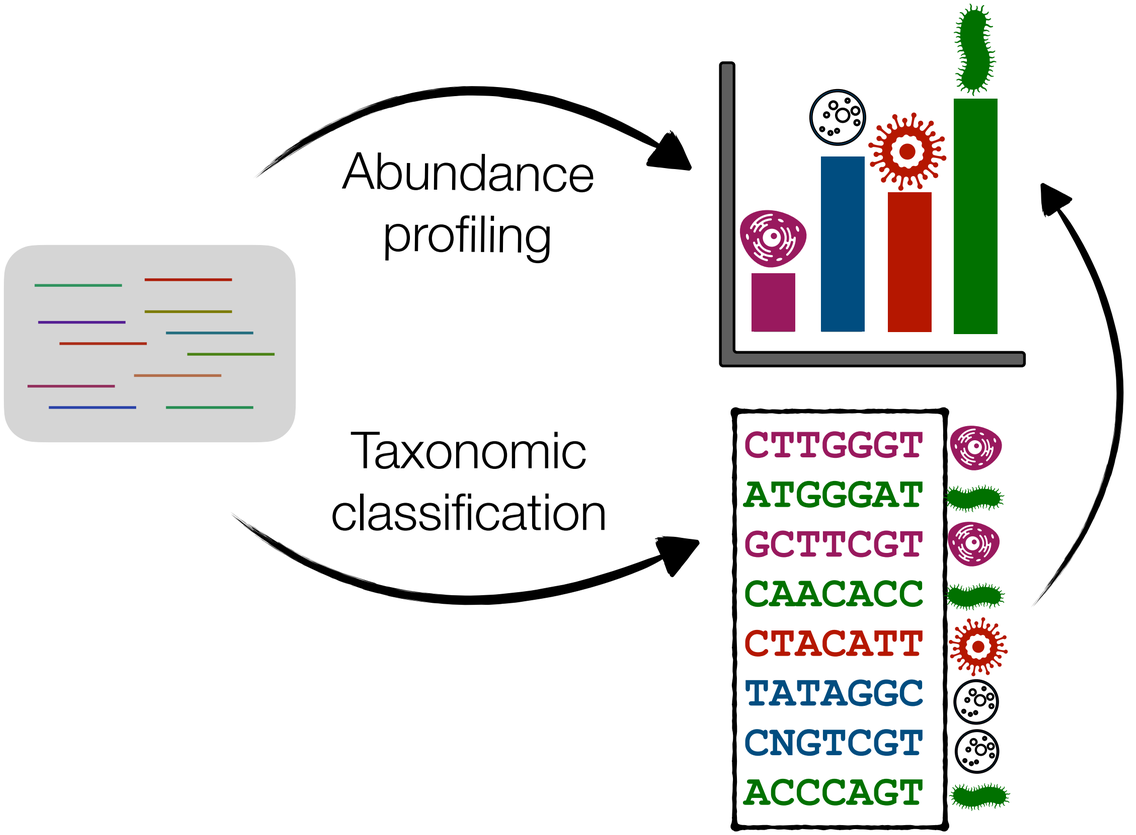
High-level goals:

- Analyze & monitor present microbial communities
- Compare microbial composition of samples
- Measure diversity and detect novel sequences

Poor man’s solution: taxonomic profiling

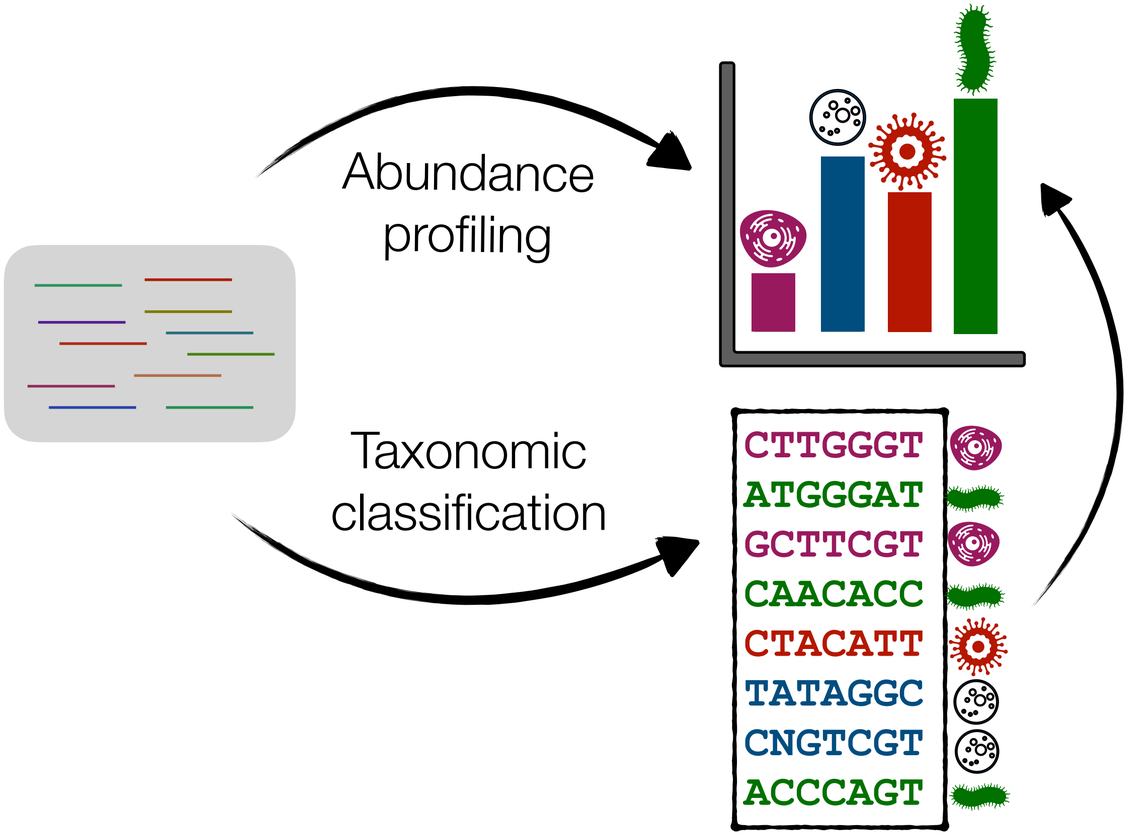


Poor man's solution: taxonomic profiling



Poor man's solution: taxonomic profiling

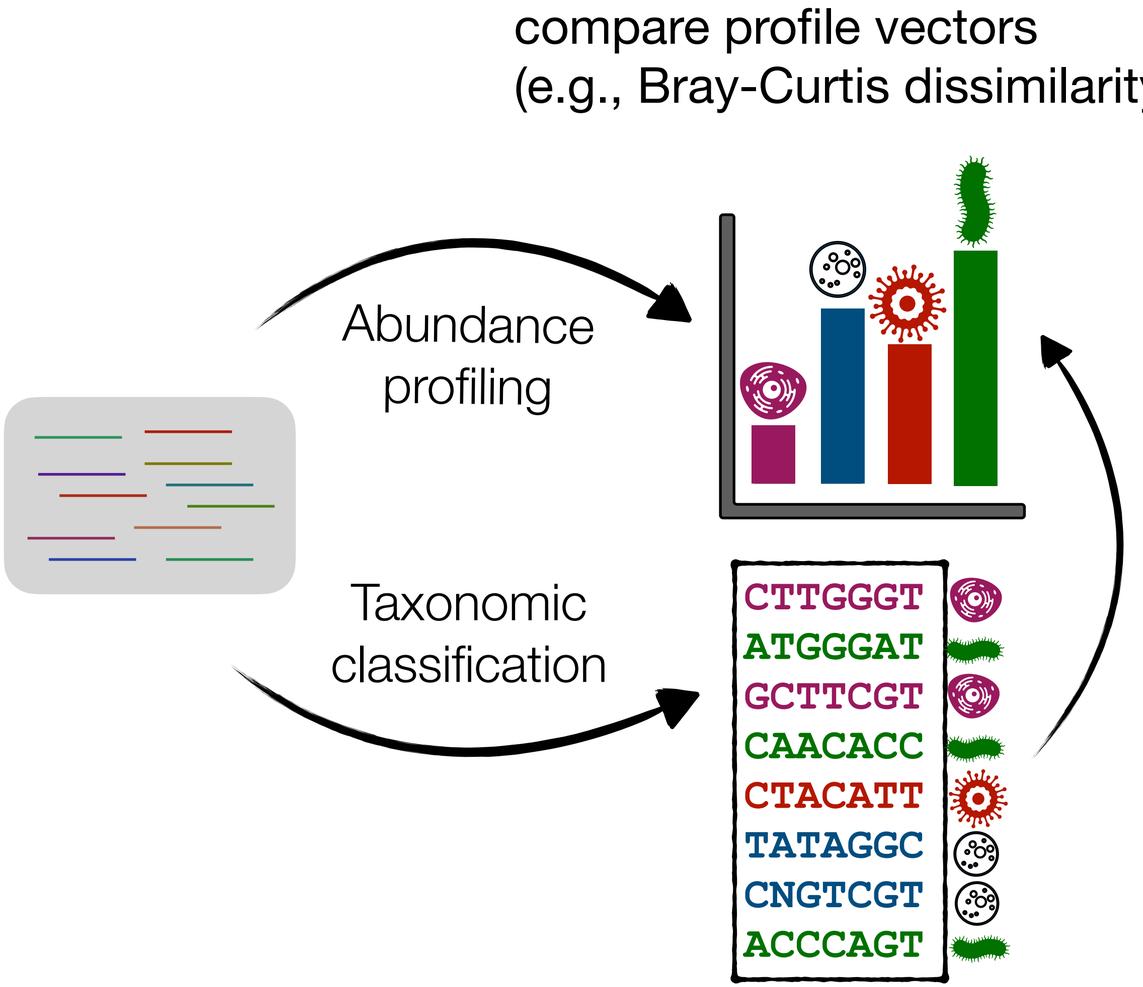
compare profile vectors
(e.g., Bray-Curtis dissimilarity)



Poor man's solution: taxonomic profiling

- **Fast and scalable methods**
  based on *k*-mer search

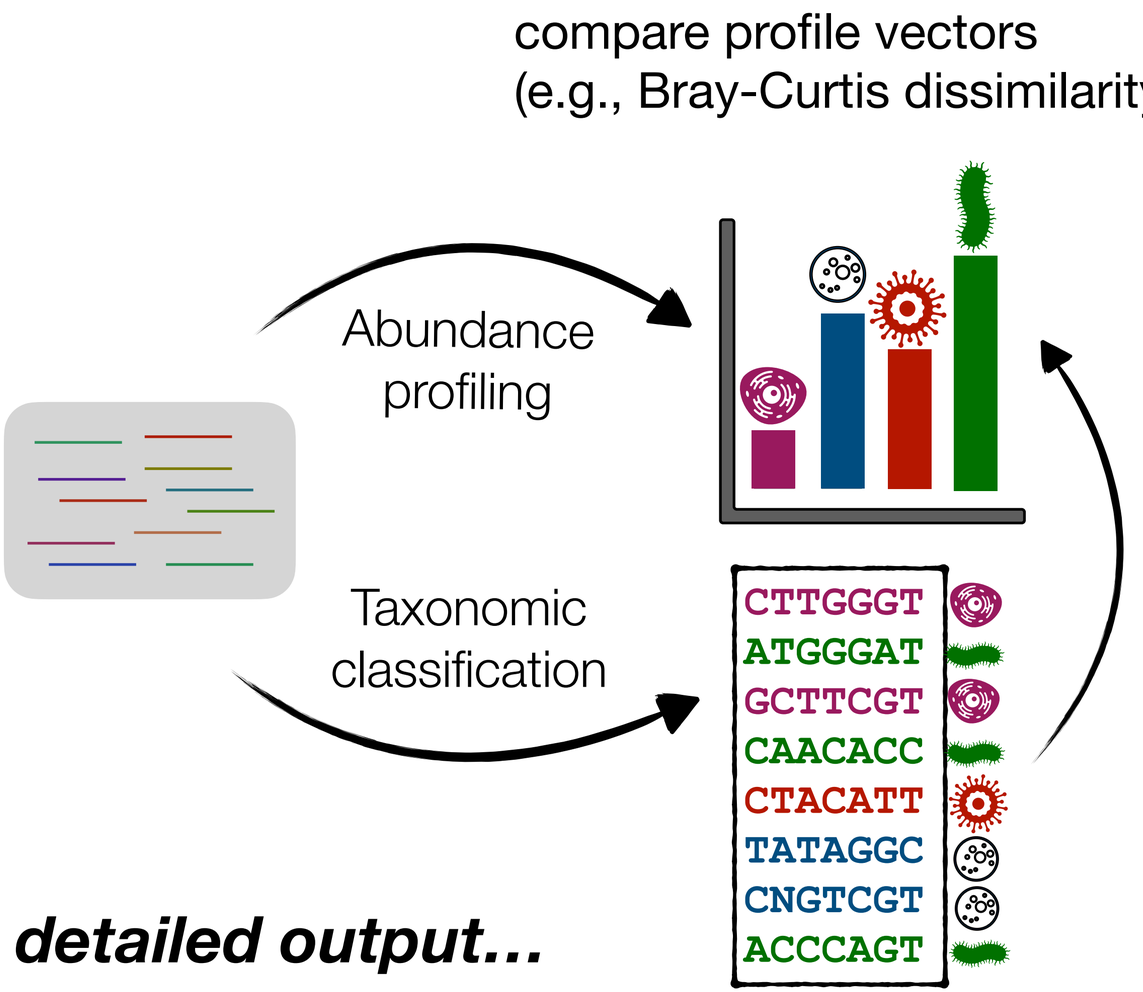
- **Limited:**
  - has low resolution
  - often ambiguous (e.g., HGT)
  - omits within-group diversity
  - no notion of *distance*, novelty?



Poor man's solution: taxonomic profiling

- **Fast and scalable methods**
  based on *k*-mer search

- **Limited:**
  - has low resolution
  - often ambiguous (e.g., HGT)
  - omits within-group diversity
  - no notion of *distance*, novelty?



We need a more detailed output…

What alternatives do we have?

What alternatives do we have?

- **Sample-wide containment analysis (using MinHash)?**

What alternatives do we have?

- **Sample-wide containment analysis (using MinHash)?**
  - higher resolution compared to taxonomic profiles **(+)**
  - no assignments for individual sequences **(-)**
  - no distances btw. queries and references **(-)**

What alternatives do we have?

- **Sample-wide containment analysis (using MinHash)?**
  - higher resolution compared to taxonomic profiles **(+)**
  - no assignments for individual sequences **(-)**
  - no distances btw. queries and references **(-)**

- **Using marker genes?**

What alternatives do we have?

- **Sample-wide containment analysis (using MinHash)?**
  - higher resolution compared to taxonomic profiles **(+)**
  - no assignments for individual sequences **(-)**
  - no distances btw. queries and references **(-)**

- **Using marker genes?**
  - aligning to a MSA is possible & and distances **(+)**
  - limited and potentially biased **(-)**
  - inability to capture novel sequences and queries **(-)**

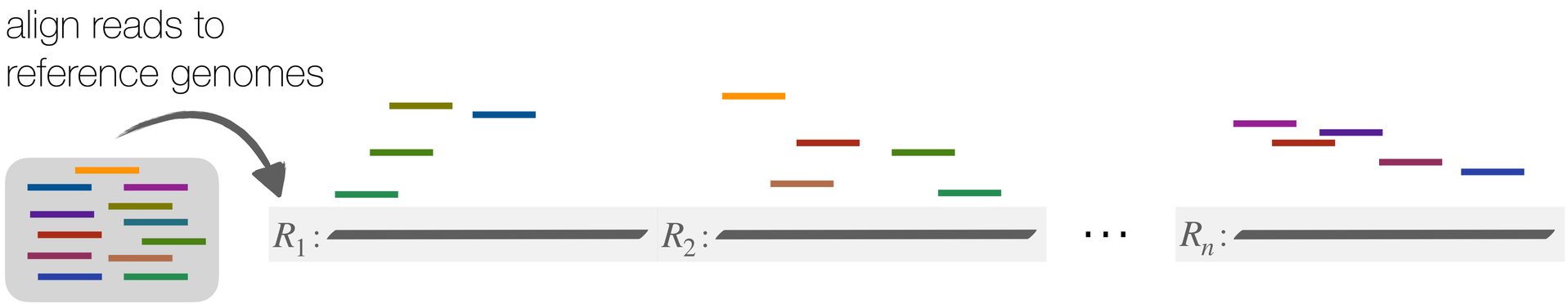
What alternatives do we have?

- **Sample-wide containment analysis (using MinHash)?**
  - higher resolution compared to taxonomic profiles **(+)**
  - no assignments for individual sequences **(-)**
  - no distances btw. queries and references **(-)**

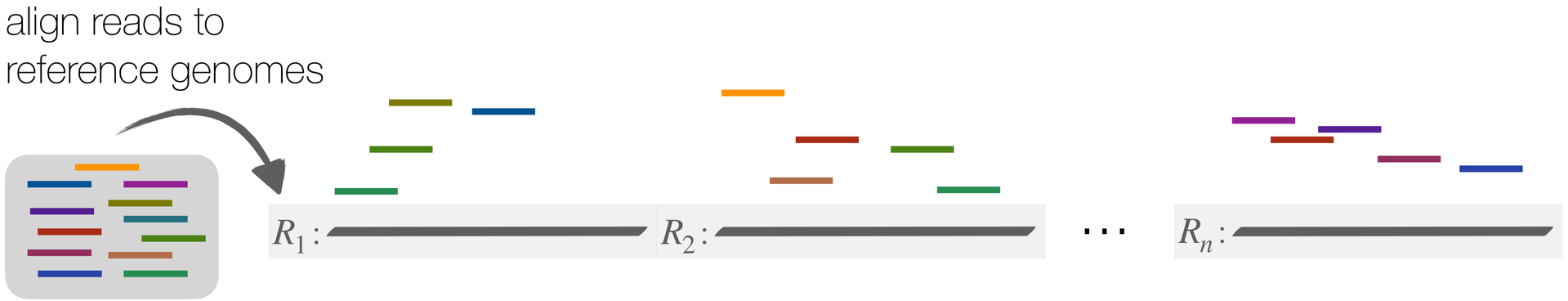
- **Using marker genes?**
  - aligning to a MSA is possible & and distances **(+)**
  - limited and potentially biased **(-)**
  - inability to capture novel sequences and queries **(-)**

- **Aligning reads to all references?**

Aligning sequences of unknown origins

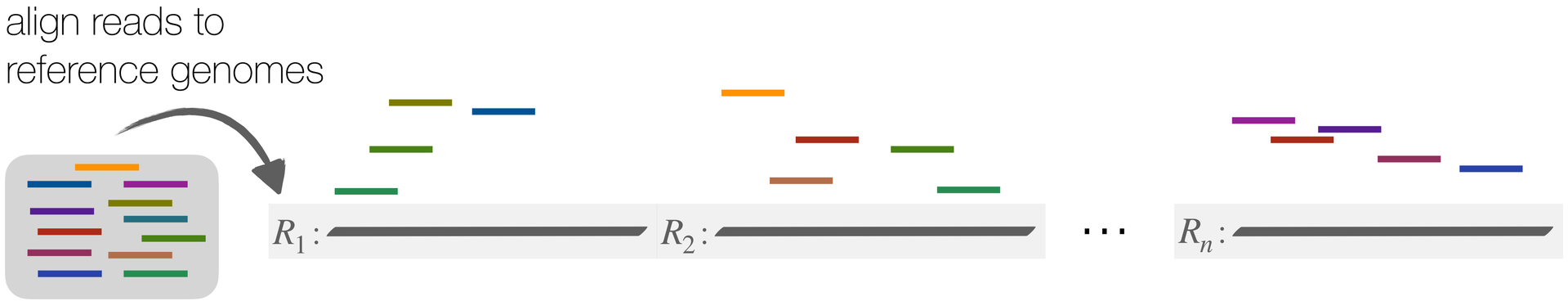


Aligning sequences of unknown origins



(+) quantifying similarity — as detailed as it gets (in fact redundant)

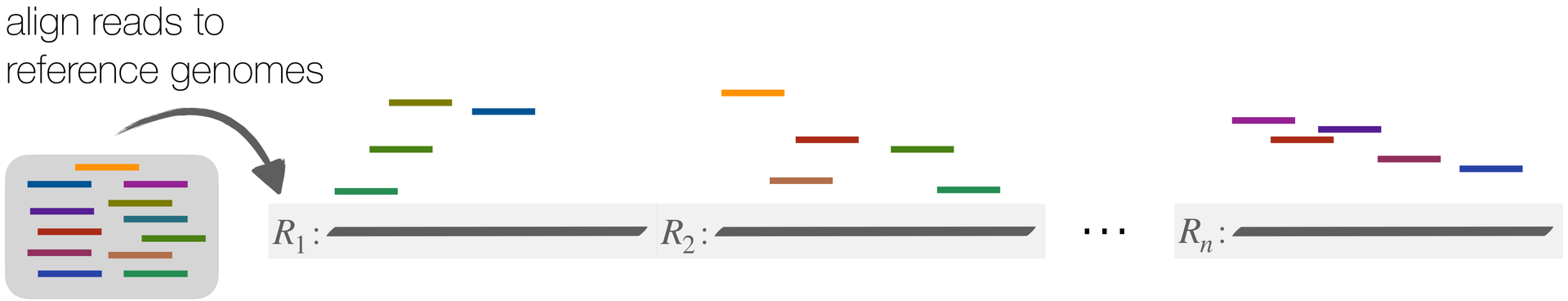
Aligning sequences of unknown origins



(+) quantifying similarity — as detailed as it gets (in fact redundant)

(+) placing sequences on a phylogeny — not new but limited

Aligning sequences of unknown origins

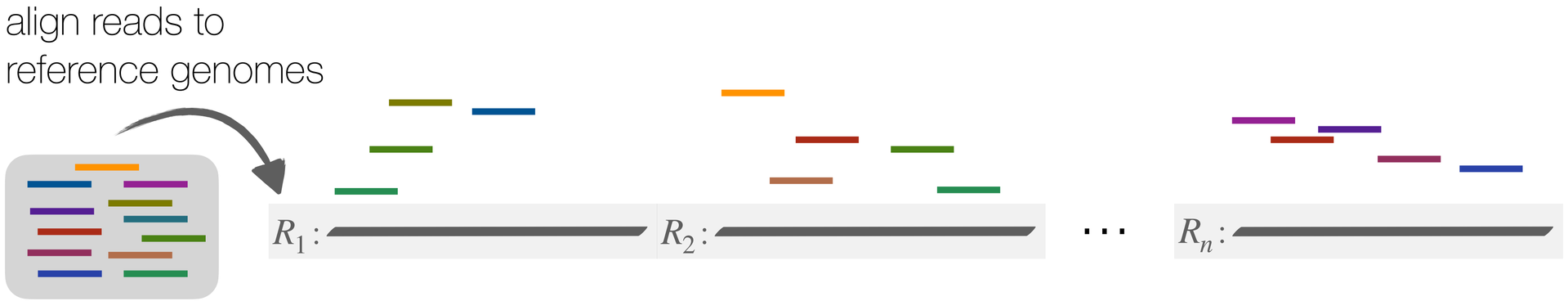


(+) quantifying similarity — as detailed as it gets (in fact redundant)

(+) placing sequences on a phylogeny — not new but limited

(-) not scalable for large n, even with efficient indexes

Aligning sequences of unknown origins



(+) quantifying similarity — as detailed as it gets (in fact redundant)

(+) placing sequences on a phylogeny — not new but limited

(-) not scalable for large n, even with efficient indexes

(-) not good for higher distances & novel sequences (>15%)

Our goal: the best of both worlds

Our goal: the best of both worlds

Given a query sequence:

Our goal: the best of both worlds

Given a query sequence:

- estimate accurate distances for references
  - scale to modern references (>100K microbial genomes)
  - akin to alignment — but without solving the difficult problem

Our goal: the best of both worlds

Given a query sequence:

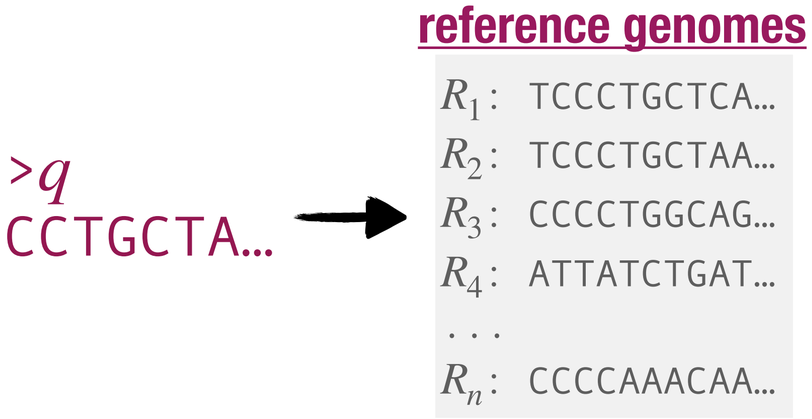
- estimate accurate distances for references
  - scale to modern references (>100K microbial genomes)
  - akin to alignment — but without solving the difficult problem
- place it on a large reference phylogeny (>100,000)
  - always relies on marker genes — with one exception

Problem statement

Problem statement

Given:

- query sequence q
- set of references $\mathscr{R} = \{R_1, \ldots, R_n\}$
- a backbone phylogeny T



Problem statement

Given:

- query sequence q
- set of references $\mathscr{R} = \{R_1, \ldots, R_n\}$
- a backbone phylogeny T

Problem statement

Given:

- query sequence q
- set of references $\mathscr{R} = \{R_1, \ldots, R_n\}$
- a backbone phylogeny T

Interpretation of the distances:

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

CCTGCTA...
...AGTTATCCCTGCTCA...

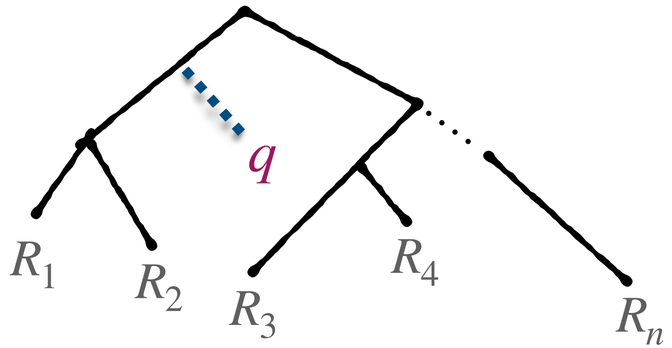
$ii)$ $\mathbb{E}[d(q, R)] \approx 1 - \text{ANI}(Q, R)$
where Q is the source genome of q

Problem statement

Given:

- query sequence q
- set of references $\mathscr{R} = \{R_1, \ldots, R_n\}$
- a backbone phylogeny T

Phylogenetic placement:



>q
CCTGCTA...

reference genomes

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

distance estimates

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

Interpretation of the distances:

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

CCTGCTA...
...AGTTATCCCTGCTCA...

ii) $\mathbb{E}[d(q, R)] \approx 1 - \mathrm{ANI}(Q, R)$
where Q is the source genome of q

Outline of the method & sub-tasks we need to solve

krepp: k-mer-based read phylogenetic placement

Outline of the method & sub-tasks we need to solve

krepp: k-mer-based read phylogenetic placement

I. Going beyond exact *k*-mer search & finding homologous *k*-mers

Outline of the method & sub-tasks we need to solve

krepp: k-mer-based read phylogenetic placement

I. Going beyond exact *k*-mer search & finding homologous *k*-mers

II. An efficient mapping between *k*-mers and reference genomes

Outline of the method & sub-tasks we need to solve

krepp: k-mer-based read phylogenetic placement

I. Going beyond exact *k*-mer search & finding homologous *k*-mers

II. An efficient mapping between *k*-mers and reference genomes

III. Modeling Hamming distances of *k*-mer matches to estimate distances

Outline of the method & sub-tasks we need to solve

krepp: k-mer-based read phylogenetic placement

I. Going beyond exact *k*-mer search & finding homologous *k*-mers

II. An efficient mapping between *k*-mers and reference genomes

III. Modeling Hamming distances of *k*-mer matches to estimate distances

IV. Finding a placement on a reference tree that best explains distances

Computing Hamming distances w/ locality sensitive hashing

Computing Hamming distances w/ locality sensitive hashing

reference *k*-mers

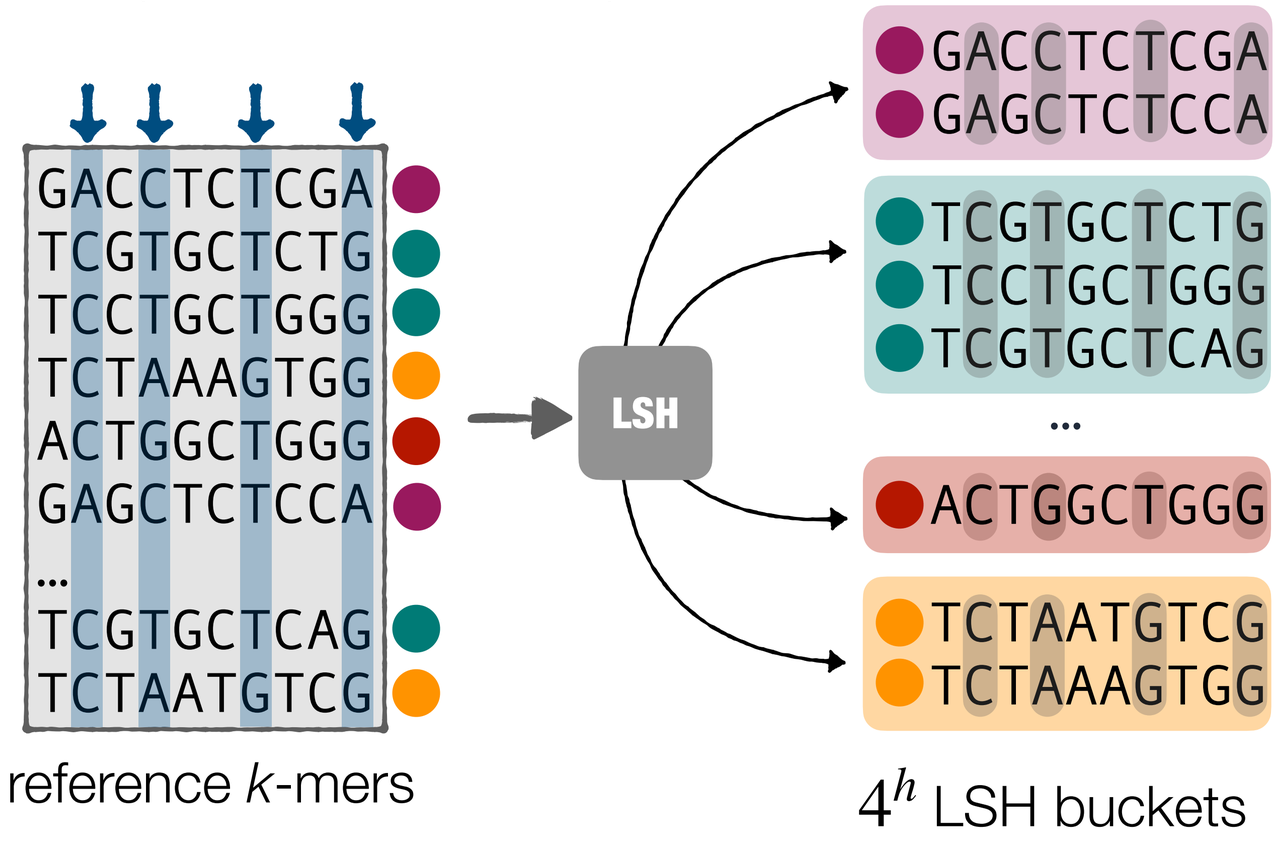
Computing Hamming distances w/ locality sensitive hashing

Select h random but fixed positions (default h: 14, k: 29)

reference *k*-mers

Computing Hamming distances w/ locality sensitive hashing

Select h random but fixed positions (default h: 14, k: 29)

