The unrealized potential of genome skimming for sample identification

Siavash Mirarab

UC, San Diego (ECE Department)

Joint work with Shahab Sarmashghi, Metin Balaban, Nora Rachtman, Kristine Bohmann, Vineet Bafna, Tom Gilbert

MOLECULAR ECOLOGY

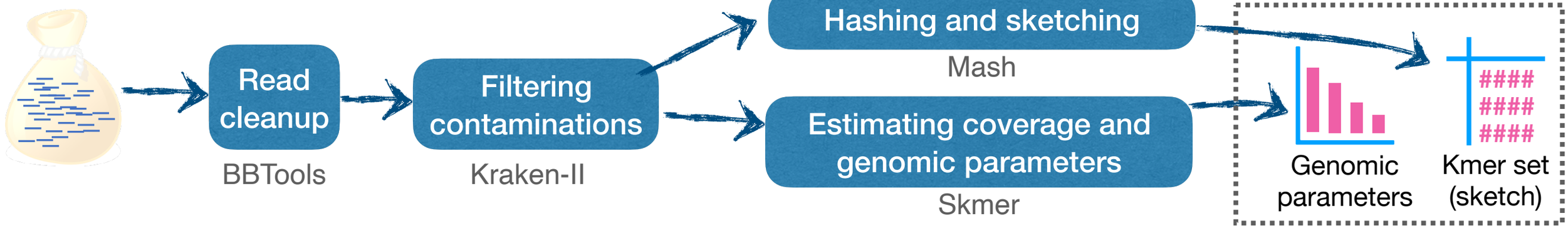
NEWS AND VIEWS | Open Access

Beyond DNA barcoding: The unrealized potential of genome skim data in sample identification

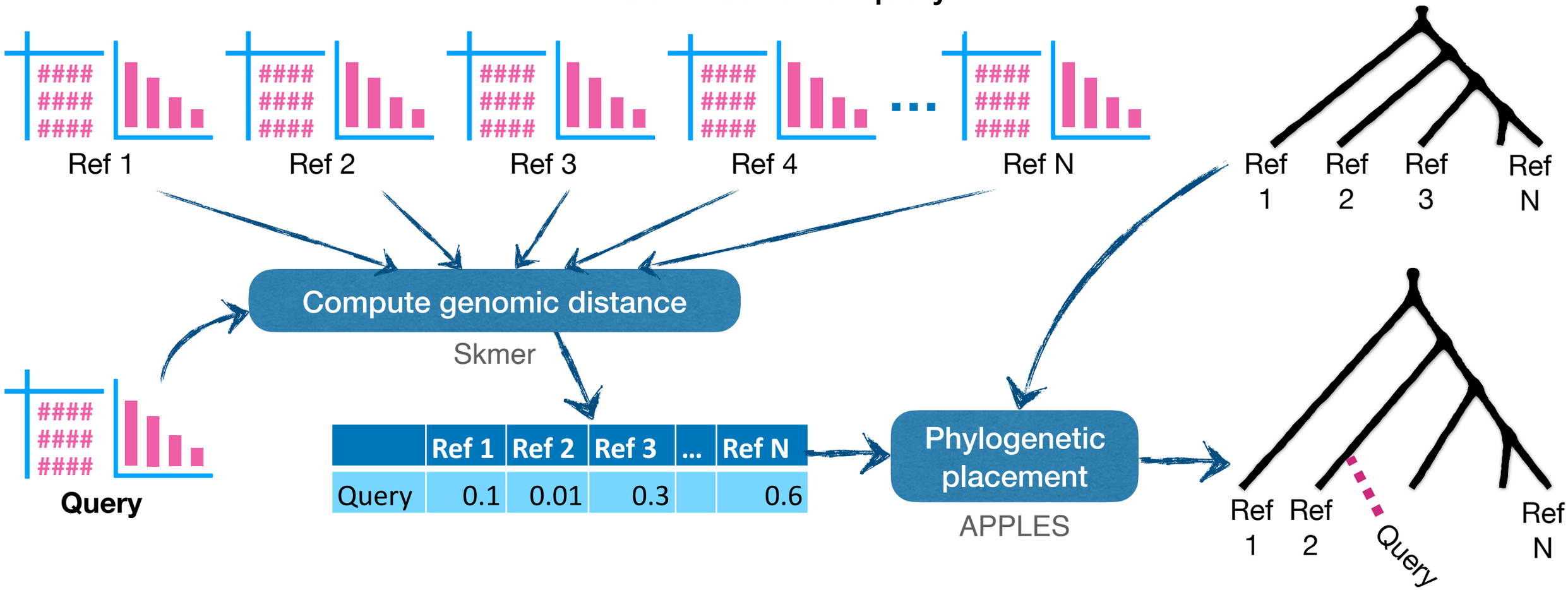
Kristine Bohmann, Siavash Mirarab, Vineet Bafna, M. Thomas P. Gilbert

