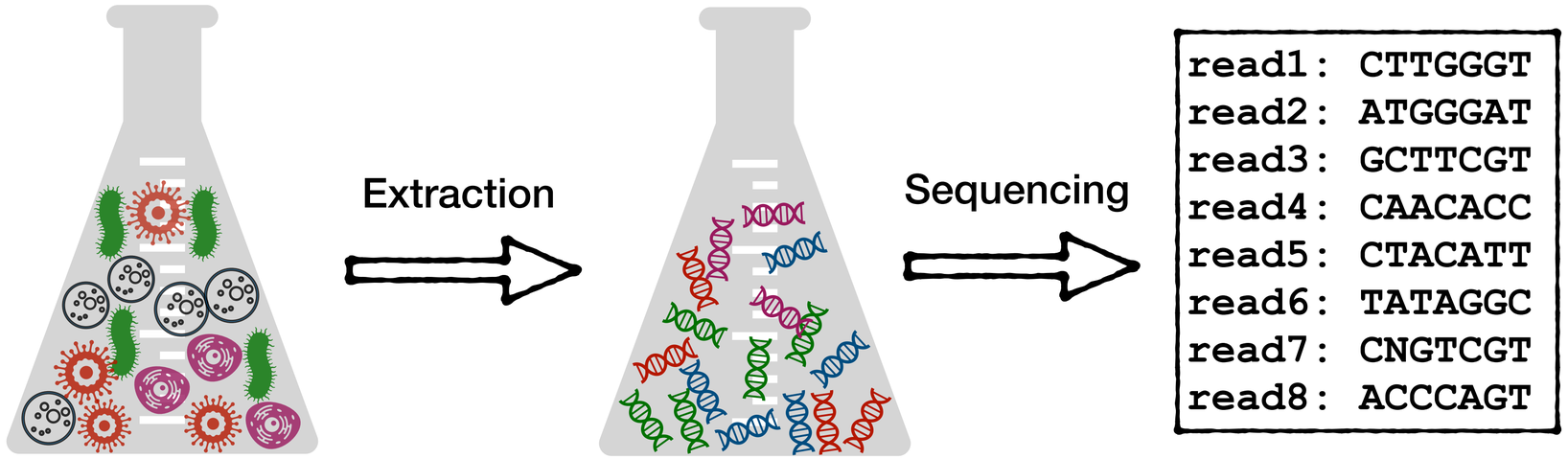
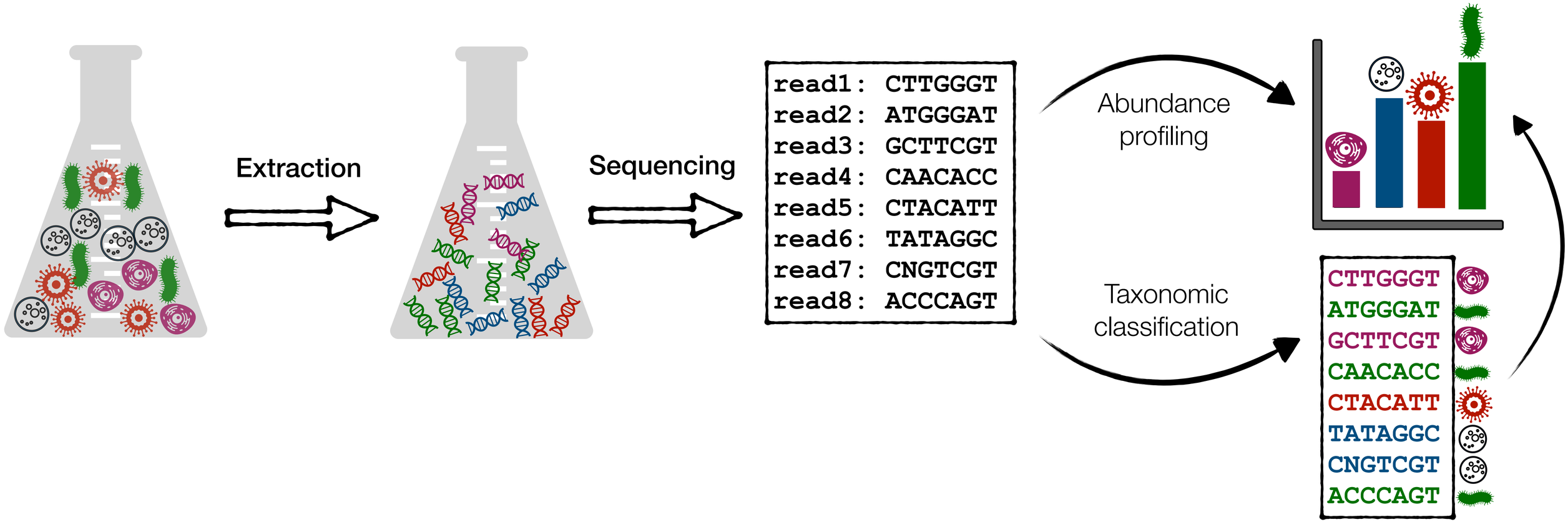
Memory-bound and taxonomy-aware k-mer selection for large reference databases

Ali Osman Berk Şapcı & Siavash Mirarab
UC San Diego

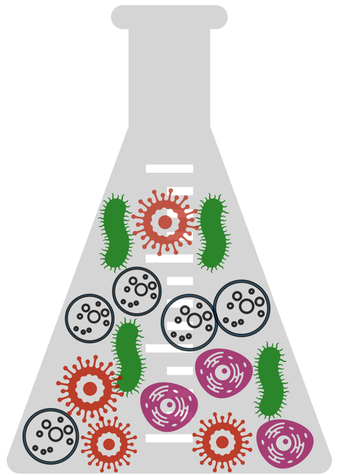
Identifying metagenomic sequences



Identifying metagenomic sequences



Identifying metagenomic sequences

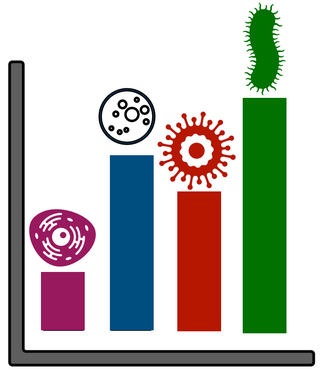


Extraction

Sequencing

```
read1: CTTGGGT
read2: ATGGGAT
read3: GCTTCGT
read4: CAACACC
read5: CTACATT
read6: TATAGGC
read7: CNGTCGT
read8: ACCCAGT
```

Abundance profiling



Taxonomic classification

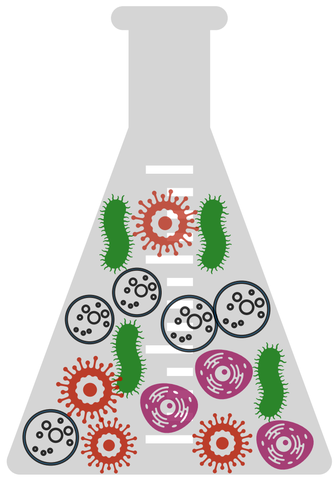
```
CTTGGGT
ATGGGAT
GCTTCGT
CAACACC
CTACATT
TATAGGC
CNGTCGT
ACCCAGT
```

find similar reference genomes

reference genomes

Species 1
```
G1: TCCCTGCTCA…
G2: TCCCTGCTCA…
G3: CAATGTGCGG…
```
Species 3
```
G4: CCCCAAACGA…
G5: GCGCGGGTTC…
```
Species 2
```
G6: AGTTGCACTA…
G7: TACCACTGTG…
G8: TACCACTGTG…
G9: CAATTAAGAA…
...
```
Species 4
```
GN: ATTATCTGAT…
```

Identifying metagenomic sequences

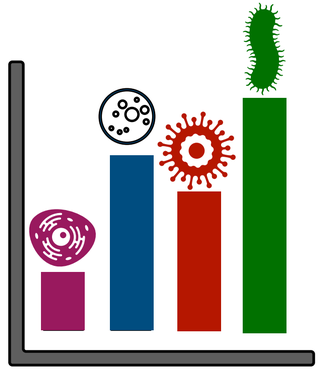


Extraction

Sequencing

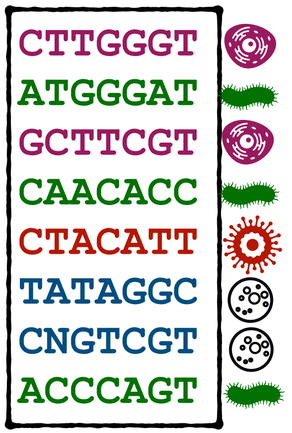
```
read1: CTTGGGT
read2: ATGGGAT
read3: GCTTCGT
read4: CAACACC
read5: CTACATT
read6: TATAGGC
read7: CNGTCGT
read8: ACCCAGT
```

Abundance profiling



Taxonomic classification

```
CTTGGGT
ATGGGAT
GCTTCGT
CAACACC
CTACATT
TATAGGC
CNGTCGT
ACCCAGT
```



reference genomes

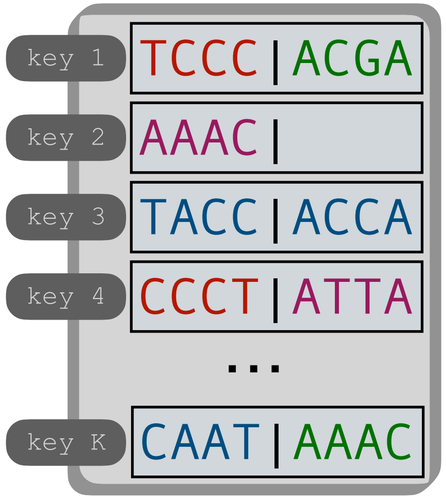
Species 1
G1: TCCCTGCTCA…
G2: TCCCTGCTCA…
G3: CAATGTGCGG…

Species 3
G4: CCCCAAACGA…
G5: GCGCGGGTTC…
G6: AGTTGCACTA…

Species 2
G7: TACCACTGTG…
G8: TACCACTGTG…
G9: CAATTAAGAA…
...

Species 4
GN: ATTATCTGAT…

build a searchable index



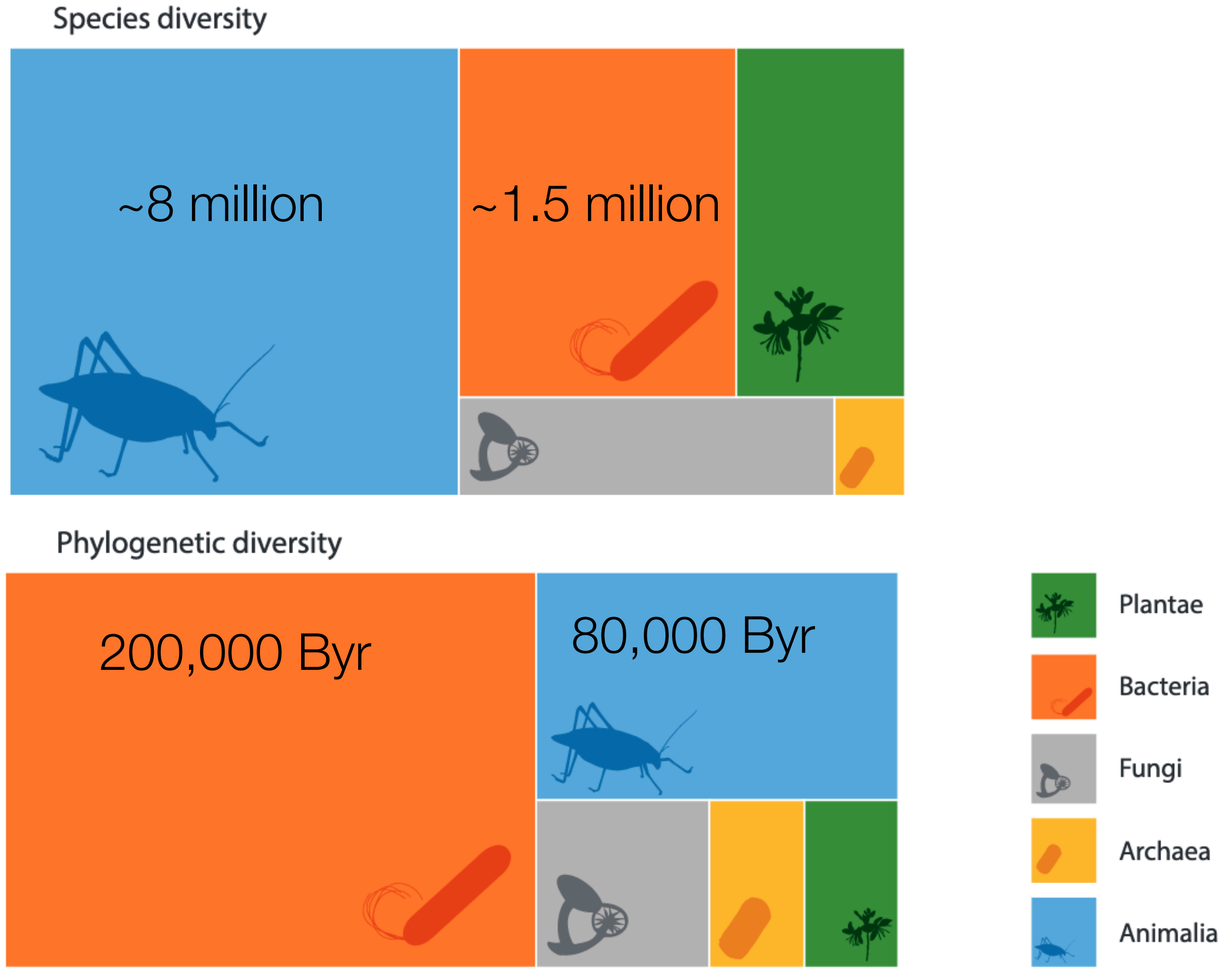
hash table

find similar reference genomes

Novel sequences challenge popular tools

- Reference databases (and indexes) remain incomplete compared to all species…

  and there is a rich diversity within species!



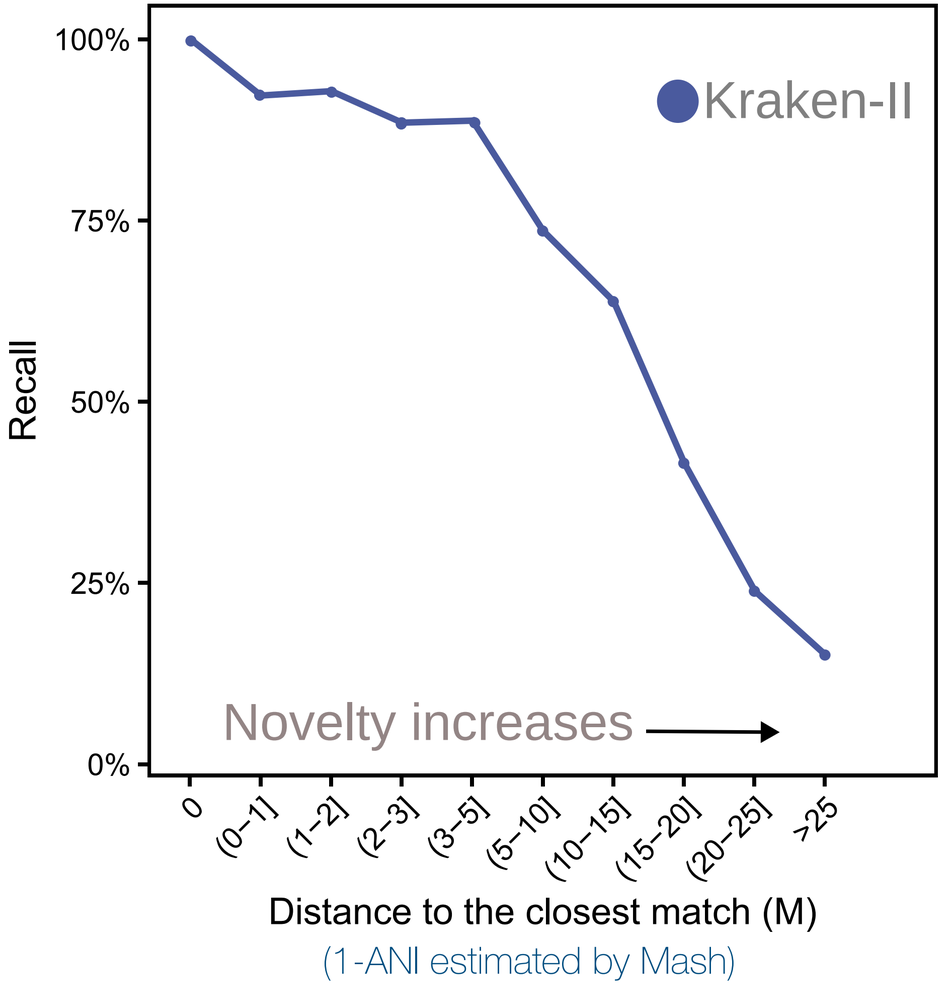
[Díaz et al., 2022]

Novel sequences challenge popular tools

- Reference databases (and indexes) remain incomplete compared to all species…

  and there is a rich diversity within species!