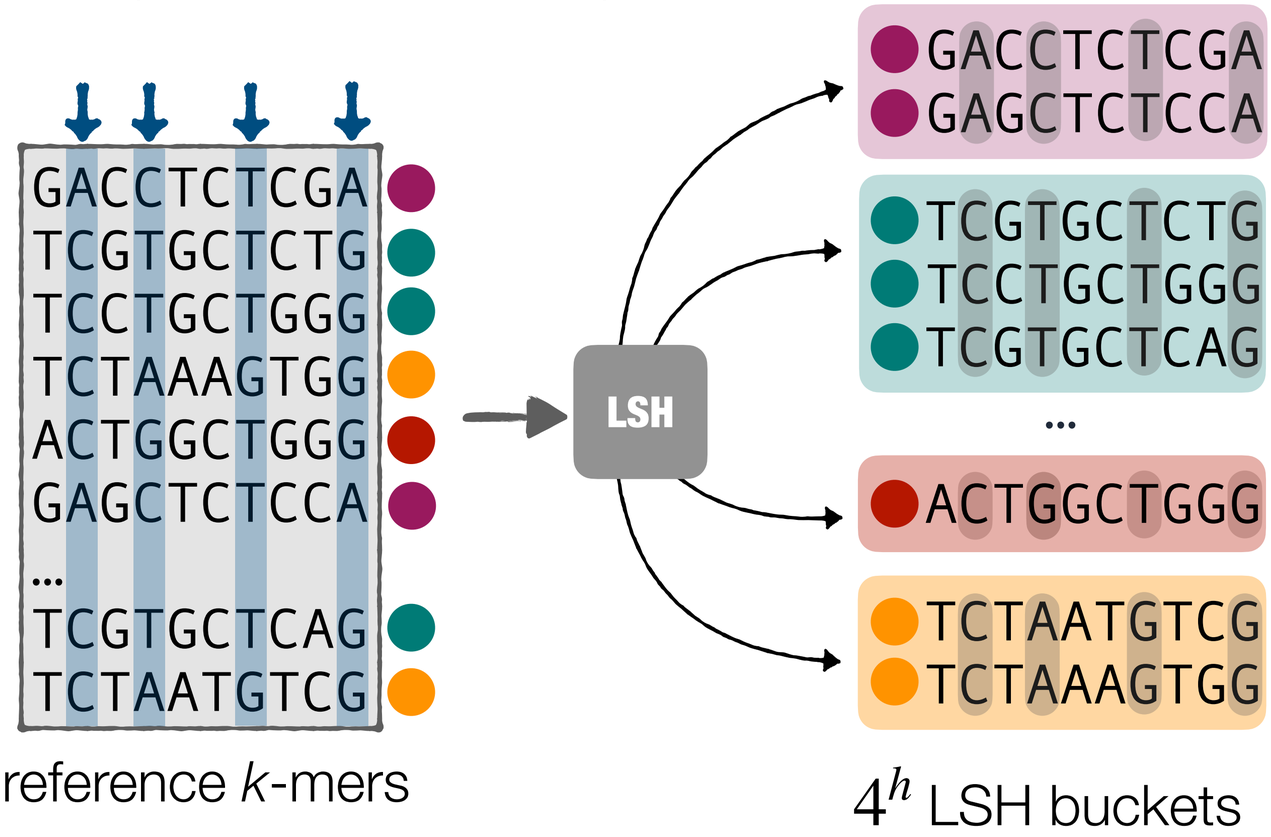


reference *k*-mers

4^h LSH buckets

Computing Hamming distances w/ locality sensitive hashing

Select h random but fixed positions (default h: 14, k: 29)



reference *k*-mers

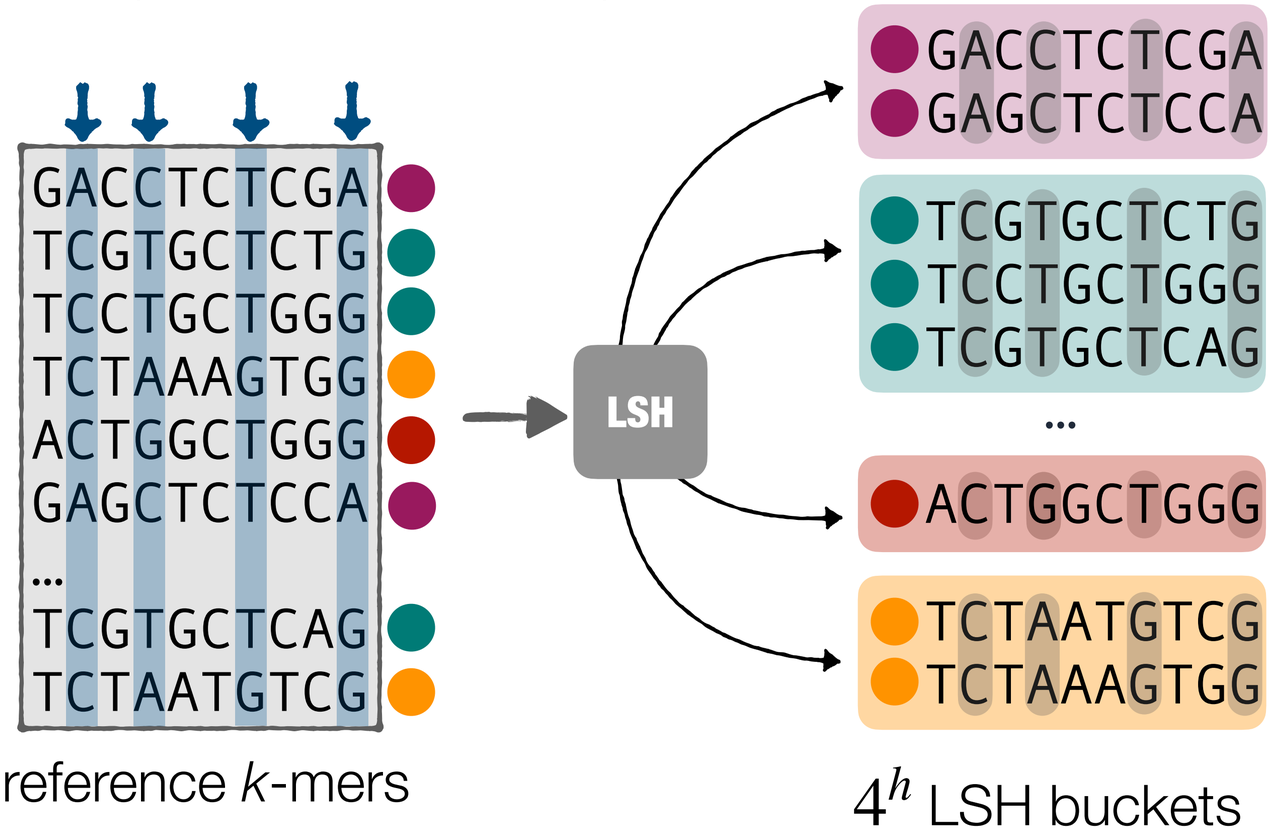
4^h LSH buckets

Given a query *k*-mer

ACCTGCTGGG

Computing Hamming distances w/ locality sensitive hashing

Select h random but fixed positions (default h: 14, k: 29)



reference *k*-mers

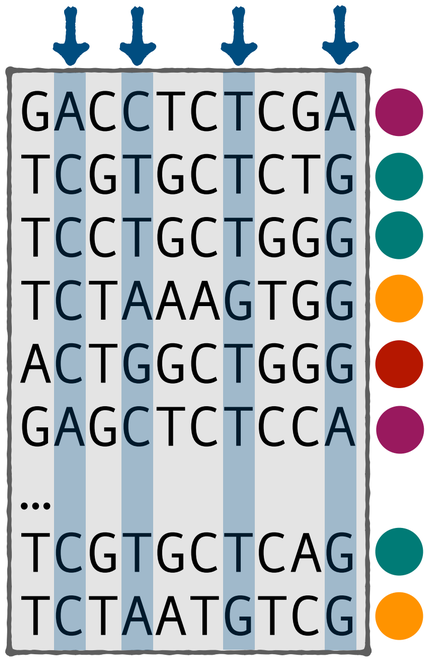
4^h LSH buckets

Given a query *k*-mer

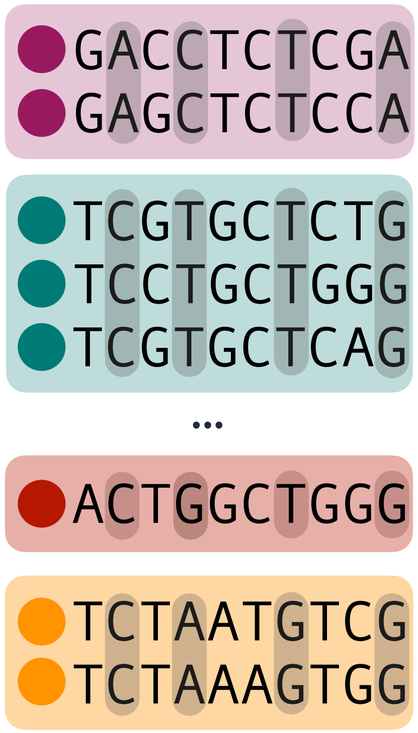
ACCTGCTGGG

Computing Hamming distances w/ locality sensitive hashing

Select h random but fixed positions (default h: 14, k: 29)



reference *k*-mers



4^h LSH buckets

Given a query *k*-mer

ACCTGCTGGG

HD

4

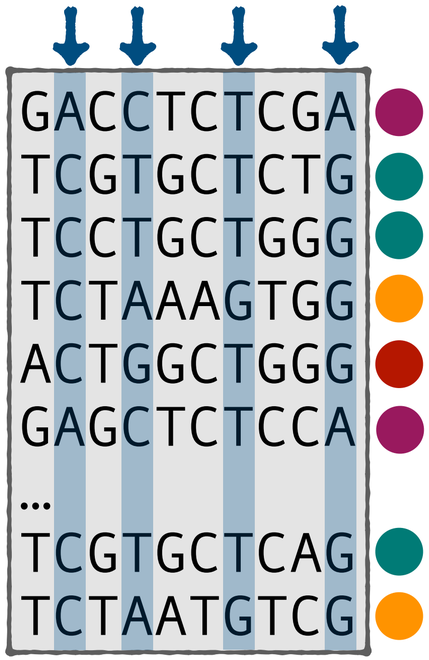
1

4

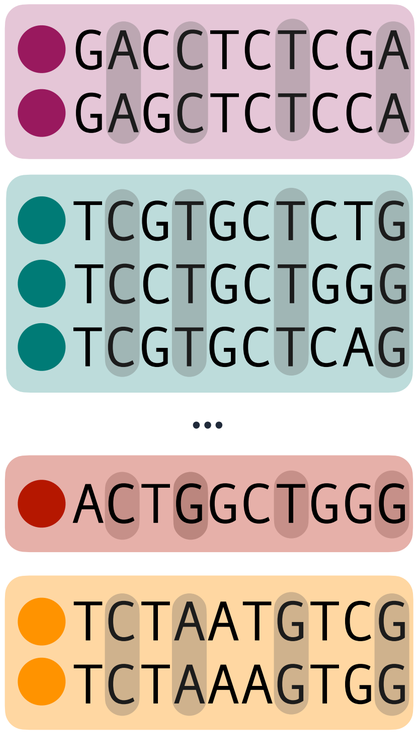
? FN at HD=2

Computing Hamming distances w/ locality sensitive hashing

Select h random but fixed positions (default h: 14, k: 29)



reference *k*-mers



4^h LSH buckets

HD

4

1

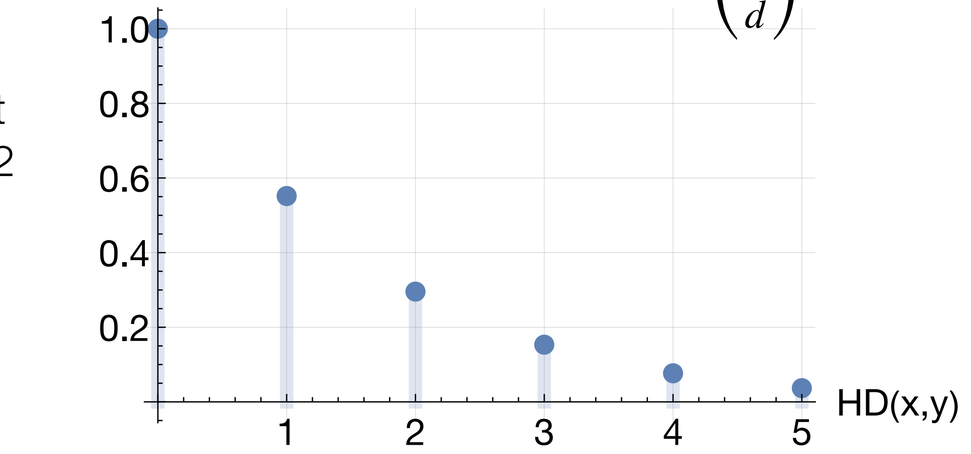
4

? FN at HD=2

Given a query *k*-mer

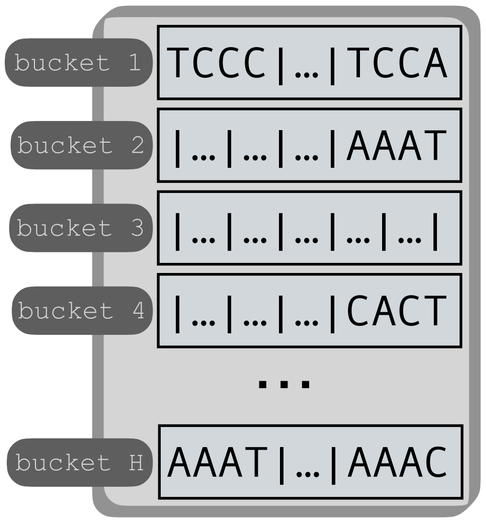
ACCTGCTGGG

match at HD=d with probability: $\frac{\binom{k-h}{d}}{\binom{k}{d}}$



Mapping indexed k-mers to reference genomes

Mapping indexed k-mers to reference genomes



LSH index

R_1

R_2

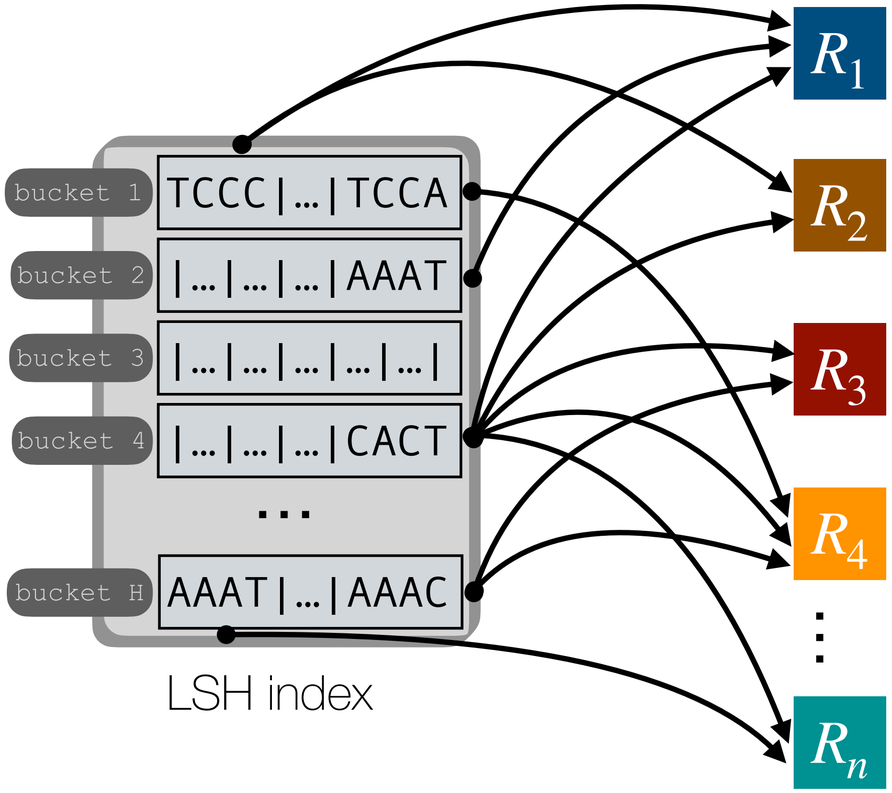
R_3

R_4

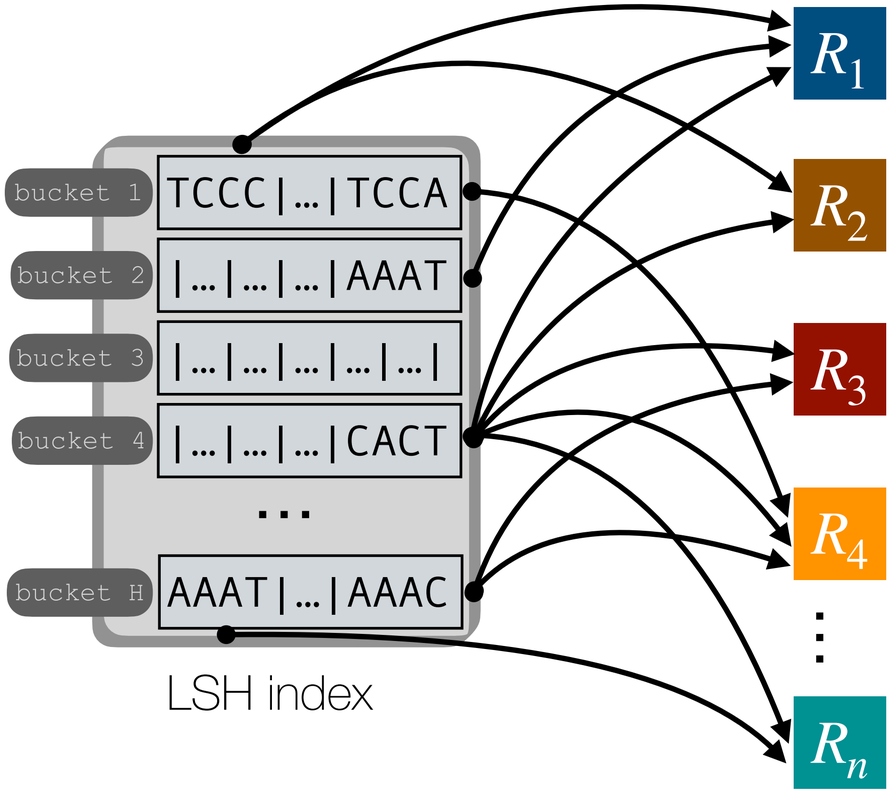
⋮

R_n

Mapping indexed k-mers to reference genomes



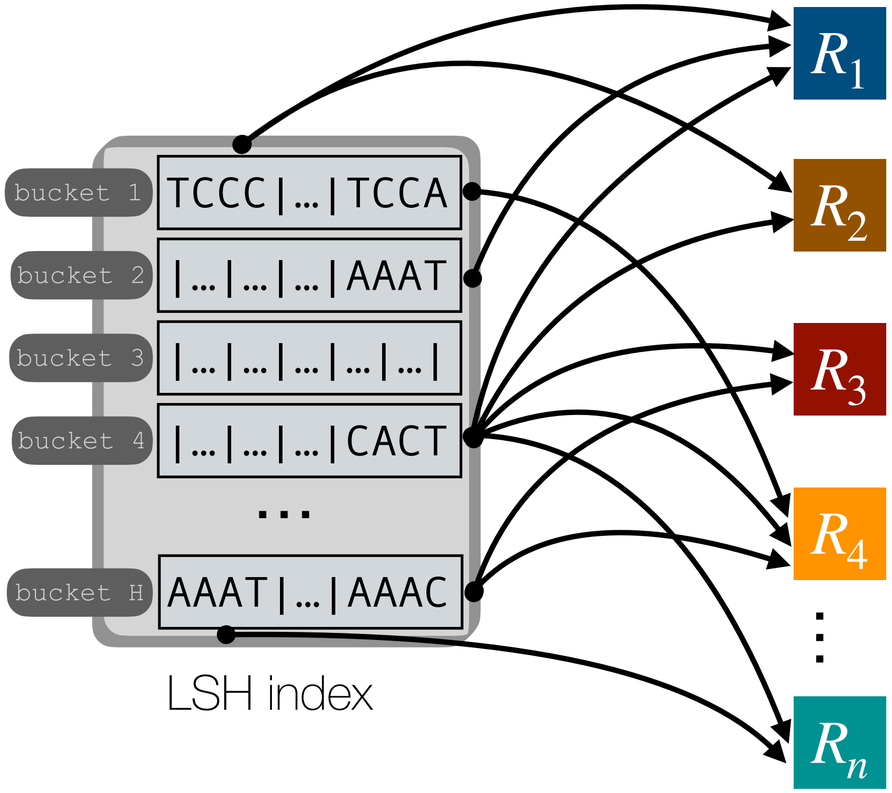
Mapping indexed k-mers to reference genomes



well studied **colored *k*-mer** problem
often represented as a sparse matrix

color: a subset of references
(including singletons)

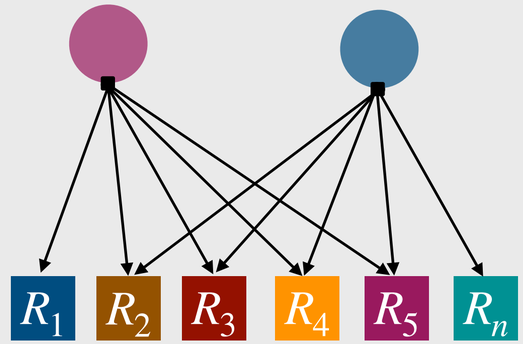
Mapping indexed k-mers to reference genomes



well studied **colored *k*-mer** problem
often represented as a sparse matrix

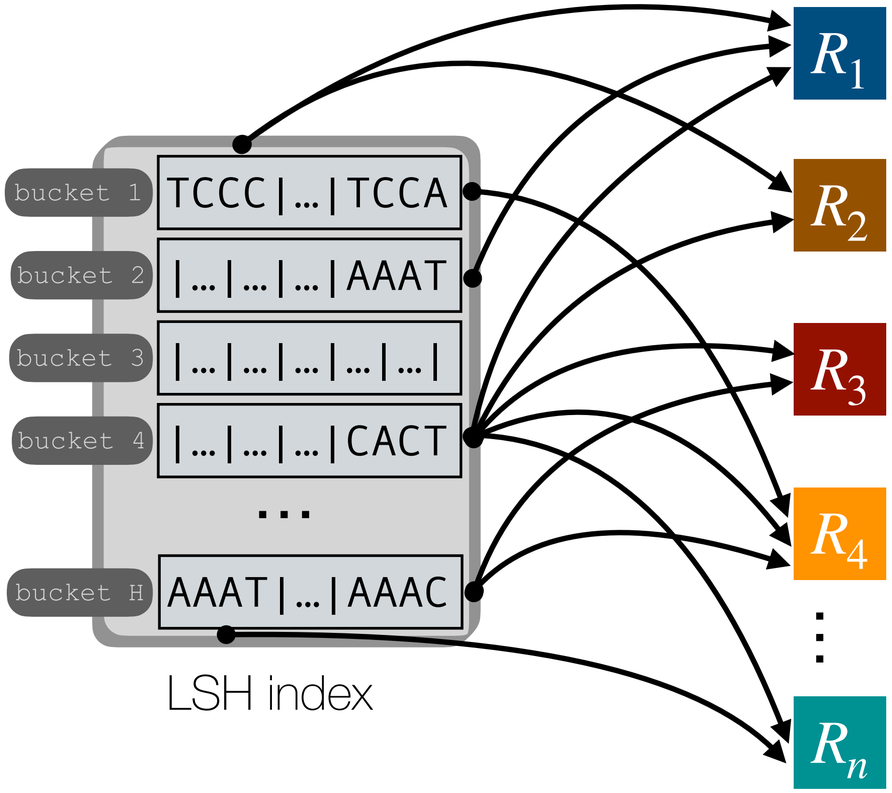
color: a subset of references
(including singletons)

$$|V| + |E| = 18$$



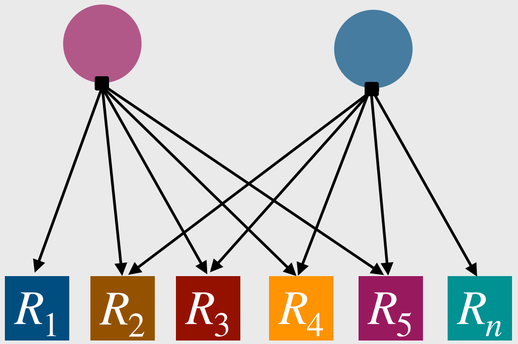
A directed bipartite graph

Mapping indexed k-mers to reference genomes



Minimize $|V| + |E|$ **?**

$$|V| + |E| = 18$$

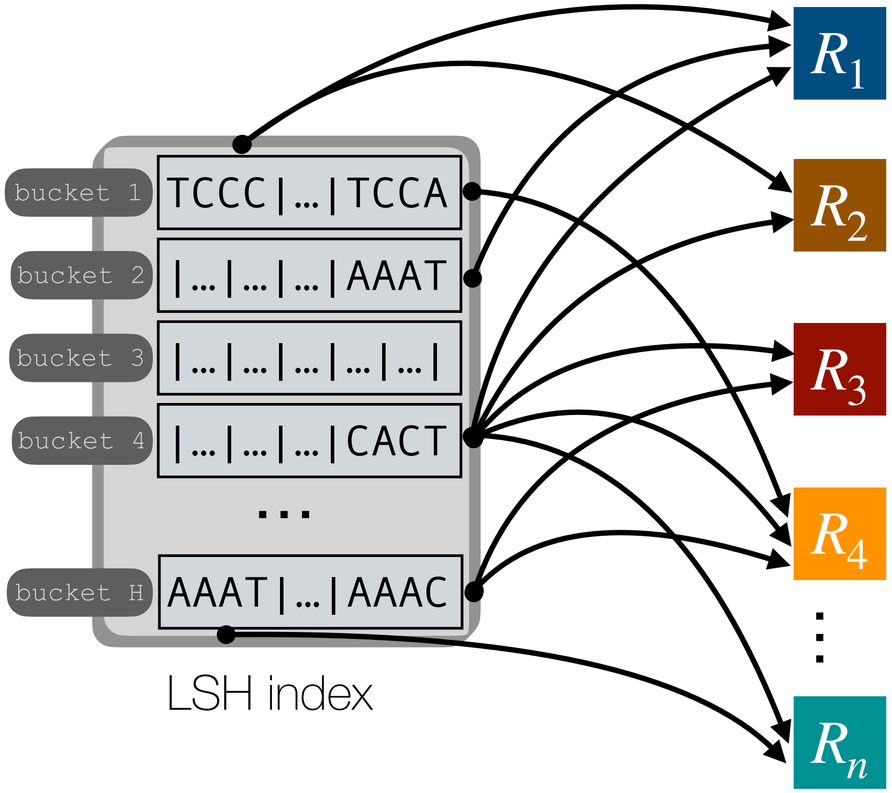


A directed bipartite graph

well studied **colored *k*-mer** problem
often represented as a sparse matrix

color: a subset of references
(including singletons)

Mapping indexed k-mers to reference genomes



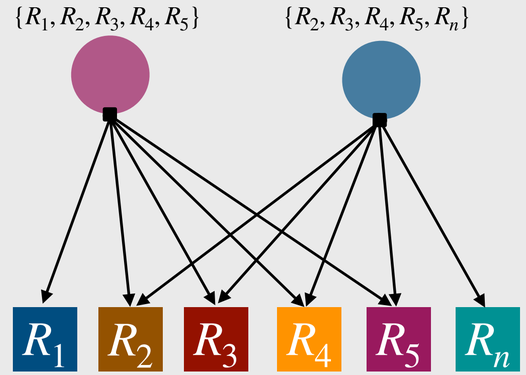
well studied **colored *k*-mer** problem
often represented as a sparse matrix

color: a subset of references
(including singletons)

Minimize $|V| + |E|$**?**

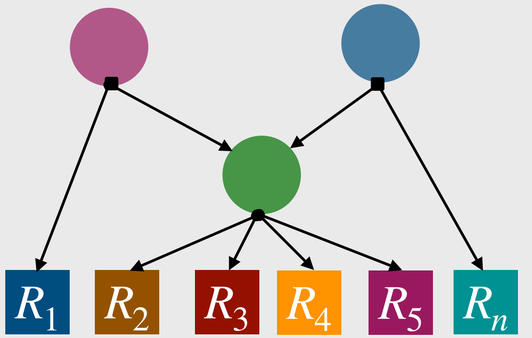
i. reduce the size by adding nodes for frequently shared subsets (*meta-colors*)

$|V| + |E| = 18$



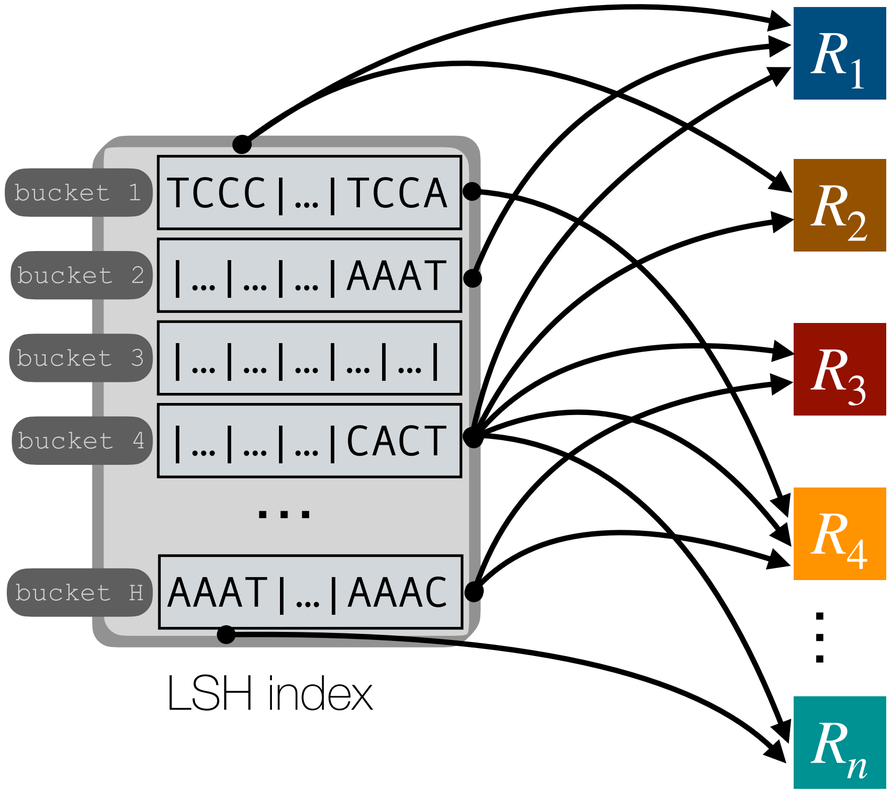
A directed bipartite graph

$|V| + |E| = 17$



DAG-based color map

Mapping indexed k-mers to reference genomes



well studied **colored *k*-mer** problem
often represented as a sparse matrix

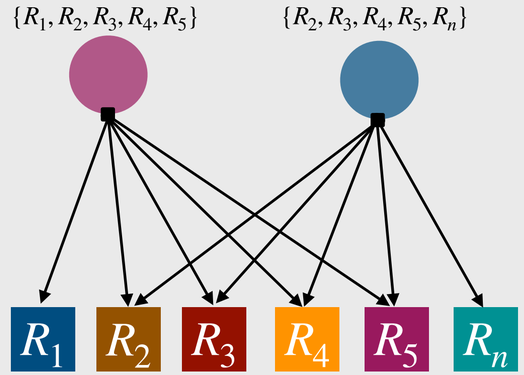
color: a subset of references
(including singletons)

Minimize $|V| + |E|$**?**

i. reduce the size by adding nodes for frequently shared subsets (*meta-colors*)

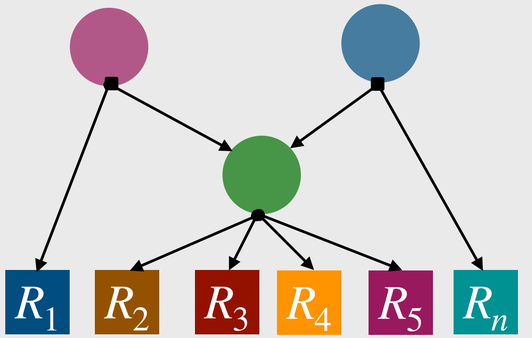
ii. explain larger color w/ smaller colors

$|V| + |E| = 18$



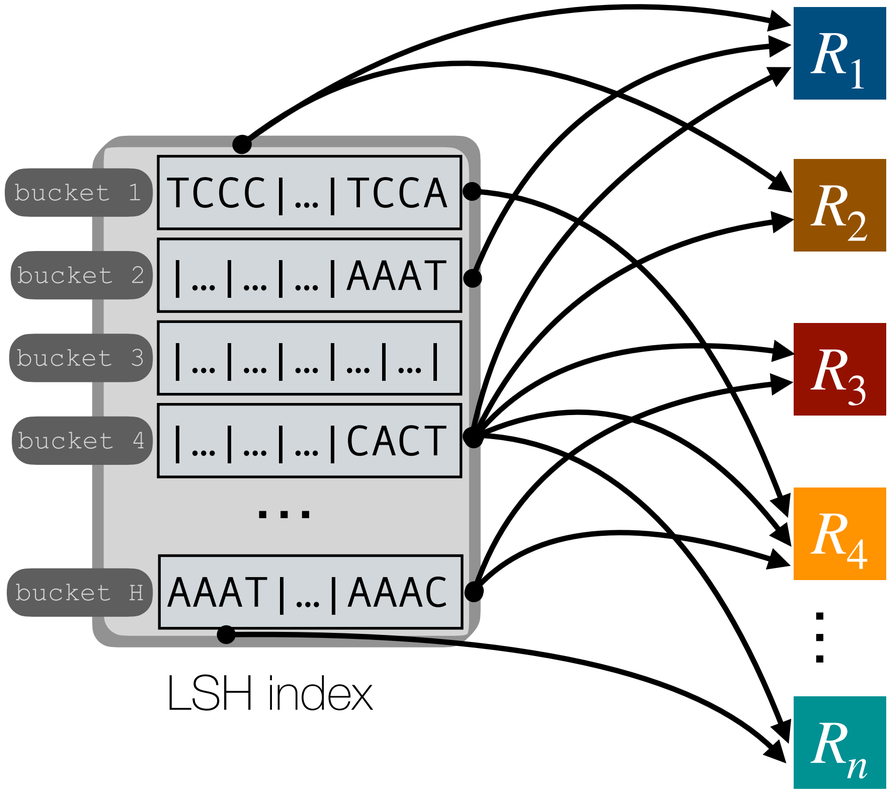
A directed bipartite graph

$|V| + |E| = 17$



DAG-based color map

Mapping indexed k-mers to reference genomes



well studied **colored *k*-mer** problem
often represented as a sparse matrix

color: a subset of references
(including singletons)

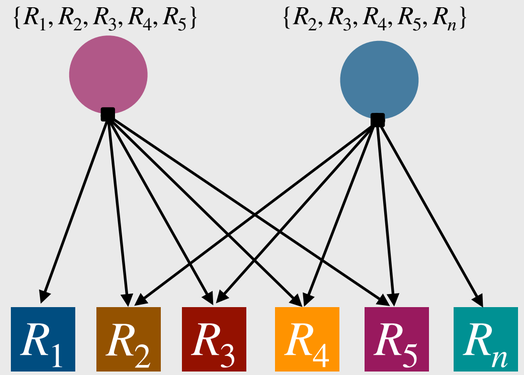
Minimize $|V| + |E|$**?**

i. reduce the size by adding nodes for frequently shared subsets (*meta-colors*)

ii. explain larger color w/ smaller colors

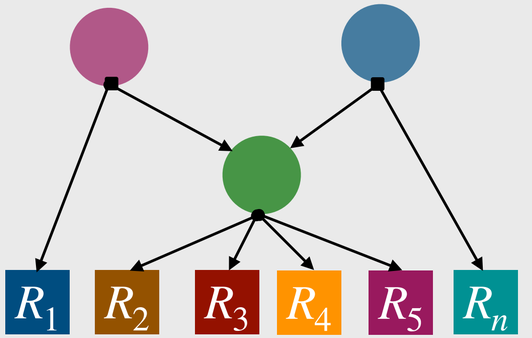
iii. use this DAG to reconstruct colors of matching *k*-mers during query time

$|V| + |E| = 18$



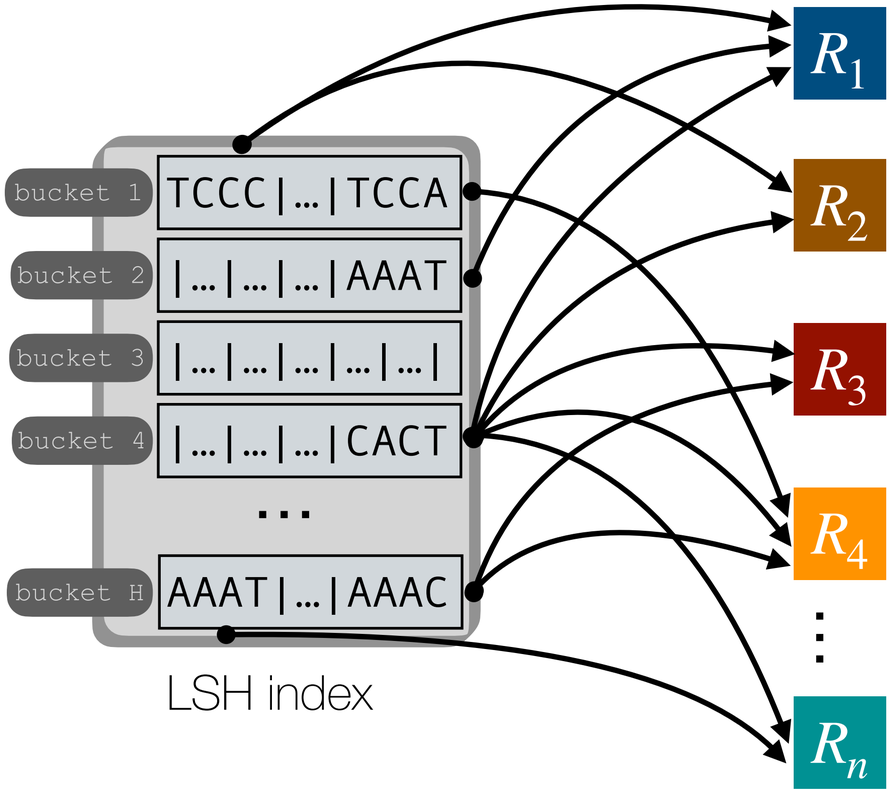
A directed bipartite graph

$|V| + |E| = 17$



DAG-based color map

Mapping indexed k-mers to reference genomes



LSH index

well studied **colored *k*-mer** problem often represented as a sparse matrix

color: a subset of references (including singletons)

Minimize $|V| + |E|$?