First published: 16 June 2020 | https://doi.org/10.1111/mec.15507

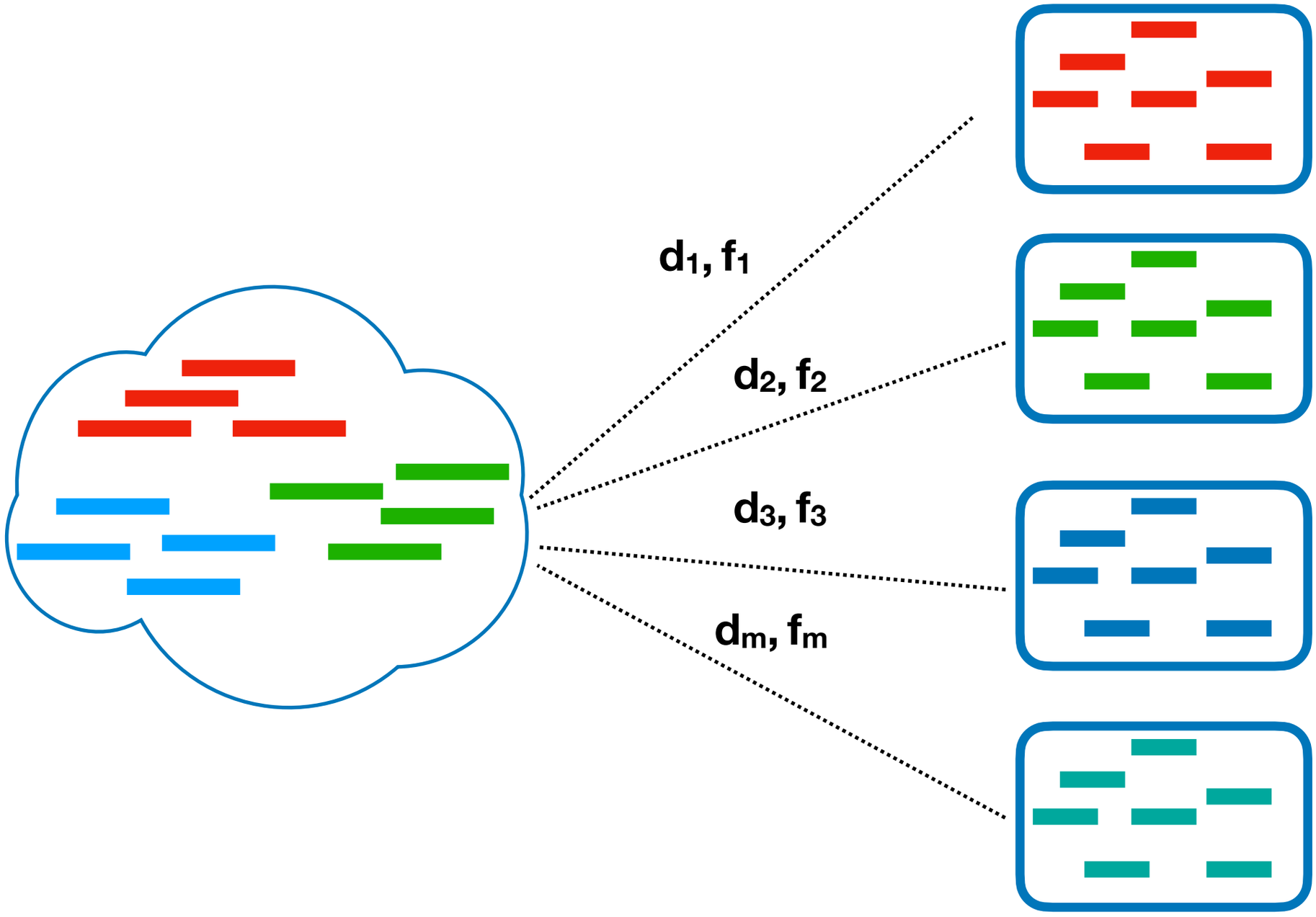
Preprocessing of query and references



Identification of query



Mixed-sample analysis



How to handle?

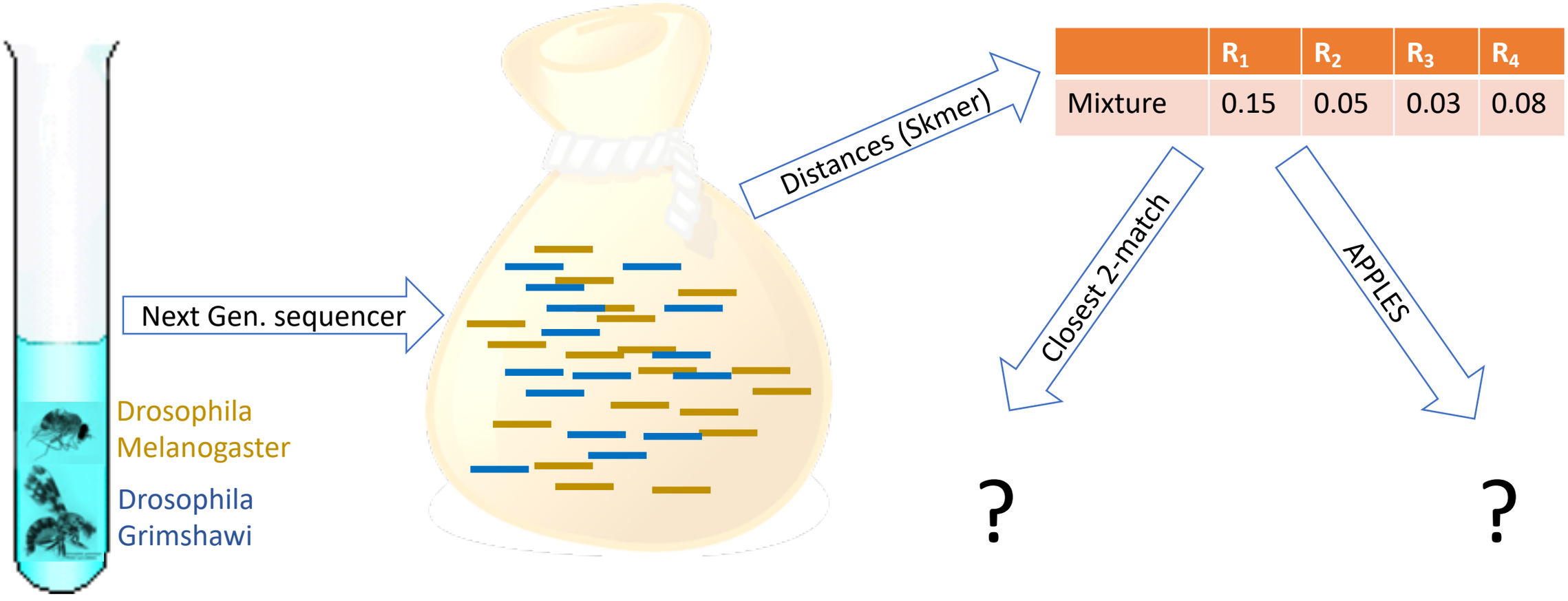
- Scenario 1: mixture of very different species (e.g., GND > 20%)

    - Divide reads into groups using classification tools (e.g., CONSULT, Kraken-II) using close (enough) references

How to handle?