- **Novel sequences:** sequences which lack a close matching reference genome



[Rachtman et al., 2019]

Solutions for identifying novel queries w/ limited resources

- find distant matches → increase sensitivity of the search
- enhance the reference set → utilize more genomes & larger databases

Solutions for identifying novel queries w/ limited resources

- find distant matches → increase sensitivity of the search
- enhance the reference set → utilize more genomes & larger databases

Computing the Hamming distances of inexact matches

CONSULT-II: accurate taxonomic identification and profiling using locality-sensitive hashing

Ali Osman Berk Şapcı [1], Eleonora Rachtman [1], Siavash Mirarab [1,2,*]

Solutions for identifying novel queries w/ limited resources

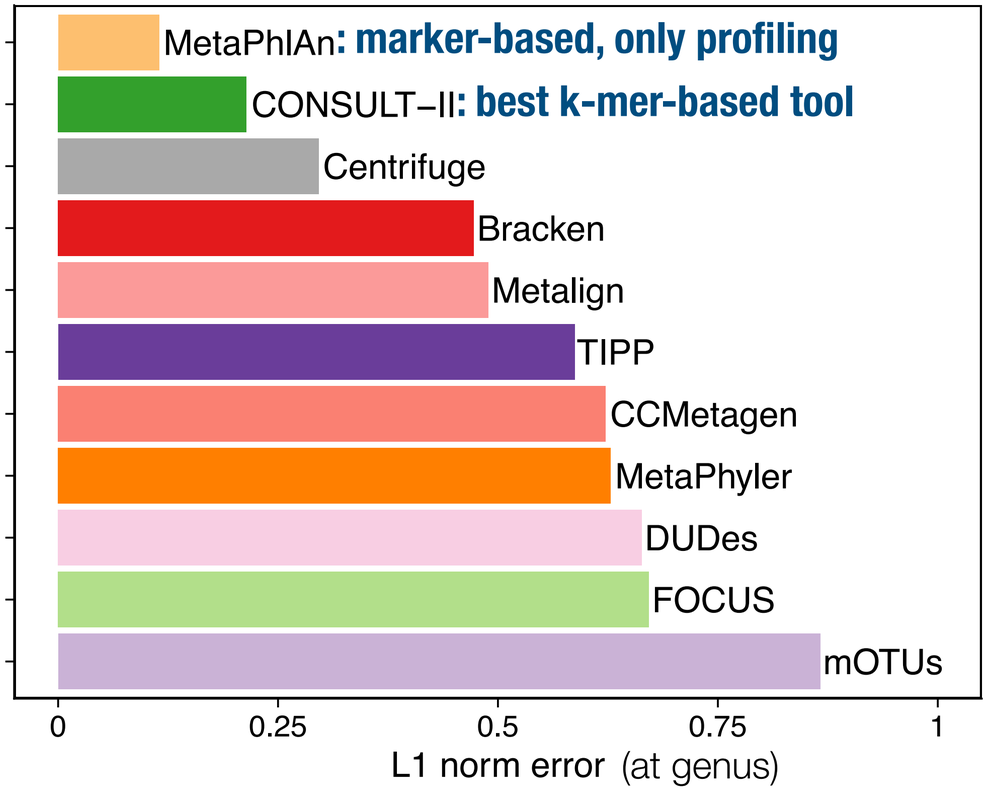
- find distant matches → increase sensitivity of the search
- enhance the reference set → utilize more genomes & larger databases

Computing the Hamming distances of inexact matches

CONSULT-II: accurate taxonomic identification and profiling using locality-sensitive hashing

Ali Osman Berk Şapcı [1], Eleonora Rachtman [1], Siavash Mirarab [1,2,*]

Strain–madness dataset [CAMI-II]



(using a RefSeq snapshot from 2019 with ~130k genomes)

Can we use more reference genomes?

- find distant matches → increase sensitivity of the search
- enhance the reference set → utilize more genomes & larger databases

Can we use more reference genomes?

- find distant matches → increase sensitivity of the search
- enhance the reference set → utilize more genomes & larger databases

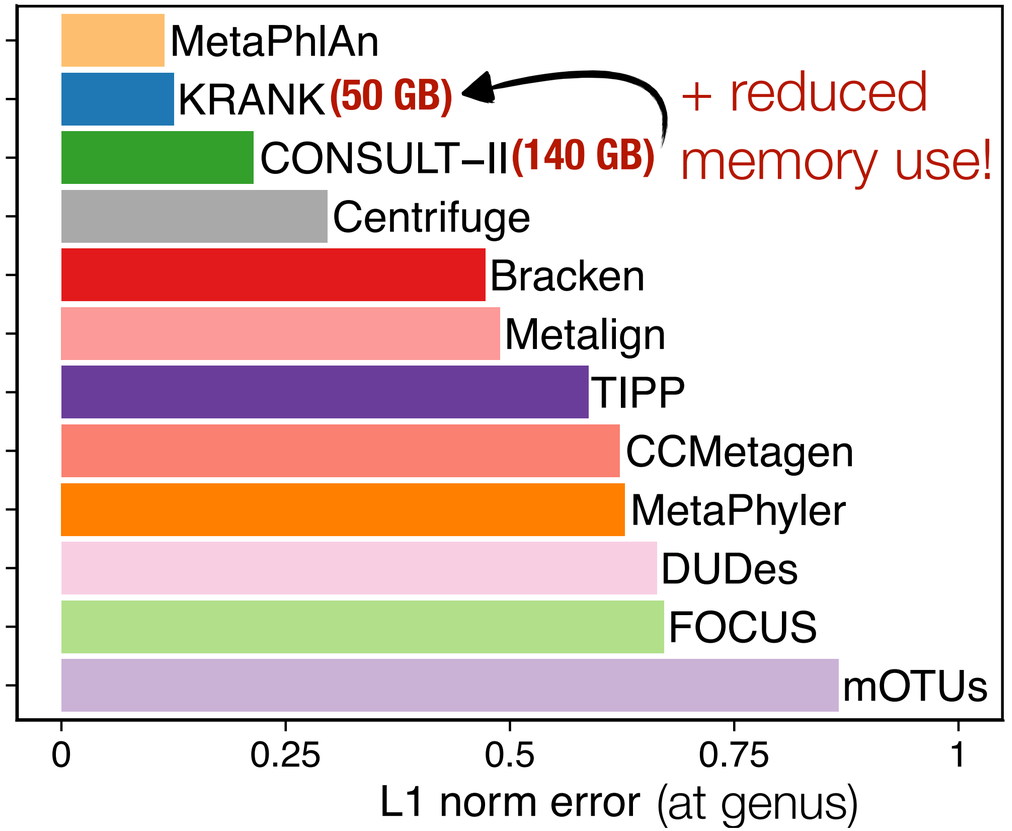
- **Challenge:** very large & diverse databases have too many *k*-mers to fit in the memory
  - Limited to a selected subset
- **This talk:** — **KRANK**
  - Selecting a representative subset of *k*-mers + classification/profiling using CONSULT-II

Can we use more reference genomes?

- find distant matches → increase sensitivity of the search
- enhance the reference set → utilize more genomes & larger databases

- **Challenge:** very large & diverse databases have too many *k*-mers to fit in the memory
  - Limited to a selected subset
- **This talk:** — **KRANK**
  - Selecting a representative subset of *k*-mers + classification/profiling using CONSULT-II

Strain–madness dataset [CAMI-II]

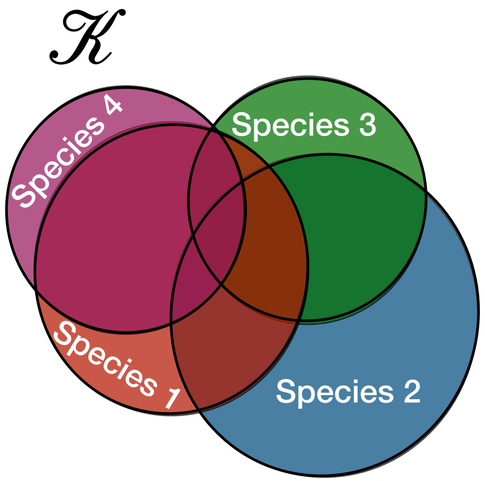


(using a RefSeq snapshot from 2019 with ~130k genomes)

Problem statement

- Given:
  1. *k*-mer set $\mathscr{K}$ of a large collection of genomes
  2. limited budget $M < |\mathscr{K}|$
  3. taxonomy

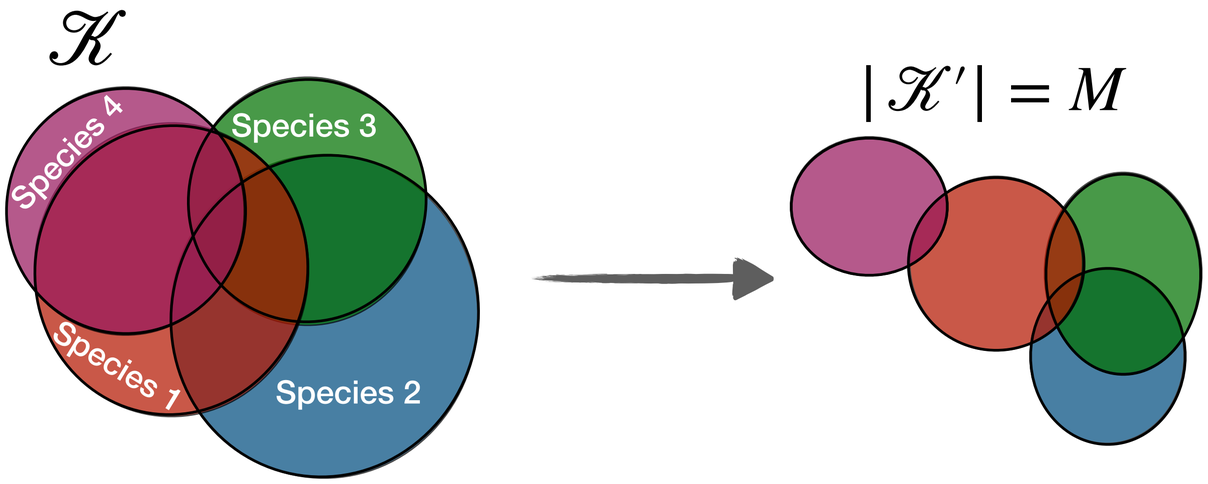
Species 1
G1: TCCCTGCTCAGTGGTATATGGTTTTTGCTA…
G2: TCCCTGCTCAGCCCCATATGGTTTTTGCTA…
Species 3
G3: CAATGTGCGGATGGCGTTACGACTTACTGG…
G4: CCCCAAACGATGCTGAAGGCTCAGGTTACA…
G5: GCGCGGGTTCCCGCCCTCAACCCGGGCCGA…
Species 2
G6: AGTTGCACTACTTCTGCGACCCAAATGCAC…
G7: TACCACTGTGTTCGTGTCATCTAGGACGGG…
G8: TACCACTGTGTTCGTGTCATCTAGGACGGG…
G9: CAATTAAGAATACCTTATATTATTGTACAC…
...
Species 4
GN: ATTATCTGATTTTATATTATGATTTTAGTA…



Problem statement

- Given:
  1. *k*-mer set $\mathscr{K}$ of a large collection of genomes
  2. limited budget $M < |\mathscr{K}|$
  3. taxonomy
- Select a subset with size M such that the collection is well represented

Species 1
G1: TCCCTGCTCAGTGGTATATGGTTTTTGCTA…
G2: TCCCTGCTCAGCCCCATATGGTTTTTGCTA…
Species 3
G3: CAATGTGCGGATGGCGTTACGACTTACTGG…
G4: CCCCAAACGATGCTGAAGGCTCAGGTTACA…
G5: GCGCGGGTTCCCGCCCTCAACCCGGGCCGA…
Species 2
G6: AGTTGCACTACTTCTGCGACCCAAATGCAC…
G7: TACCACTGTGTTCGTGTCATCTAGGACGGG…
G8: TACCACTGTGTTCGTGTCATCTAGGACGGG…
G9: CAATTAAGAATACCTTATATTATTGTACAC…
...
Species 4
GN: ATTATCTGATTTTATATTATGATTTTAGTA…



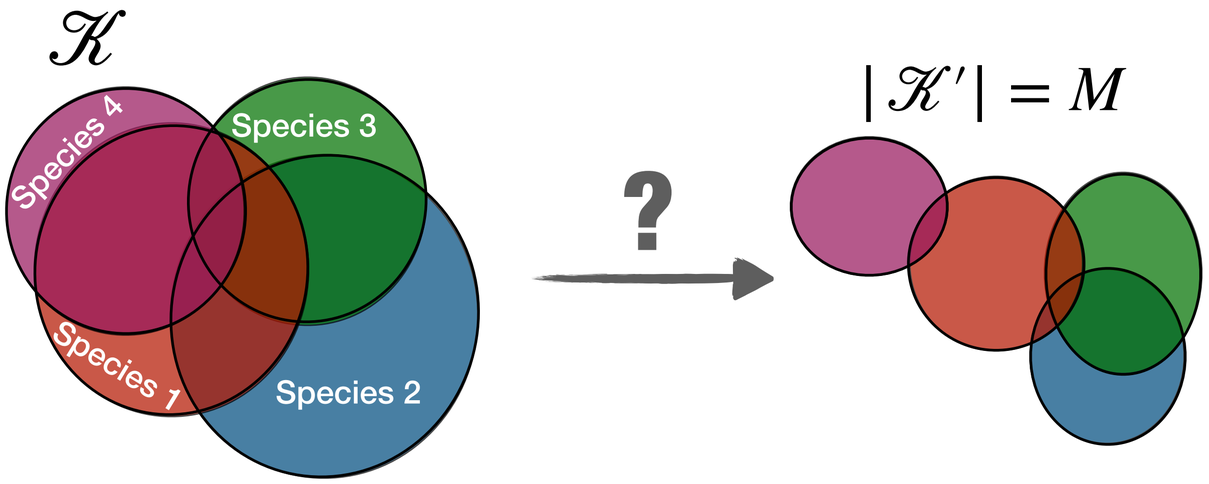
Problem statement

- Given:
  1. *k*-mer set $\mathscr{K}$ of a large collection of genomes
  2. limited budget $M < |\mathscr{K}|$
  3. taxonomy

high accuracy in taxonomic identification

- Select a subset with size M such that the collection is well represented

Species 1
G1: TCCCTGCTCAGTGGTATATGGTTTTTGCTA…
G2: TCCCTGCTCAGCCCCATATGGTTTTTGCTA…
Species 3
G3: CAATGTGCGGATGGCGTTACGACTTACTGG…
G4: CCCCAAACGATGCTGAAGGCTCAGGTTACA…
G5: GCGCGGGTTCCCGCCCTCAACCCGGGCCGA…
Species 2
G6: AGTTGCACTACTTCTGCGACCCAAATGCAC…
G7: TACCACTGTGTTCGTGTCATCTAGGACGGG…
G8: TACCACTGTGTTCGTGTCATCTAGGACGGG…
G9: CAATTAAGAATACCTTATATTATTGTACAC…
...
Species 4
GN: ATTATCTGATTTTATATTATGATTTTAGTA…