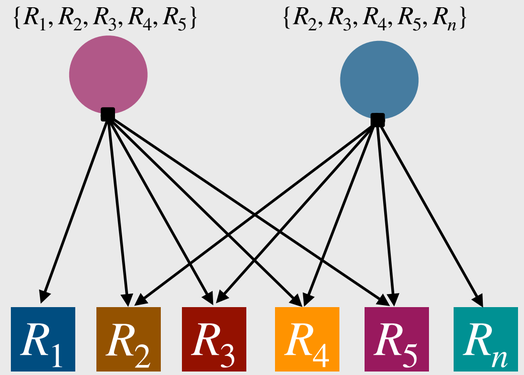
i. reduce the size by adding nodes for frequently shared subsets (*meta-colors*)

ii. explain larger color w/ smaller colors

iii. use this DAG to reconstruct colors of matching *k*-mers during query time

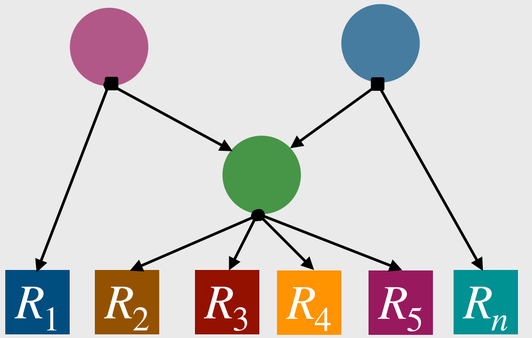
We use a heuristic guided by the phylogeny to build an efficient DAG.

$|V| + |E| = 18$



A directed bipartite graph

$|V| + |E| = 17$



DAG-based color map

Finding homologous k-mers of indexed references

Finding homologous k-mers of indexed references

Given a query sequence;

```
>ID_XYZ
ATACCTAGGAGTACGGGAC
```

Finding homologous k-mers of indexed references

Given a query sequence;

```
>ID_XYZ
ATACCTAGGAGTACGGGAC
```

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

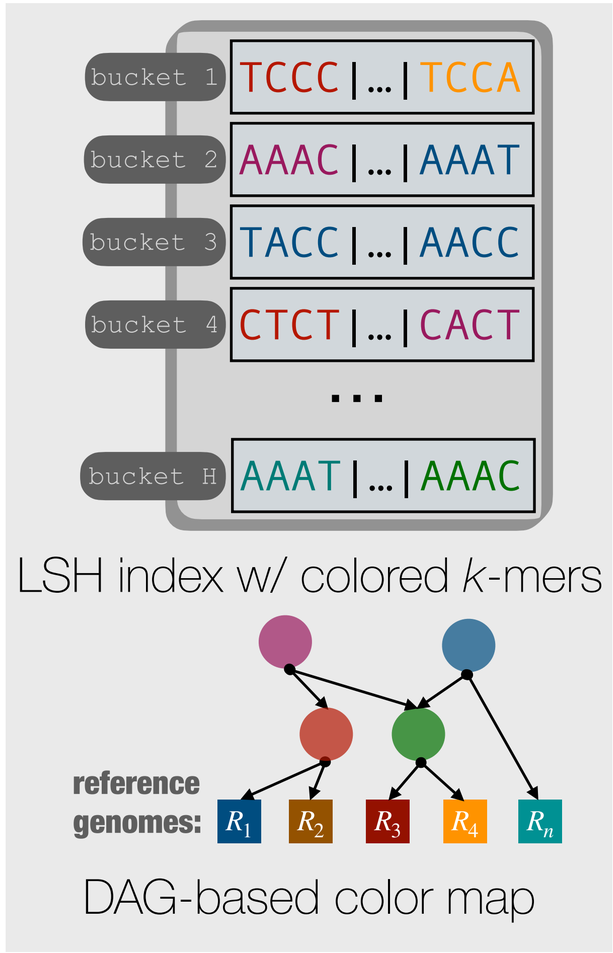
Finding homologous k-mers of indexed references

Given a query sequence;

```
>ID_XYZ
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



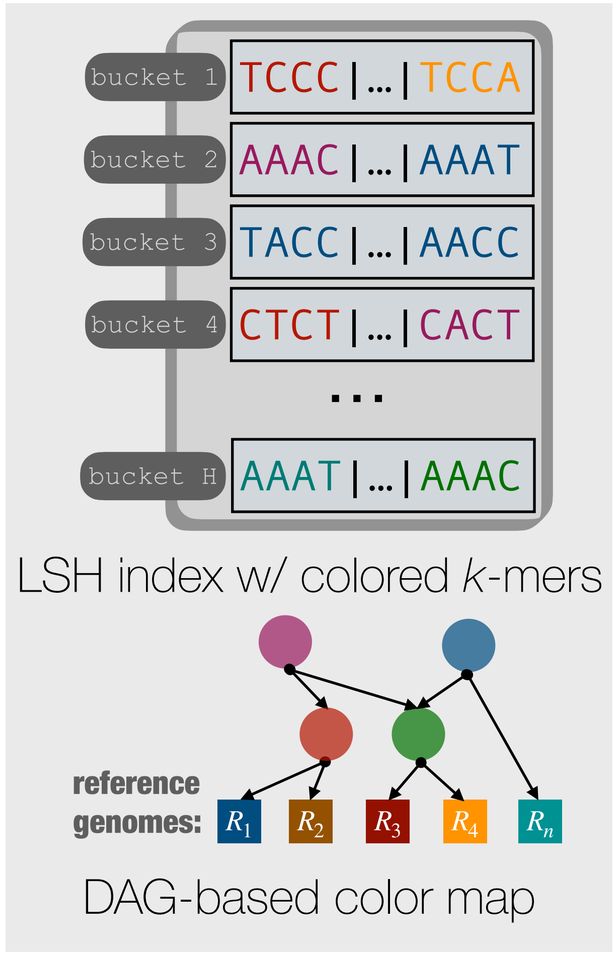
Finding homologous k-mers of indexed references

Given a query sequence;

```
>ID_XYZ
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 closest HD match as homologous

sparse table

	R_1	R_2	...	R_n
1	0	1	...	-
2	1	4	...	4
3	-	-	...	-
4	-	2	...	-
5	0	0	...	3
...	...	...	...	...
L-k+1	3	0	...	4

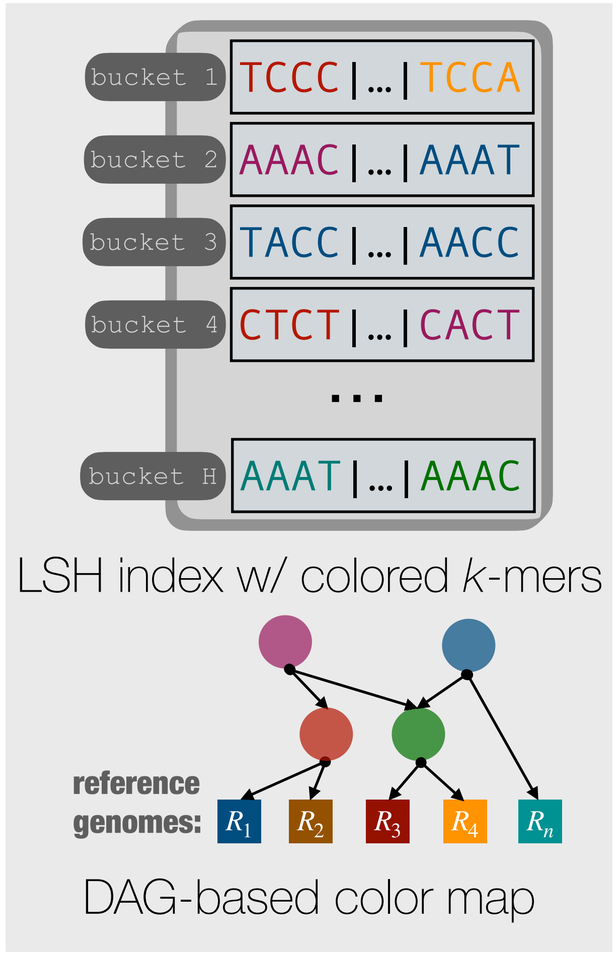
Finding homologous k-mers of indexed references