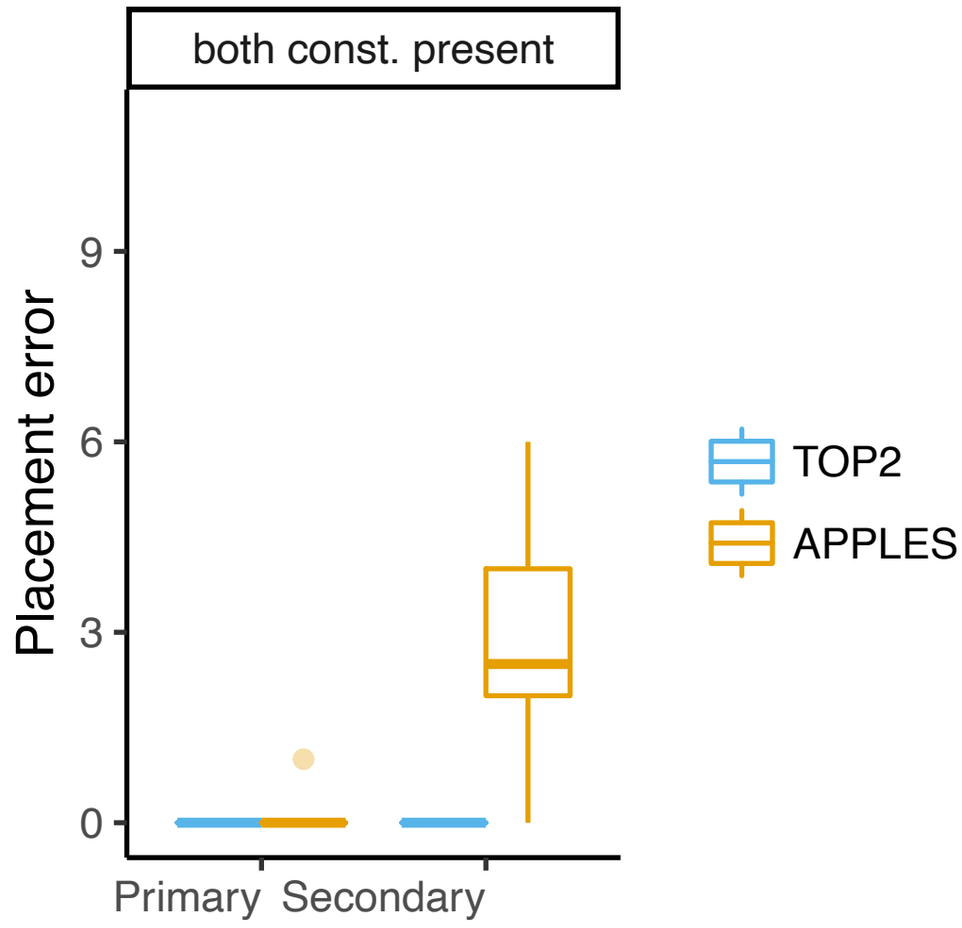
- Scenario 1: mixture of very different species (e.g., GND > 20%)
  - Divide reads into groups using classification tools (e.g., CONSULT, Kraken-II) using close (enough) references
- Scenario 2: mixture of very close species
  - If you have reference genomes of constituents, you can always do read mapping
  - But what if you don’t?

How about a query that is a mixture of two genomes?



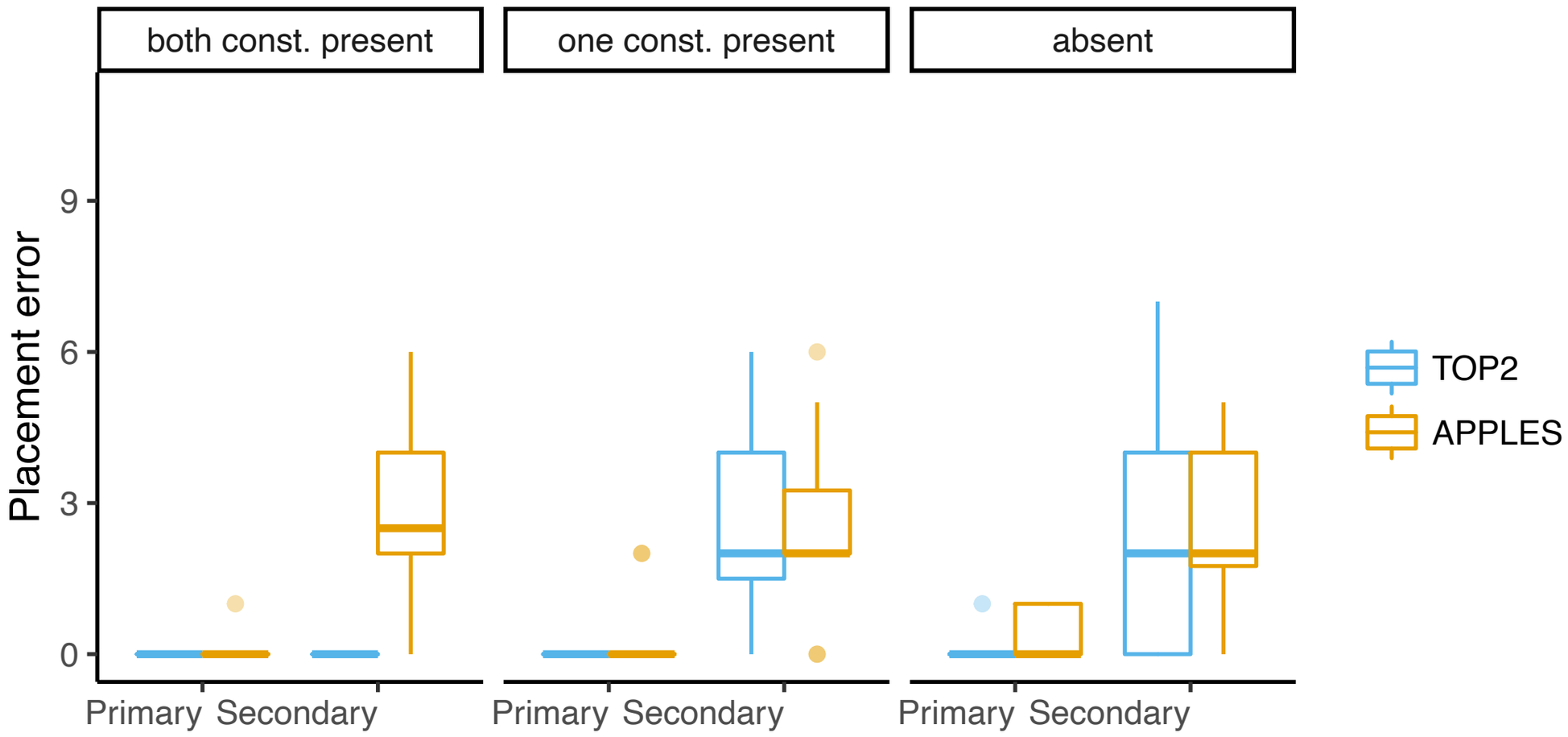
Do existing methods work?