Reducing the reference set by selecting k-mers

```
G1: TCCCTGC
     CCCTGCT
      CCTGCTC
       CTGCTCA…
G2: TCGCTAC
     CGCTACG
      GCTACGC
       CTACGCG…
G3: CAATGTG
     AATGTGC
      ATGTGCG
       TGTGCGG…
G5: GCGCGGG
     CGCGGGT
      GCGGGTT
       CGGGTTC…
G4: CCCCAAA
     CCCAAAC
      CCAAACG
       CAAACGT…
```

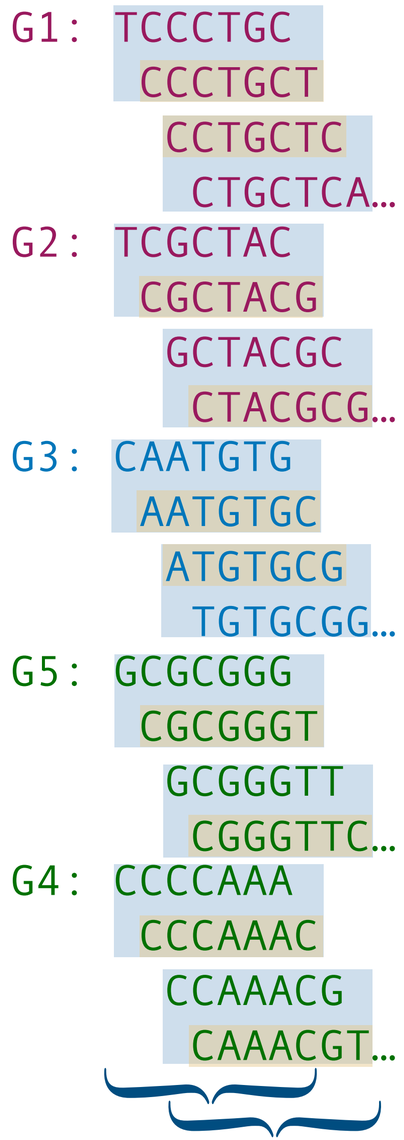
Reducing the reference set by selecting k-mers

- **Baseline:** random selection

```
G1: TCCCTGC
     CCCTGCT
      CCTGCTC
       CTGCTCA…
G2: TCGCTAC
     CGCTACG
      GCTACGC
       CTACGCG…
G3: CAATGTG
     AATGTGC
      ATGTGCG
       TGTGCGG…
G5: GCGCGGG
     CGCGGGT
      GCGGGTT
       CGGGTTC…
G4: CCCCAAA
     CCCAAAC
      CCAAACG
       CAAACGT…
```

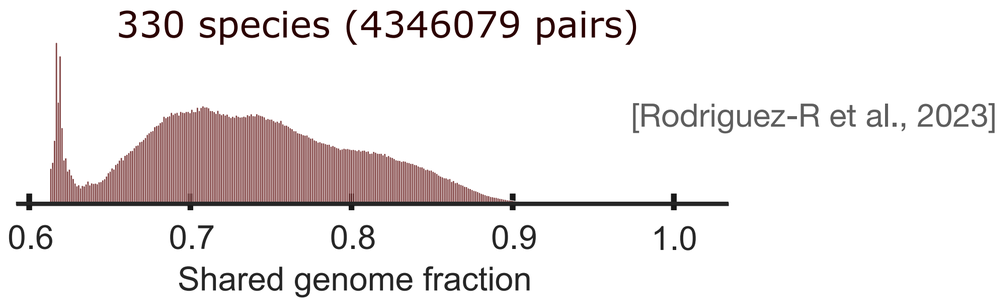
Reducing the reference set by selecting k-mers

- **Baseline:** random selection
- **Minimizers:** selecting one among overlapping *k*-mers with a sliding window

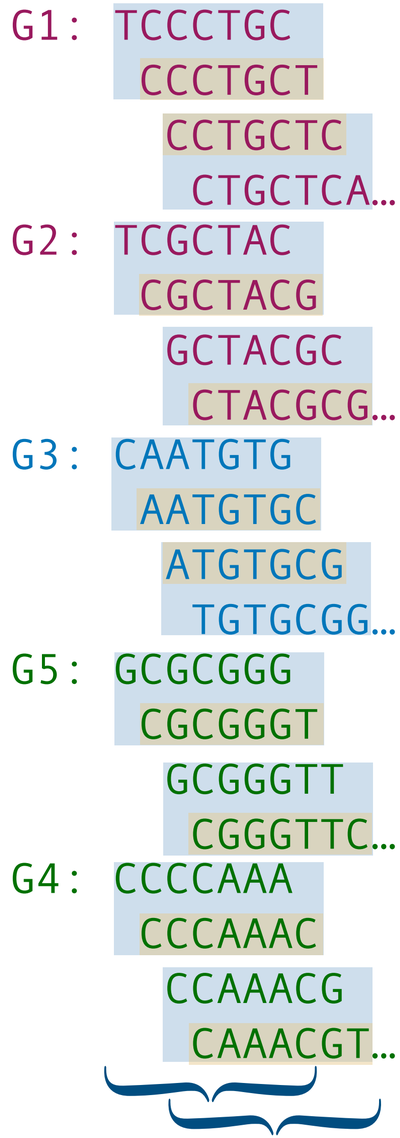


Reducing the reference set by selecting k-mers

- **Baseline:** random selection
- **Minimizers:** selecting one among overlapping *k*-mers with a sliding window
- Even with minimizers, number of distinct *k*-mers grows fast with the number of genomes.

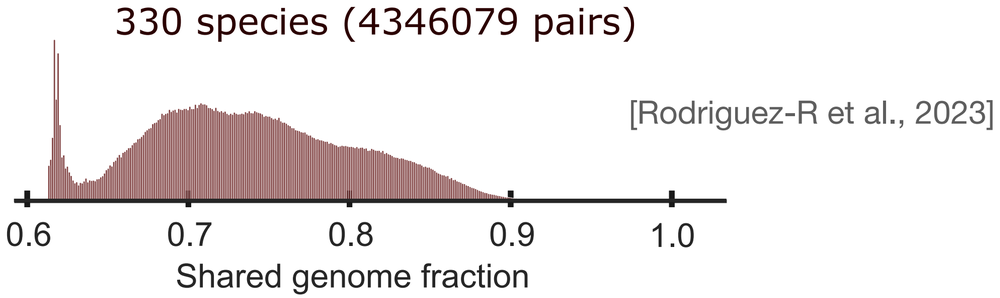


[Rodriguez-R et al., 2023]



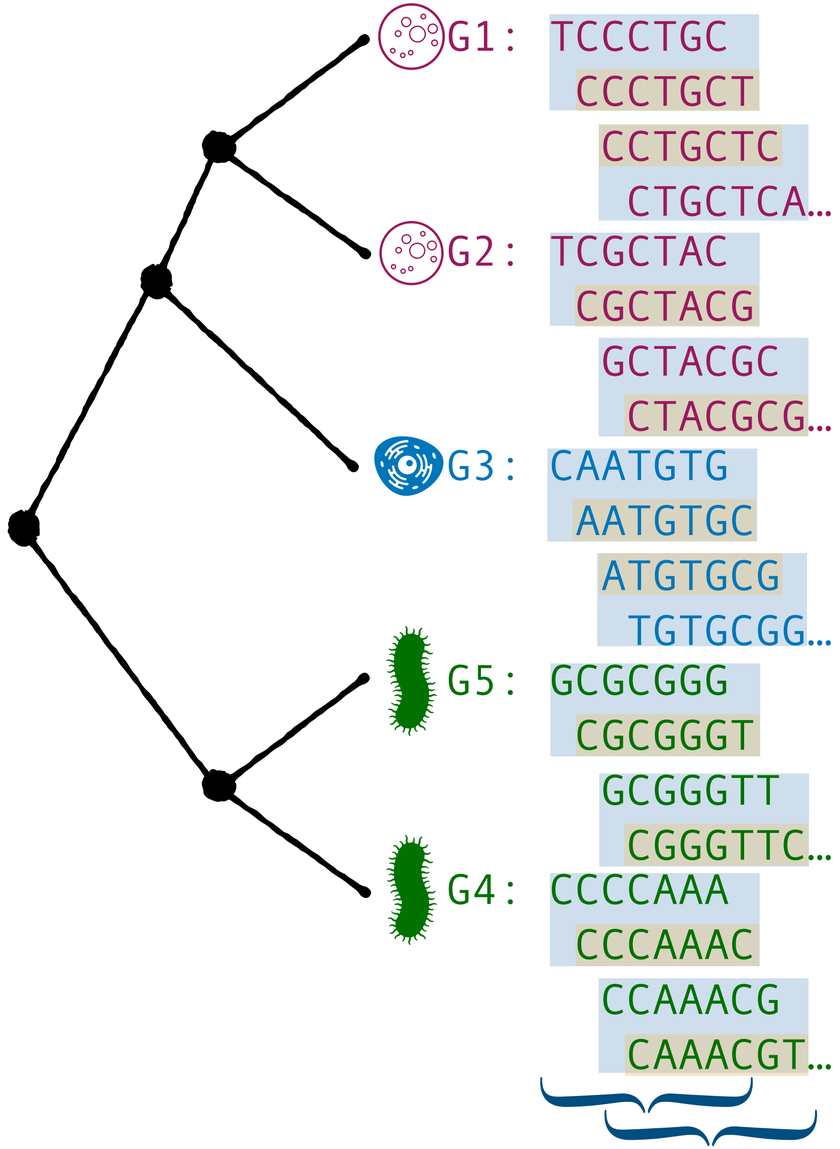
Reducing the reference set by selecting k-mers

- **Baseline:** random selection
- **Minimizers:** selecting one among overlapping *k*-mers with a sliding window
- Even with minimizers, number of distinct *k*-mers grows fast with the number of genomes.



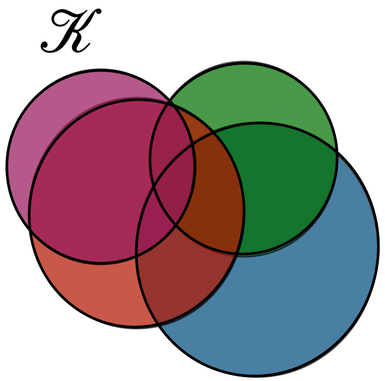
[Rodriguez-R et al., 2023]

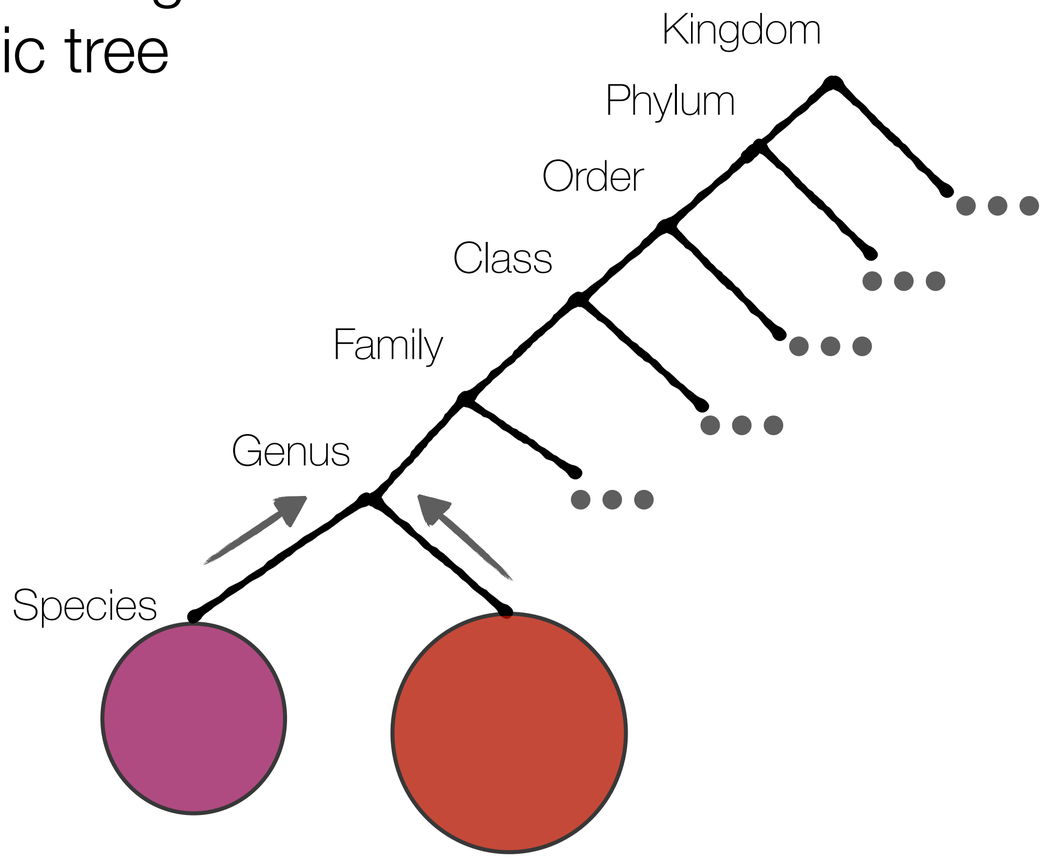
- Additionally, exploit the evolutionary dimension



KRANK selects a representative *k*-mer subset in a memory-bound manner!