Given a query sequence;

```
>ID_XYZ
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 closest HD match as homologous

sparse table

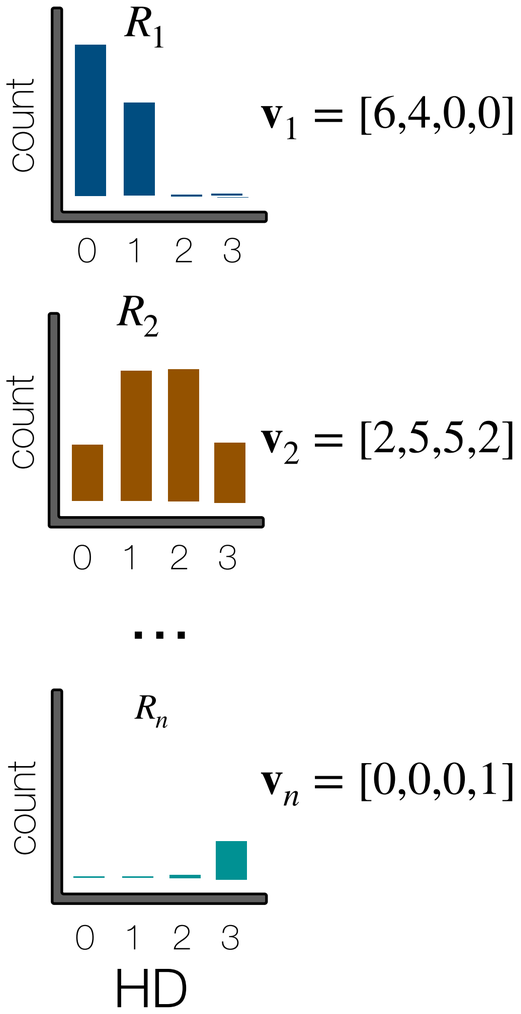
	R_1	R_2	...	R_n
1	0	1	...	-
2	1	4	...	4
3	-	-	...	-
4	-	2	...	-
5	0	0	...	3
...	...	...	...	...
L-k+1	3	0	...	4

summarize as a histogram for each reference

Likelihood of k-mer matches & Hamming distances

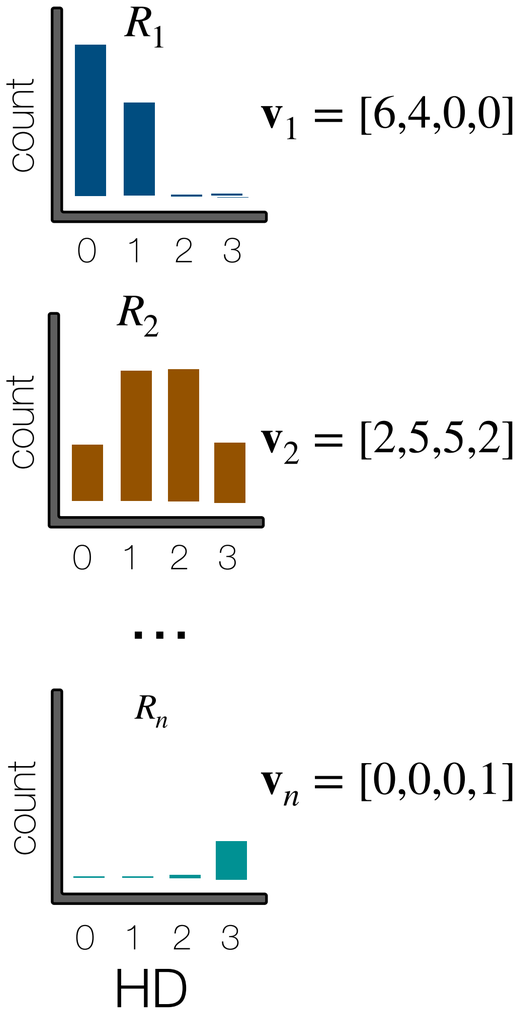
Likelihood of k-mer matches & Hamming distances

Hamming distance histograms



Likelihood of k-mer matches & Hamming distances

Hamming distance histograms



Goal: compute the probability of q having distance D to R_i

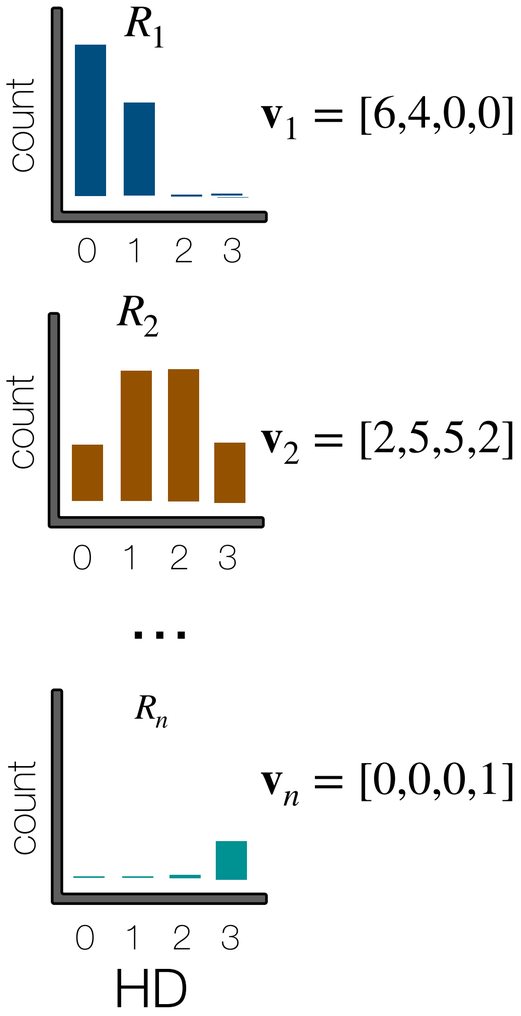
- $\mathbf{v}_i$: match count for each HD up to δ
- u_i: number of mismatches $(L-k+1) - \sum v_i$

Likelihood of distance D to reference R_i

$$\mathscr{L}_i(D; k, h, \delta, u_i, \mathbf{v}_i) =$$

Likelihood of k-mer matches & Hamming distances

Hamming distance histograms



Goal: compute the probability of q having distance D to R_i

- $\mathbf{v}_i$: match count for each HD up to δ
- u_i: number of mismatches $(L-k+1) - \sum v_i$

The Statistics of *k*-mers from a Sequence Undergoing a Simple Mutation Process Without Spurious Matches

Authors: Antonio Blanca, Robert S. Harris, David Koslicki, and Paul Medvedev | AUTHORS INFO & AFFILIATIONS

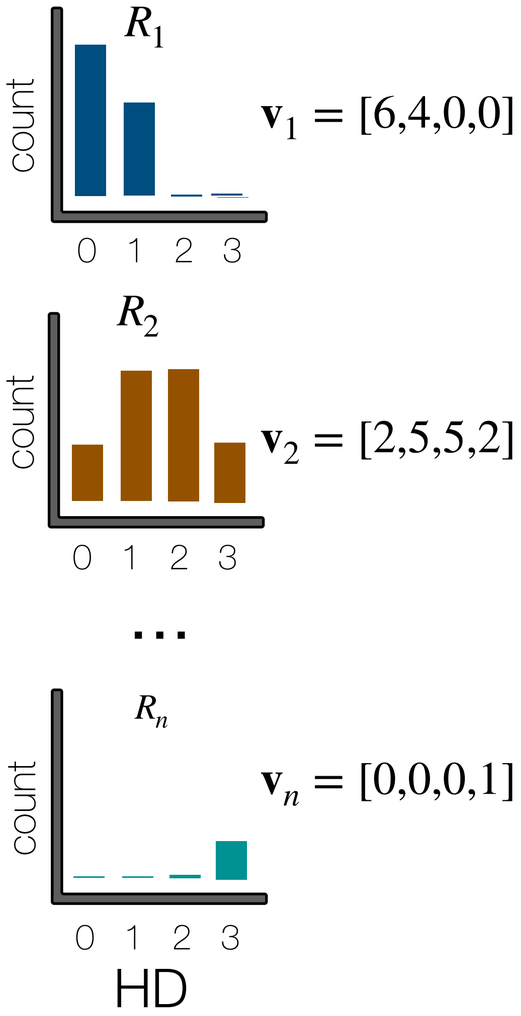
Publication: Journal of Computational Biology • https://doi.org/10.1089/cmb.2021.0431

Likelihood of distance D to reference R_i

$$\mathscr{L}_i(D; k, h, \delta, u_i, \mathbf{v}_i) =$$

Likelihood of k-mer matches & Hamming distances

Hamming distance histograms



Goal: compute the probability of q having distance D to R_i

- $\mathbf{v}_i$: match count for each HD up to δ
- u_i: number of mismatches $(L-k+1) - \sum v_i$

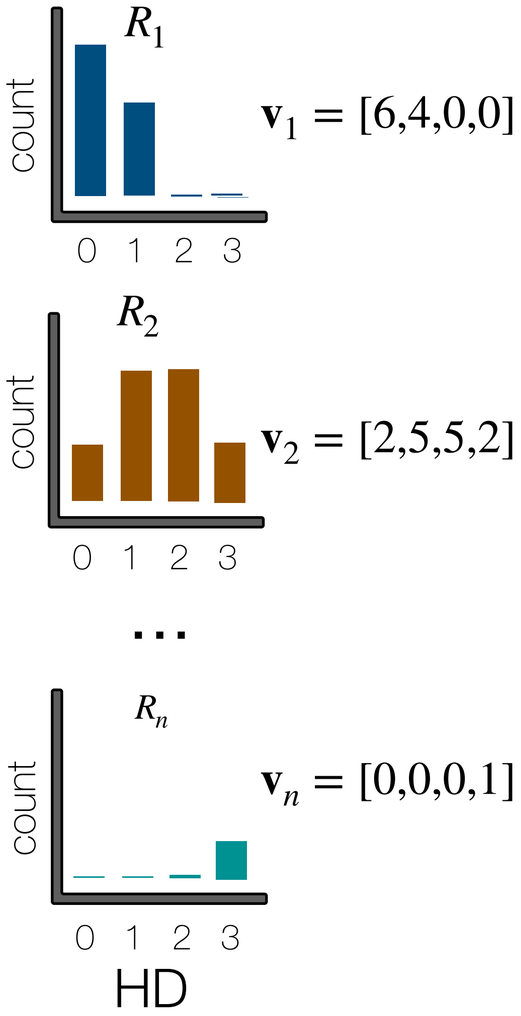
Independence assumption: treat q as a bag of $L-k+1$ k-mers

Likelihood of distance D to reference R_i

$$\mathscr{L}_i(D; k, h, \delta, u_i, \mathbf{v}_i) =$$

Likelihood of k-mer matches & Hamming distances

Hamming distance histograms



Goal: compute the probability of q having distance D to R_i

- $\mathbf{v}_i$: match count for each HD up to δ
- u_i: number of mismatches $(L-k+1) - \sum v_i$

Independence assumption: treat q as a bag of $L-k+1$ k-mers

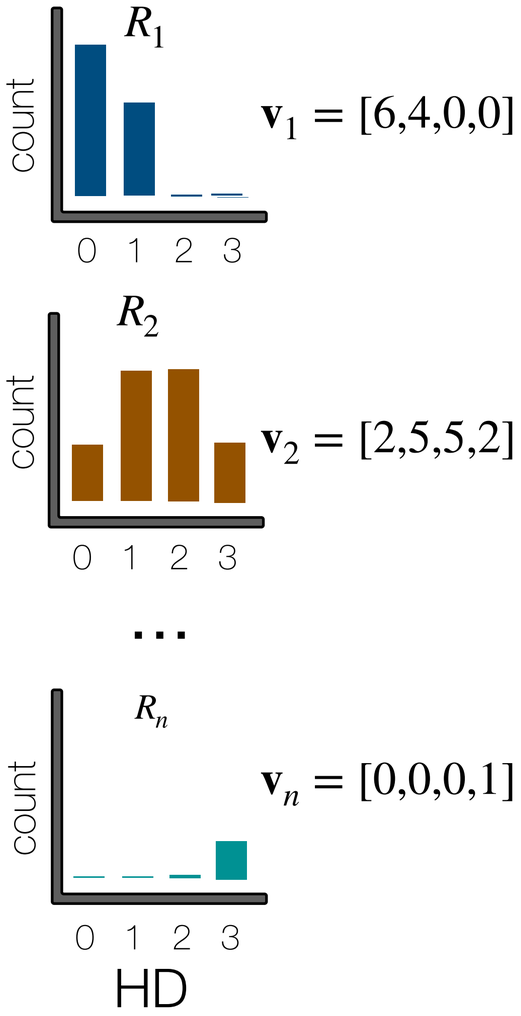
Likelihood of distance D to reference R_i

$$\mathscr{L}_i(D;k,h,\delta,u_i,\mathbf{v}_i) = P_{miss}(D;k,h,\delta)^{u_i}$$

Probability of having u_i mismatches in total

Likelihood of k-mer matches & Hamming distances

Hamming distance histograms



Goal: compute the probability of q having distance D to R_i

- $\mathbf{v}_i$: match count for each HD up to δ
- u_i: number of mismatches $(L-k+1) - \sum v_i$

Independence assumption: treat q as a bag of $L-k+1$ k-mers

Likelihood of distance D to reference R_i

$$\mathscr{L}_i(D;k,h,\delta,u_i,\mathbf{v}_i) = P_{miss}(D;k,h,\delta)^{u_i} \prod_{d=0}^{\delta} P_{match}(D;d,k,h)^{v_{i,d}}$$

Probability of having u_i mismatches in total

Probability of having $v_{i,d}$ matches w/ HD $= d$

Observing homologous k-mers with varying HDs

Observing homologous k-mers with varying HDs

$$P_{match}(D; d, k, h) =$$

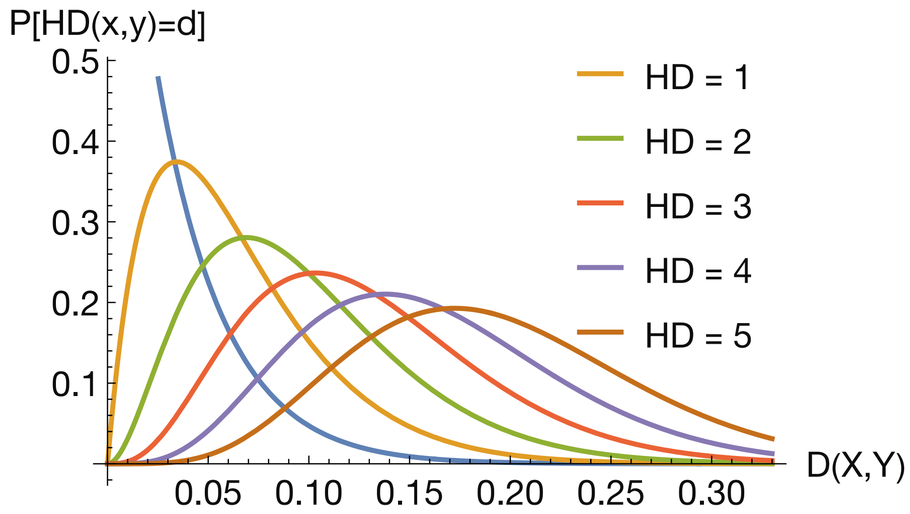
Observing homologous k-mers with varying HDs

$$P_{match}(D; d, k, h) = \qquad P_{mutate}(D; d, k)$$

Observing homologous k-mers with varying HDs

$$P_{match}(D; d, k, h) = \qquad P_{mutate}(D; d, k)$$

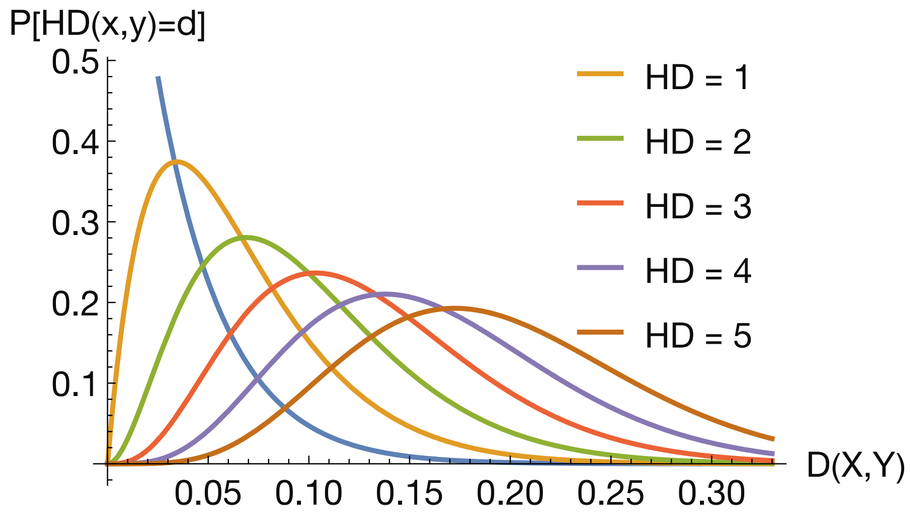
$$D^d(1 - D)^{(k-d)} \binom{k}{d}$$



Observing homologous k-mers with varying HDs

$$P_{match}(D; d, k, h) = P_{mutate}(D; d, k)\ P_{collide}(d, k, h)$$

$$D^d(1 - D)^{(k-d)} \binom{k}{d}$$

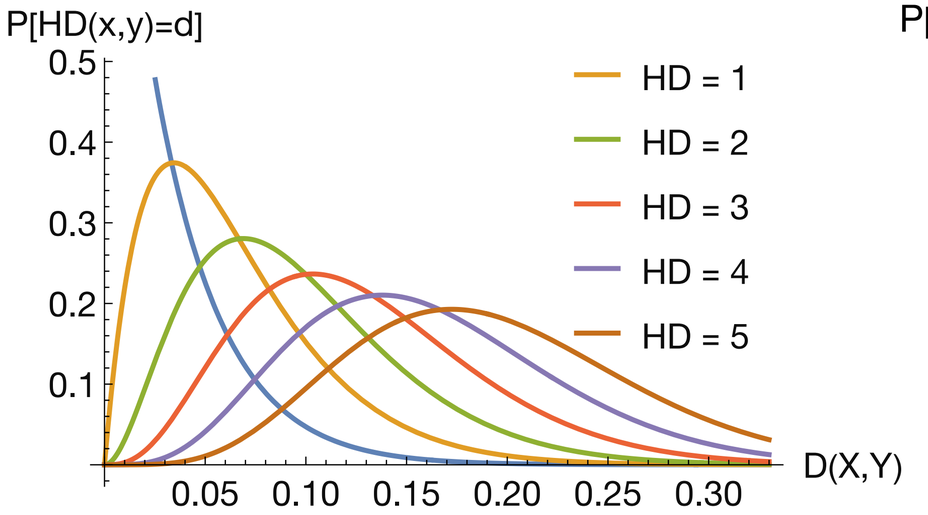


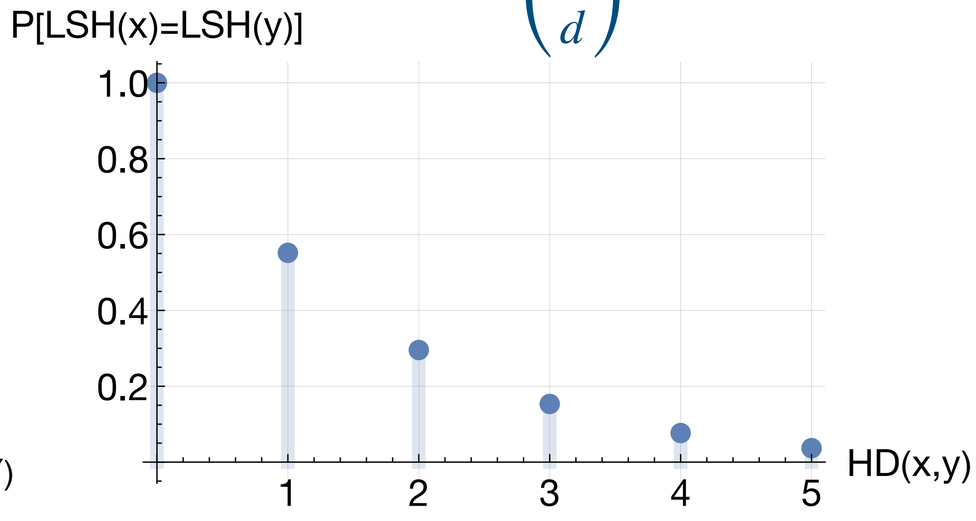
Observing homologous k-mers with varying HDs

$$P_{match}(D; d, k, h) = P_{mutate}(D; d, k)\ P_{collide}(d, k, h)$$

$$D^d(1 - D)^{(k-d)}\binom{k}{d}$$

$$\frac{\binom{k-h}{d}}{\binom{k}{d}}$$



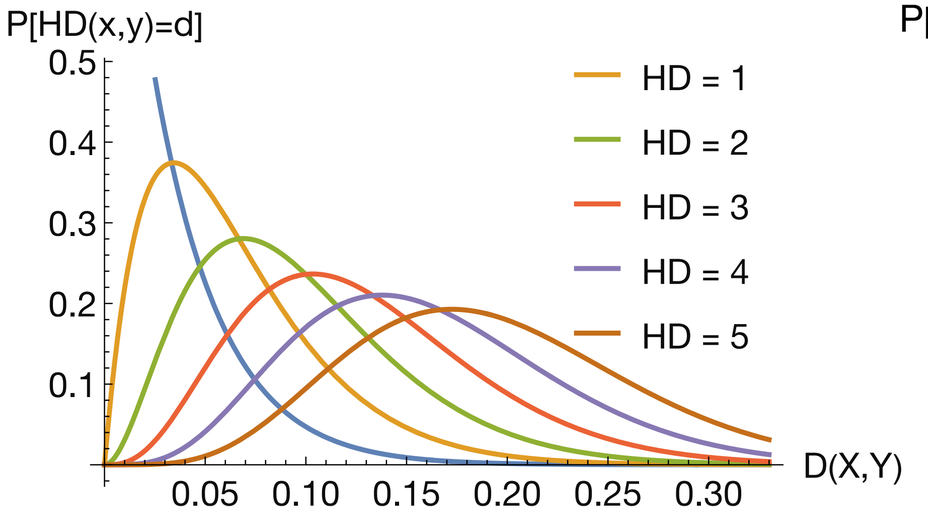


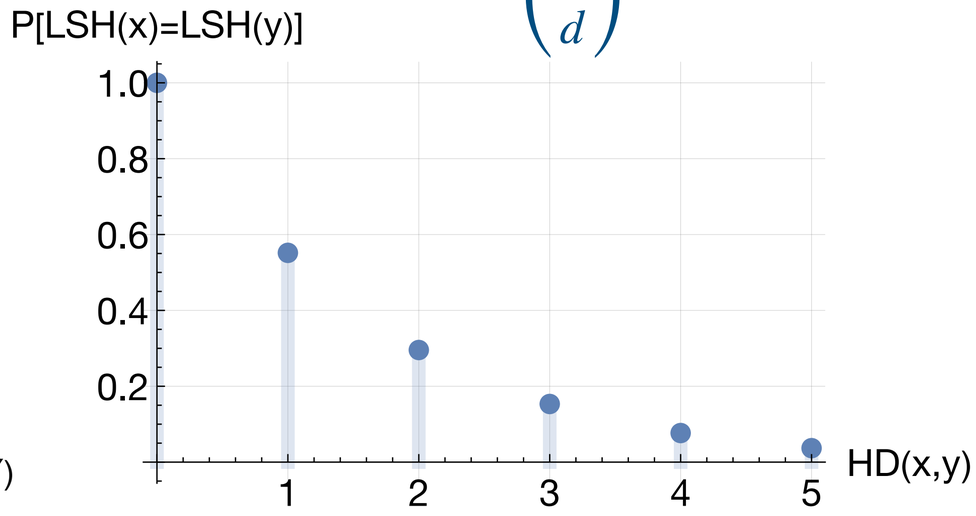
Observing homologous k-mers with varying HDs