- When both constituents are present
  - The closest two species give correct placements
  - APPLES finds one of the two



Do existing methods work?

- When constituents are not present, they can find only one.



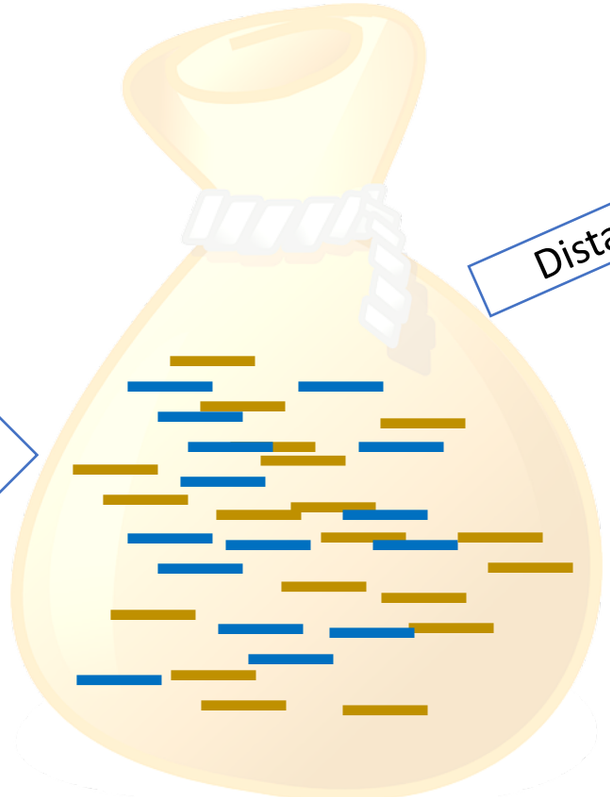
Distance decovolution

Deconvolve the distances of the mixture to a set of references to constituent parts

Drosophila Melanogaster

Drosophila Grimshawi

Short read sequencer



Distances (Mash)

	R_1	R_2	R_3	R_4
Mixture	0.15	0.05	0.03	0.08

Magical Deconvolver

	R_1	R_2	R_3	R_4
D. Melan.	0.16	0.10	0.09	0.08

	R_1	R_2	R_3	R_4
D. Grims.	0.15	0.04	0.03	0.10

Reverse engineer

	R_1	R_2	R_3	R_4
D. Melan.	0.16	0.10	0.09	0.08

+

	R_1	R_2	R_3	R_4
D. Grims.	0.15	0.04	0.03	0.10

=

	A	B	C	D
Mixture	?	?	?	?

- Can we obtain mixture distances given constituent distances to reference?

Reverse engineer

	R_1	R_2	R_3	R_4
D. Melan.	0.16	0.10	0.09	0.08

+

	R_1	R_2	R_3	R_4
D. Grims.	0.15	0.04	0.03	0.10

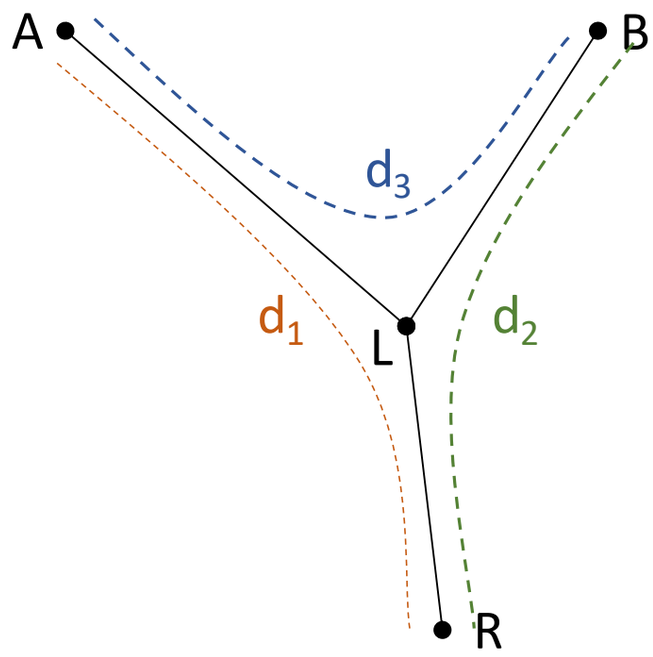
=

	A	B	C	D
Mixture	?	?	?	?

- Can we obtain mixture distances given constituent distances to reference?

Distance of mixture

$$(1-\hat{d}_{MR})^k = 2\frac{(1-d_1)^k+(1-d_2)^k-\left(1-\frac{d_1+d_2+d_3}{2}\right)^k}{3-(1-d_3)^k}$$



Thus …

- Theorem: When the reference library includes both constituents: the smallest two distances are to the constituent references.

Thus …

- Theorem: When the reference library includes both constituents: the smallest two distances are to the constituent references.

- When one or both constituents are missing:

- For each pair of edges on the tree,

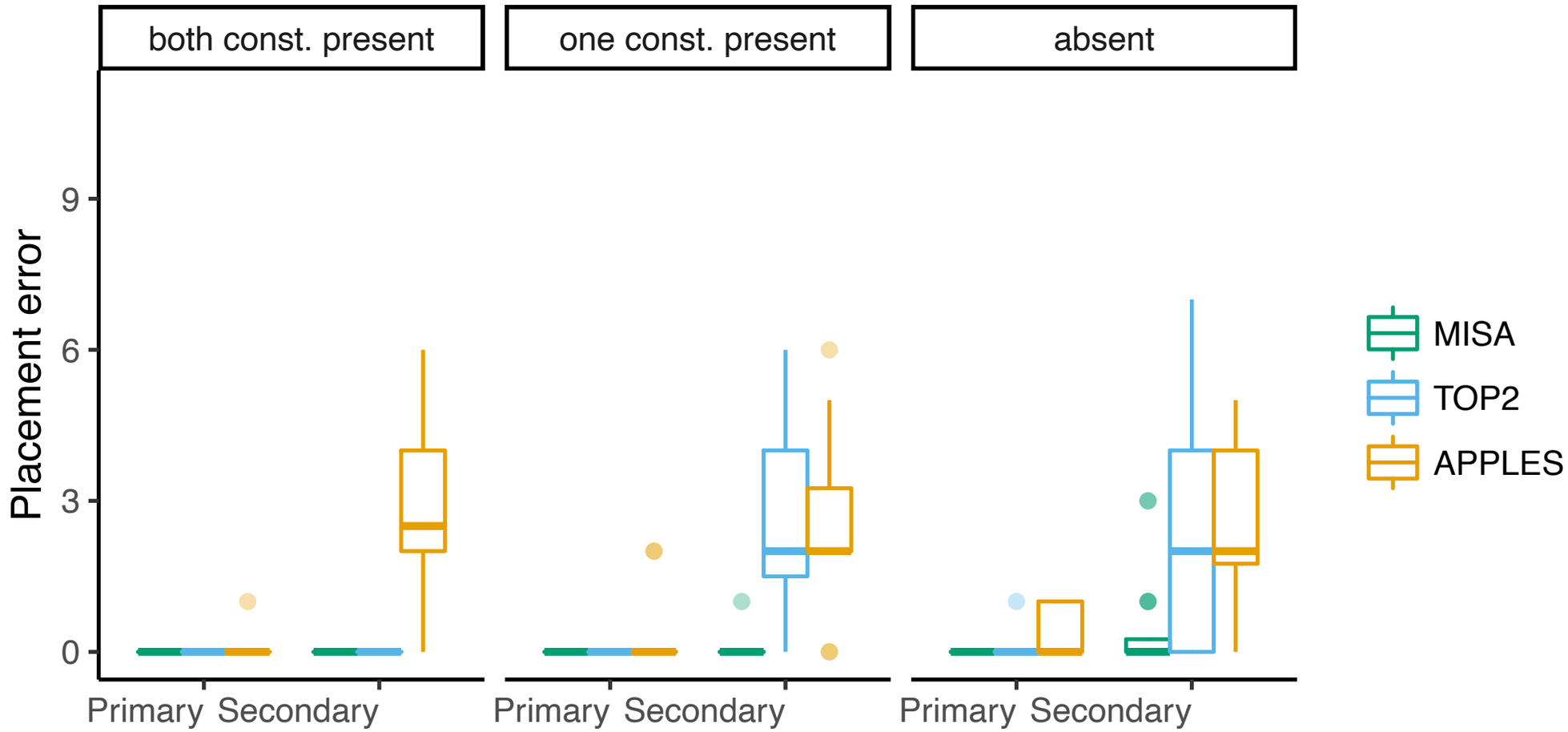
    - Numerically find (1) branch lengths and (2) deconvolved distances that minimize the phylogenetic placement error while *enforcing* the reversed engineered formula

- Return the pair of edges with minimum distance error

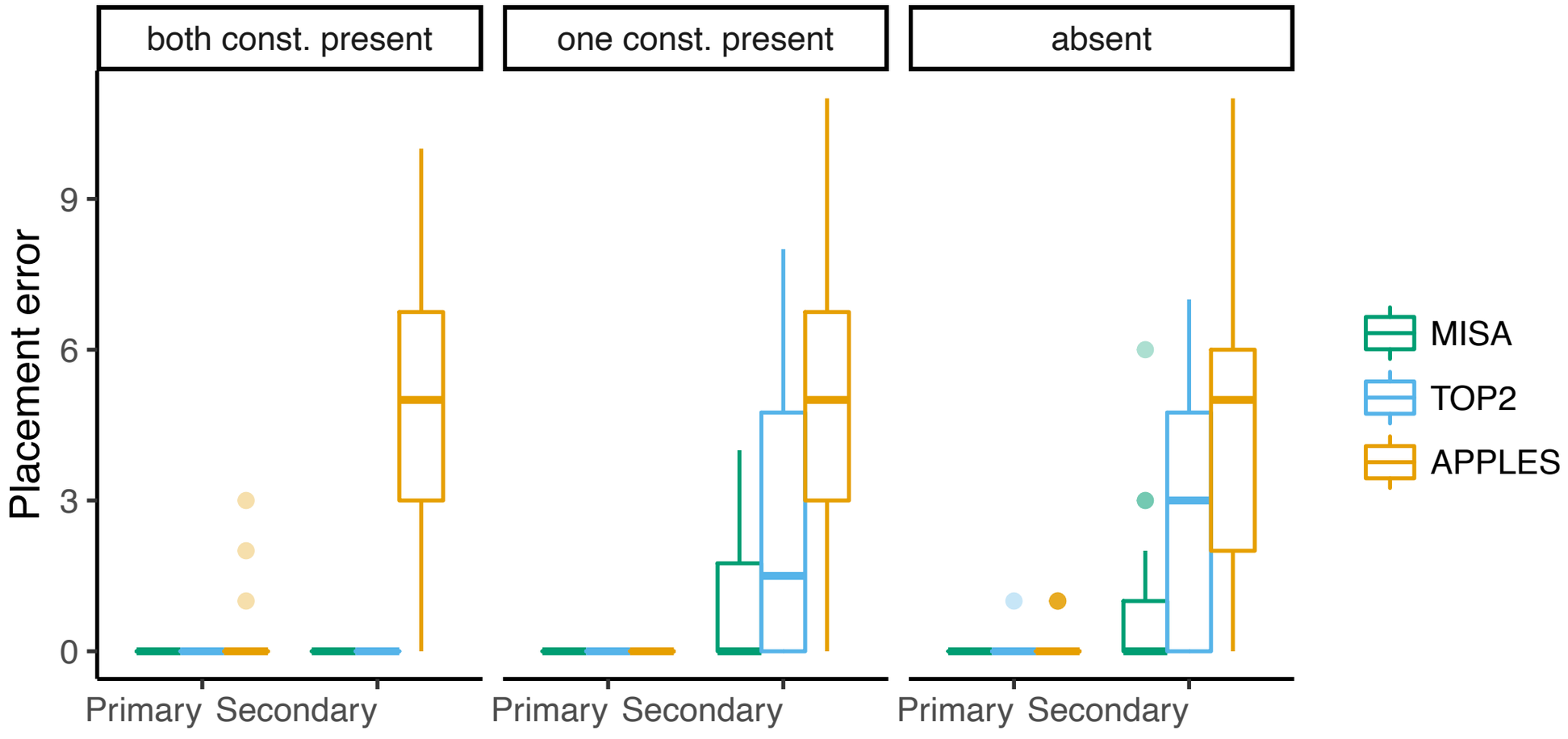
MISA

- Software:
  - https://github.com/balabanmetin/misa
- Paper:
  - M. Balaban, and S. Mirarab. “Phylogenetic Double Placement of Mixed Samples.” Bioinformatics Vol. 36, no. Supplement_1 (2020): pp. i335–43. doi:10.1093/bioinformatics/btaa489.