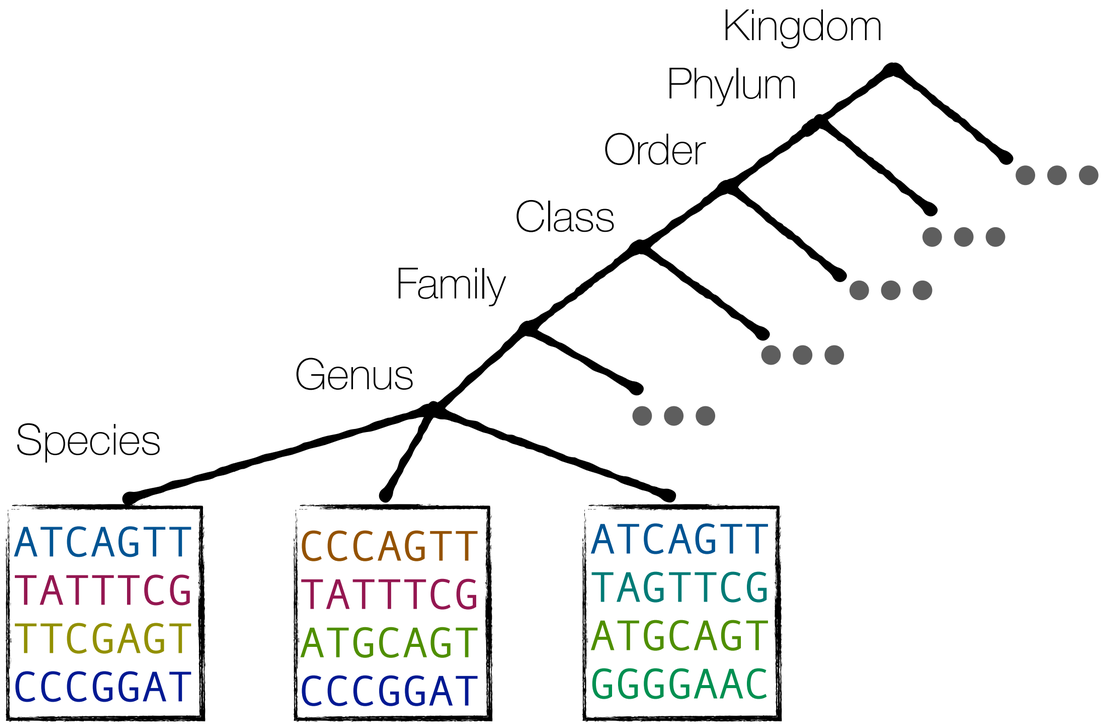
Core idea: hierarchical subsampling through a post order traversal of the taxonomic tree





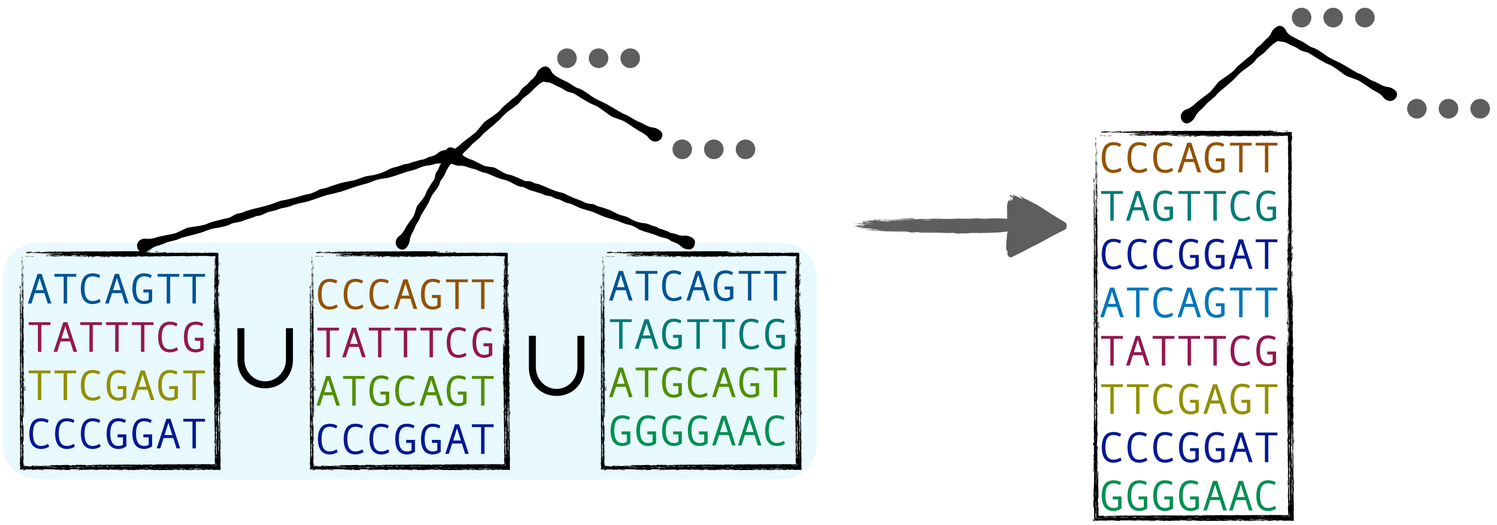
At leaves: k-mer sets of species

- Extract reference *k*-mers for each species.



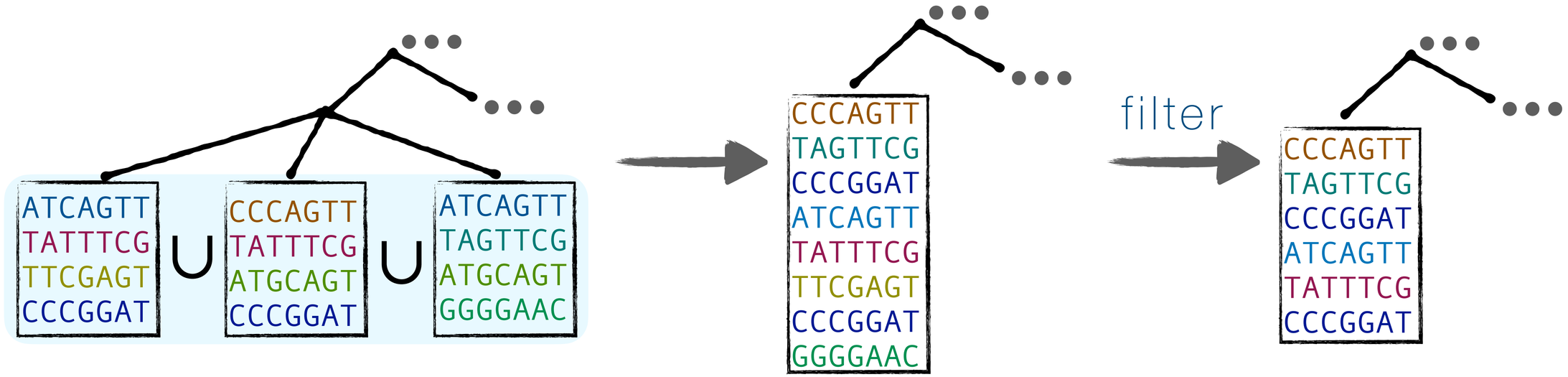
At internal nodes: gradual filtering of k-mers

- Recursively take the union of sibling taxa.



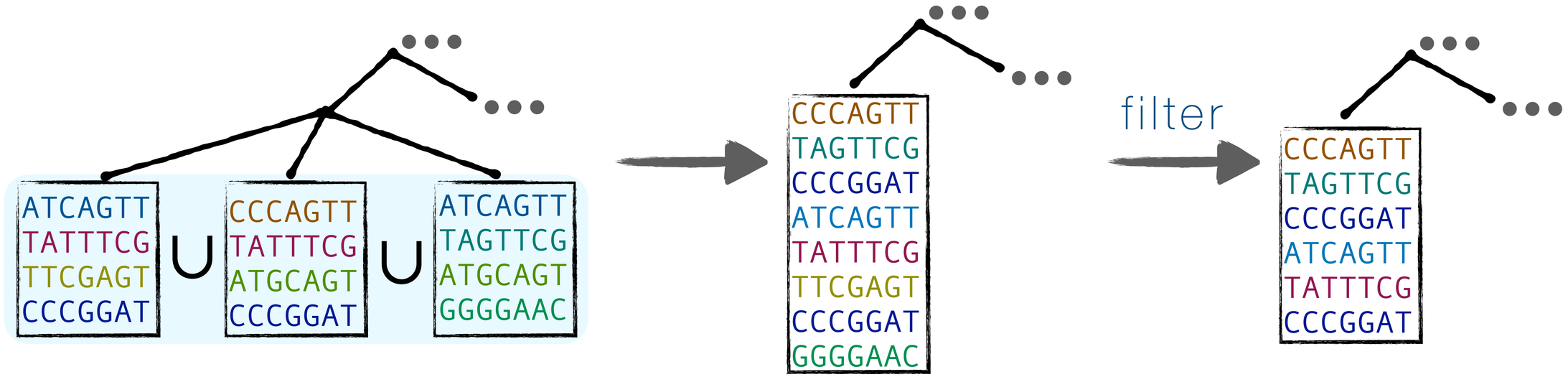
At internal nodes: gradual filtering of k-mers

- Recursively take the union of sibling taxa.
- Filter *some number* of *k*-mers based on *a ranking*.



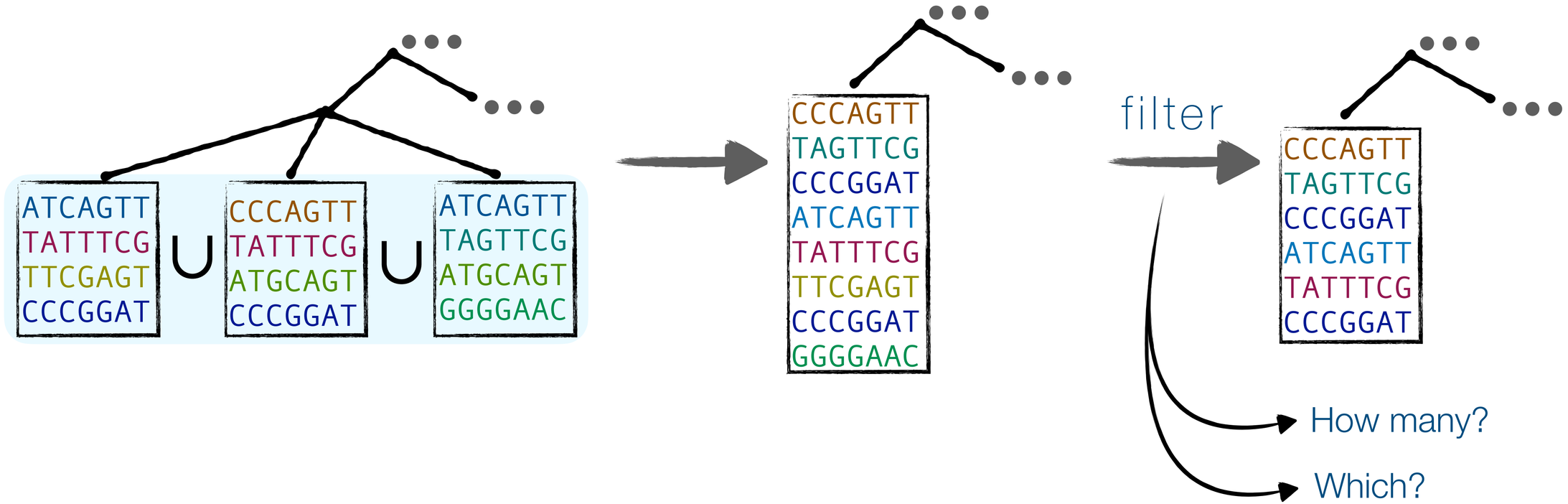
At internal nodes: gradual filtering of k-mers

- Recursively take the union of sibling taxa.
- Filter *some number* of *k*-mers based on *a ranking*.
- At the root, we obtain the final library with size M.



At internal nodes: gradual filtering of k-mers

- Recursively take the union of sibling taxa.
- Filter *some number* of *k*-mers based on *a ranking*.
- At the root, we obtain the final library with size M.



Q1: How many *k*-mers should we remove from each node/taxon?

Q2: How do we rank *k*-mers to assess which one(s) should be kept?

- **Baseline:** no gradual filtering — wait & keep M randomly at the root

- **Baseline:** no gradual filtering — wait & keep M randomly at the root

Given total budget M,
$\mathbb{E}[$# of selected k-mers for a taxon $t]$ is

$$M \frac{|\mathcal{K}_t|}{|\mathcal{K}|}$$

$|\mathcal{K}_t|$: set of k-mers under the taxon t

$|\mathcal{K}|$: set of all reference k-mers

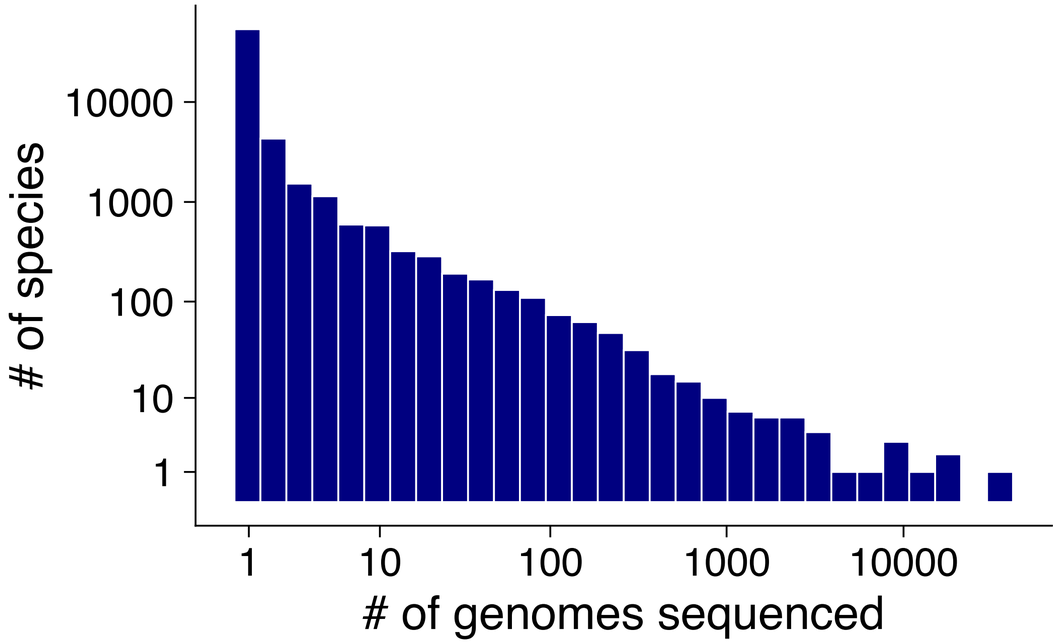
- **Baseline:** no gradual filtering — wait & keep M randomly at the root

Given total budget M,
$\mathbb{E}[\#$ of selected k-mers for a taxon $t]$ is

$$M \frac{|\mathcal{K}_t|}{|\mathcal{K}|}$$

- $|\mathcal{K}_t|$: set of k-mers under the taxon t
- $|\mathcal{K}|$: set of all reference k-mers

- Proportional contribution →
  - taxa with low sampling get little representation
  - highly-sampled groups dominates (e.g., *E. coli*)



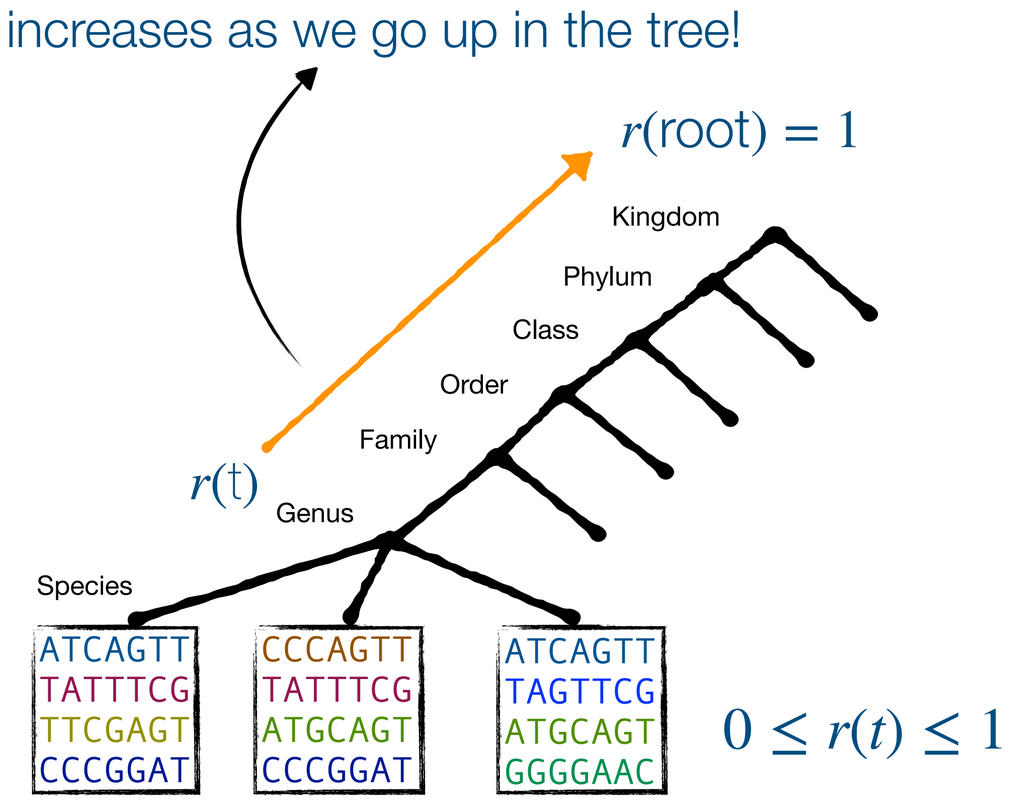
Gradual filtering is making some decisions earlier

Goal: remove *k*-mers from bloated taxa earlier & delay decisions for smaller taxa.