$$P_{match}(D;d,k,h) = \rho_i \, P_{mutate}(D;d,k) \;\; P_{collide}(d,k,h)$$

$$D^d(1-D)^{(k-d)}\binom{k}{d}$$

$$\frac{\binom{k-h}{d}}{\binom{k}{d}}$$





Observing homologous k-mers with varying HDs

$$P_{match}(D; d, k, h) = \rho_i \, P_{mutate}(D; d, k) \; P_{collide}(d, k, h)$$

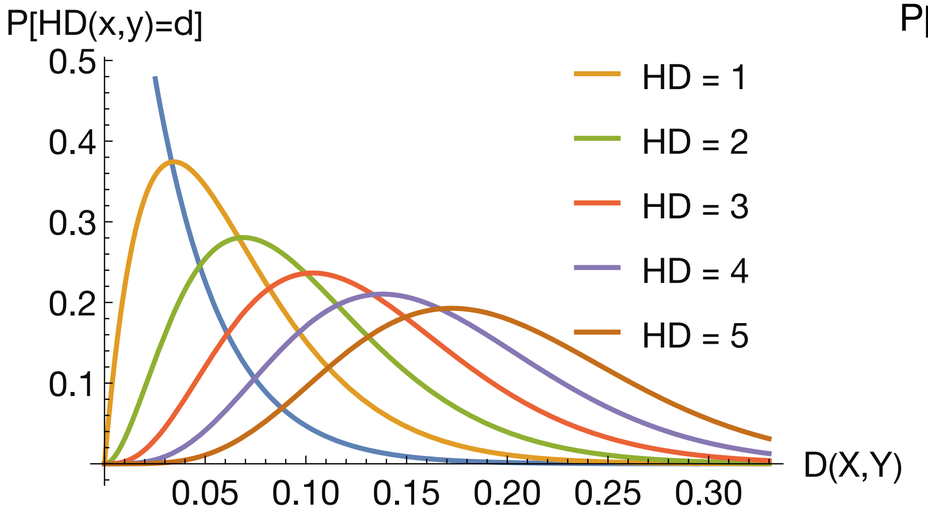
$$\rho_i = \frac{\text{\# of subsampled}}{\text{\# of distinct}}$$

precomputed for R_i

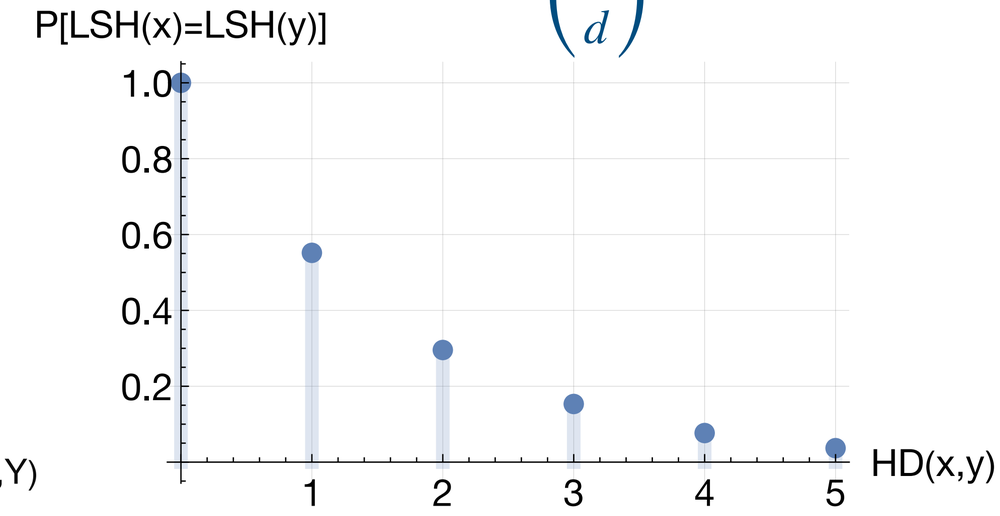
→ not all *k*-mers have to be indexed:

- minimizers
- FracMinHash
- …

$$D^d (1 - D)^{(k-d)} \binom{k}{d}$$



$$\frac{\binom{k-h}{d}}{\binom{k}{d}}$$



Multiple events could lead to a mismatch might

Multiple events could lead to a mismatch might

A mismatch occurs for two homologous *k*-mers (query x and reference y)

$$P_{miss}(D; d, k, h, \delta) =$$

Multiple events could lead to a mismatch might

A mismatch occurs for two homologous *k*-mers (query x and reference y)

- if y is not indexed: $1 - \rho$

$$P_{miss}(D; d, k, h, \delta) =$$

$$(1 - \rho)$$

Multiple events could lead to a mismatch might

A mismatch occurs for two homologous *k*-mers (query x and reference y)

- if y is not indexed: $1-\rho$
- if y is indexed (with probability ρ), but either:

$$P_{miss}(D;d,k,h,\delta) =$$

$$(1-\rho)+\rho\Big($$

$$\Big)$$

Multiple events could lead to a mismatch might

A mismatch occurs for two homologous *k*-mers (query x and reference y)

- if y is not indexed: $1 - \rho$
- if y is indexed (with probability ρ), but either:
  - $\mathrm{HD}(x, y) > \delta$ *or* $\mathrm{LSH}(x) \neq \mathrm{LSH}(y)$!

$$P_{miss}(D; d, k, h, \delta) =$$

$$(1 - \rho) + \rho \left(\sum_{d=\delta+1}^{k} P_{mutate}(D; d, k) \right)$$

Multiple events could lead to a mismatch might

A mismatch occurs for two homologous *k*-mers (query x and reference y)

- if y is not indexed: $1 - \rho$
- if y is indexed (with probability ρ), but either:
  - $\mathrm{HD}(x, y) > \delta$ *or* $\mathrm{LSH}(x) \neq \mathrm{LSH}(y)$!

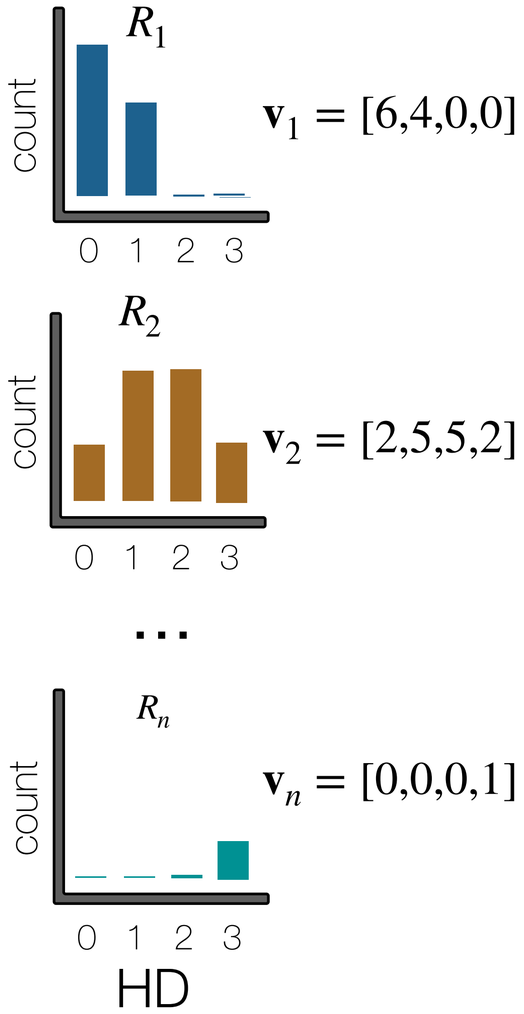
$$P_{miss}(D; d, k, h, \delta) =$$

$$(1 - \rho) + \rho \left(\sum_{d=\delta+1}^{k} P_{mutate}(D; d, k) + \sum_{d=0}^{\delta} P_{mutate}(D; d, k)(1 - P_{collide}(d, k, h)) \right)$$

Maximum likelihood estimation of distances

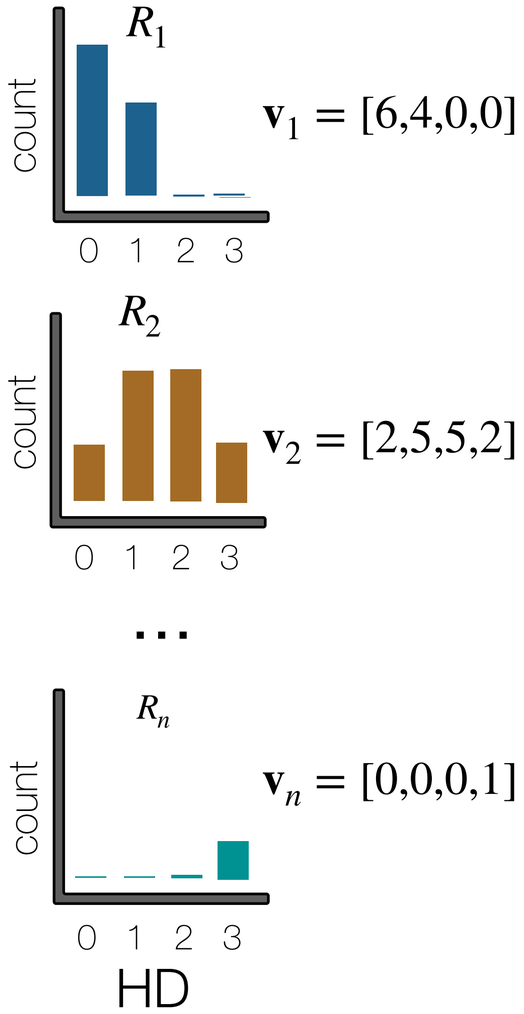
Maximum likelihood estimation of distances

Hamming distance histograms



Maximum likelihood estimation of distances

Hamming distance histograms

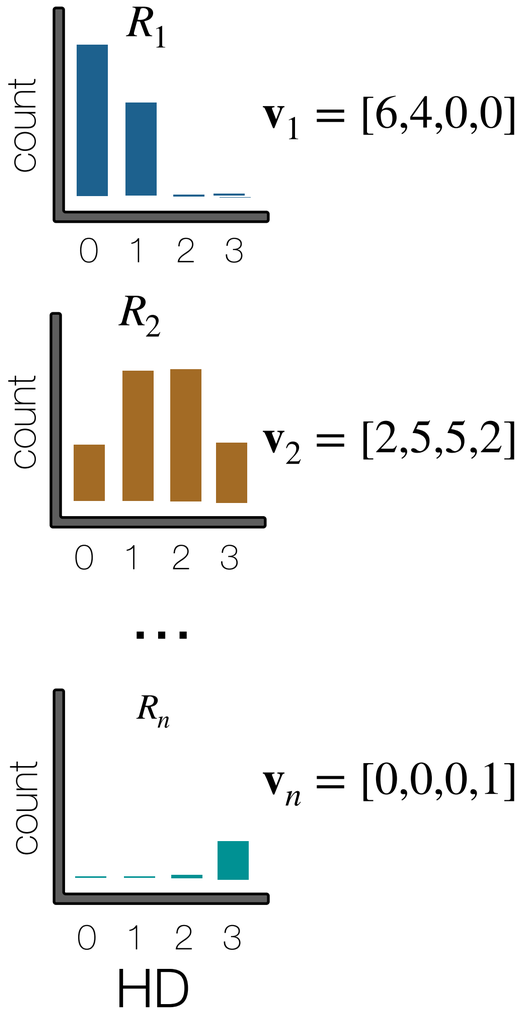


Optimize $-\log \mathscr{L}_i$ w.r.t. D for each hitting reference R_i:

$$\arg\max_D u_i \log P_{miss}(D;k,h,\delta) + \sum_{d=0}^{\delta} v_{i,d}\left(d\log(D) + (k-d)\log(1-D)\right)$$

Maximum likelihood estimation of distances

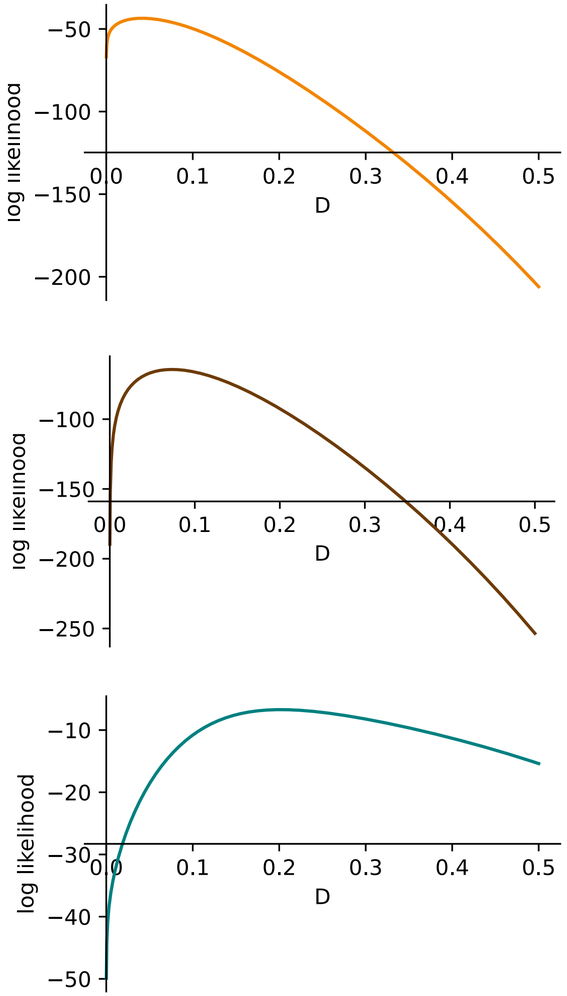
Hamming distance histograms



single variable & convex with a sensible choice of parameters

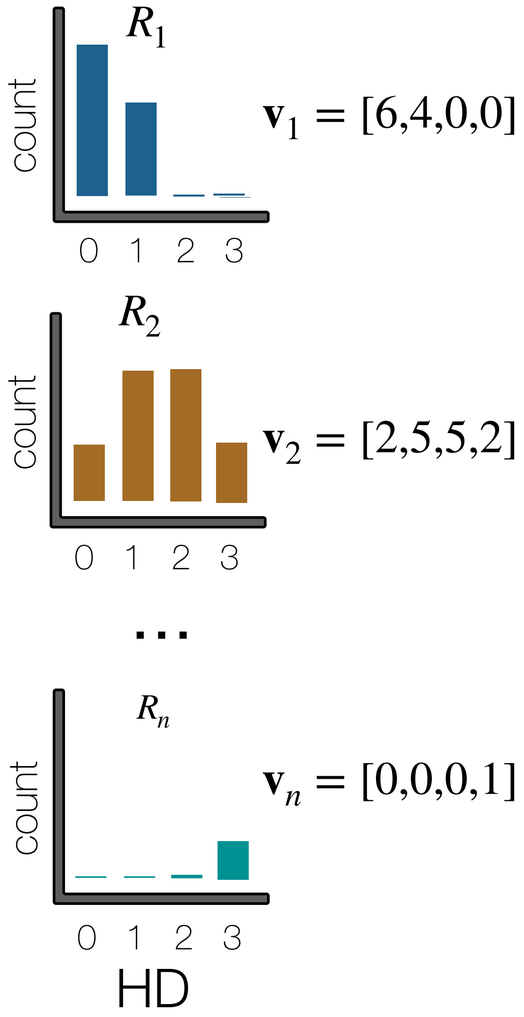
Optimize $-\log \mathscr{L}_i$ w.r.t. D for each hitting reference R_i:

$$\arg\max_D u_i \log P_{miss}(D;k,h,\delta) + \sum_{d=0}^{\delta} v_{i,d}\left(d\log(D) + (k-d)\log(1-D)\right)$$



Maximum likelihood estimation of distances

Hamming distance histograms

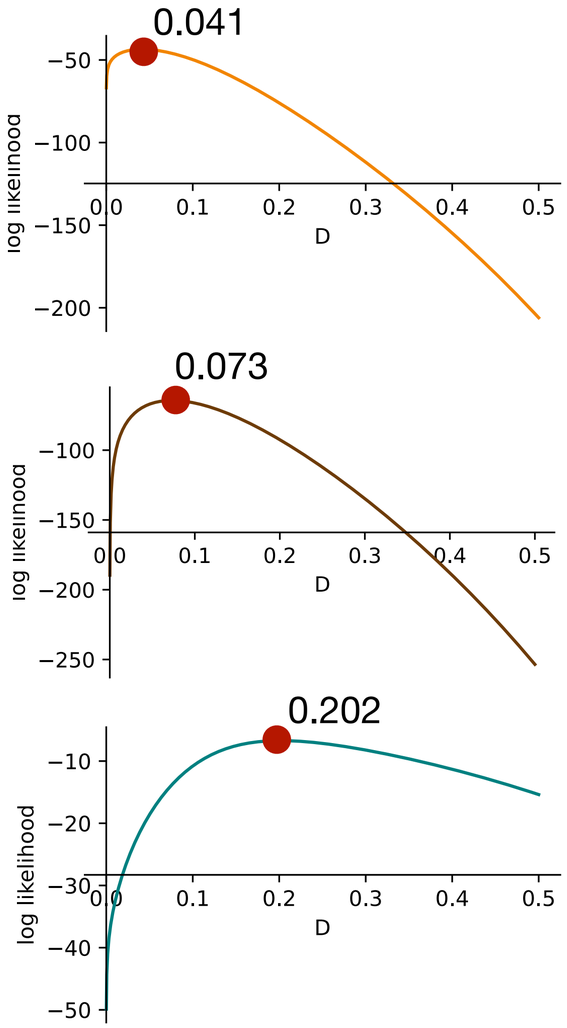


single variable & convex with a sensible choice of parameters

Optimize $-\log \mathscr{L}_i$ w.r.t. D for each hitting reference R_i:

$$\arg\max_D u_i \log P_{miss}(D;k,h,\delta) + \sum_{d=0}^{\delta} v_{i,d}\left(d\log(D) + (k-d)\log(1-D)\right)$$

using Brent's Method



krepp estimates read-level distances accurately

krepp estimates read-level distances accurately

default: 29-mer
minimizers of 35-mers

150 bp short reads

(Hamming distance) / (seq. length)

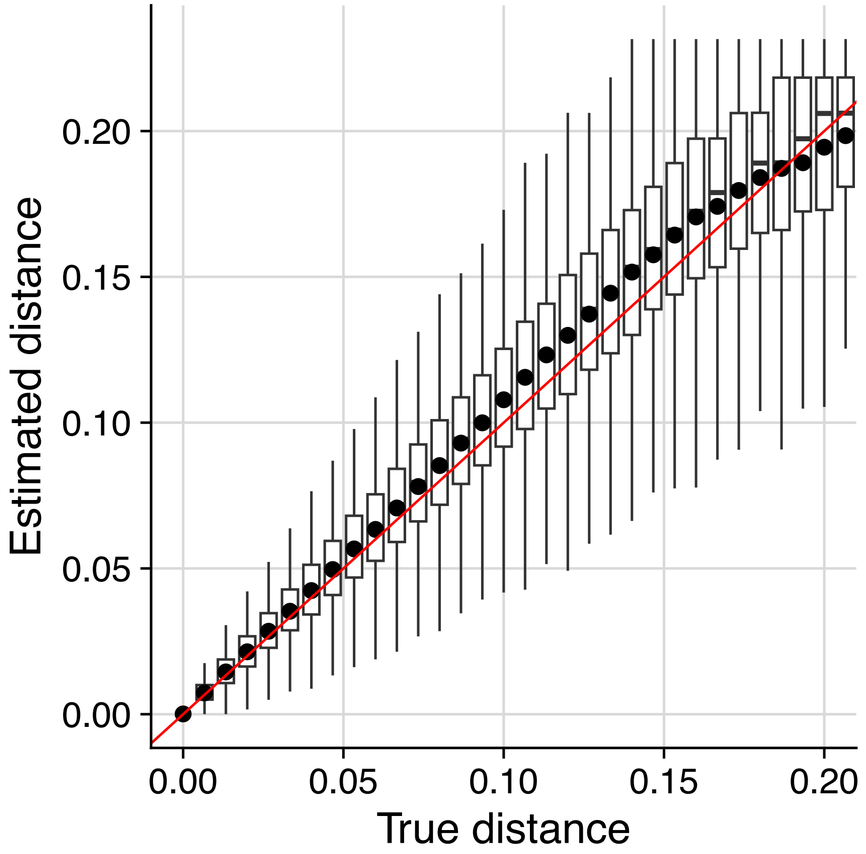
krepp estimates read-level distances accurately

default: 29-mer
minimizers of 35-mers

- Simulated genomes and known coordinates
- Highly accurate (despite some noise)

150 bp short reads

(Hamming distance) / (seq. length)



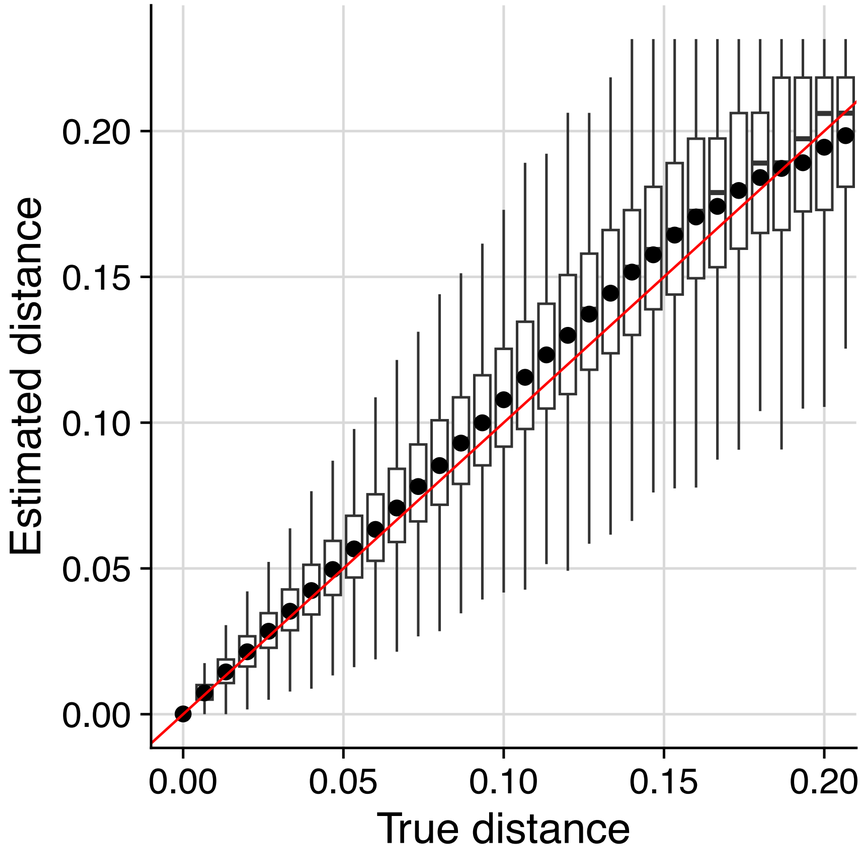
krepp estimates read-level distances accurately

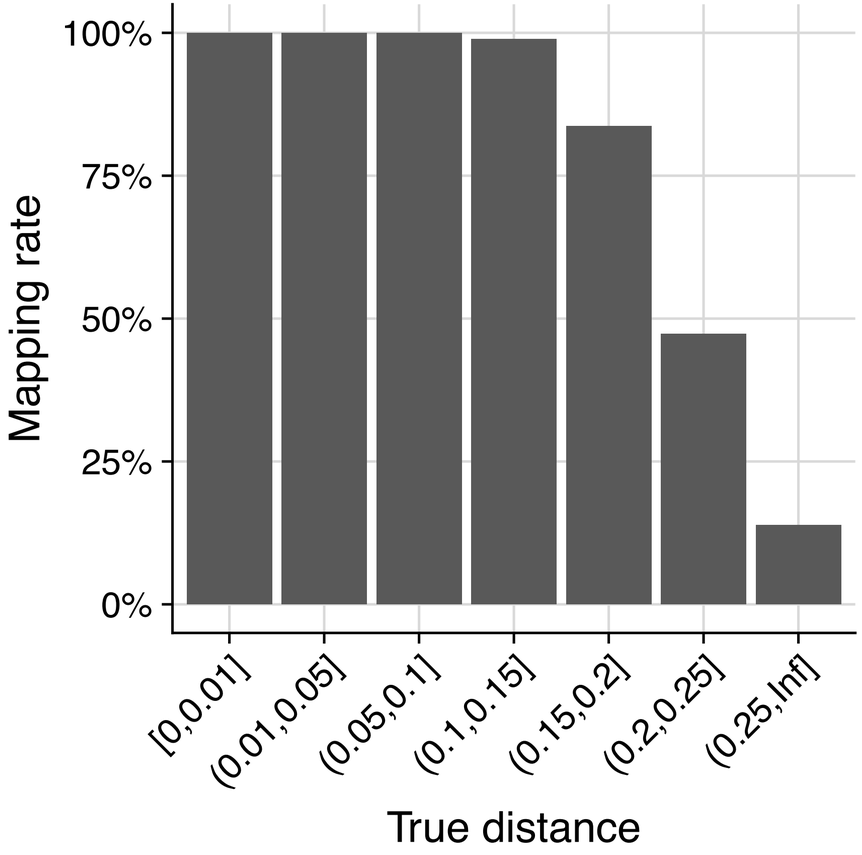
default: 29-mer
minimizers of 35-mers

- Simulated genomes and known coordinates