Revisiting results



Mixtures of skims



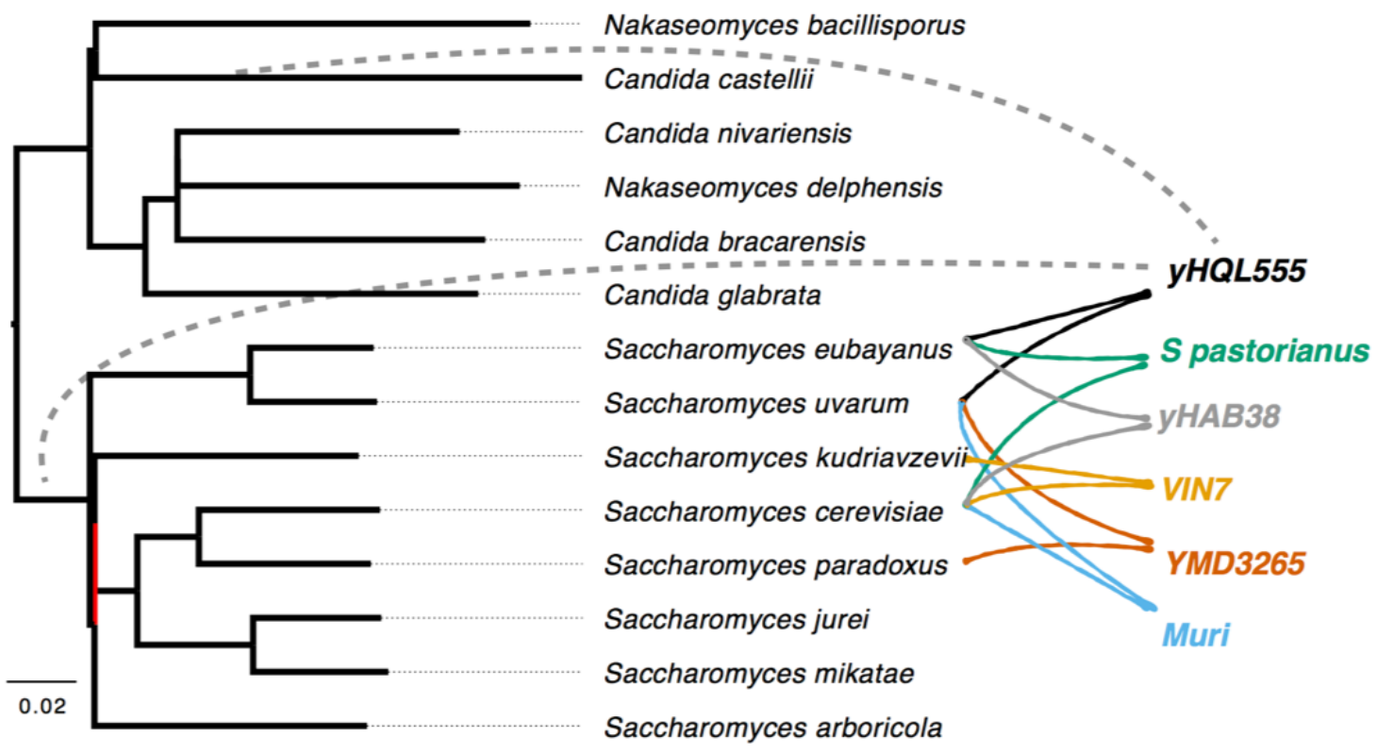
Can even detects (recent allopolyploid) hybrids

- 15 Pure yeast species from NCBI.
  6 hybrids with ncestors are among 15

- Perfect accuracy when one or both constituents are present

- 5/6 correct when both constituents are absent

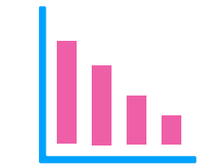
  - Still error is <2 edges



Tutorial

How about repeats?

- Recall estimating parameters from genome skims
  - Genome length + coverage

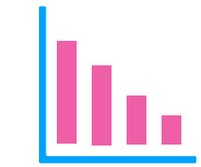


Genomic parameters

X. Li and M. S. Waterman, 2003, Genome research
Hozza et al., 2015,String Processing and Information Retrieval

How about repeats?

- Recall estimating parameters from genome skims
  - Genome length + coverage
- Repeat spectra
  - Especially important for population analysis
  - Previous works have focused on high coverage, or used parametric models for repeat spectra



Genomic parameters

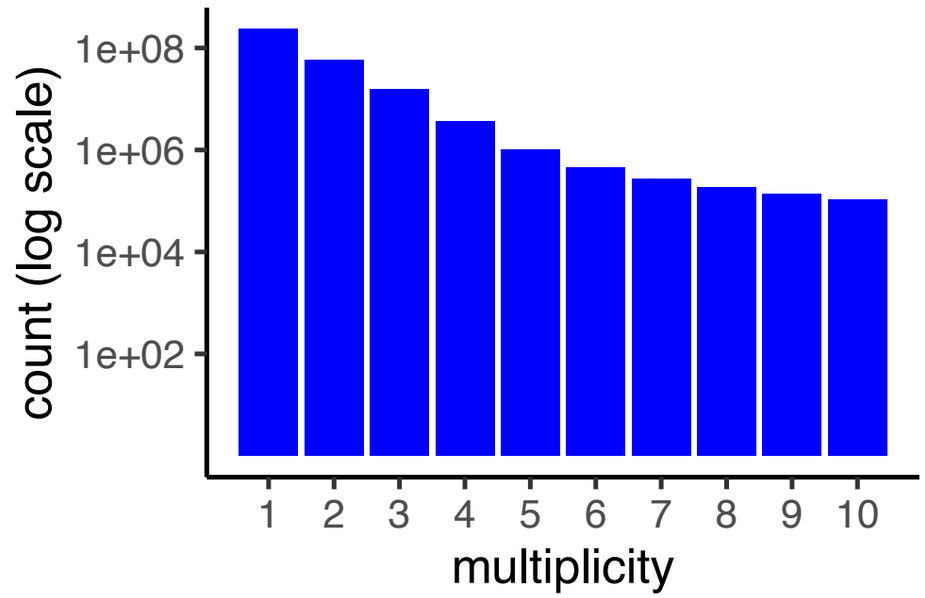
X. Li and M. S. Waterman, 2003, Genome research
Hozza et al., 2015,String Processing and Information Retrieval