Gradual filtering is making some decisions earlier

Goal: remove *k*-mers from bloated taxa earlier & delay decisions for smaller taxa.

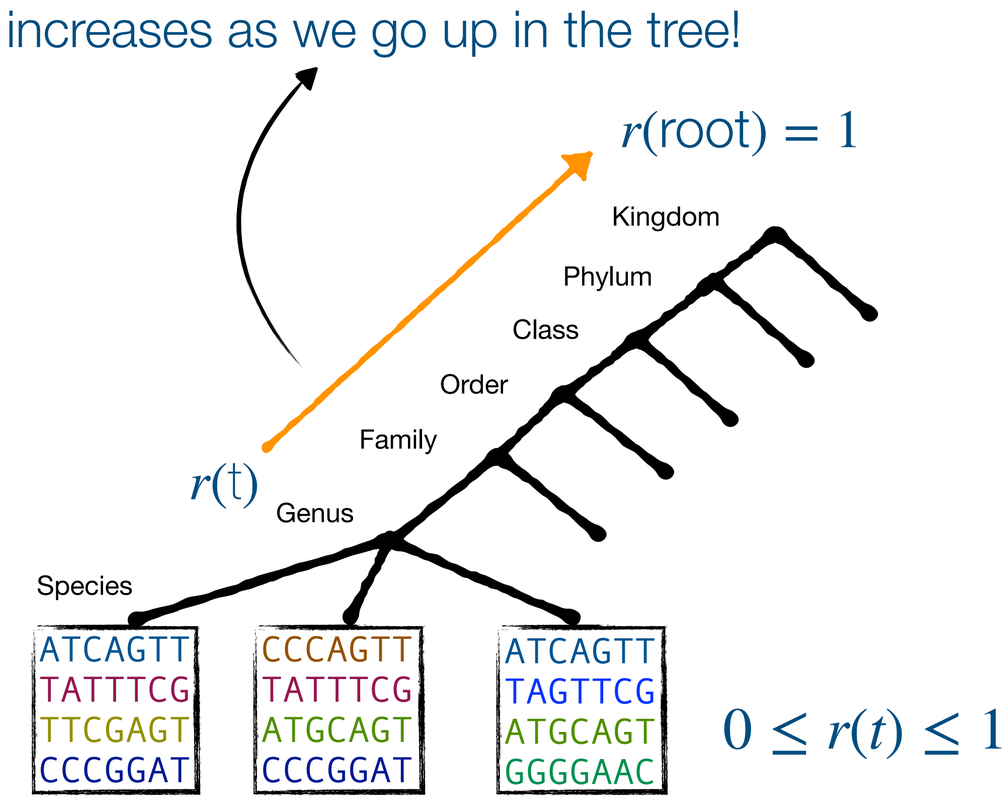
- Adaptive size constraint, $r(t)M$, on internal nodes



Gradual filtering is making some decisions earlier

Goal: remove *k*-mers from bloated taxa earlier & delay decisions for smaller taxa.

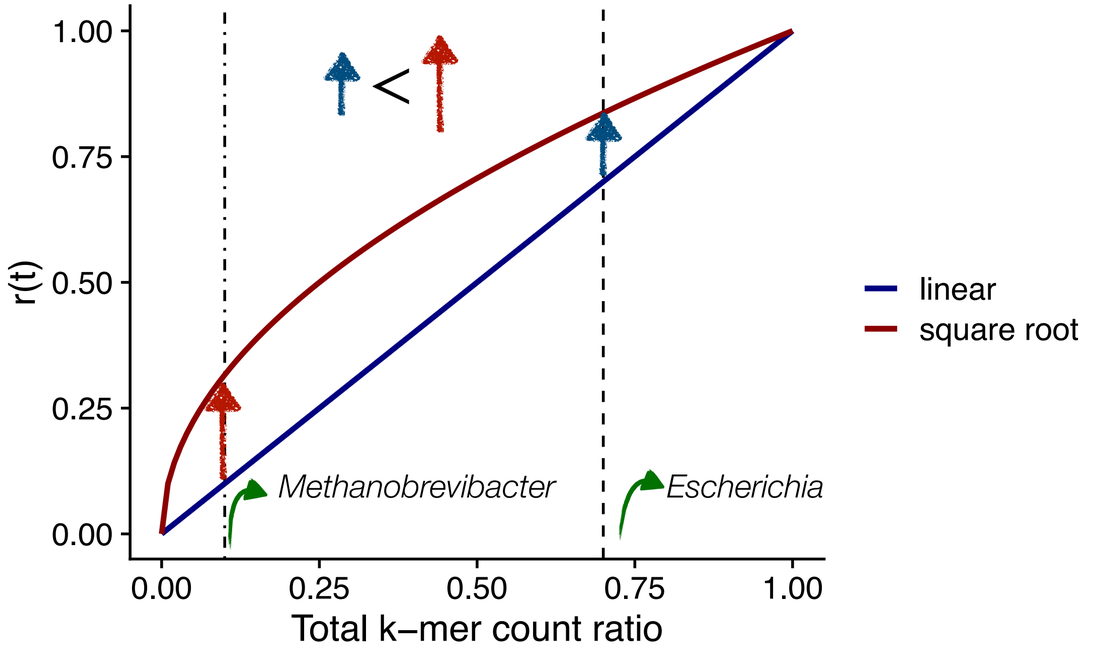
- Adaptive size constraint, $r(t)M$, on internal nodes
- $r(t)$ is a heuristic: square root of ratio of *k*-mers under t



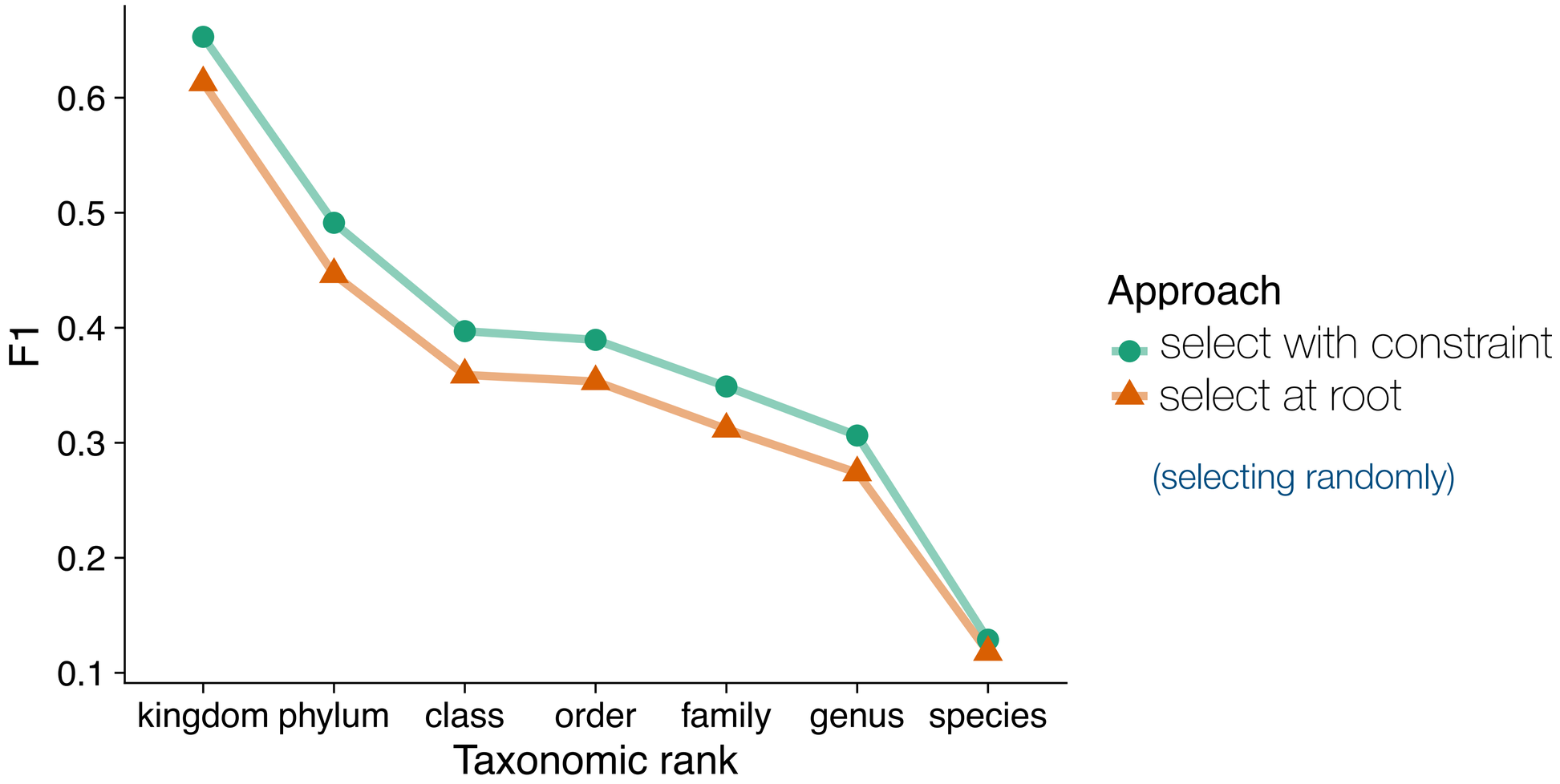
Gradual filtering is making some decisions earlier

Goal: remove *k*-mers from bloated taxa earlier & delay decisions for smaller taxa.

- Adaptive size constraint, $r(t)M$, on internal nodes
- $r(t)$ is a heuristic: square root of ratio of *k*-mers under t
- Concavity of $r(t)$ favors fewer taxa with k-mers (less diversity or sparsely sampled)



Adaptive size constraint improves classification



(empirical analysis using 3.2Gb, in WoL-v1 with 9k species, 10k genomes)

Q1: How many *k*-mers should we remove from each node/taxon?

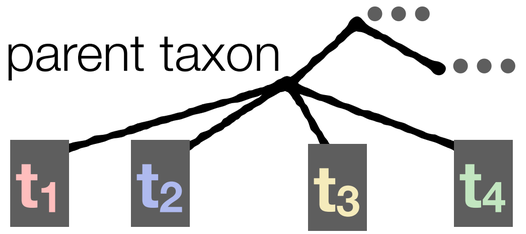
Q2: How do we rank *k*-mers to assess which one(s) should be kept?

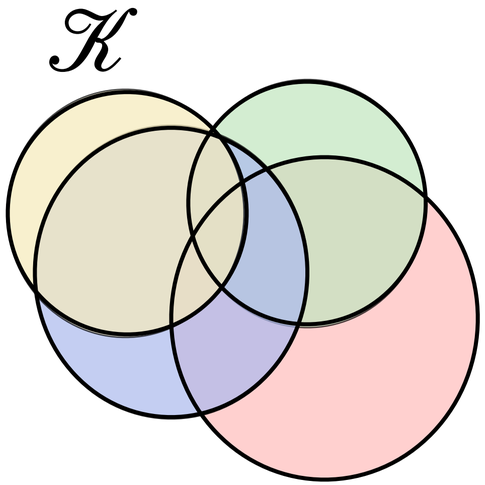
Which k-mers would provide better representation?

Baseline: select randomly until the constraint is satisfied.

Which k-mers would provide better representation?

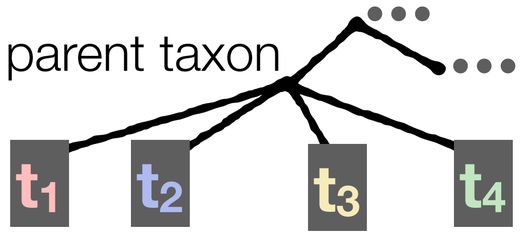
Baseline: select randomly until the constraint is satisfied.

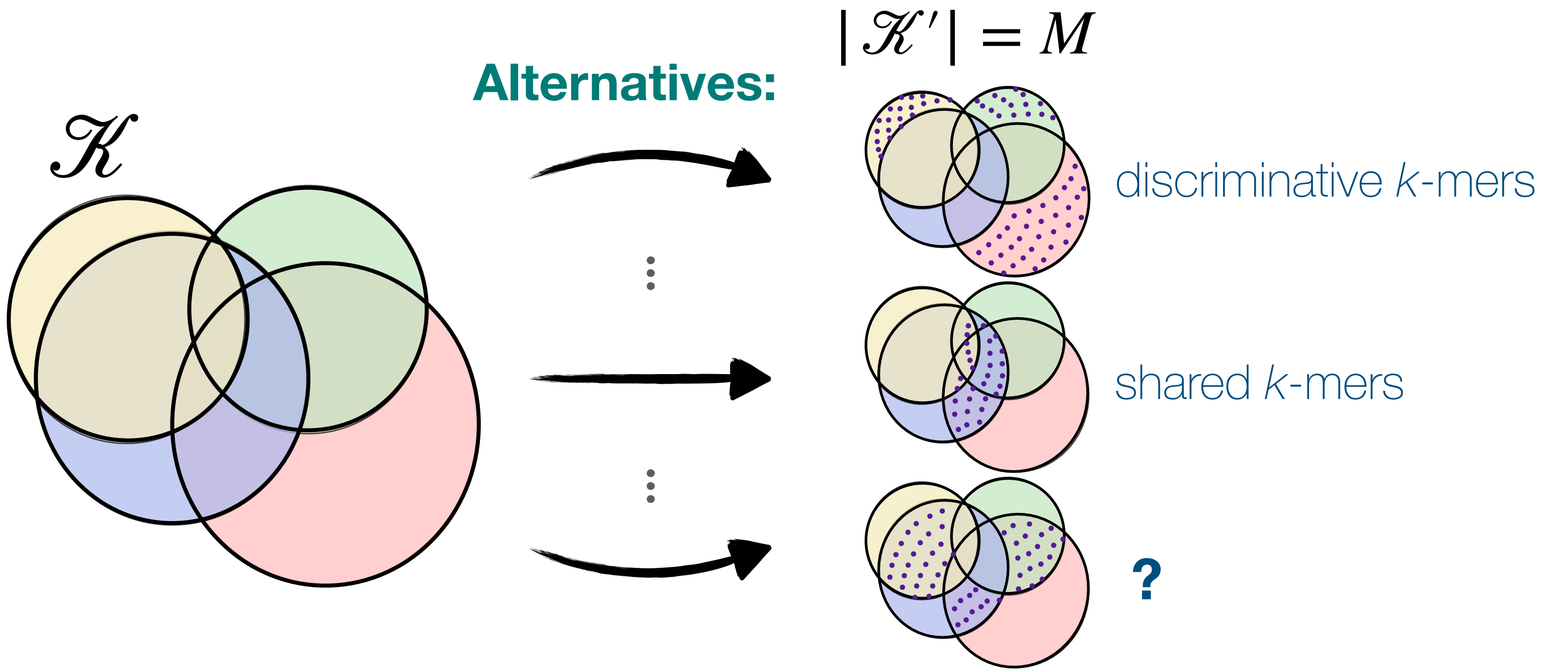




Which k-mers would provide better representation?