- Highly accurate (despite some noise)

150 bp short reads

(Hamming distance) / (seq. length)





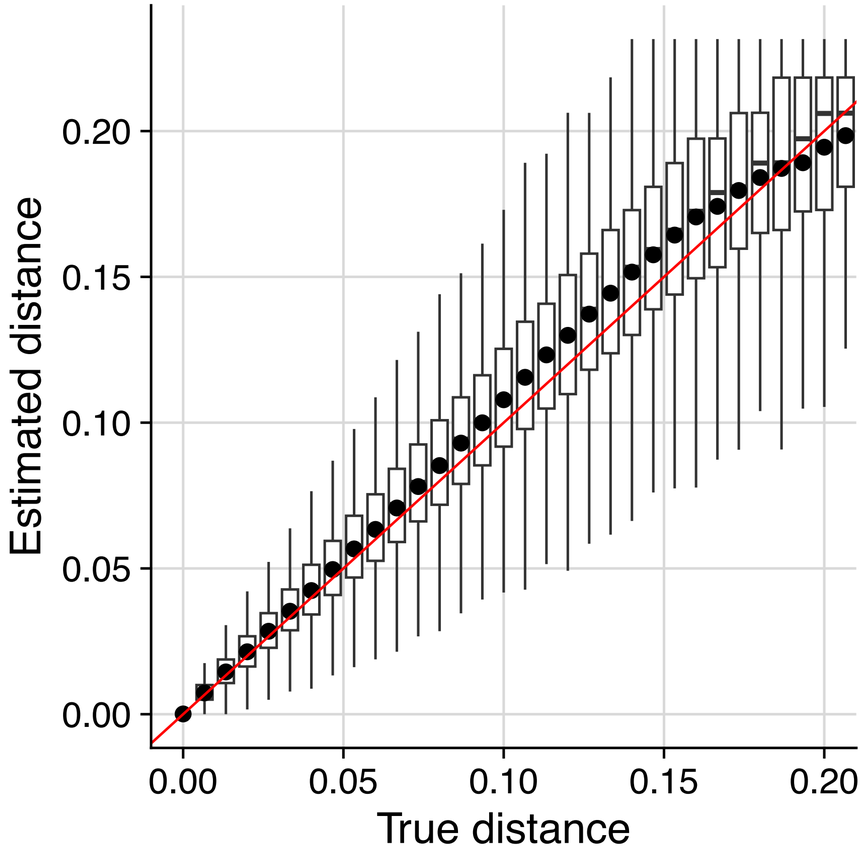
krepp estimates read-level distances accurately

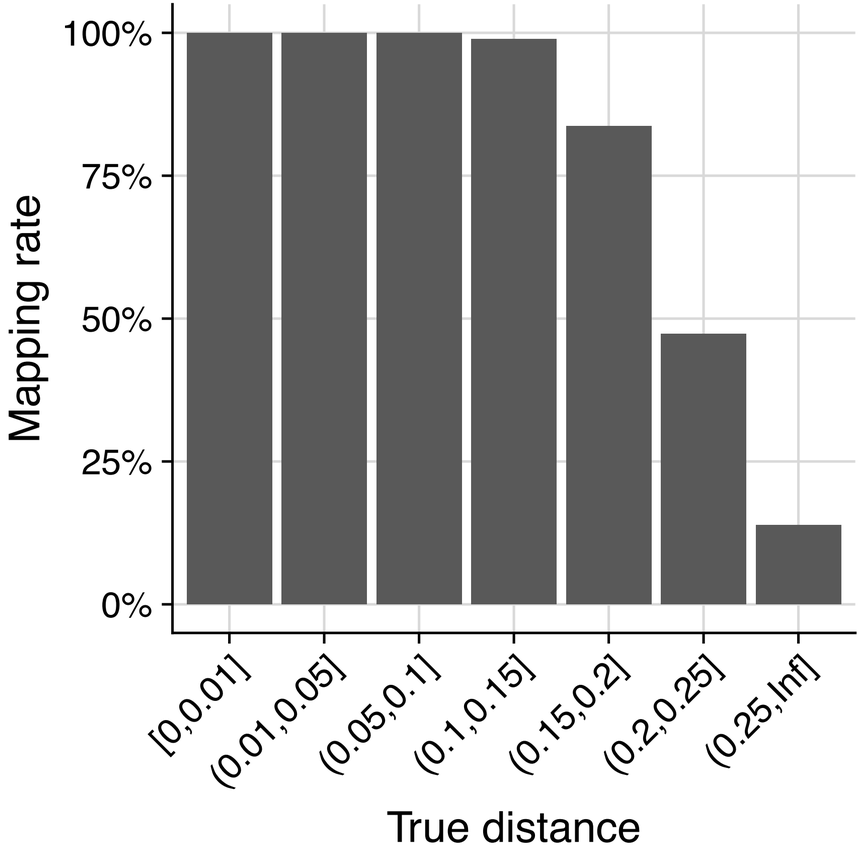
default: 29-mer
minimizers of 35-mers

- Simulated genomes and known coordinates
- Highly accurate (despite some noise)
- Slight overestimation bias for high distances

150 bp short reads

(Hamming distance) / (seq. length)





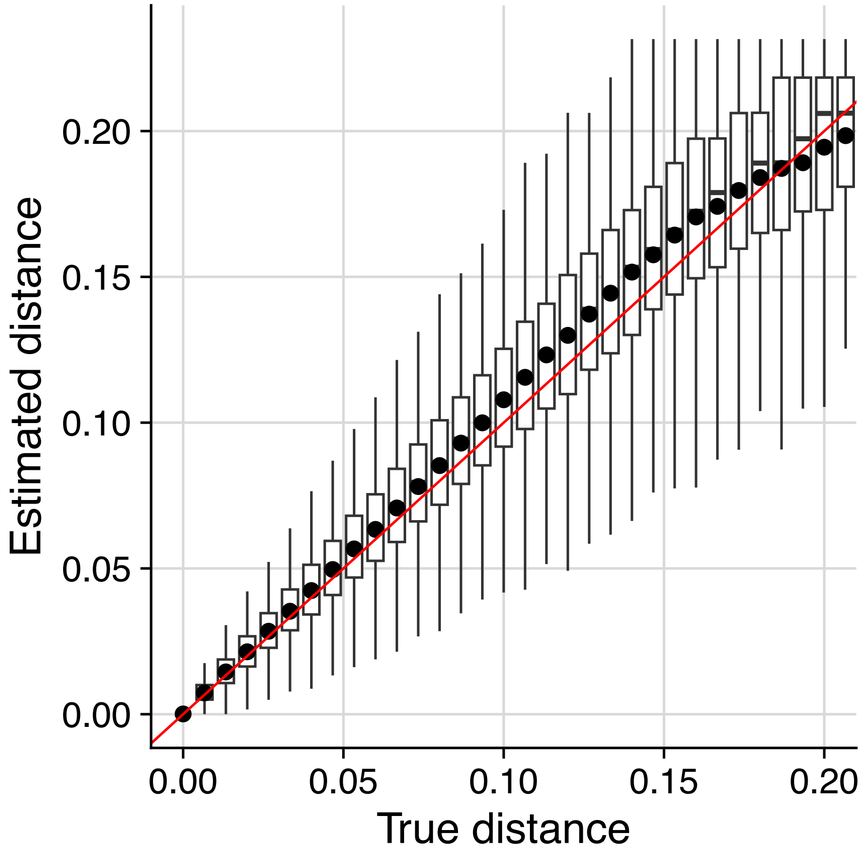
krepp estimates read-level distances accurately

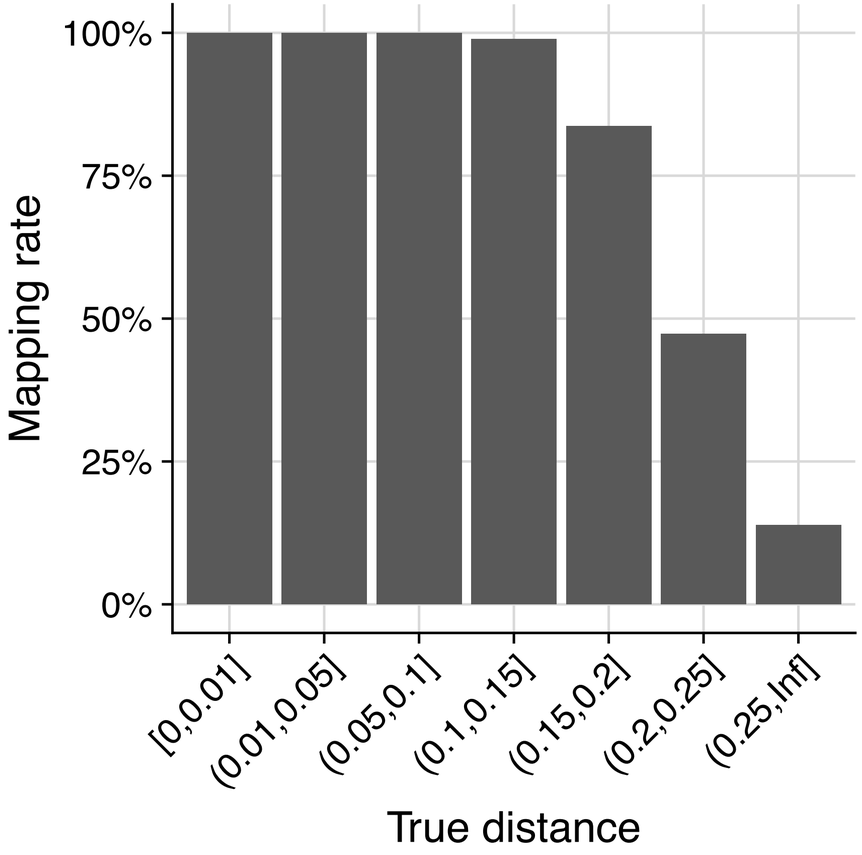
default: 29-mer
minimizers of 35-mers

- Simulated genomes and known coordinates
- Highly accurate (despite some noise)
- Slight overestimation bias for high distances
- subsampling ↑
  bias ↓, noise ↑
  mapping rate ↓

150 bp short reads

(Hamming distance) / (seq. length)



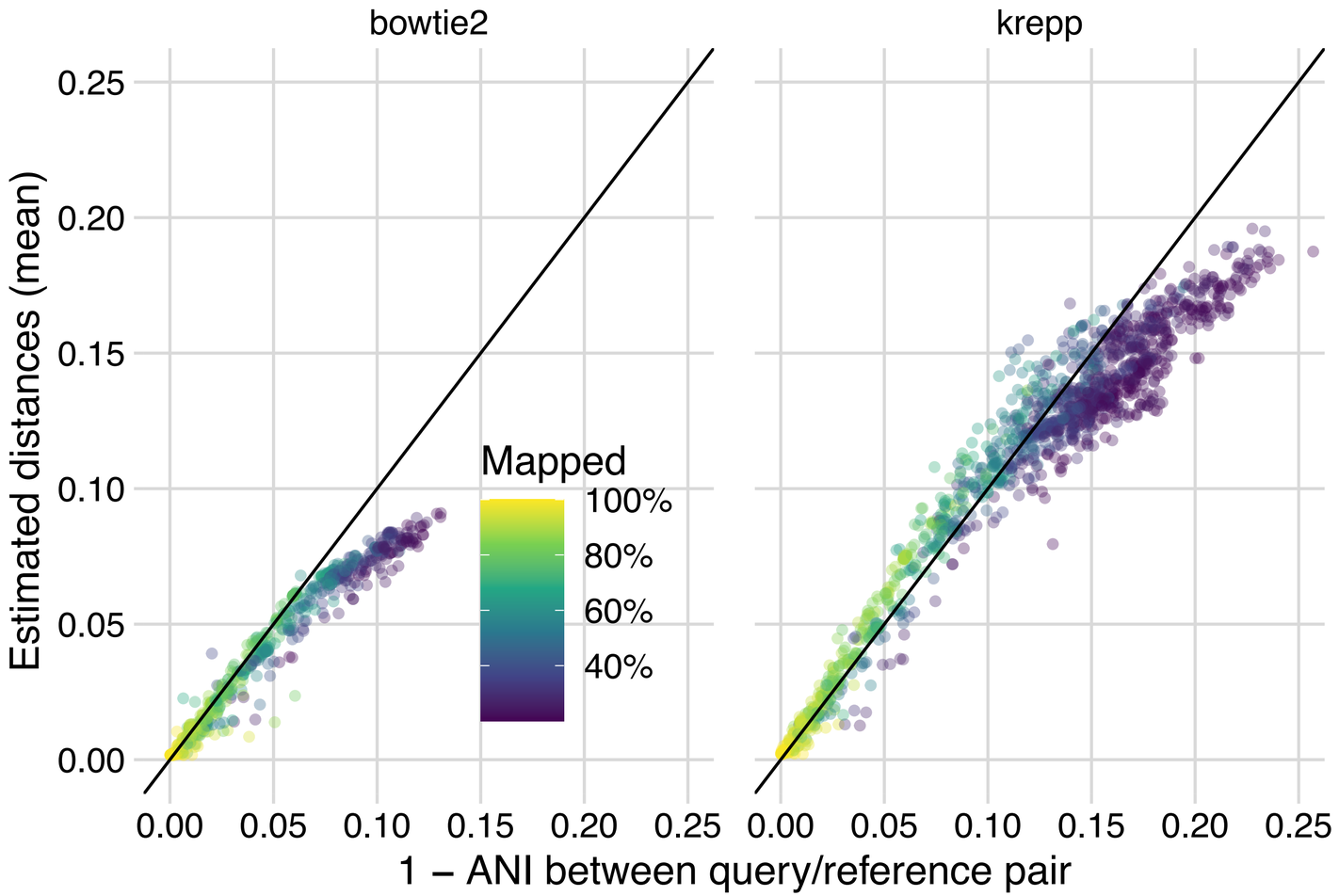


krepp matches genome-wide nucleotide identity on average

krepp matches genome-wide nucleotide identity on average

Reference: Web of Life (v2)
16,000 microbial genomes

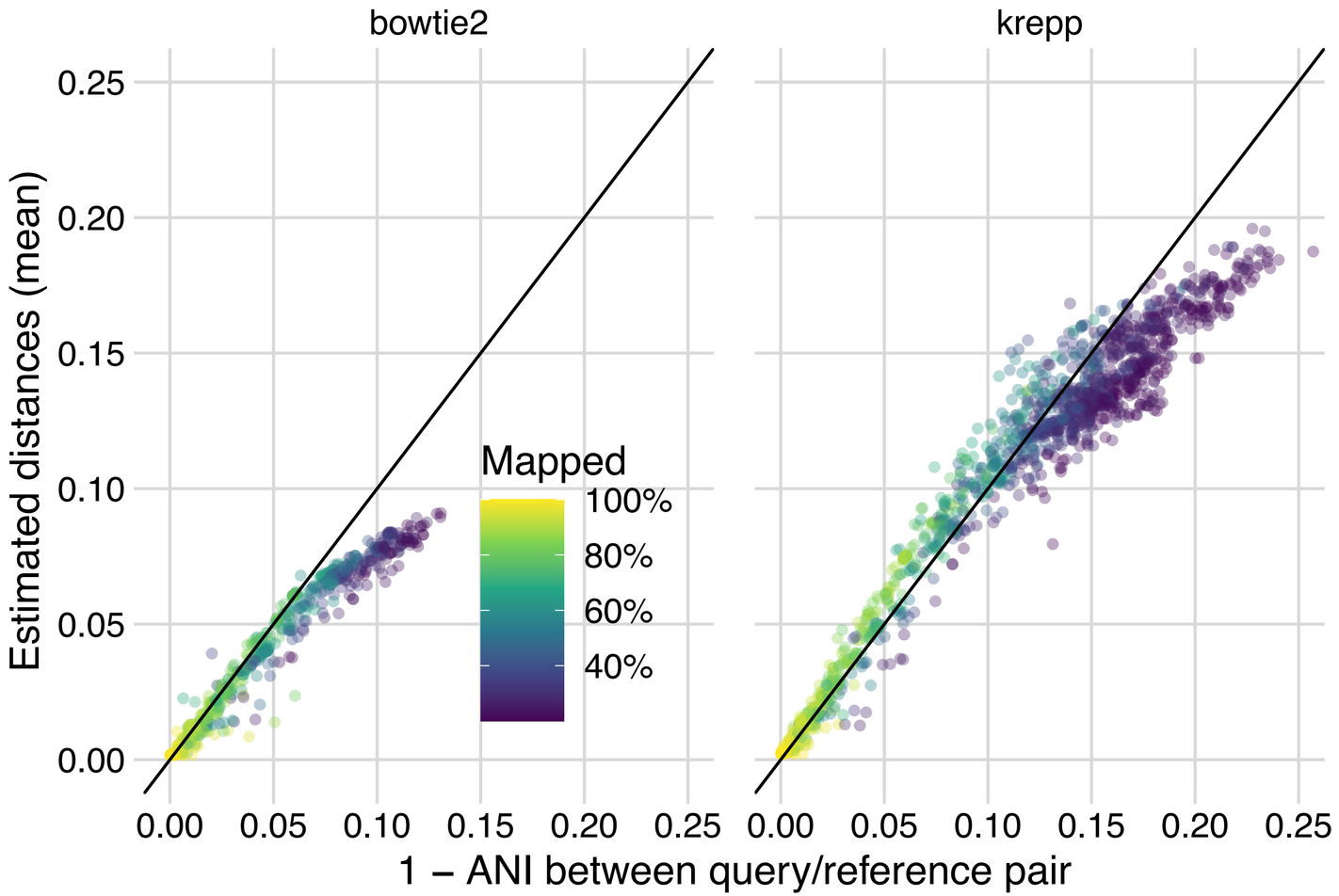
- Real query/reference genomes w/ simulated short reads



krepp matches genome-wide nucleotide identity on average

Reference: Web of Life (v2)
16,000 microbial genomes

- Real query/reference genomes w/ simulated short reads

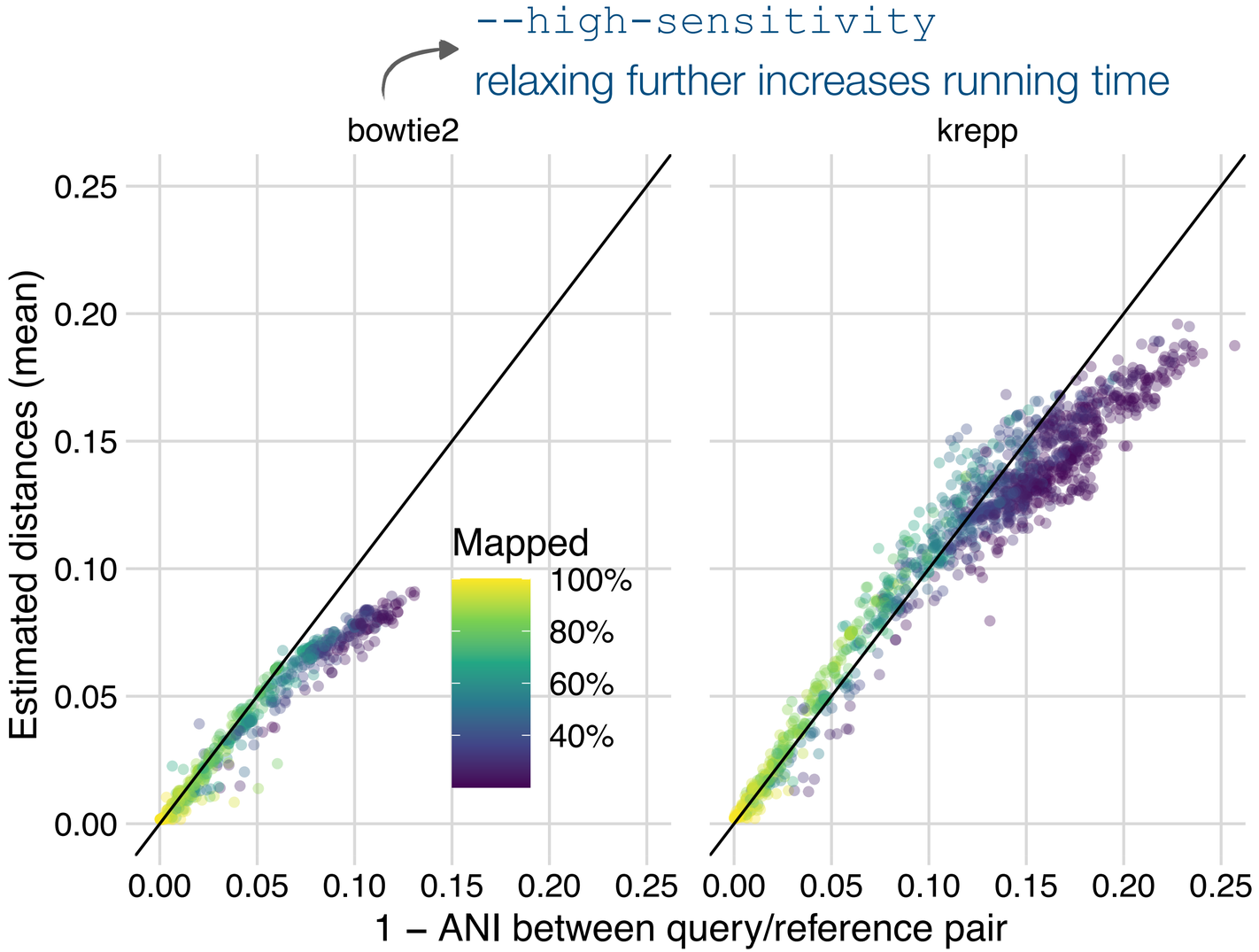


MinHash-based — similar w/ orthoANI etc.

krepp matches genome-wide nucleotide identity on average

Reference: Web of Life (v2)
16,000 microbial genomes

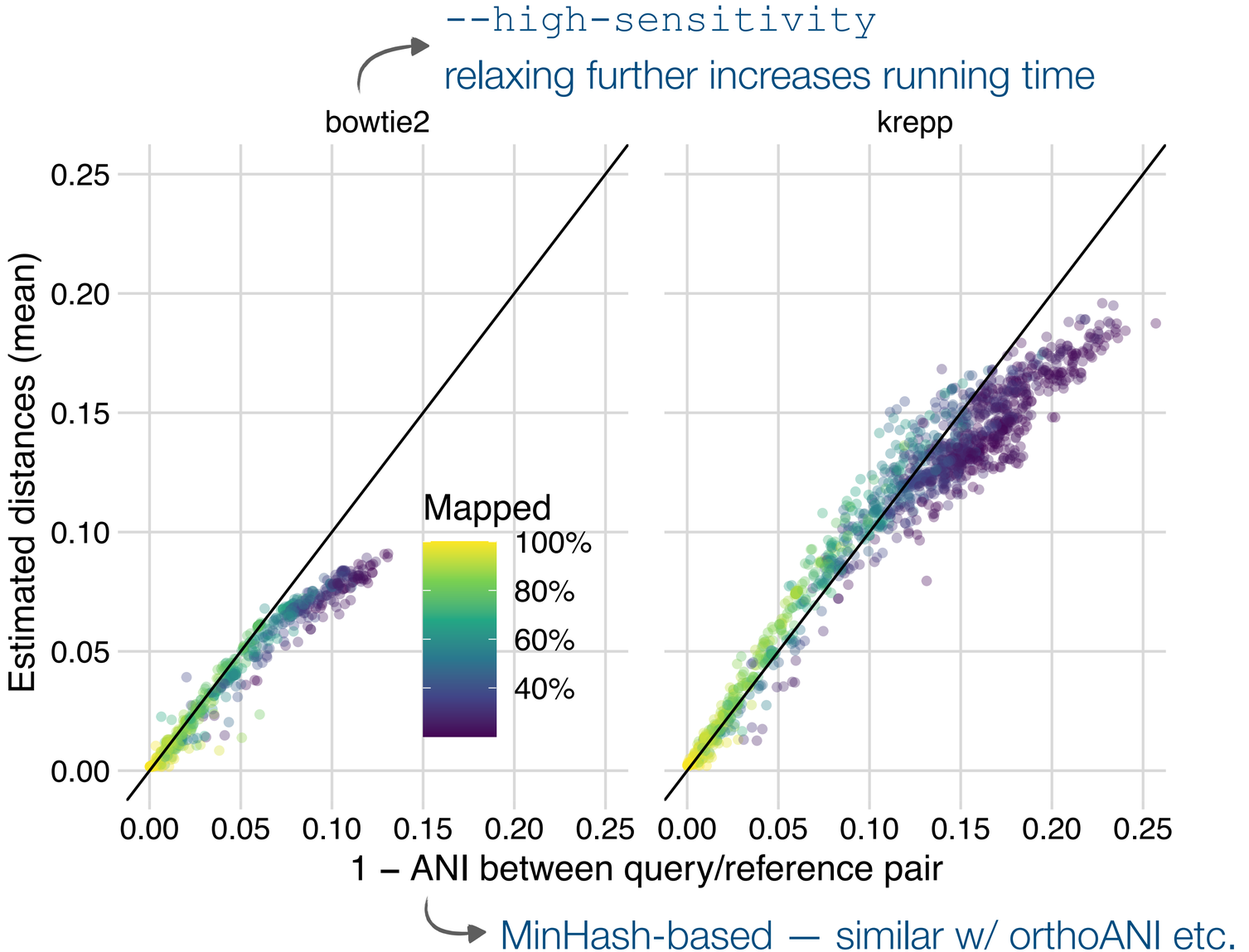
- Real query/reference genomes w/ simulated short reads



krepp matches genome-wide nucleotide identity on average

Reference: Web of Life (v2)
16,000 microbial genomes

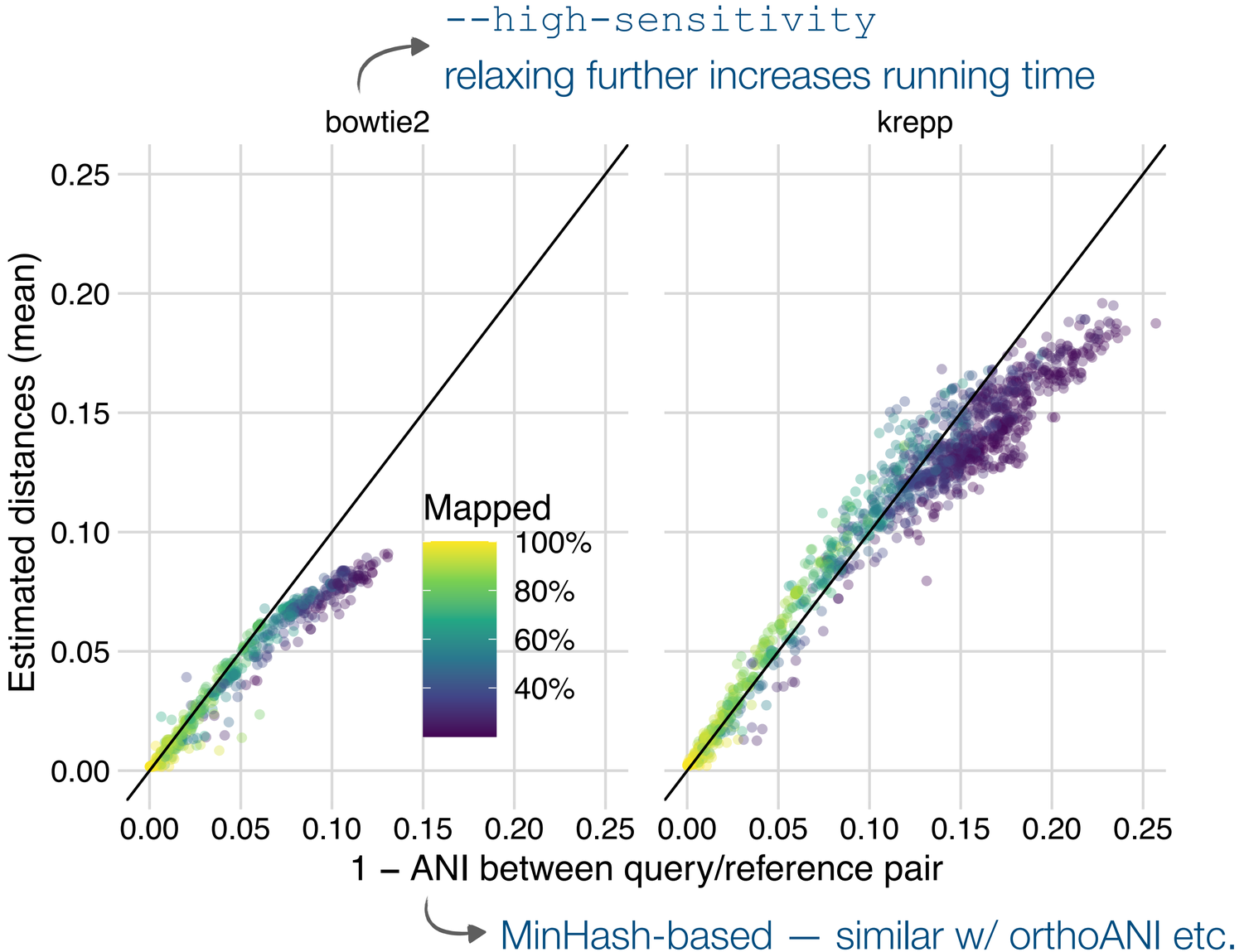
- Real query/reference genomes w/ simulated short reads
- *krepp* extends to more distant reference genomes accurately



krepp matches genome-wide nucleotide identity on average

Reference: Web of Life (v2)
16,000 microbial genomes

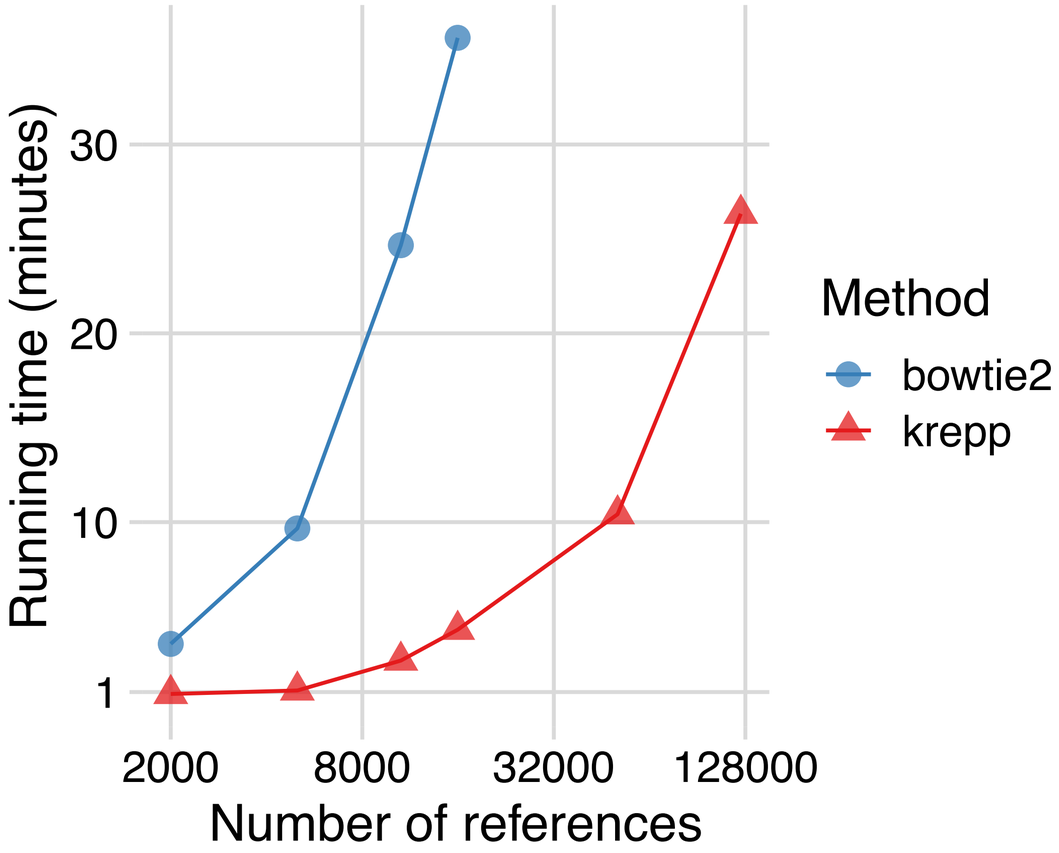
- Real query/reference genomes w/ simulated short reads
- *krepp* extends to more distant reference genomes accurately
- ANI↓: mapping rate ↓
  bias towards similar regions ↑



Scalability:
Avoiding the more difficult problem & effective parallelization

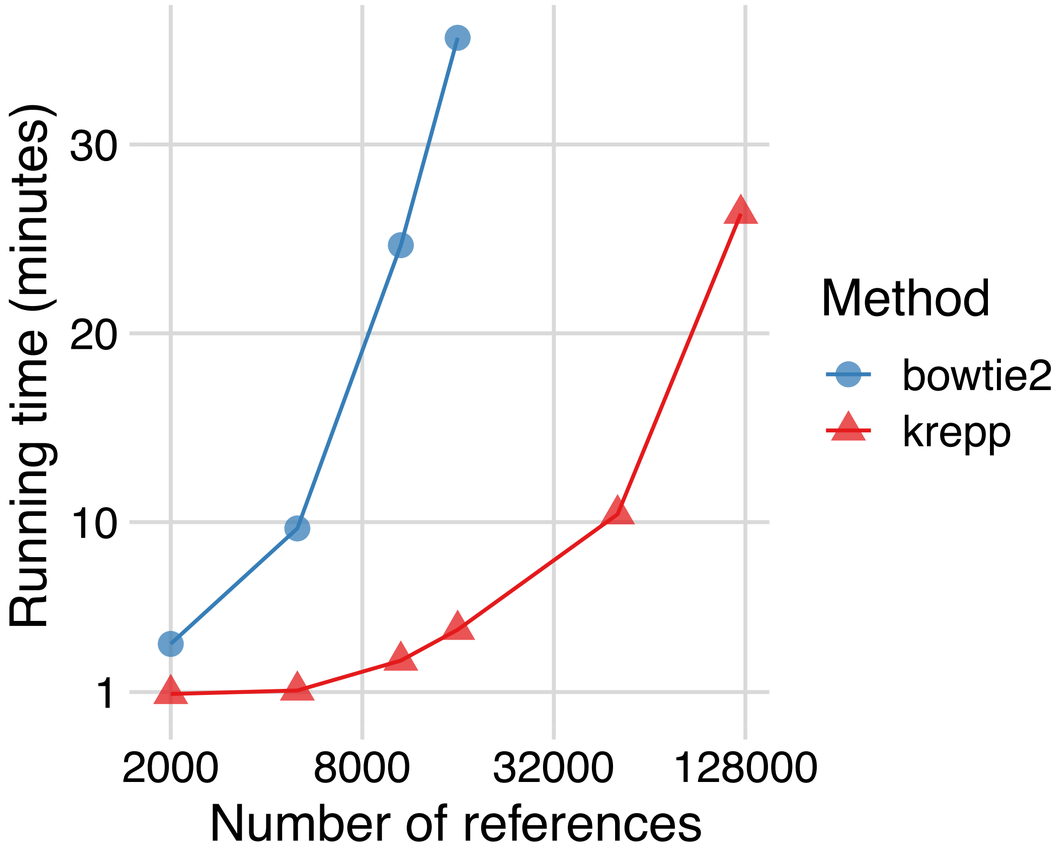
Scalability:
Avoiding the more difficult problem & effective parallelization

Mapping 10M reads (16 threads):

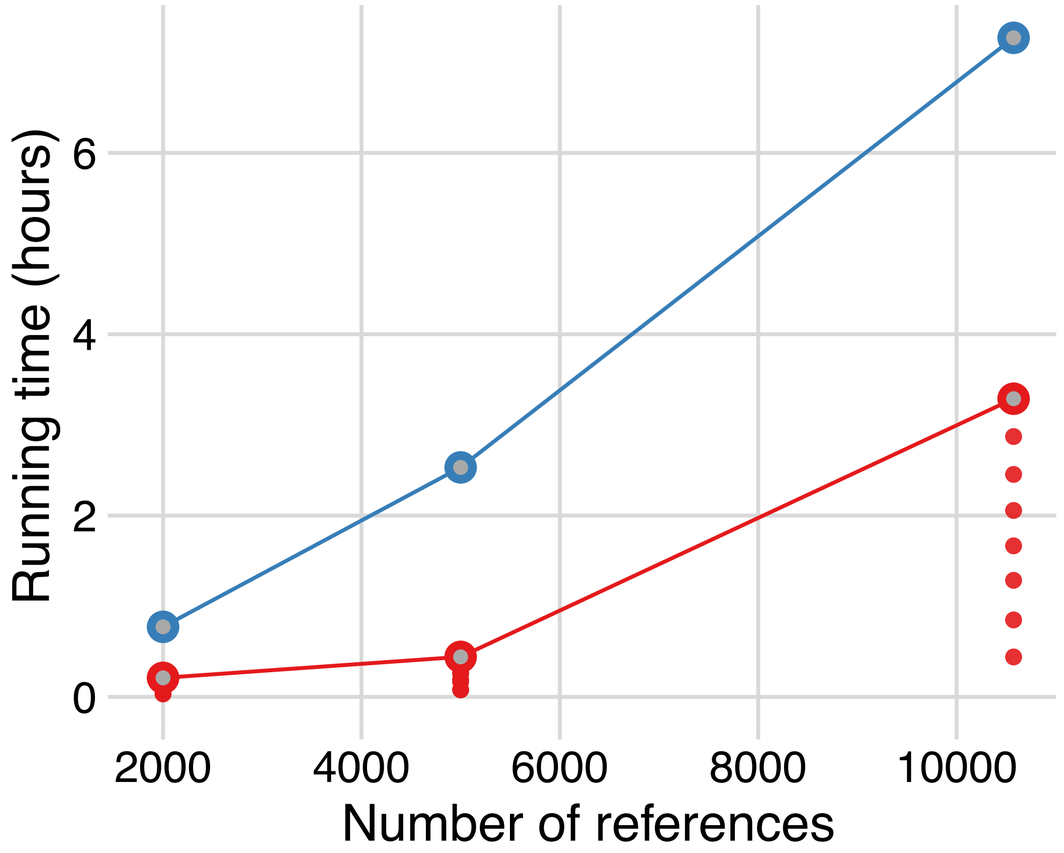


Scalability:
Avoiding the more difficult problem & effective parallelization

Mapping 10M reads (16 threads):

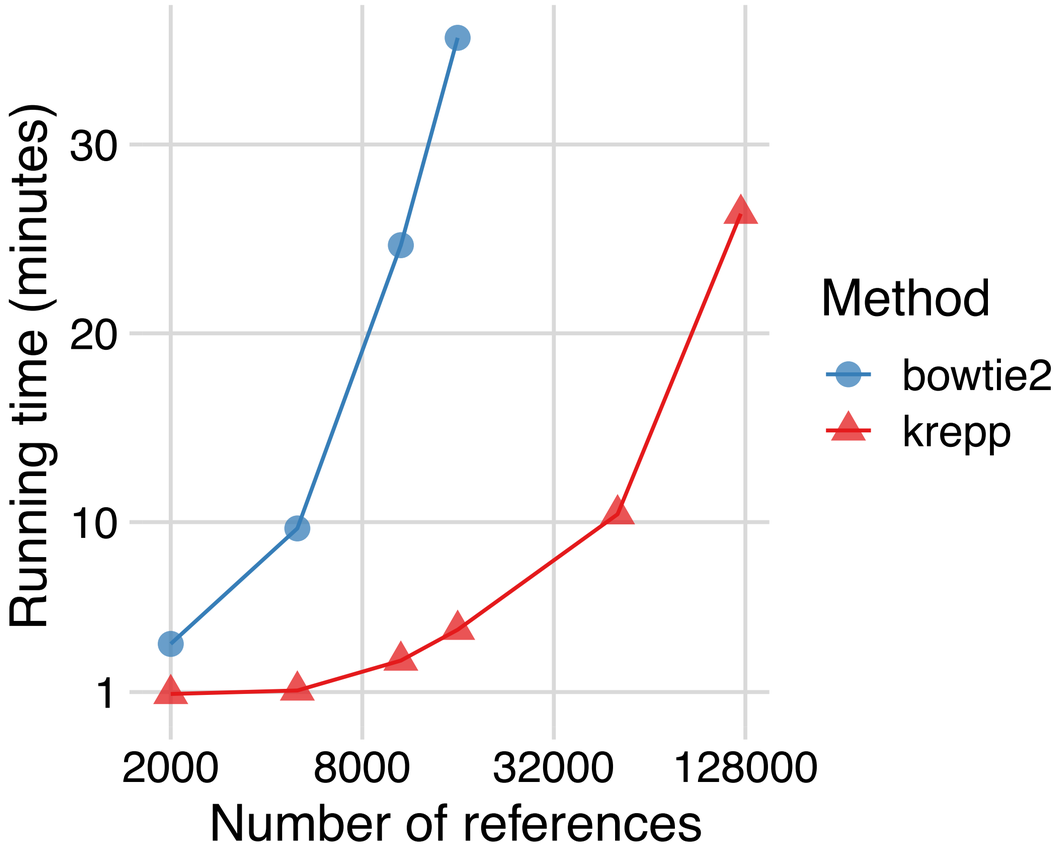


Indexing microbial genomes (32 threads):

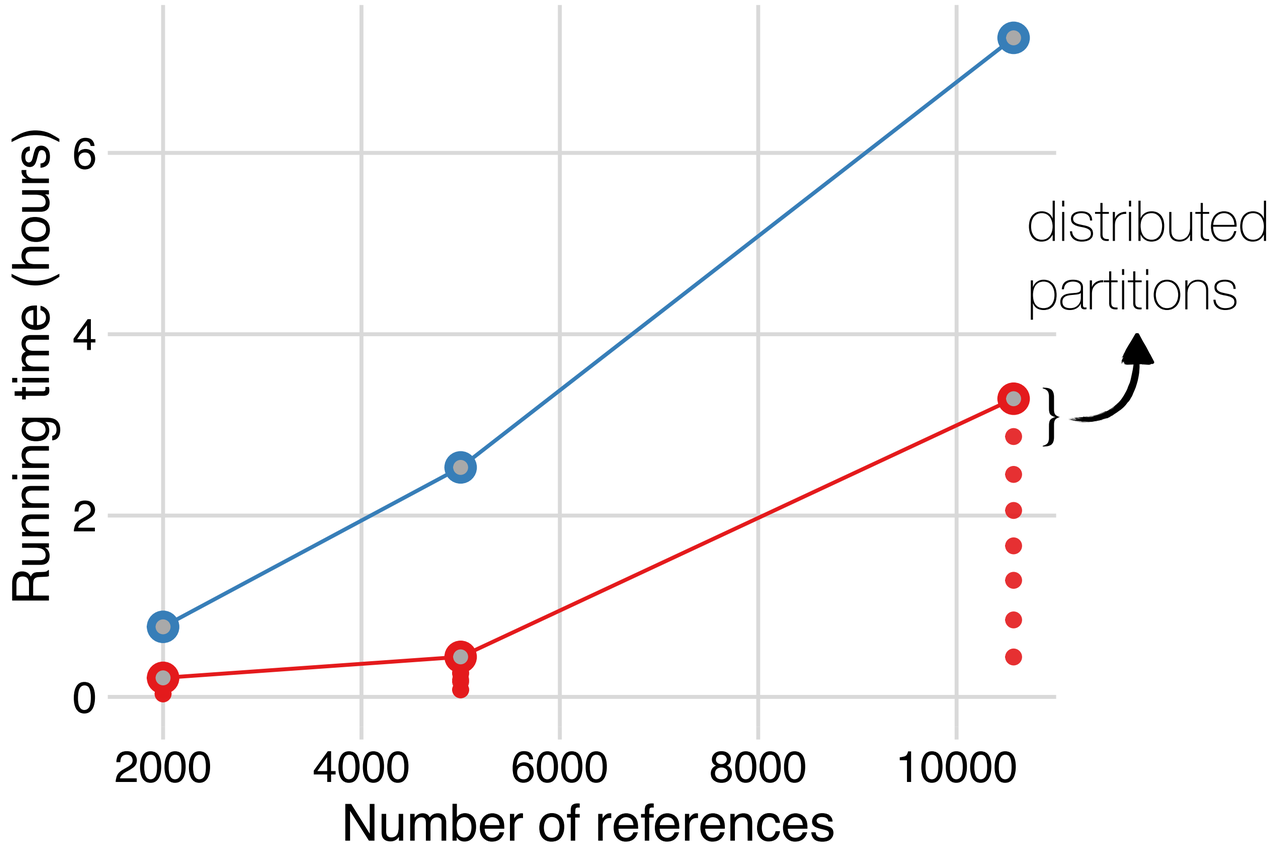


Scalability:
Avoiding the more difficult problem & effective parallelization

Mapping 10M reads (16 threads):

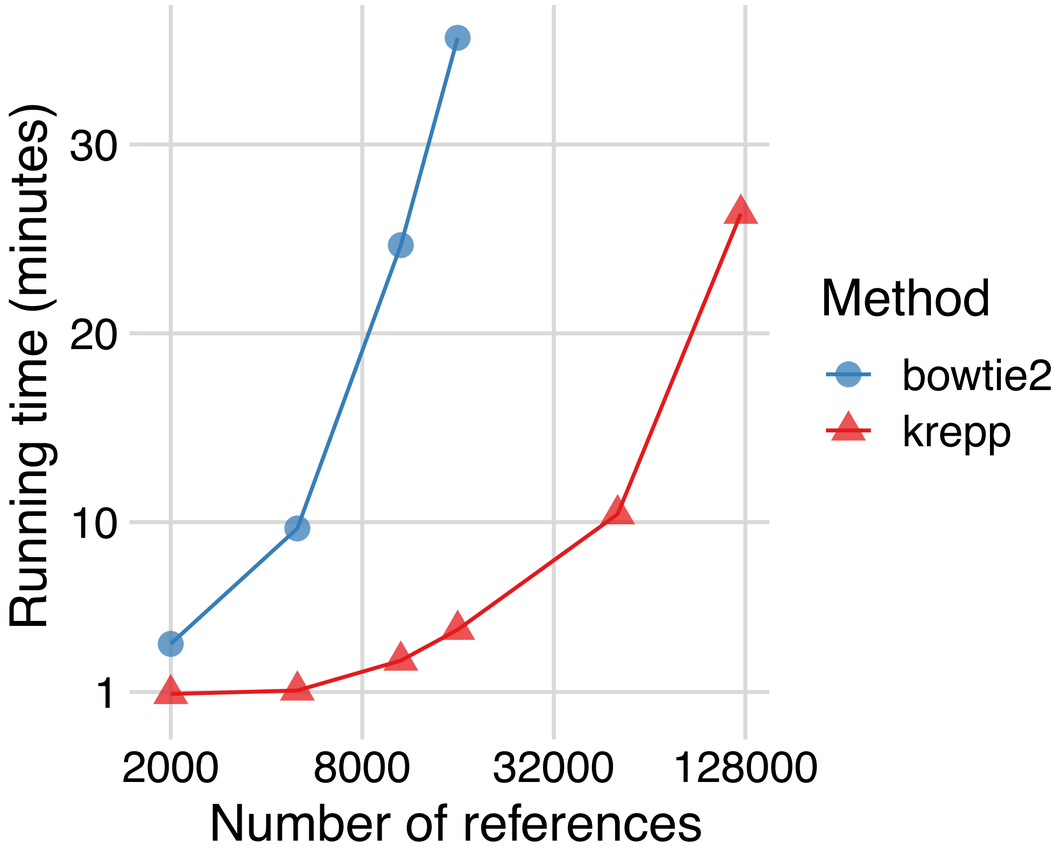


Indexing microbial genomes (32 threads):

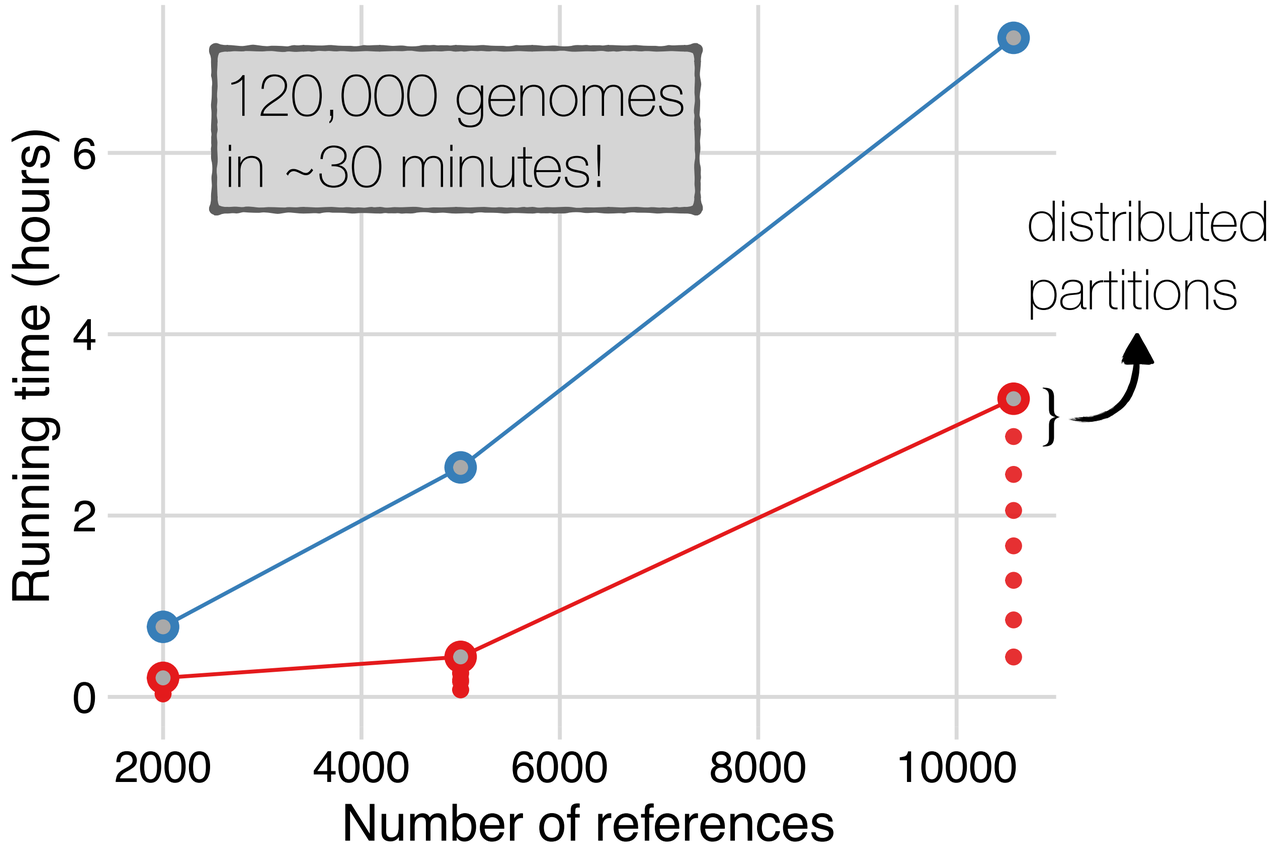


Scalability:
Avoiding the more difficult problem & effective parallelization

Mapping 10M reads (16 threads):



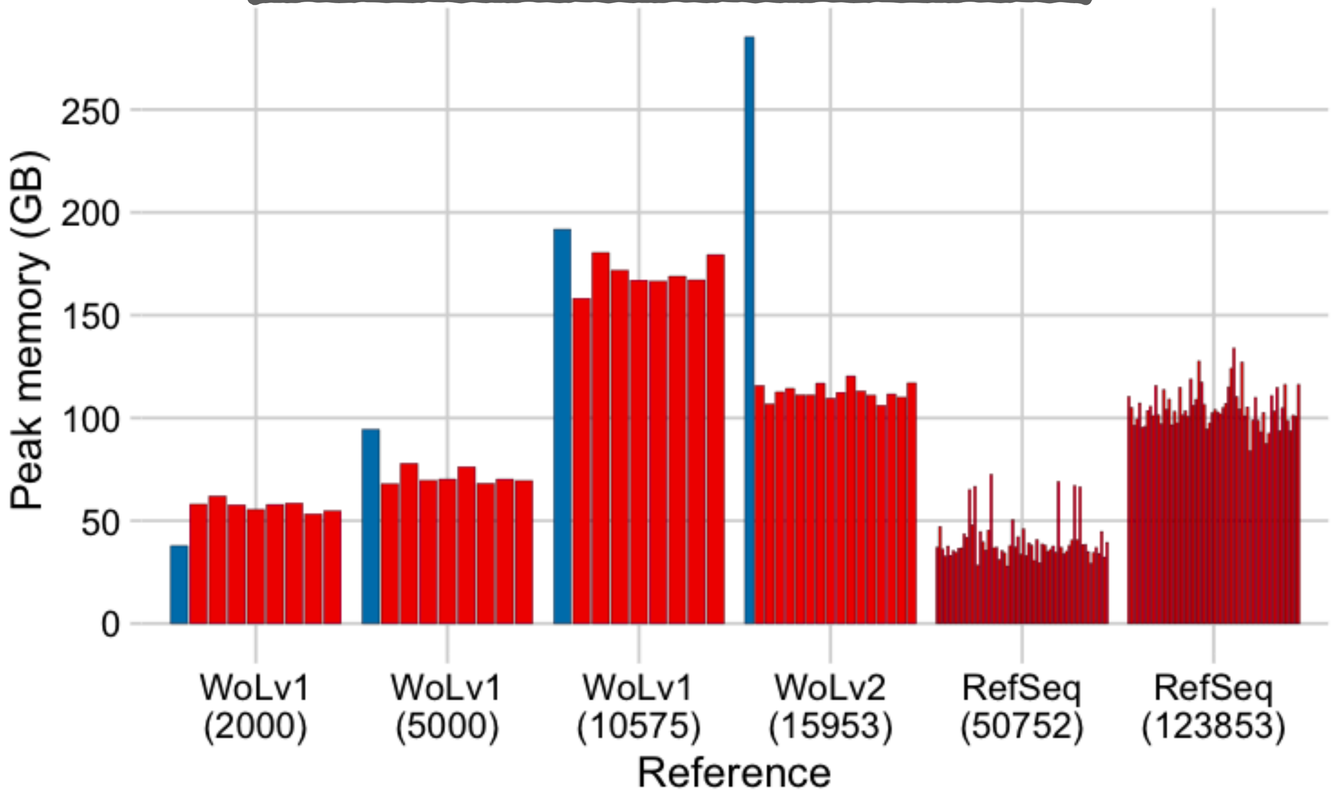
Indexing microbial genomes (32 threads):



Scalability:
krepp can be distributed and has flexible memory requirements

Indexing microbial genomes (32 threads):

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



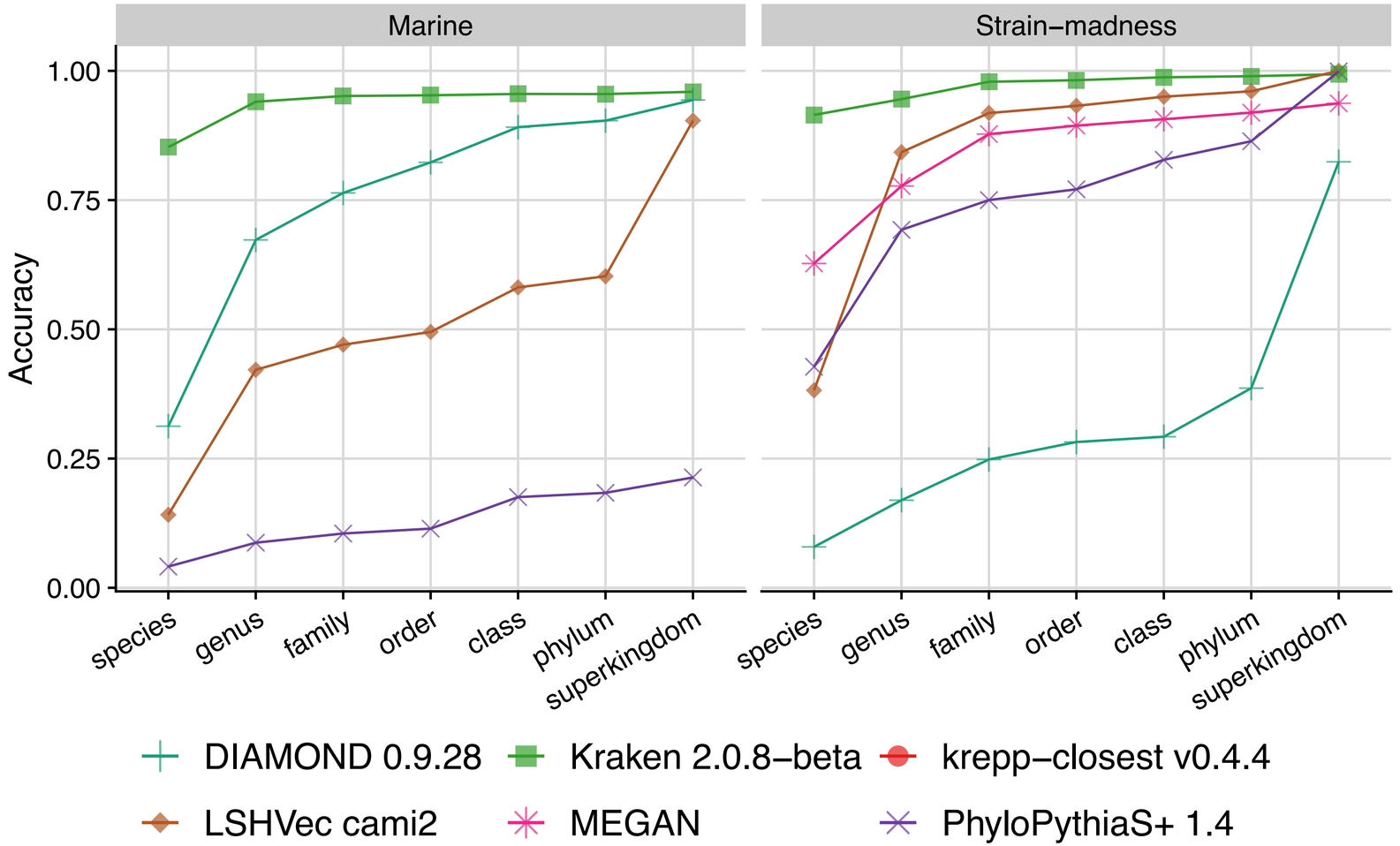
What about longer sequences?
Taxonomic binning of contigs in CAMI-II

What about longer sequences?
Taxonomic binning of contigs in CAMI-II

- **CAMI-II:** Binning contigs to taxonomic groups

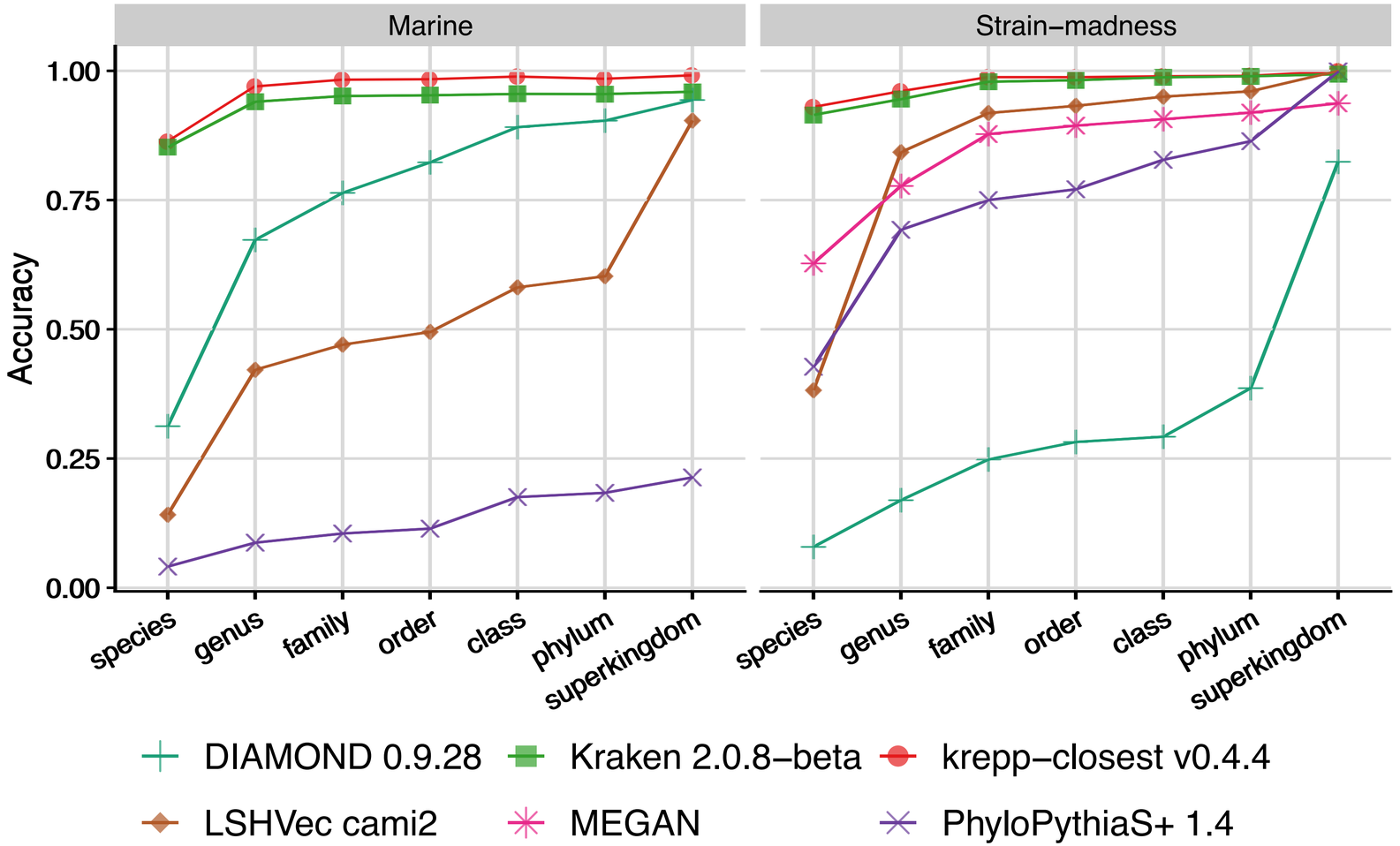
What about longer sequences?
Taxonomic binning of contigs in CAMI-II

- **CAMI-II:** Binning contigs to taxonomic groups
- Simulated data mimicking two different environments



What about longer sequences?
Taxonomic binning of contigs in CAMI-II

- **CAMI-II:** Binning contigs to taxonomic groups
- Simulated data mimicking two different environments
- *krepp* assigns each contig to the closest genome's group



Dealing with uncertainty: statistically distinguishability