K-mer repeat spectra

- The histogram of k-mer counts

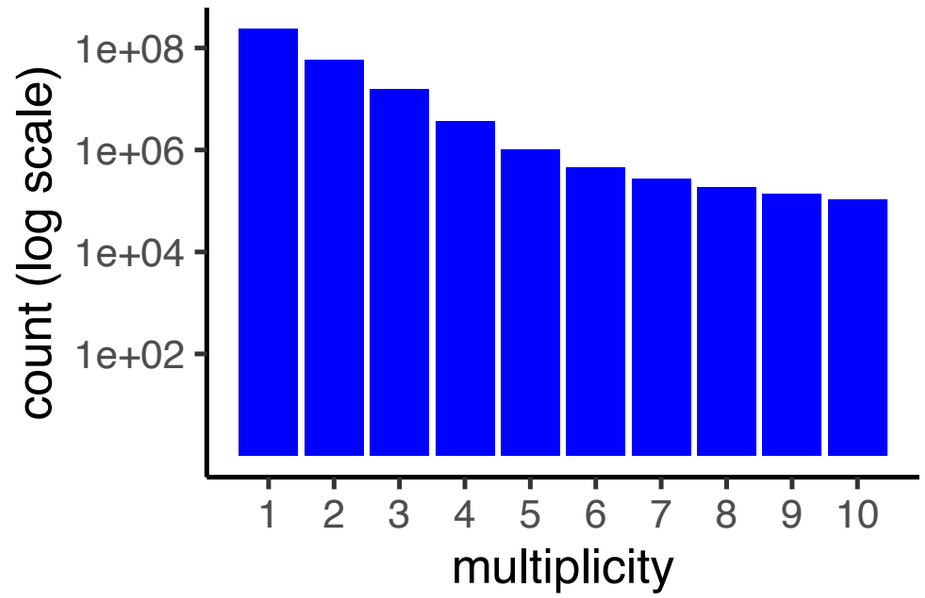
K-mer repeat spectra

- The histogram of k-mer counts



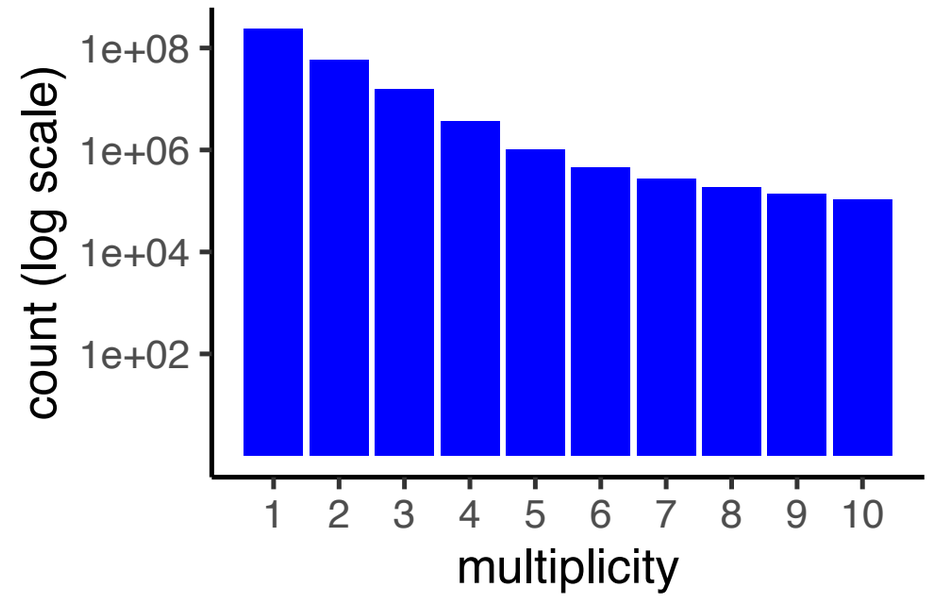
K-mer repeat spectra

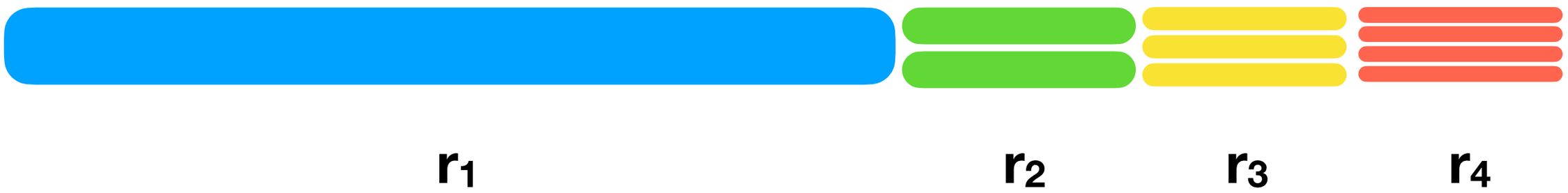
- The histogram of k-mer counts
- $$L = \sum_{j=1} j \cdot r_j$$