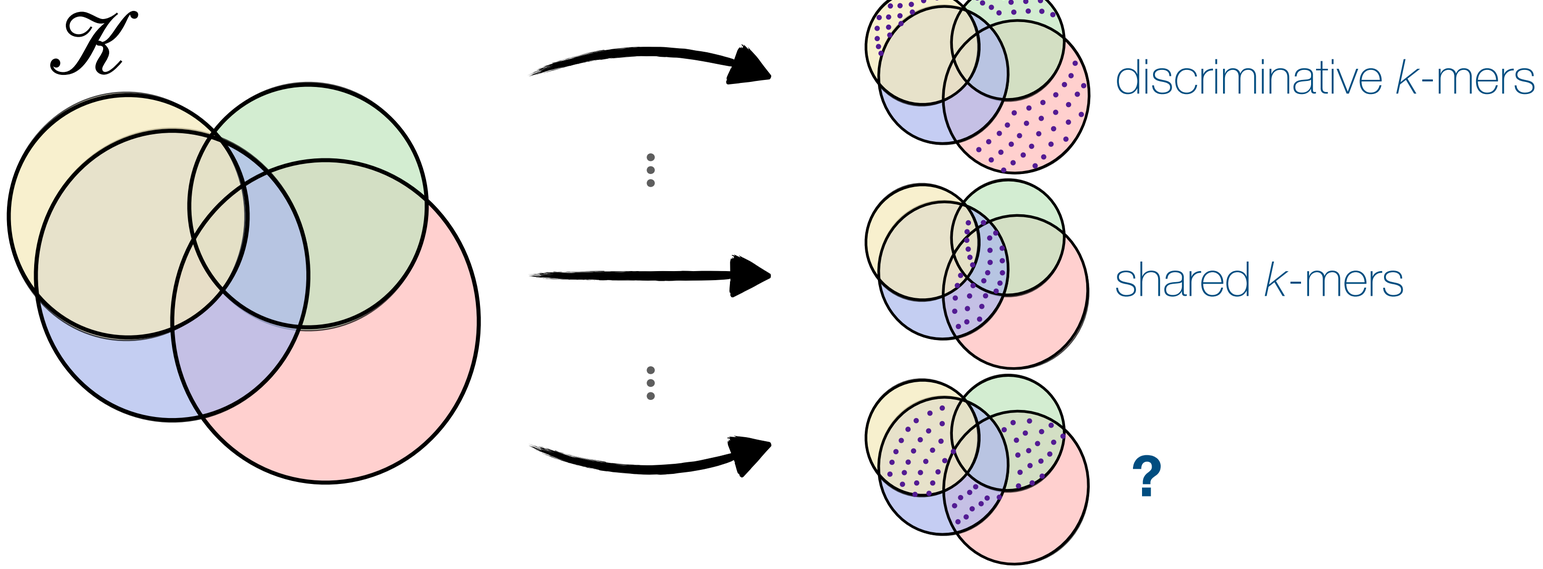
Baseline: select randomly until the constraint is satisfied.

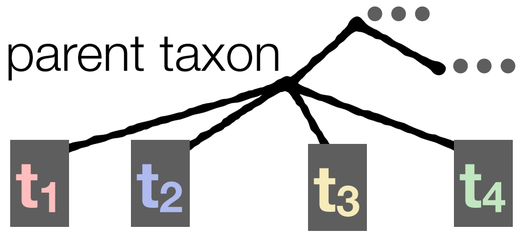




Which k-mers would provide better representation?

Baseline: select randomly until the constraint is satisfied.



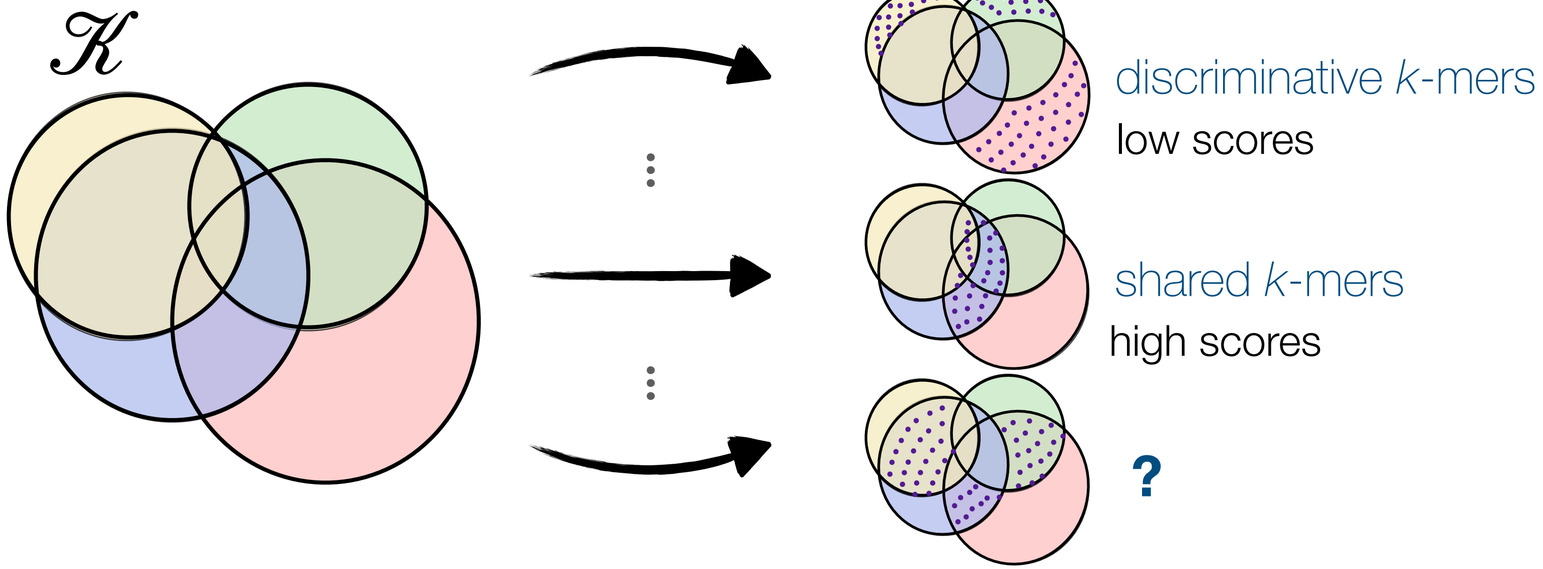


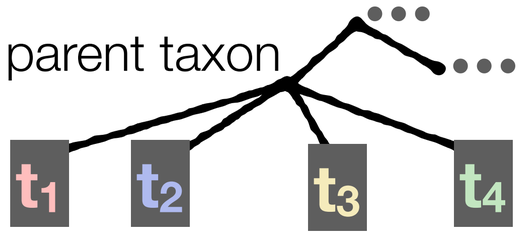
# of species under t with k-mer x

	x_1	x_2	x_3	...	$x_{\lvert\mathcal{K}'\rvert}$
t_1	4	7	0	...	3
t_2	0	0	2	...	0
t_3	0	0	1	...	1
t_4	2	2	1	...	0
Score:	**6**	**9**	**4**	**...**	**4**

Which k-mers would provide better representation?

Baseline: select randomly until the constraint is satisfied.

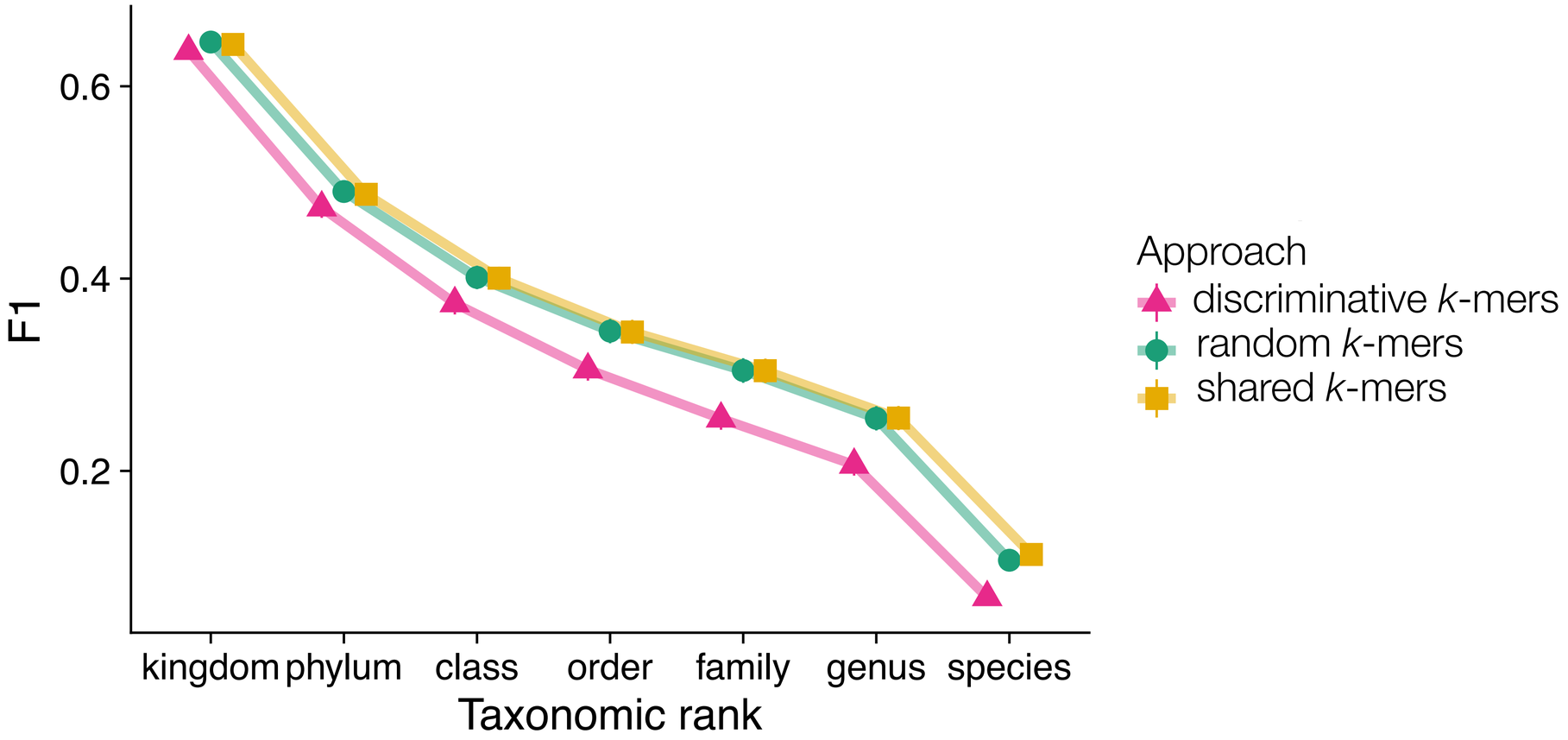




# of species under t with k-mer x

	x_1	x_2	x_3	...	$x_{\lvert\mathcal{K}\rvert}$
t_1	4	7	0	...	3
t_2	0	0	2	...	0
t_3	0	0	1	...	1
t_4	2	2	1	...	0
Score:	**6**	**9**	**4**	**...**	**4**

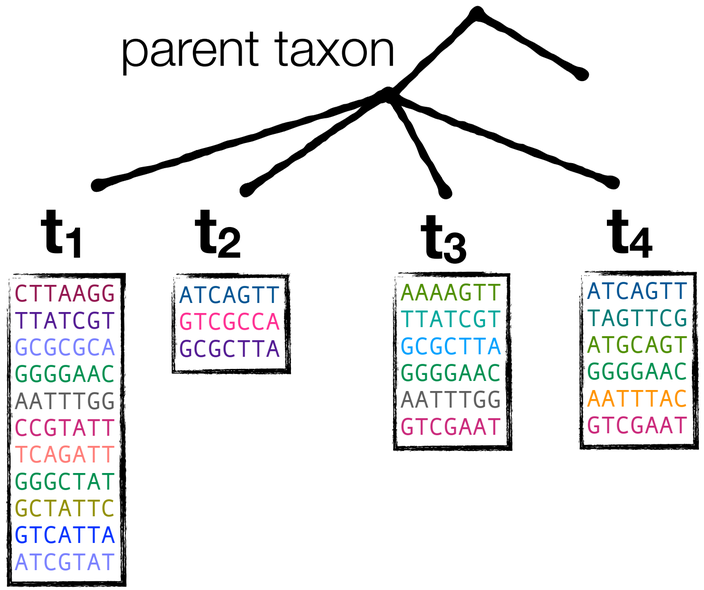
Neither discriminative nor shared k-mers improve the baseline



(empirical analysis using 3.2Gb, in WoL-v1 with 9k species, 10k genomes)

Incorporating taxon coverage in ranking

Intuition: keep shared *k*-mers but ensure no group is left uncovered.

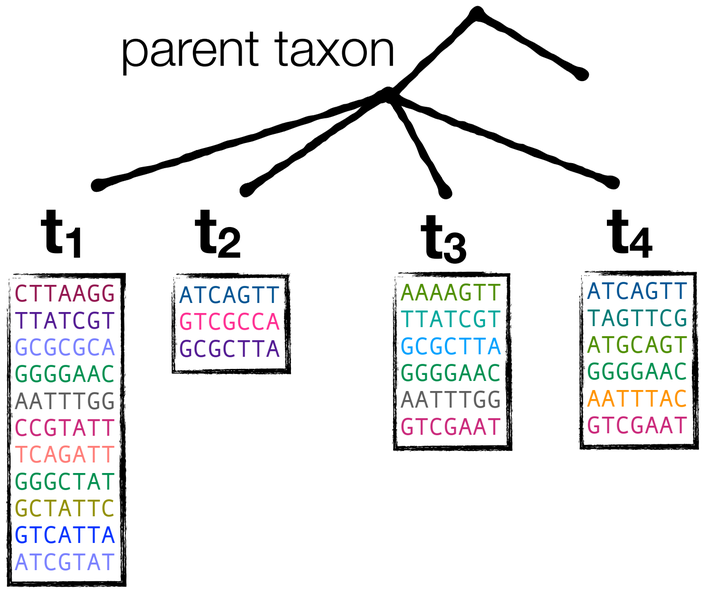


t_1: Afford to remove more!

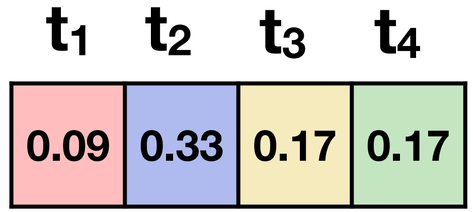
t_2: Needs to be prioritized!

Incorporating taxon coverage in ranking

Intuition: keep shared *k*-mers but ensure no group is left uncovered.