Dealing with uncertainty: statistically distinguishability

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

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:

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

D: alternative distance

$i*$: closest reference

Dealing with uncertainty: statistically distinguishability

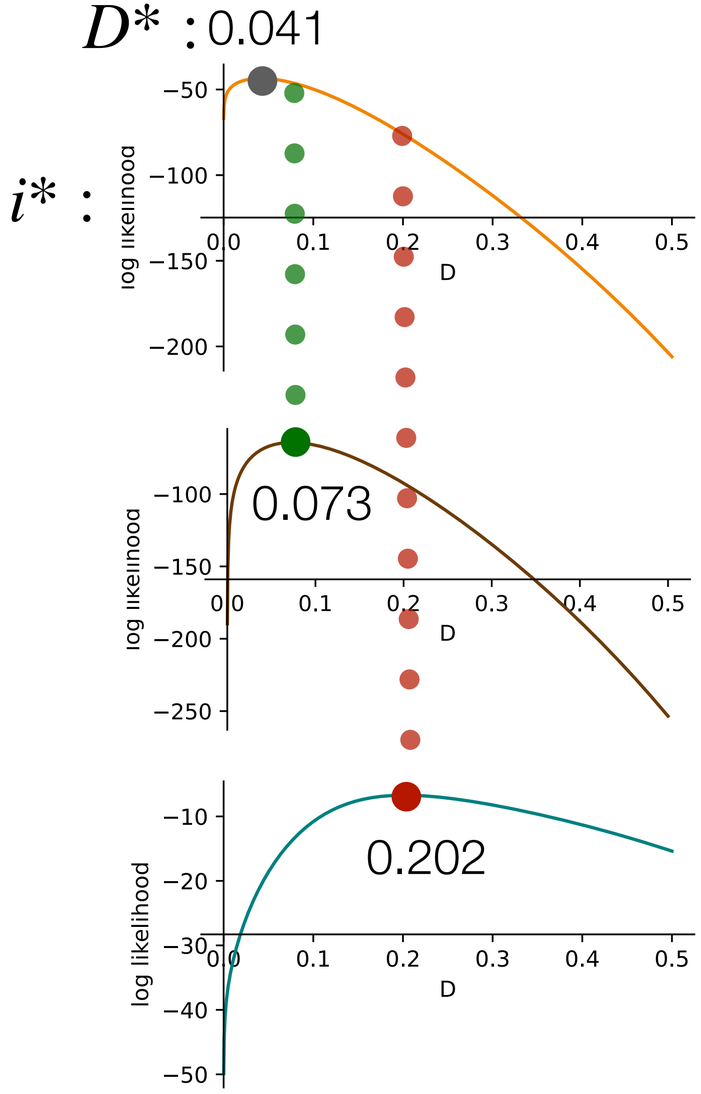
- short reads — **low signal**
- high distances — fewer matching *k*-mers
- small differences may not be statistically meaningful
  - **test distinguishability**

likelihood-ratio test
with the closest reference:

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

D: alternative distance

$i*$: 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:

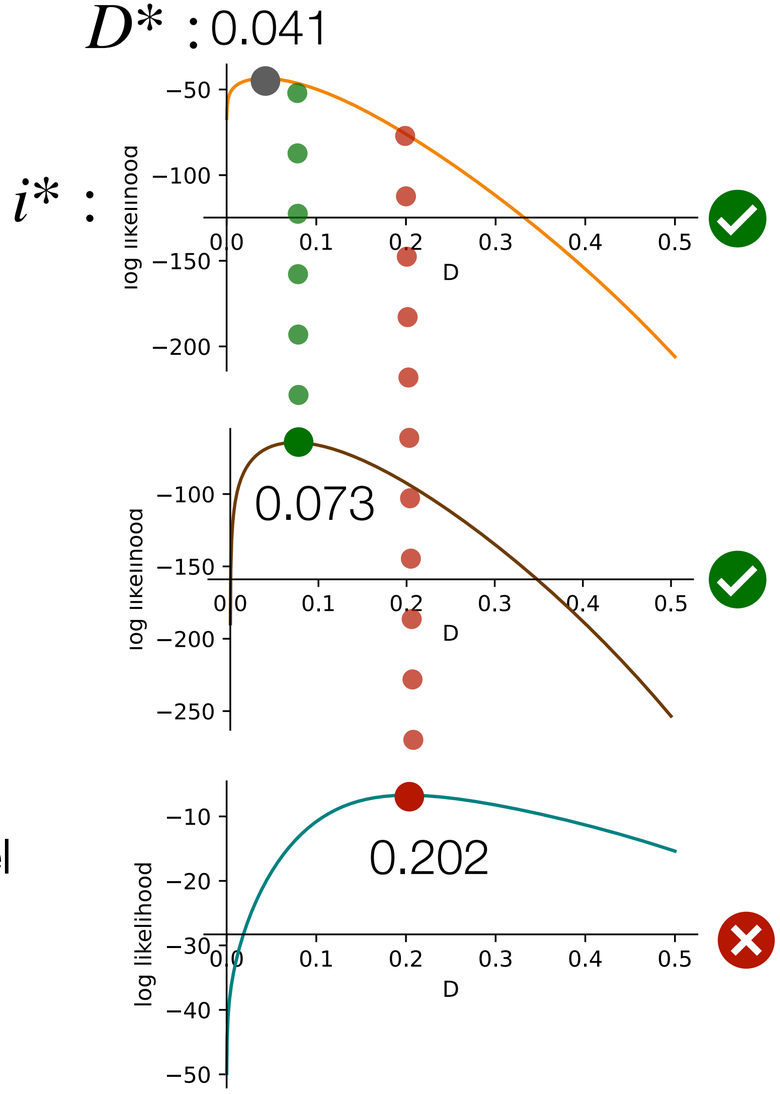
$$\lambda_{LR} = \frac{\mathscr{L}_{i*}(D; k, h, \delta, u_{i*}, \mathbf{v}_{i*})}{\mathscr{L}_{i*}(D^*; k, h, \delta, u_{i*}, \mathbf{v}_{i*})}$$

D: alternative distance

i^*: closest reference

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

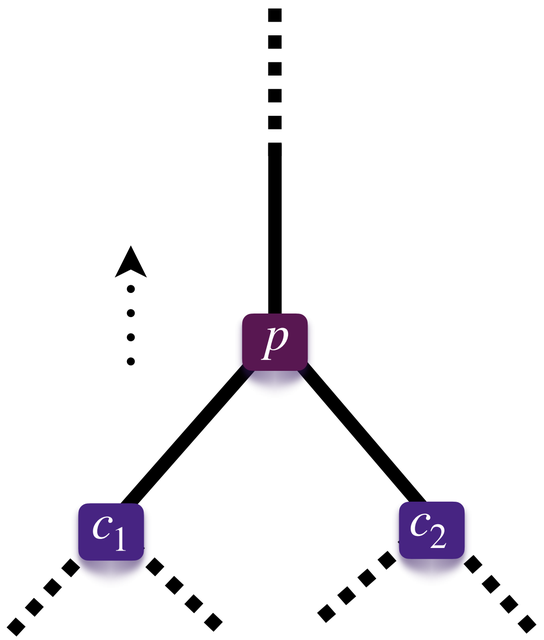
- select a significance level (default: α=90%)



Defining clade distances & branch length agnostic placement

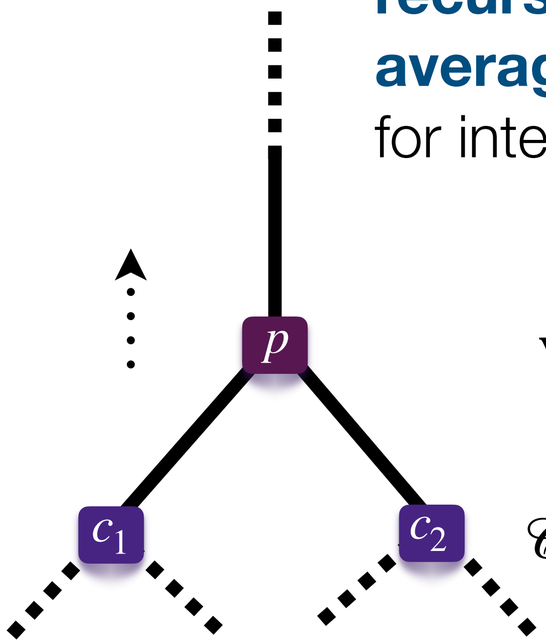
Defining clade distances & branch length agnostic placement

A notion of distance for internal nodes:



Defining clade distances & branch length agnostic placement

A notion of distance for internal nodes:



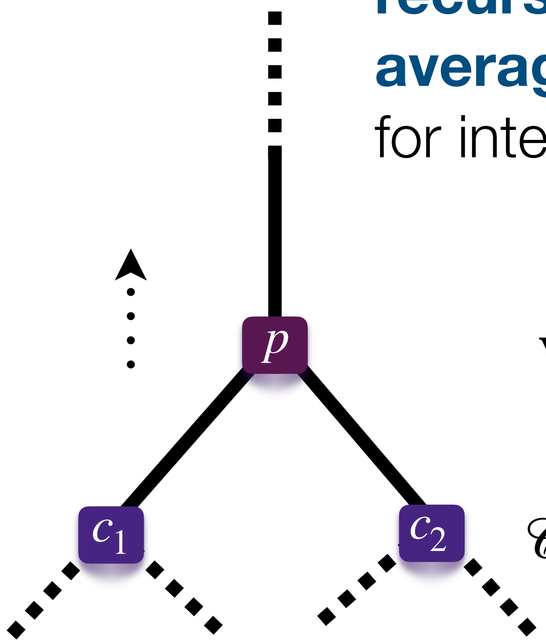
recursively compute **average** HD histograms for internal nodes

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

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

Defining clade distances & branch length agnostic placement

A notion of distance for internal nodes:



recursively compute average HD histograms for internal nodes

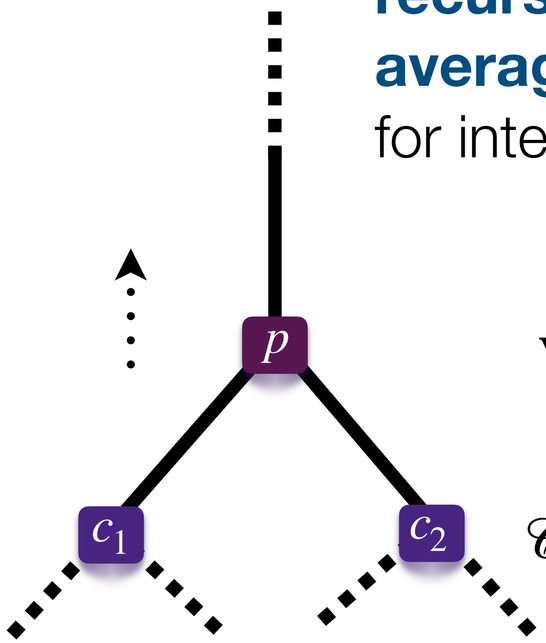
$$\mathbf{v}_p = \frac{\sum_{c \in \mathscr{C}(p)} \mathbf{v}_c}{|\mathscr{C}(p)|}$$

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

use the same likelihood model and log-likelihood ratio test

Defining clade distances & branch length agnostic placement

A notion of distance for internal nodes:

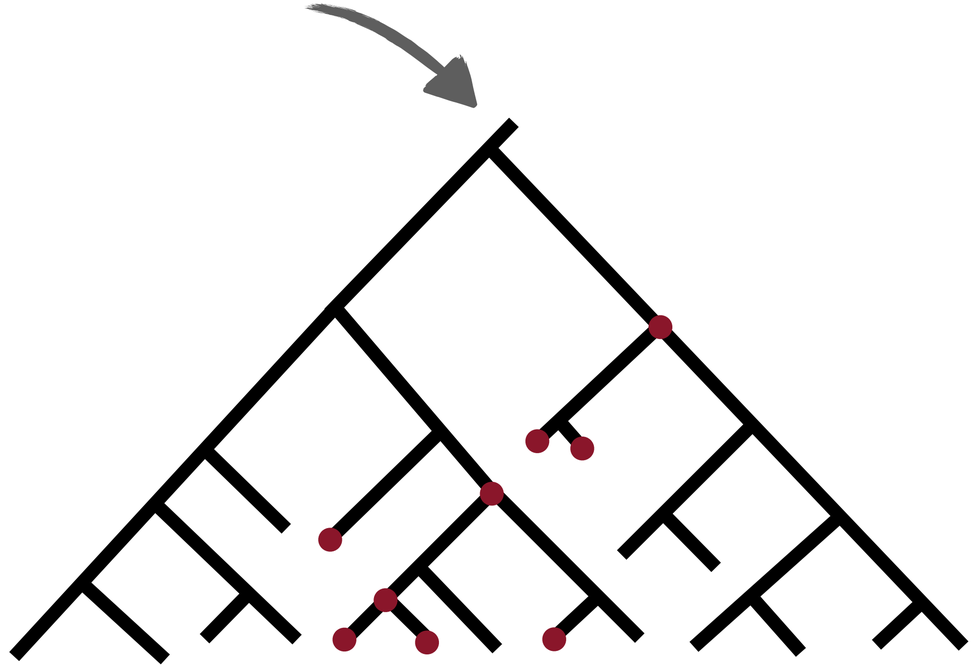


recursively compute average HD histograms for internal nodes

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

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

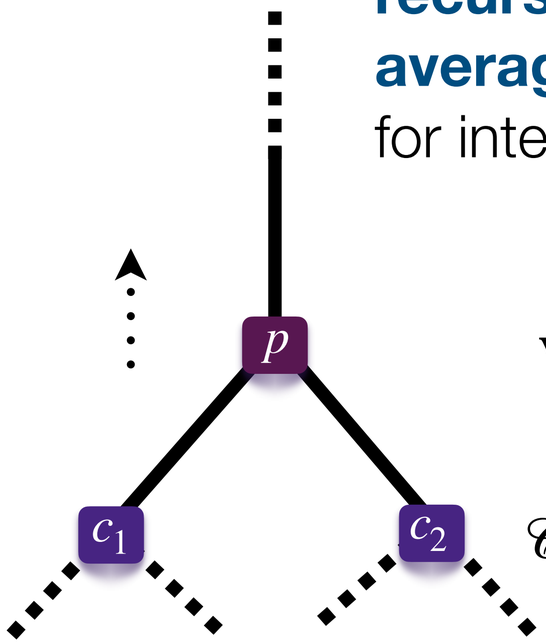
use the same likelihood model and log-likelihood ratio test



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

Defining clade distances & branch length agnostic placement

A notion of distance for internal nodes:



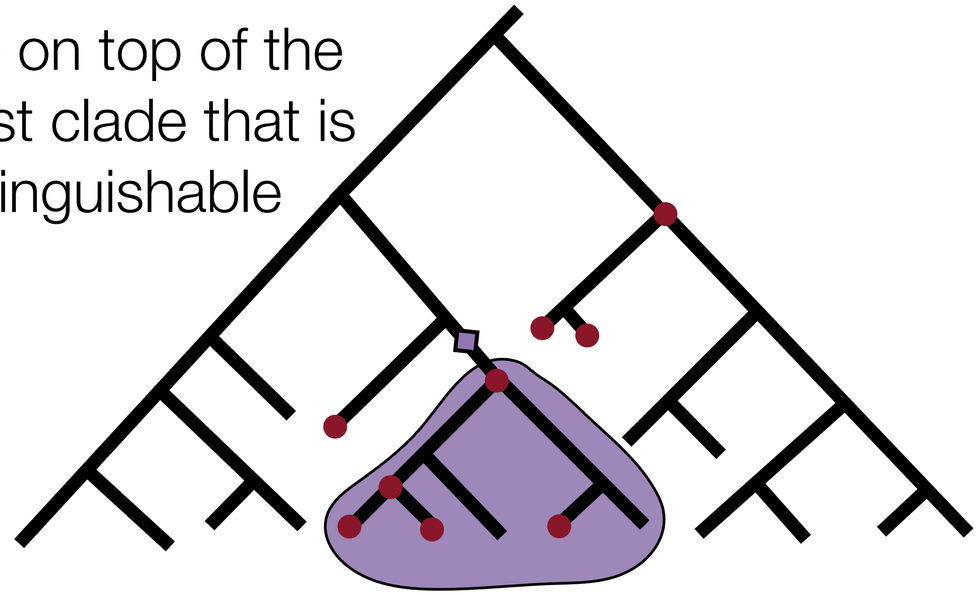
recursively compute average HD histograms for internal nodes

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

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

use the same likelihood model and log-likelihood ratio test

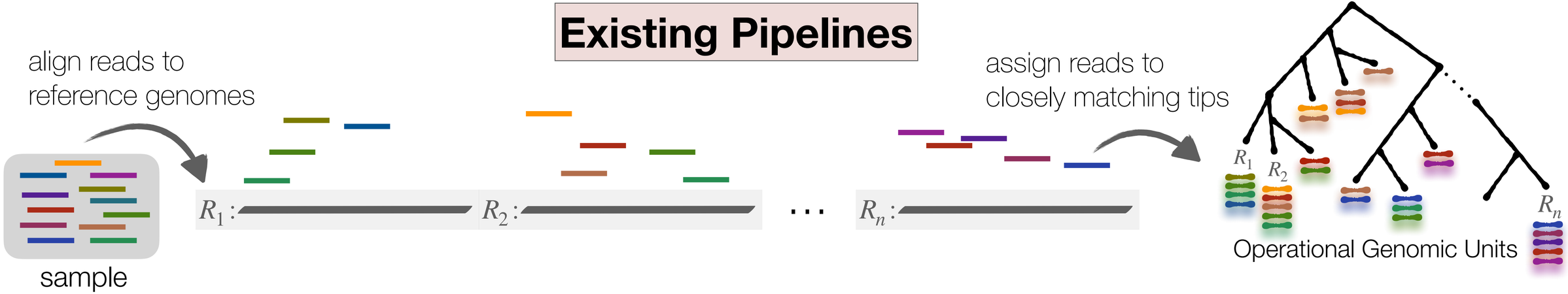
place on top of the largest clade that is indistinguishable



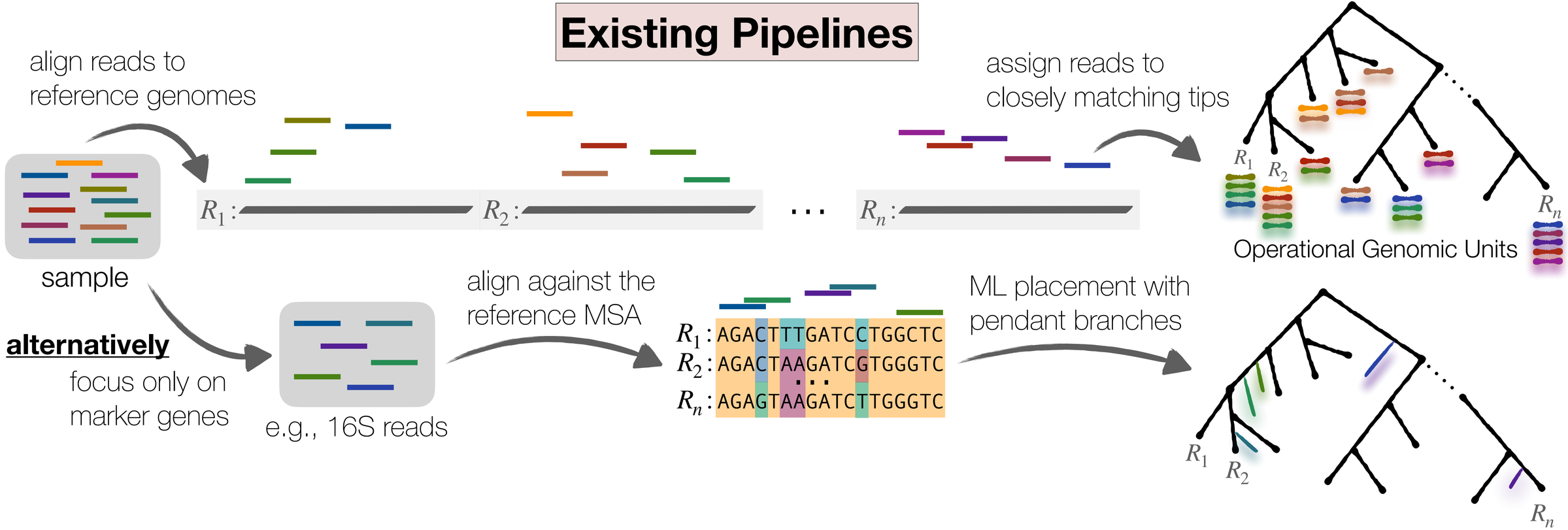
● : indistinguishable w.r.t. the closest reference

Existing Pipelines

Existing Pipelines



Existing Pipelines



Existing Pipelines

align reads to reference genomes

sample

R_1 : R_2 : ... R_n :

assign reads to closely matching tips

R_1 R_2 ... R_n

Operational Genomic Units

alternatively

focus only on marker genes

e.g., 16S reads

align against the reference MSA

R_1 : AGACTTTGATCCTGGCTC
R_2 : AGACTAAGATCGTGGGTC
...
R_n : AGAGTAAGATCTTGGGTC

ML placement with pendant branches

R_1 R_2 ... R_n

krepp

compute HD of *k*-mer matches

sample

TCCC|...|TCCA
AAAC|...|AAAT
TACC|...|AACC
...
AAAT|...|AAAC

k-mer index

R_1 R_2 R_3 ... R_n

color index

R_i

0 1 2 3

$\hat{D}$

likelihood model

reference genomes:

R_1 : ATGTAACGAGGCAGTATCCG
R_2 : CAATGGCGGTTTTTTACCCT
...
R_n : GCGAGGATCCGAGAAGGCGC

i) estimate distances of reads to *all* references

R_1	R_2	...	R_n
0.01	0.02	...	NA
0.03	0.12	...	0.06
NA	0.05	...	0.02
0.18	0.04	...	0.21
0.02	NA	...	0.05
NA	0.19	...	0.01
NA	NA	...	0.02

R_1 R_2 ... R_n

Phylogenetic Placements