Scalable heuristic: down-weight the impact of taxa that are highly covered among surviving k-mers:

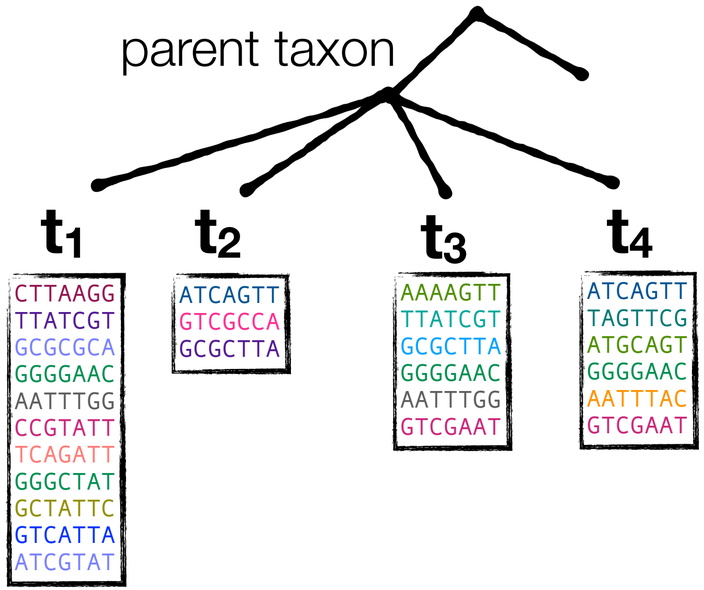


t_1: Afford to remove more!

t_2: Needs to be prioritized!

Incorporating taxon coverage in ranking

Intuition: keep shared *k*-mers but ensure no group is left uncovered.

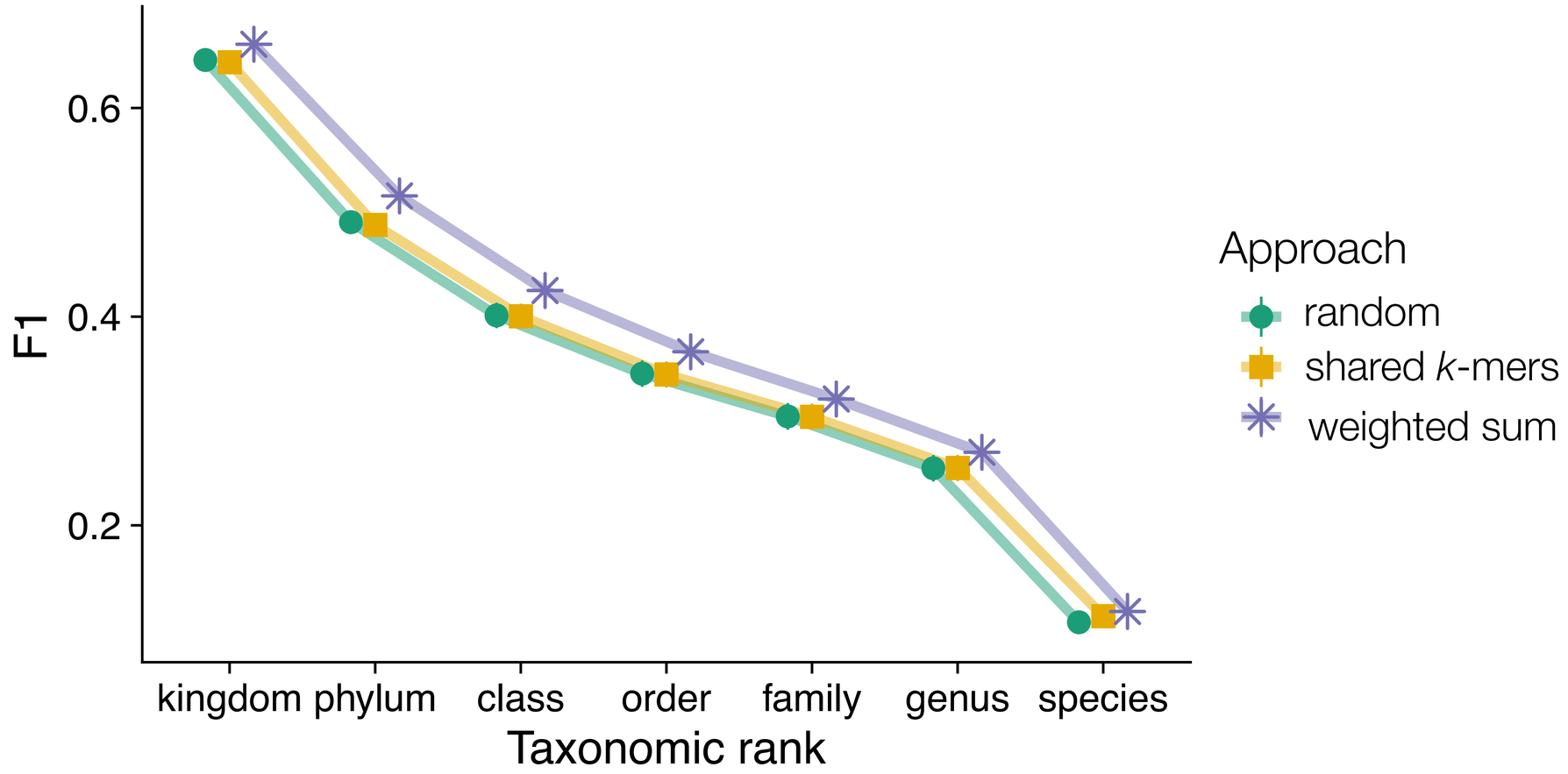


Scalable heuristic: down-weight the impact of taxa that are highly covered among surviving k-mers:

# of species under t with k-mer x

weights of taxa		x_1	x_2	x_3	...	$x_{\lvert\mathcal{K}\rvert}$
0.09	t_1	4	7	0	...	3
0.33	t_2	0	0	2	...	0
0.17	t_3	0	0	1	...	1
0.17	t_4	2	2	1	...	0
	Score:	**0.7**	**0.97**	**1**	**...**	**0.44**

Neither discriminative nor shared k-mers improve the baseline



(empirical analysis using 3.2Gb, in WoL-v1 with 9k species, 10k genomes)

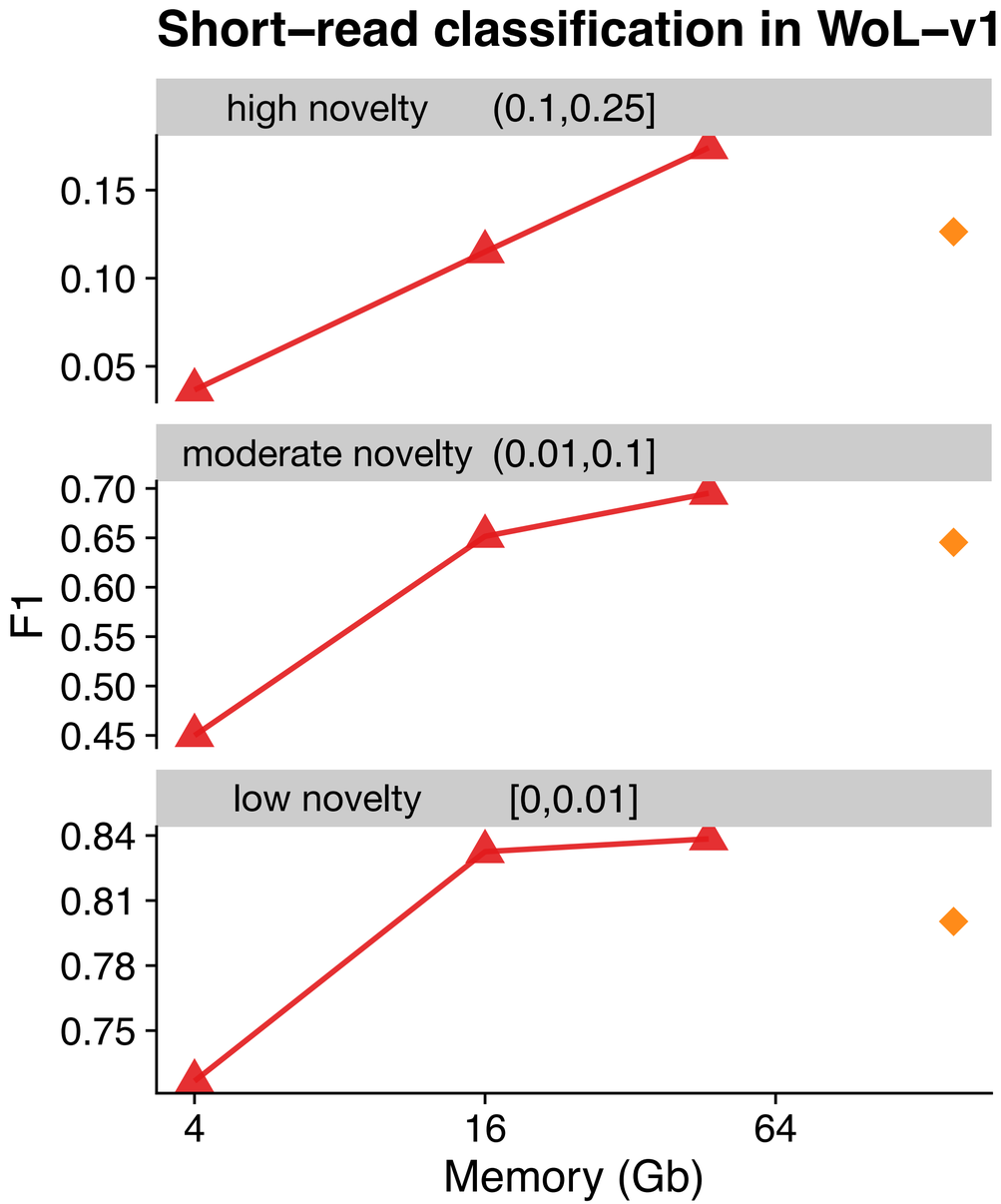
- **KRANK** puts all these heuristics together:
  - weighted-sum ranking + adaptive size constraint
  - other minor tricks
  - highly optimized and scalable implementation

KRANK builds lightweight and robust reference libraries

- Simulated reads across different novelty levels.

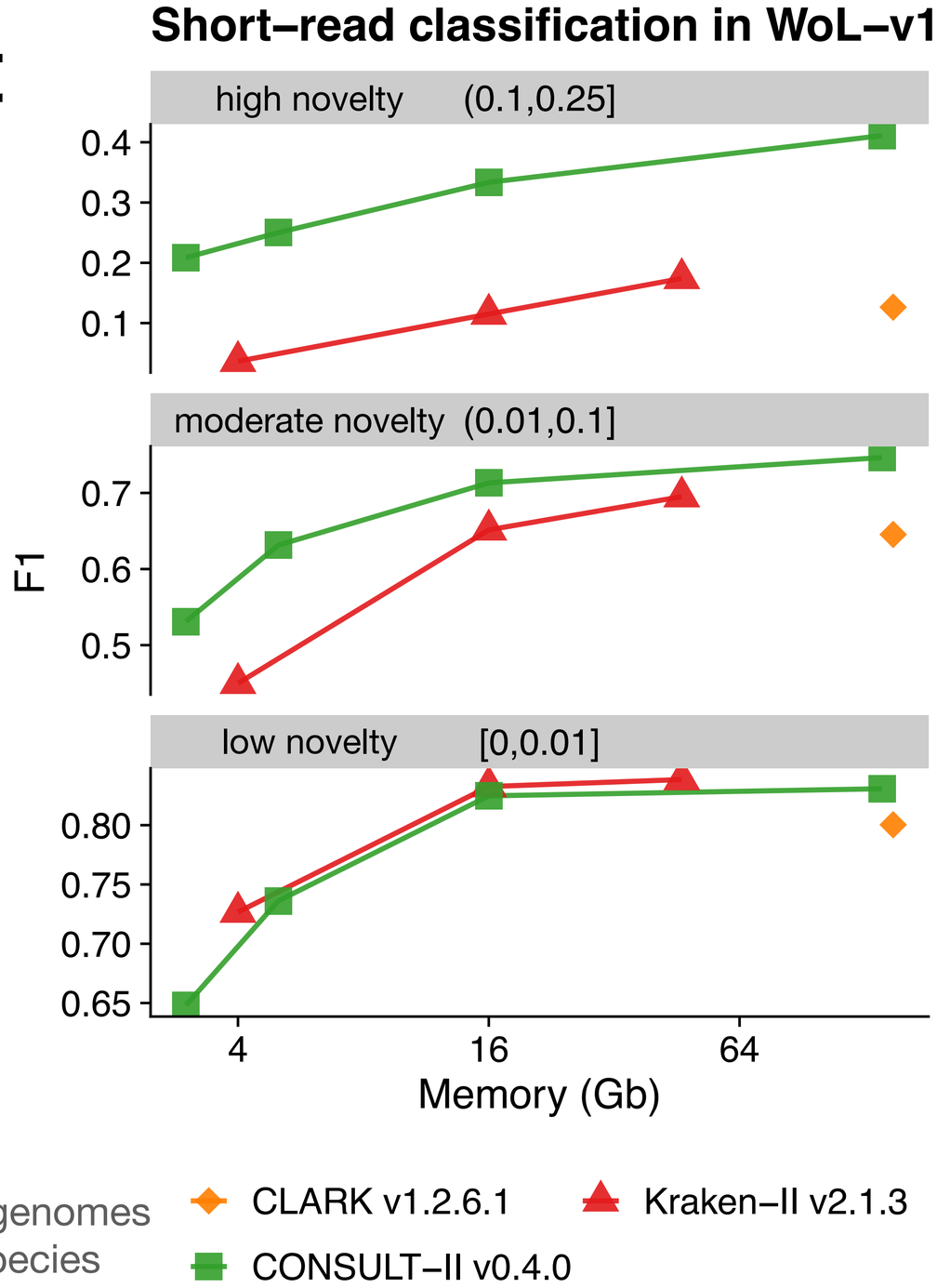
KRANK builds lightweight and robust reference libraries

- Simulated reads across different novelty levels.
- Adjusting the memory usage and observing the impact on the performance.



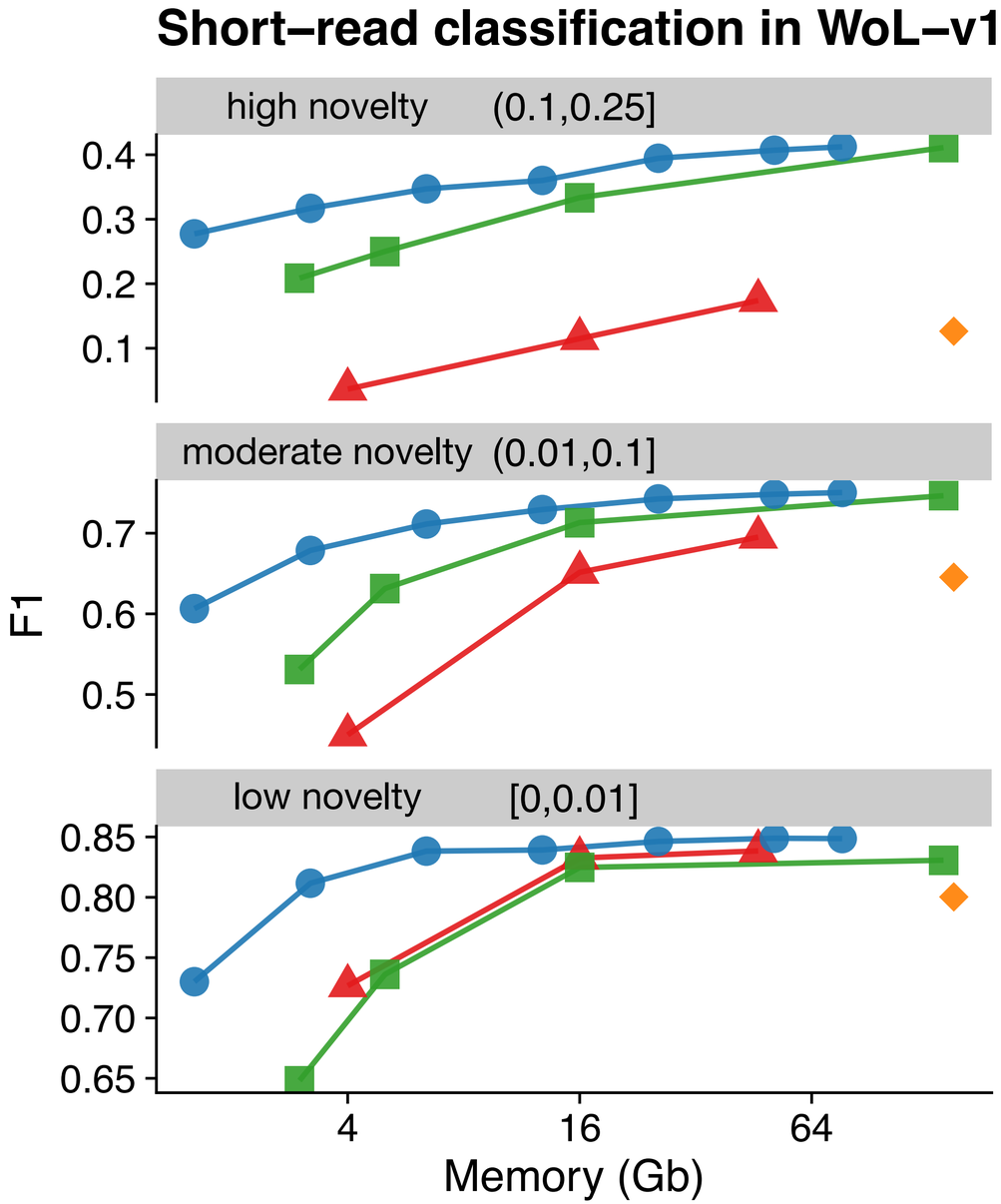
KRANK builds lightweight and robust reference libraries

- Simulated reads across different novelty levels.
- Adjusting the memory usage and observing the impact on the performance.