ii) place reads anywhere on the backbone phylogeny

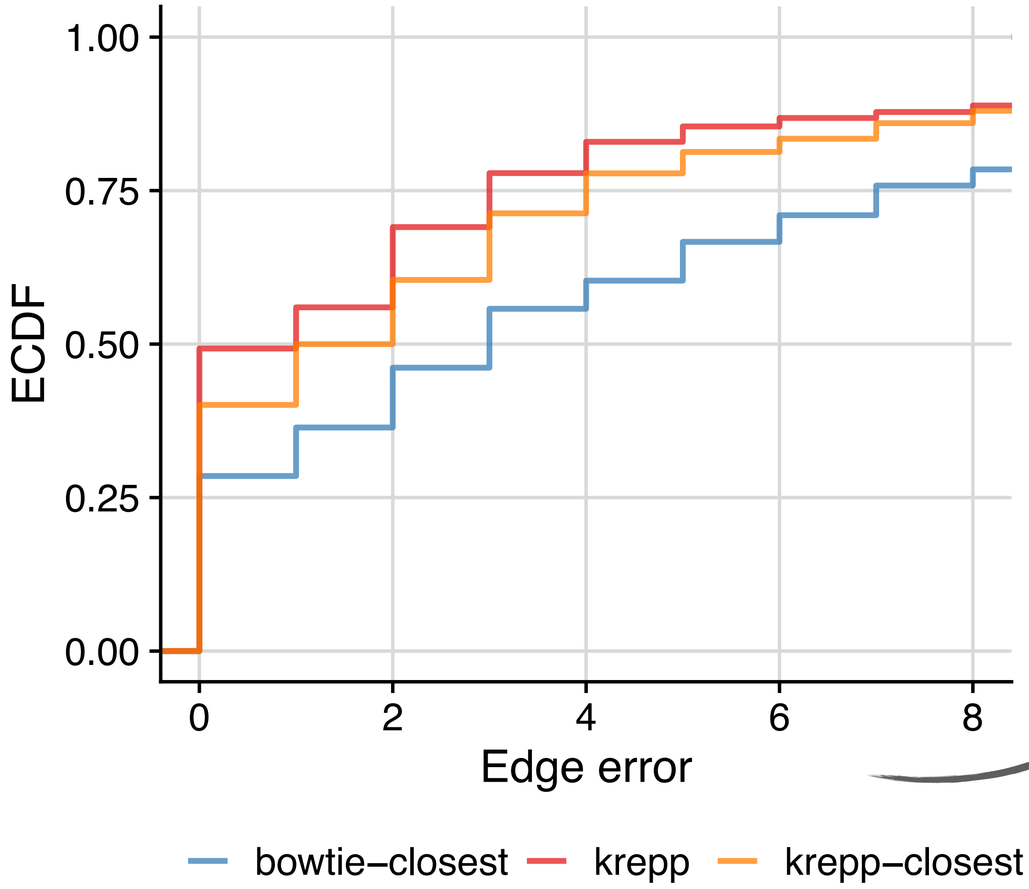
krepp’s heuristic improves closest-tip placement

krepp's heuristic improves closest-tip placement

- Leave all out: 100 diverse taxa
- WoL-v2 tree with 15,952 leaves

krepp's heuristic improves closest-tip placement

- Leave all out: 100 diverse taxa
- WoL-v2 tree with 15,952 leaves
- Outperforms baselines:
  on the closest, on the LCA, etc.
- >80% of all reads within four
  edges of the correct node



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

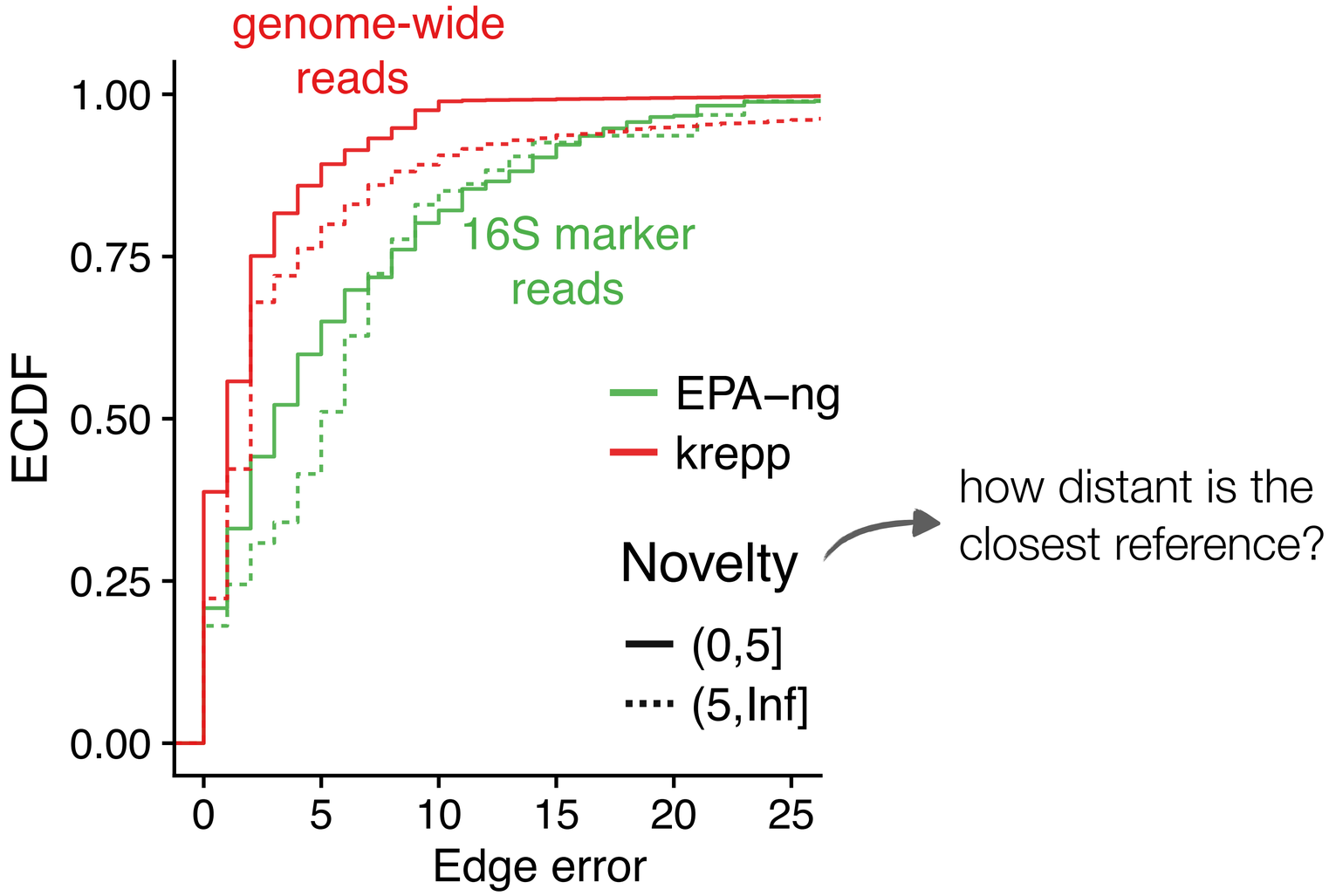
krepp places genome-wide reads more accurately than ML-based placement of 16S reads

krepp places genome-wide reads more accurately than ML-based placement of 16S reads

- Leave all out —100 diverse taxa
- WoL-v1 tree — 10,575 leaves

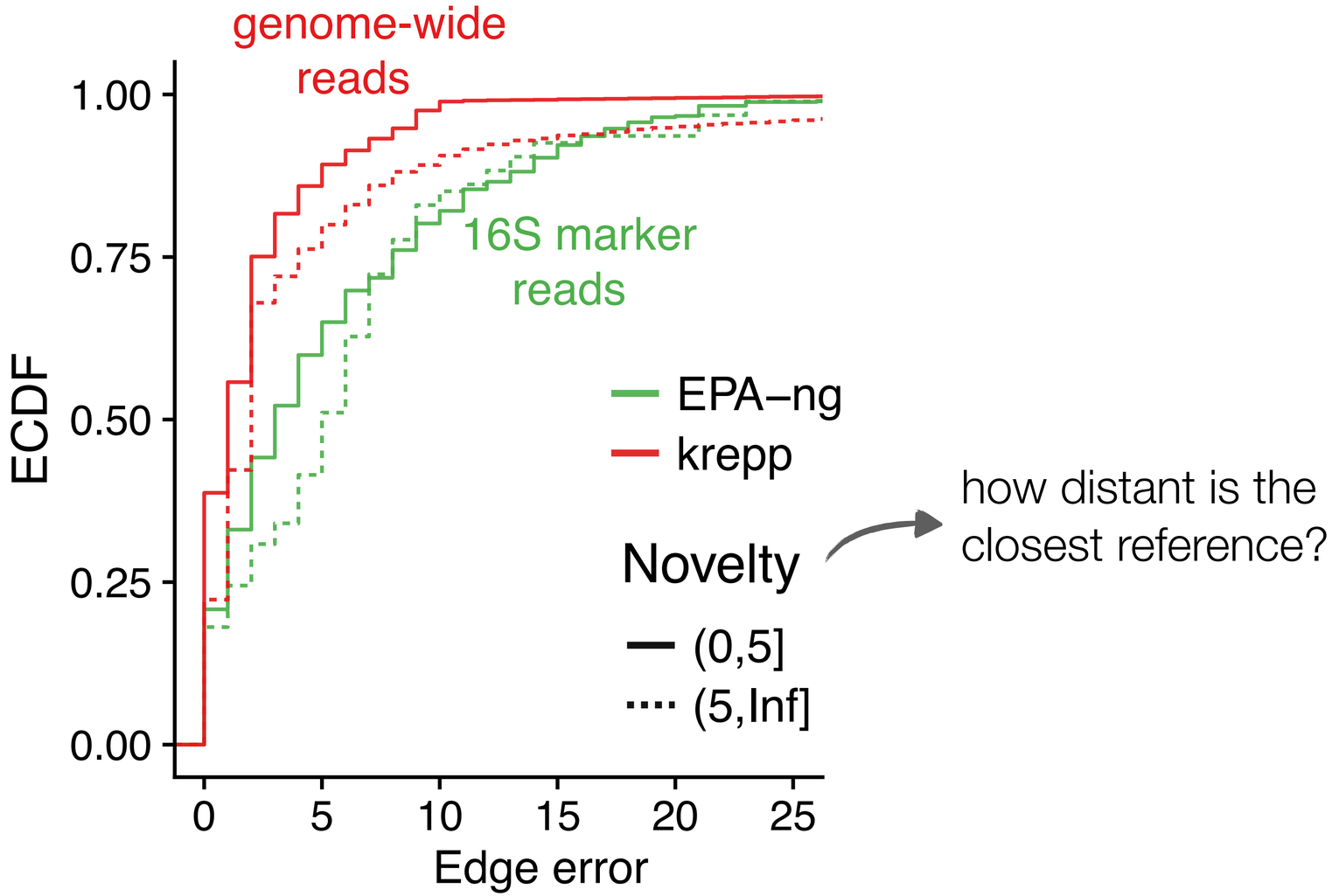
krepp places genome-wide reads more accurately than ML-based placement of 16S reads

- Leave all out —100 diverse taxa
- WoL-v1 tree — 10,575 leaves



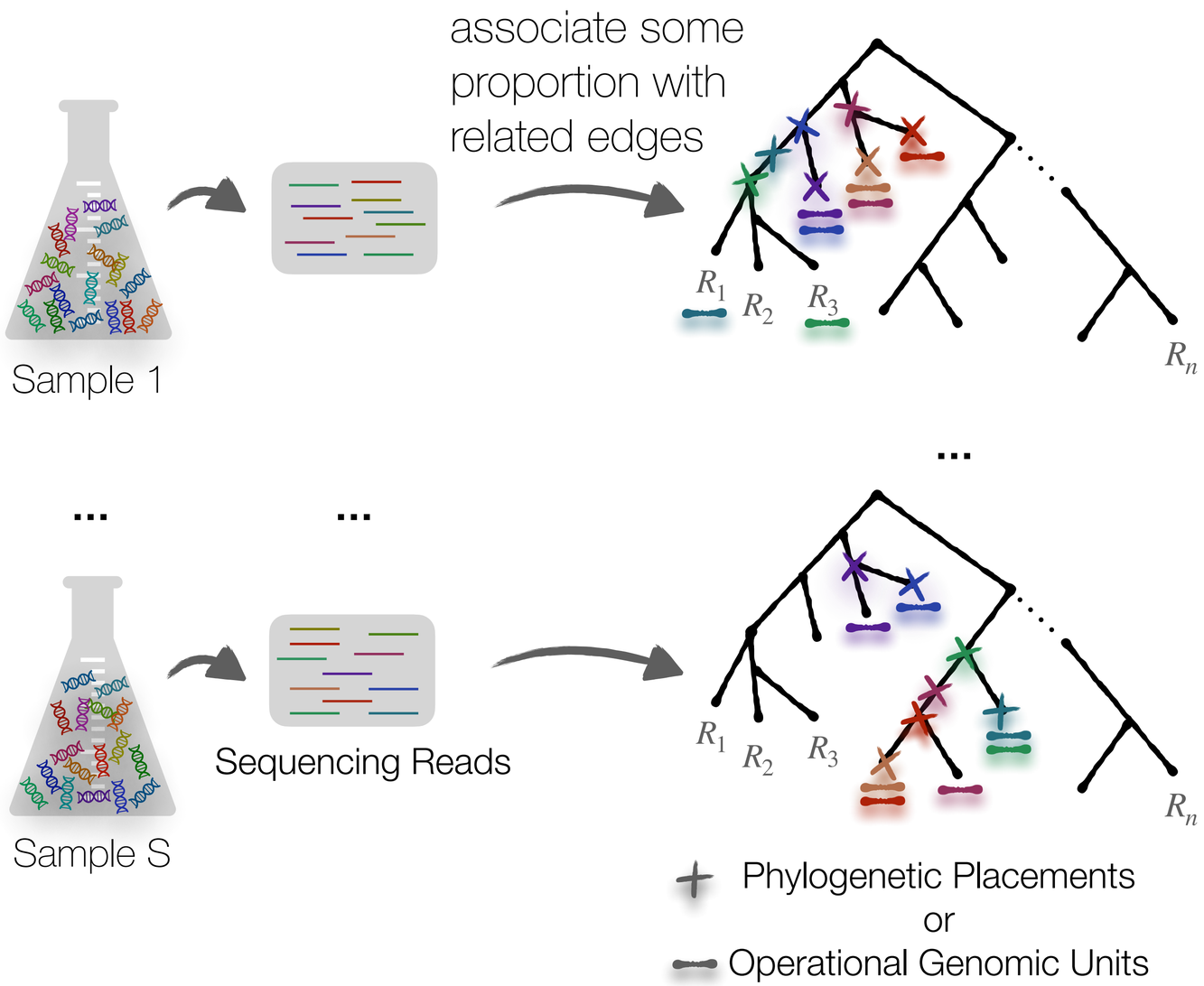
krepp places genome-wide reads more accurately than ML-based placement of 16S reads

- Leave all out —100 diverse taxa
- WoL-v1 tree — 10,575 leaves
- 2.4 versus 5.6 edges (average)
- *krepp* places 86% of all reads

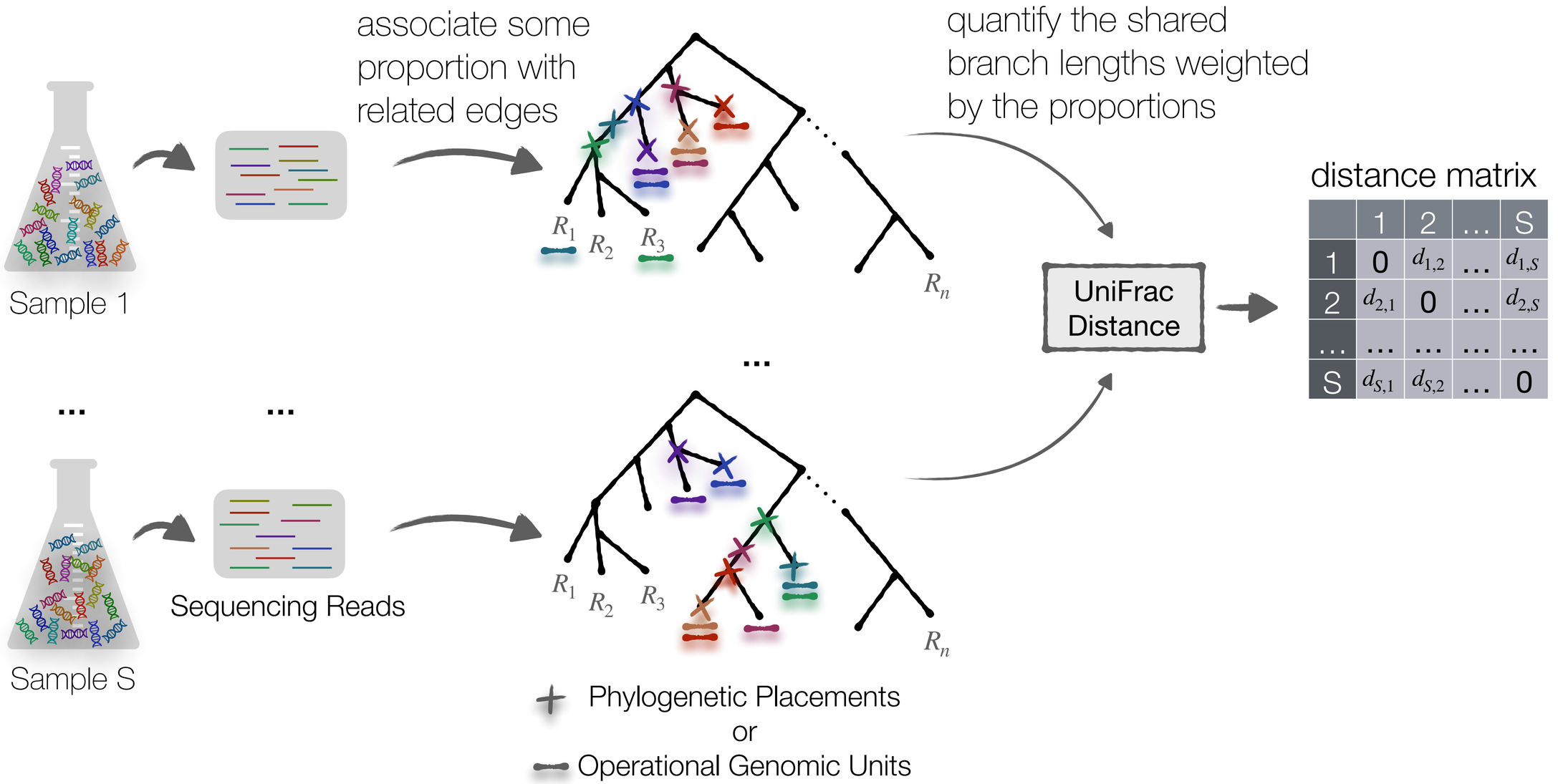


Characterization of metagenomic samples on a backbone phylogeny

Characterization of metagenomic samples on a backbone phylogeny

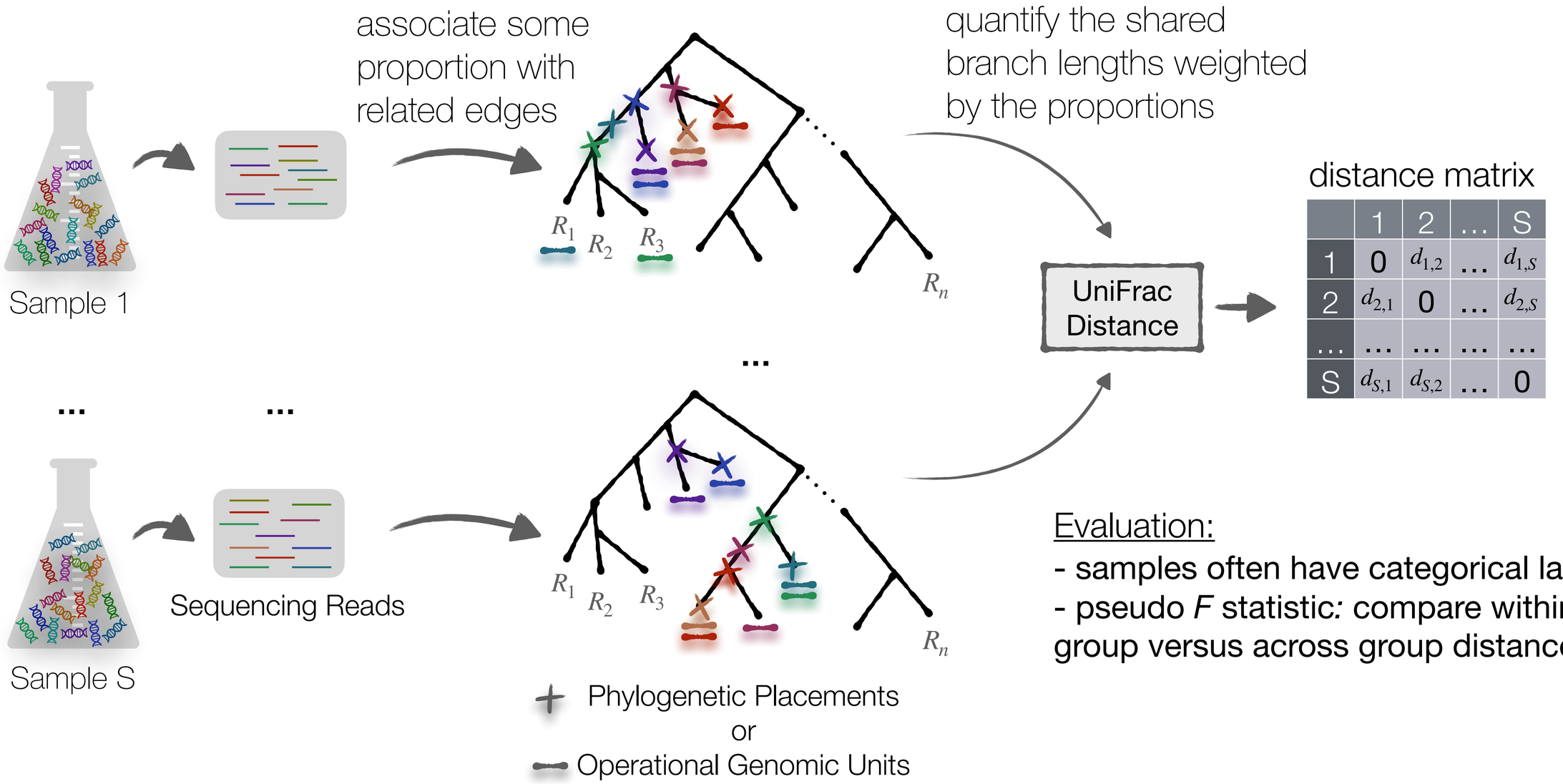


Characterization of metagenomic samples on a backbone phylogeny



	1	2	...	S
1	0	$d_{1,2}$	...	$d_{1,S}$
2	$d_{2,1}$	0	...	$d_{2,S}$
...	...	...	...	...
S	$d_{S,1}$	$d_{S,2}$	...	0

Characterization of metagenomic samples on a backbone phylogeny

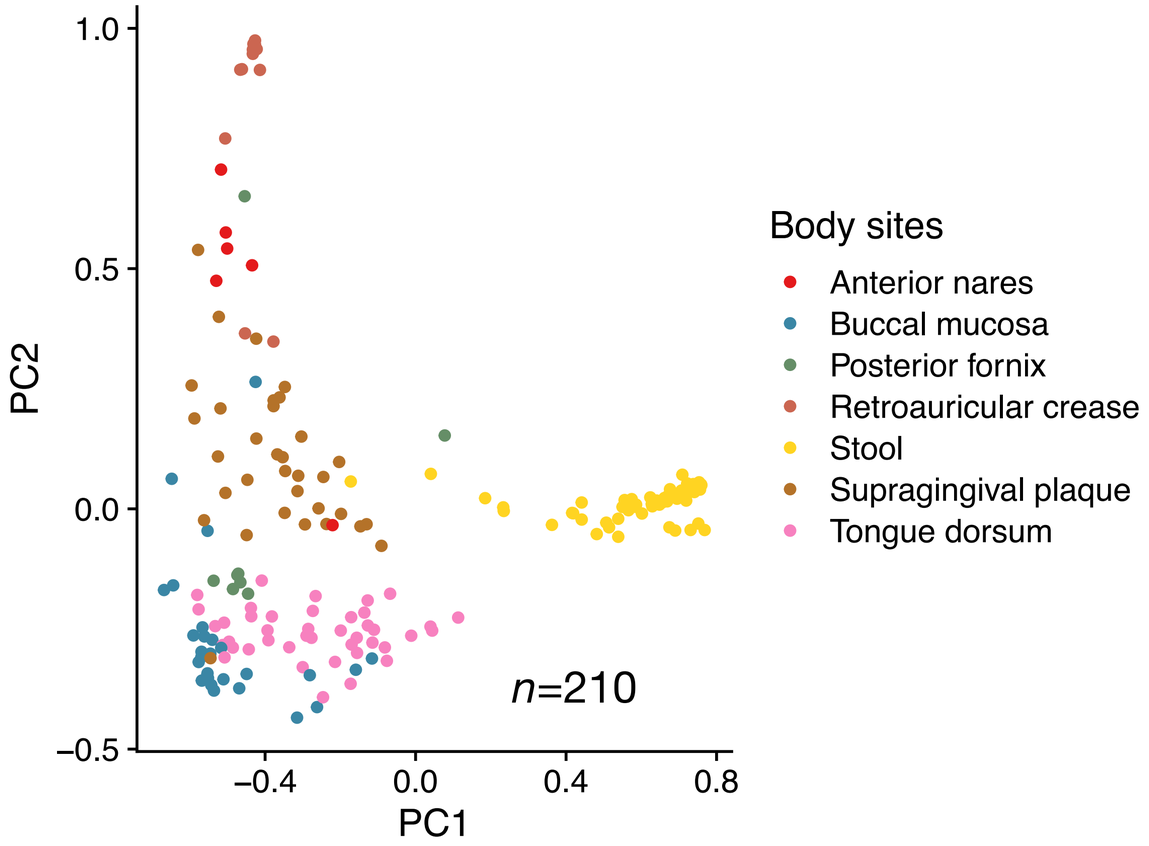


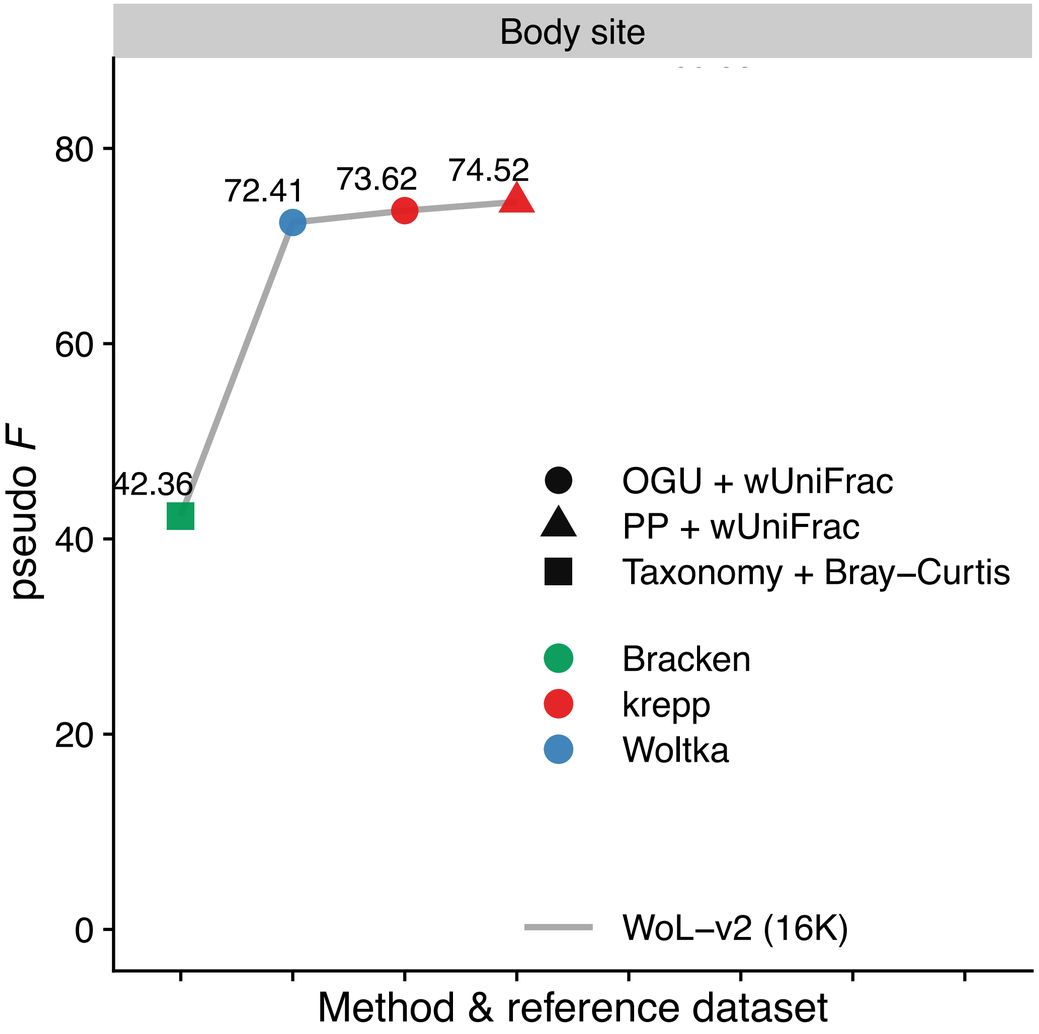
	1	2	…	S
1	0	$d_{1,2}$	…	$d_{1,S}$
2	$d_{2,1}$	0	…	$d_{2,S}$
…	…	…	…	…
S	$d_{S,1}$	$d_{S,2}$	…	0

Evaluation:

- samples often have categorical labels
- pseudo *F* statistic*:* compare within group versus across group distances

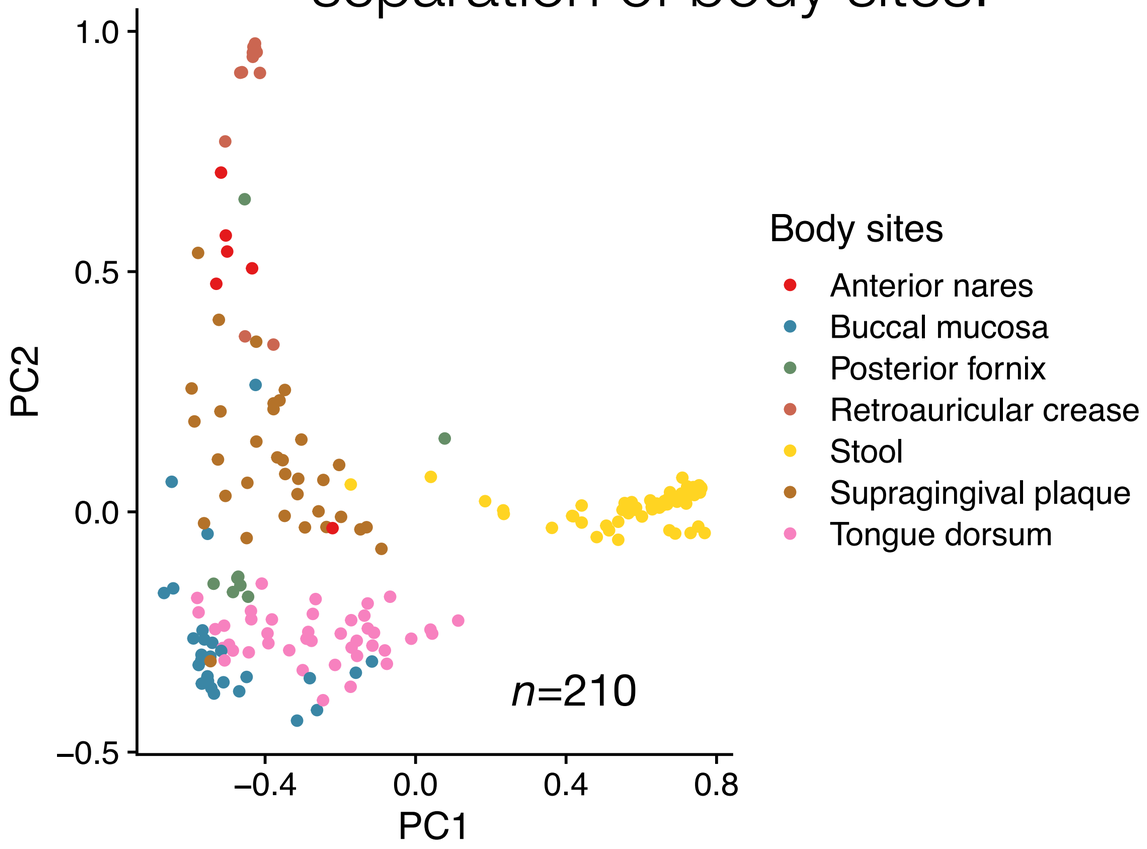
Analyzing human microbiome with larger references

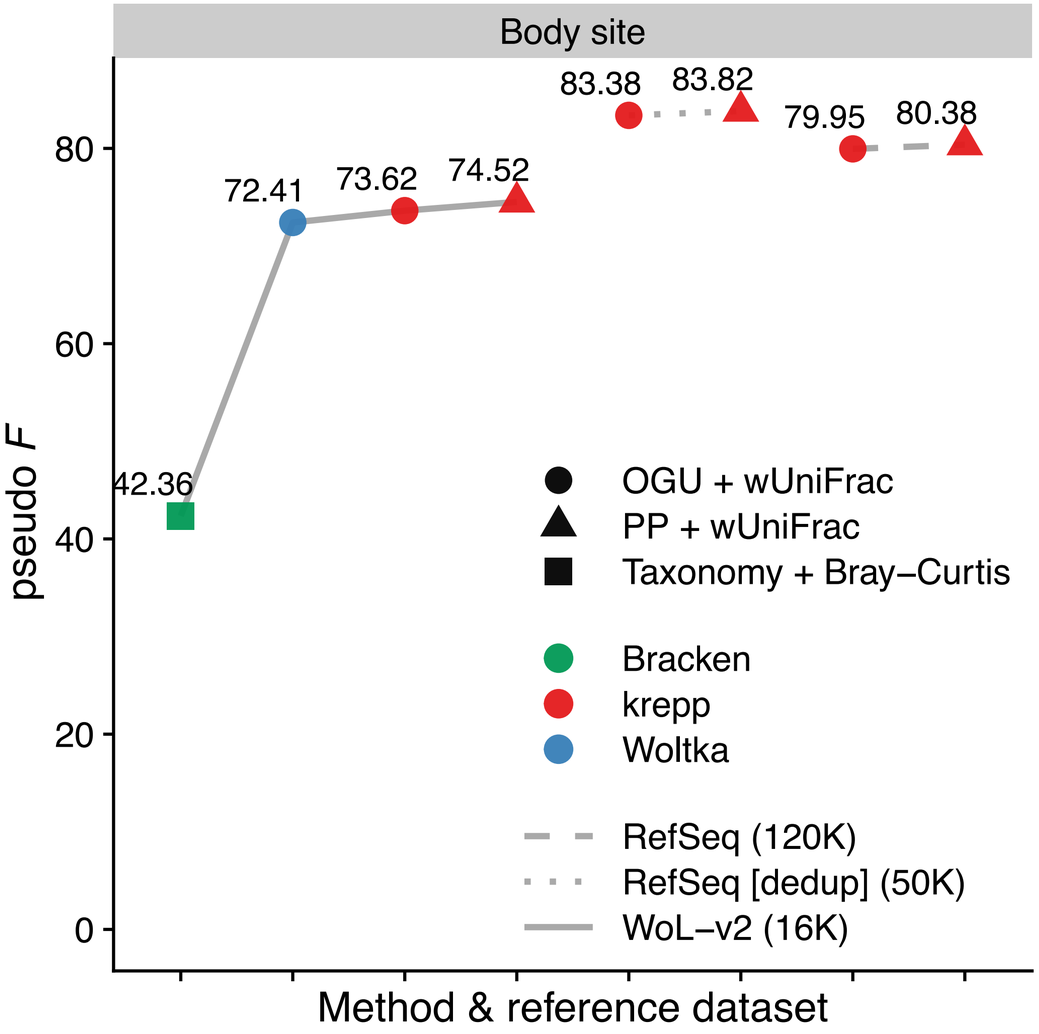




Analyzing human microbiome with larger references

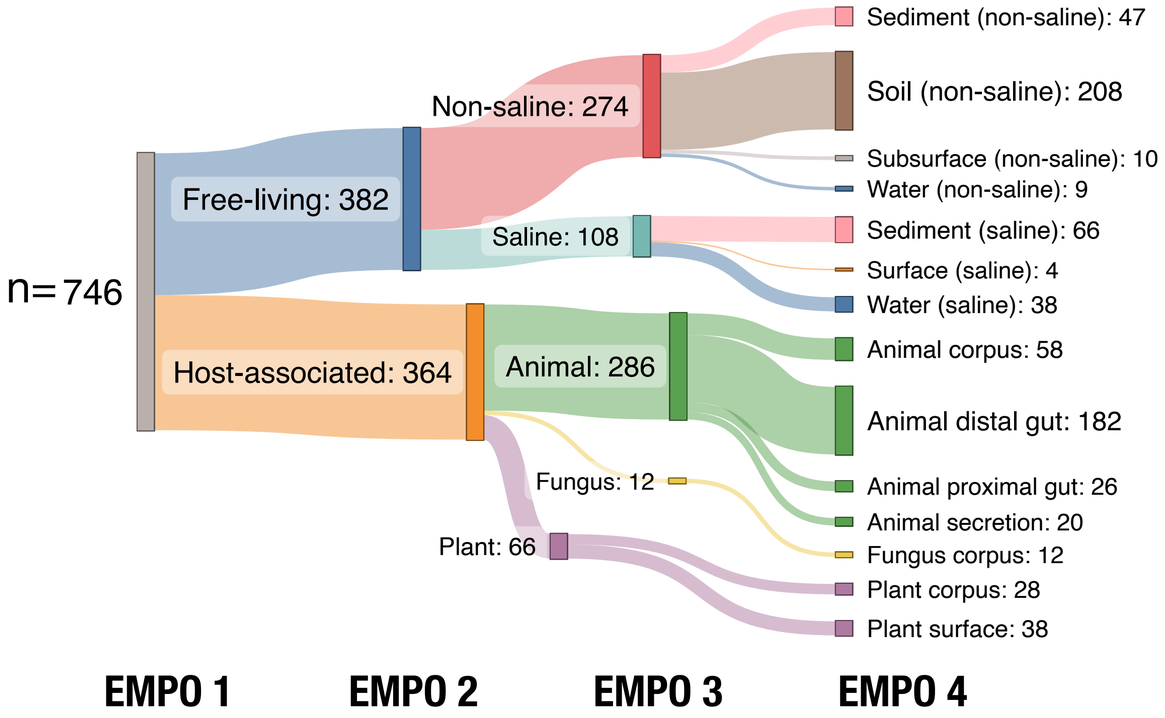
Scaling to large references further **improves** separation of body sites.



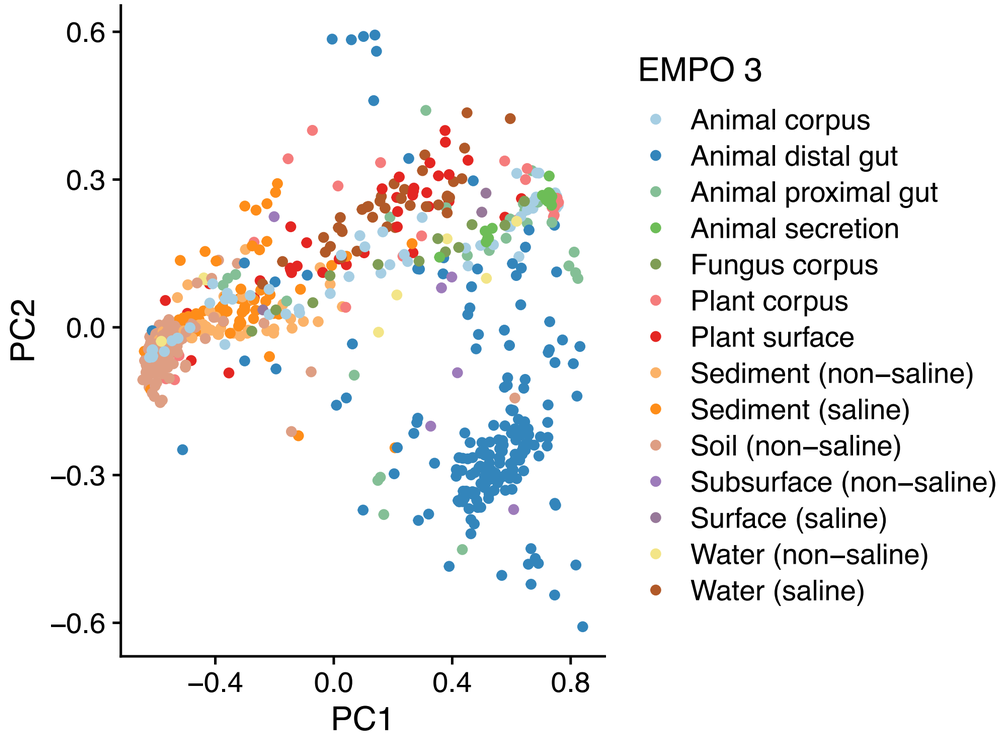


Improved characterization of earth's microbiome

Hierarchical categorization
of earth microbiome samples:

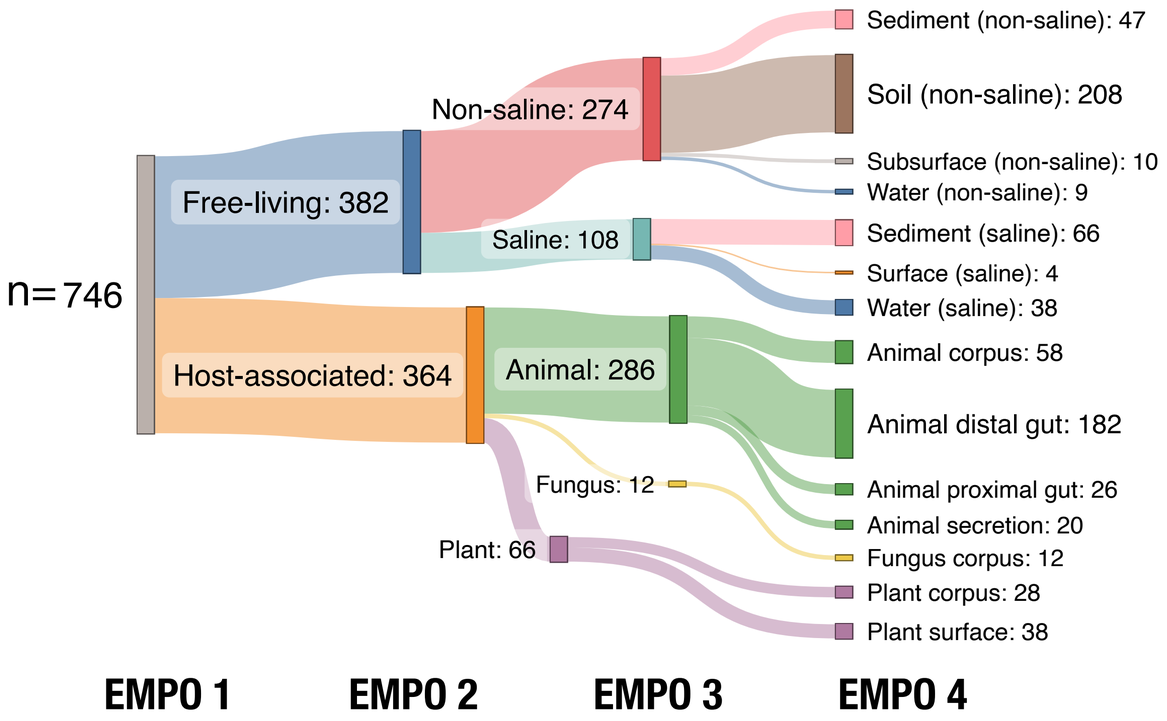


Reference: Web of Life (v1)
11,000 microbial genomes

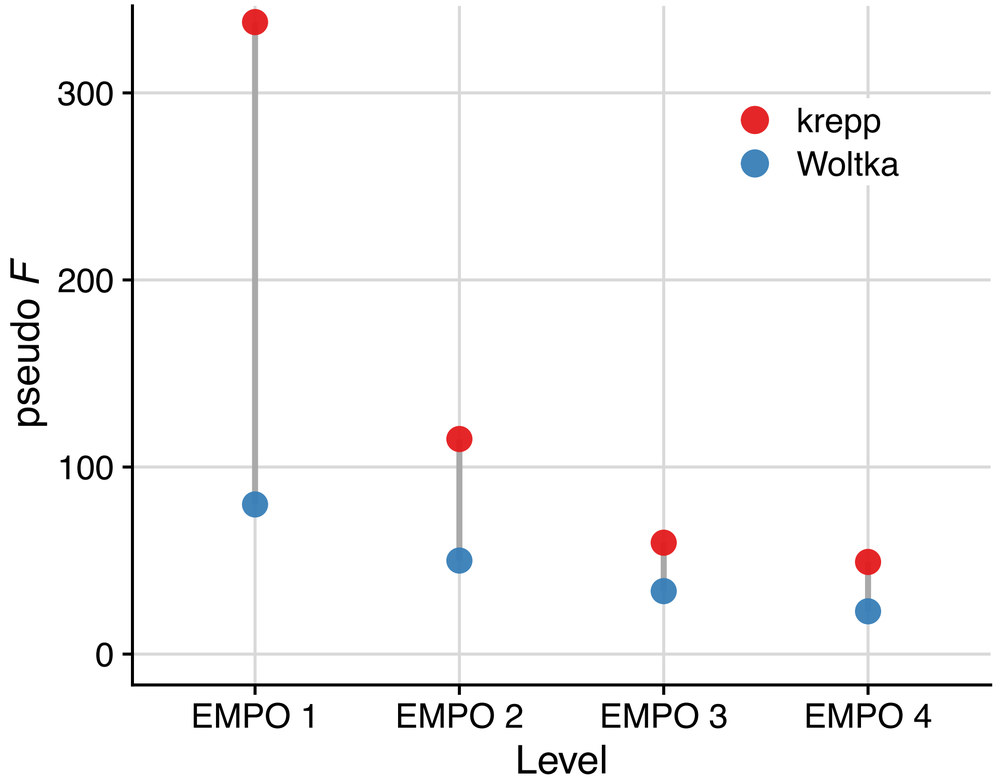


Improved characterization of earth's microbiome

Hierarchical categorization
of earth microbiome samples:



Reference: Web of Life (v1)
11,000 microbial genomes



Conclusion

krepp

- estimates read to genome distances >10x faster than alignment
- extends to more distant references and maps novel reads
- is the only method that can accurately place genome-wide reads
  - One exception: App-SpaM; cannot scale beyond 50-100 microbial genomes
- offers significantly better characterization of metagenomic samples

Siavash Mirarab

Siavash Mirarab

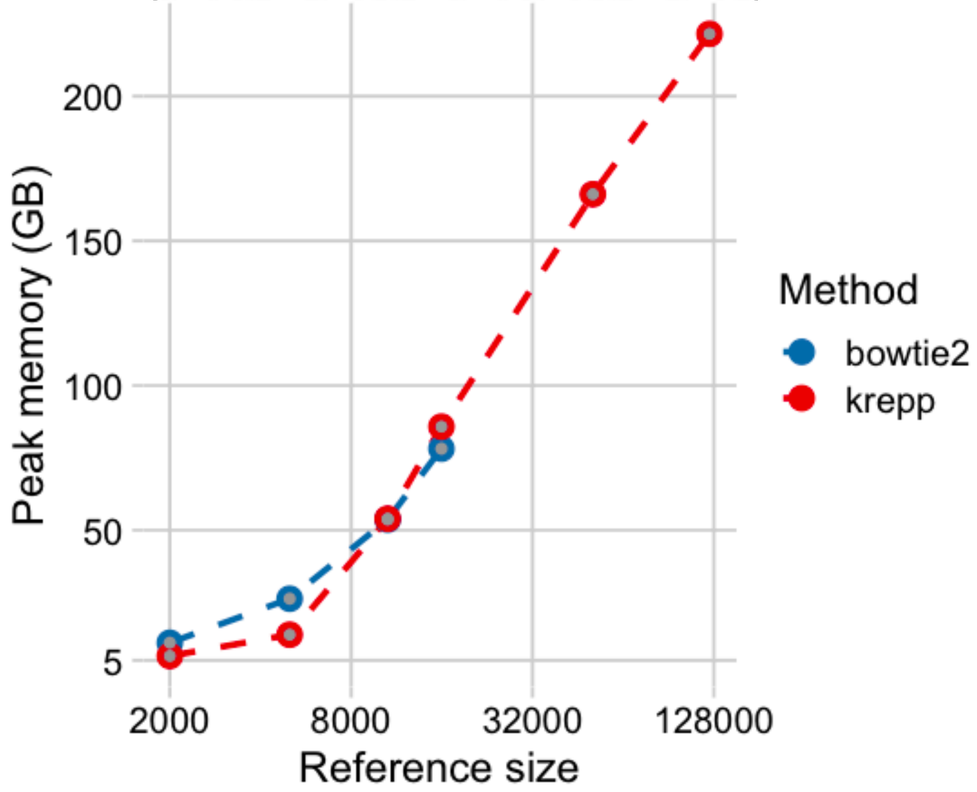
Questions?

Extra Slides

Scalability:
krepp can be distributed and has flexible memory requirements

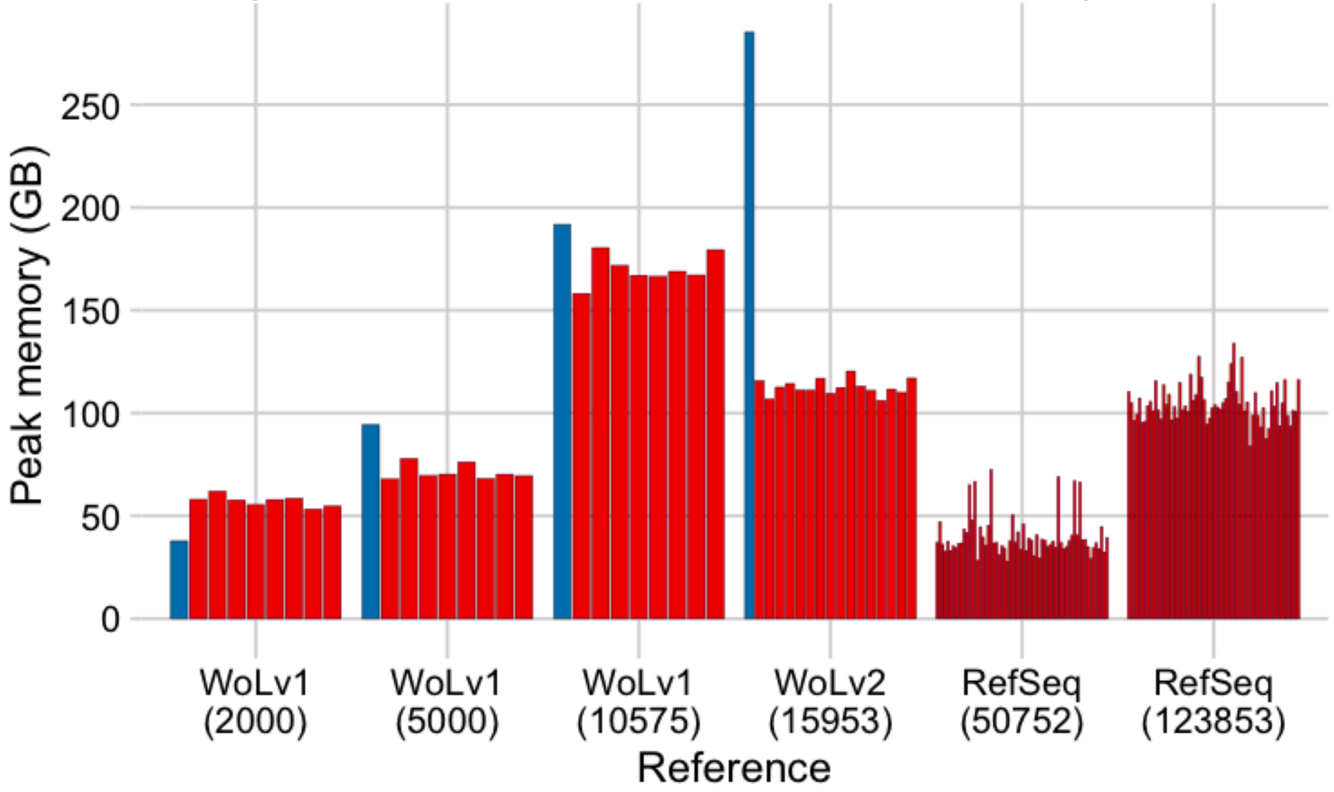
Mapping 10M reads (16 threads):

reducing memory use
w/ further subsampling...



Indexing microbial genomes (32 threads):

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



Let b_i be the length of the branch i and p_i^A and p_i^B are the taxa proportions descending from the branch i for community A and B, respectively.

$$d(A,B) = \frac{\sum_i^n b_i |p_i^A - p_i^B|}{\sum_i^n b_i (p_i^A + p_i^B)}$$

$$SS_T = \frac{1}{N}\sum_{i=1}^{N-1}\sum_{j=i+1}^{N} d_{ij}^2 \qquad SS_W = \frac{1}{n}\sum_{i=1}^{N-1}\sum_{j=i+1}^{N} d_{ij}^2 \delta_{ij}$$

$$SS_A = SS_T - SS_W$$

$$F = \frac{\left(\frac{SS_A}{p-1}\right)}{\left(\frac{SS_W}{N-p}\right)}$$

Multiple permutations to get a p-value!

N is the number of groups, p is the number of objects in each group.