KRANK builds lightweight and robust reference libraries

- Simulated reads across different novelty levels.
- Adjusting the memory usage and observing the impact on the performance.
- KRANK preserves the same level of robust performance with much smaller *k*-mer subsets!

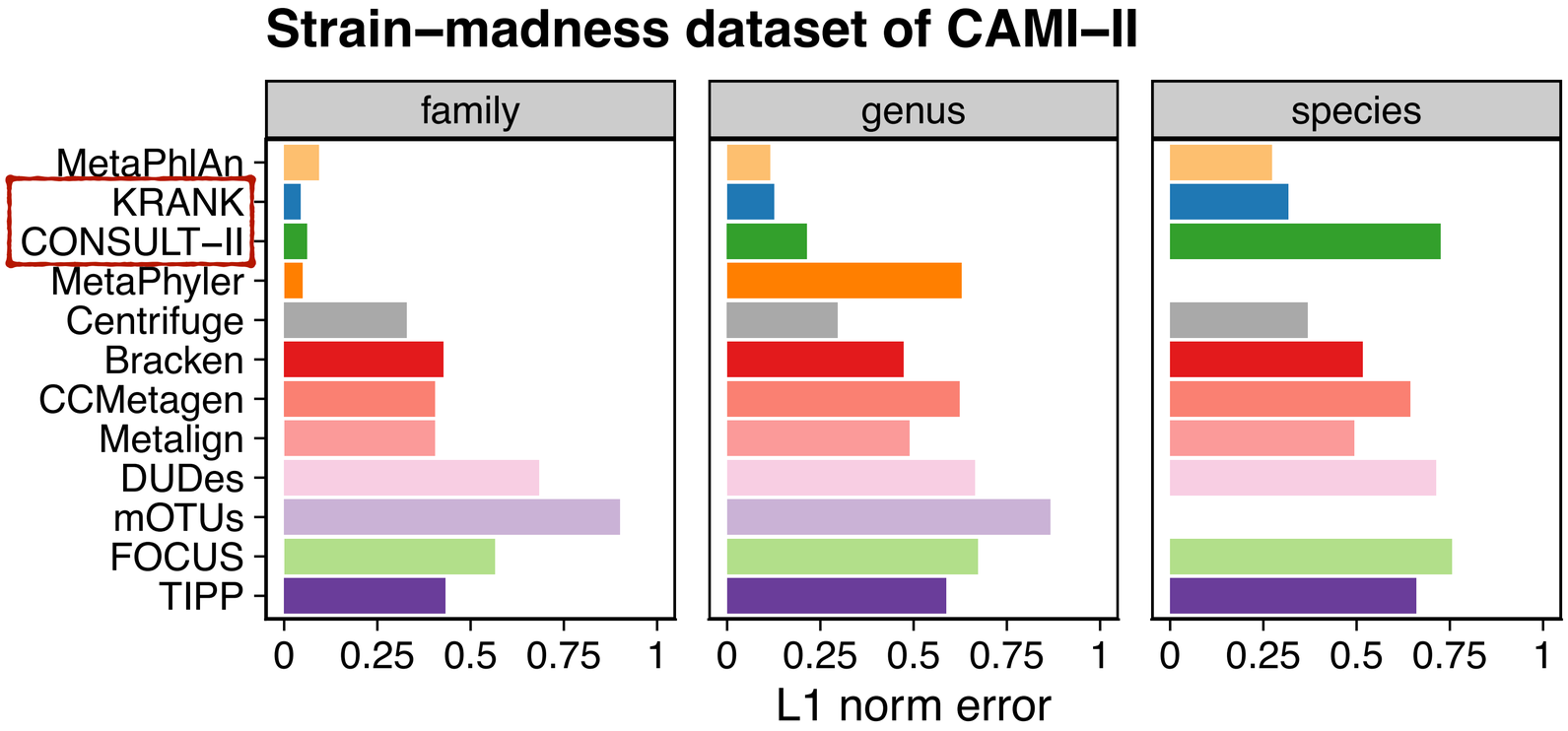


Boosting the performance in CAMI-II with a smaller subset

- Library construction: 3-hours (36 nodes × 14 cores) for RefSeq genomes (2019)

Boosting the performance in CAMI-II with a smaller subset

- Library construction: 3-hours (36 nodes × 14 cores) for RefSeq genomes (2019)
- Consistently improves CONSULT-II across all ranks

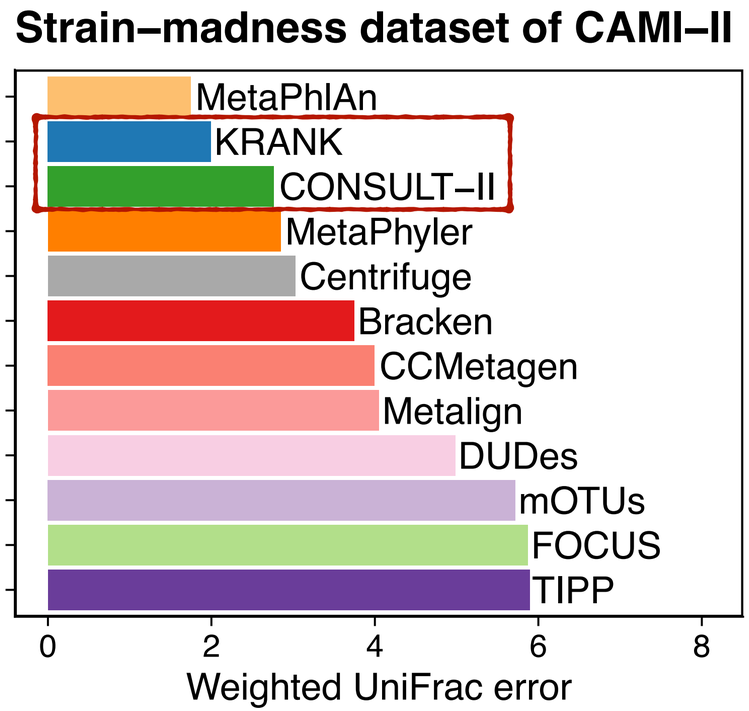


CONSULT-II: 140Gb

KRANK: 51Gb

Boosting the performance in CAMI-II with a smaller subset

- Library construction: 3-hours (36 nodes ✕ 14 cores) for RefSeq genomes (2019)
- Consistently improves CONSULT-II across all ranks
- Second-best tool according to rank-invariant UniFrac error



CONSULT-II: 140Gb

KRANK: 51Gb

- KRANK uses taxonomy to subsample large k-mer databases
  - based on carefully chosen heuristics
  - Used in combination with minimizers

- Future work includes:
  - exploring alternatives strategy a more principled approach,
    - better modeling of imbalance
    - using a phylogenetic tree
  - pairing KRANK with other classification methods
  - pairing with sketching algorithms

- KRANK uses taxonomy to subsample large k-mer databases
  - based on carefully chosen heuristics
  - Used in combination with minimizers

- Future work includes:
  - exploring alternatives strategy a more principled approach,
    - better modeling of imbalance
    - using a phylogenetic tree
  - pairing KRANK with other classification methods
  - pairing with sketching algorithms

Extra Slides

The case against discriminative k-mers

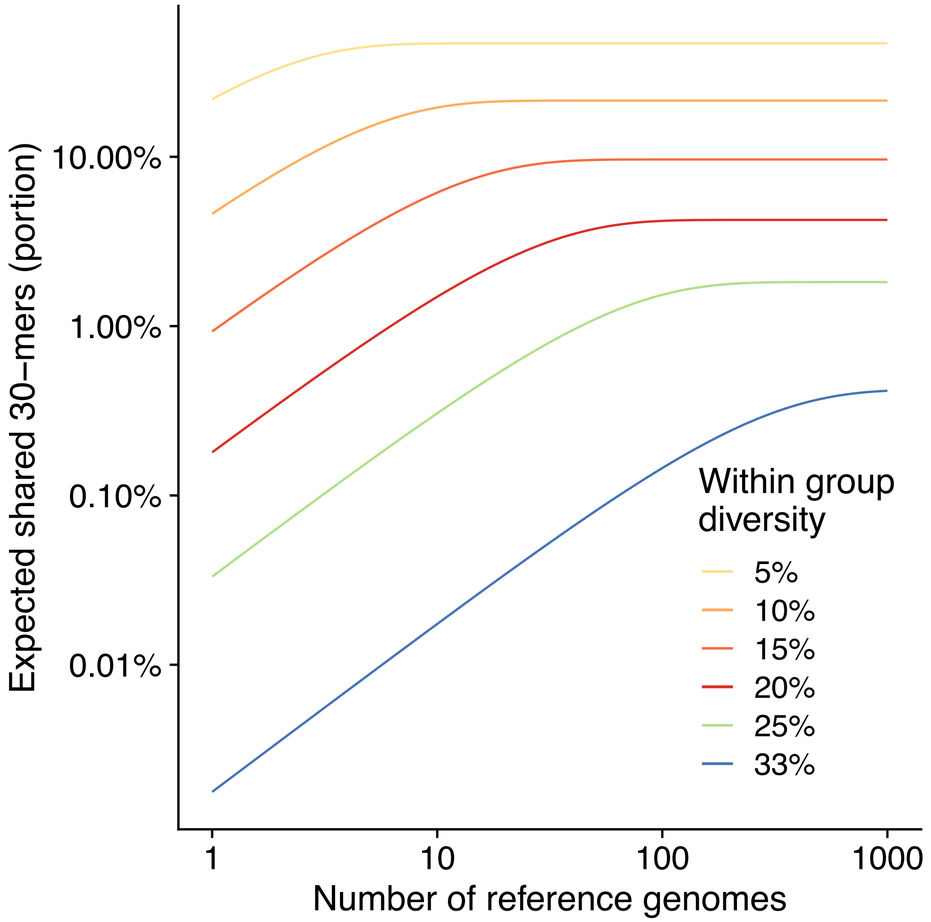
- **Problem:** considerably small portion of *k*-mers are shared within a group! (it gets worse for upper ranks)
  - **Claim:** Removing common *k*-mers will make it difficult to find matches!

Given a query genome, what is the expected portion of shared *k*-mers in a reference set with N genomes within $2d$ distance?

$$(1-d)^k\left(1-(1-(1-d)^k)^N\right)$$

k-mer from the ancestor stays same

k-mer from the ancestor changes in all N



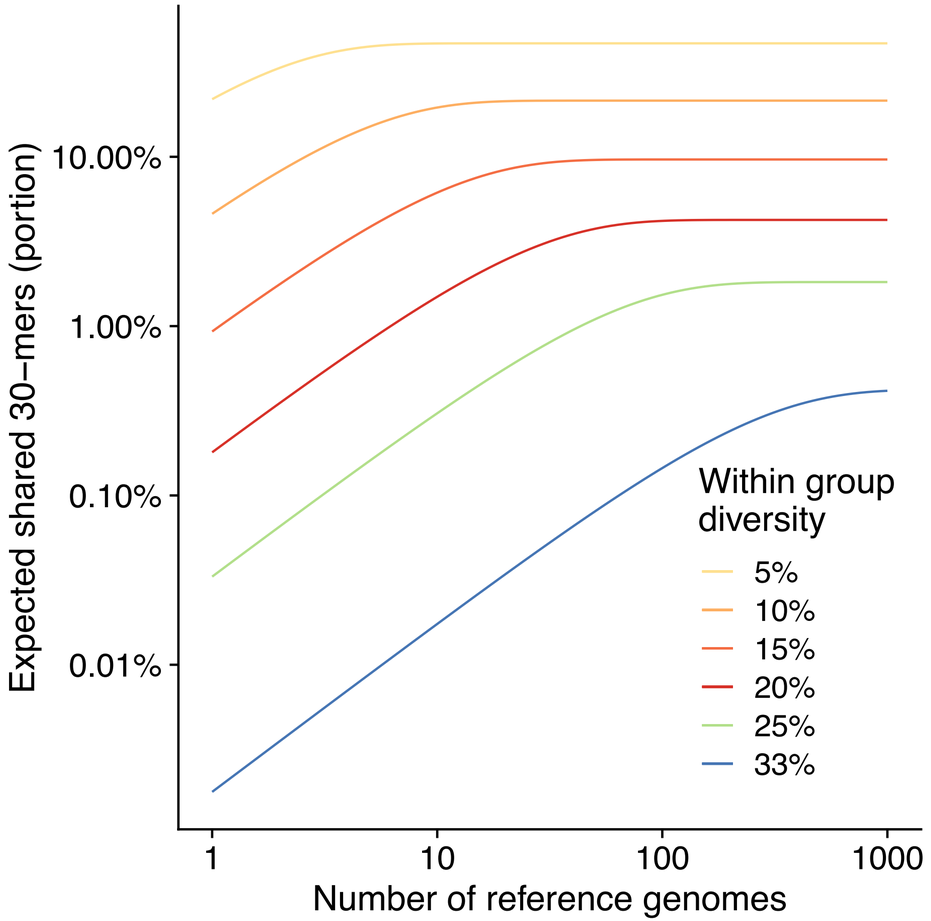
The case against discriminative k-mers

- **Problem:** considerably small portion of *k*-mers are shared within a group! (it gets worse for upper ranks)
  - **Claim:** Removing common *k*-mers will make it difficult to find matches!

Example: within d = 20% diversity (~genus)

- $N = 5$: 0.7% of query 30-mers,
- $N \to \infty$: 4.2% of query 30-mers,

will be found in at least one reference.



Bonus: compact k-mer encodings

CONSULT-II used 2 bits per letter: 64bit for 32-mers.

We only compute HD between *k*-mers that have the same hash value!

We do not need h positions used to compute LSH; they are already the same!

Just drop LSH positions and store the rest: $k = 32, h = 16 \rightarrow$ 32bit

Improvements are pronounced at higher ranks

- KRANK 13Gb competes with CONSULT-II 144Gb.
- Novel queries were accurately classified at higher ranks.
- With little memory, KRANK+CONSULT-II is highly sensitive.