K-mer repeat spectra

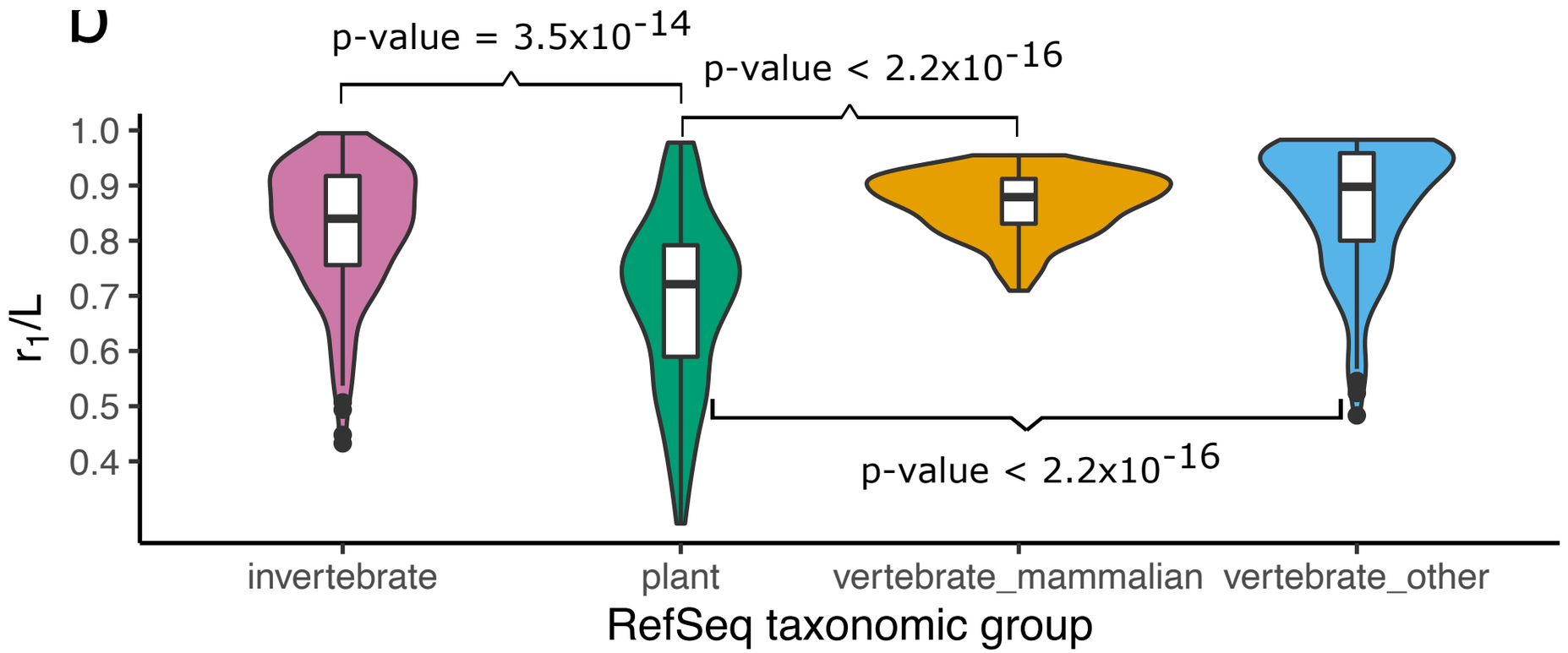
- The histogram of k-mer counts
- $$L = \sum_{j=1} j \cdot r_j$$



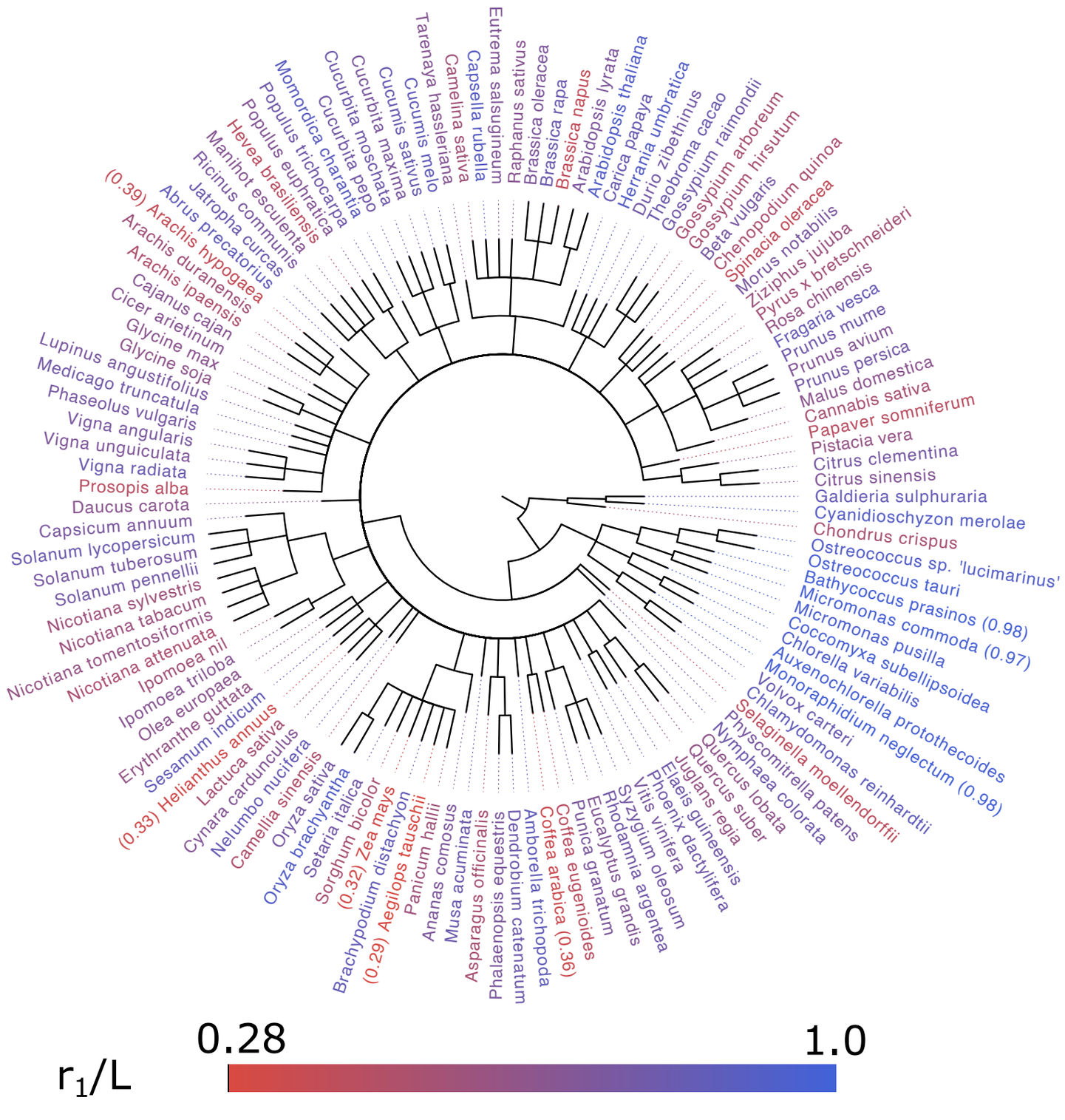


Repetitiveness across taxonomy

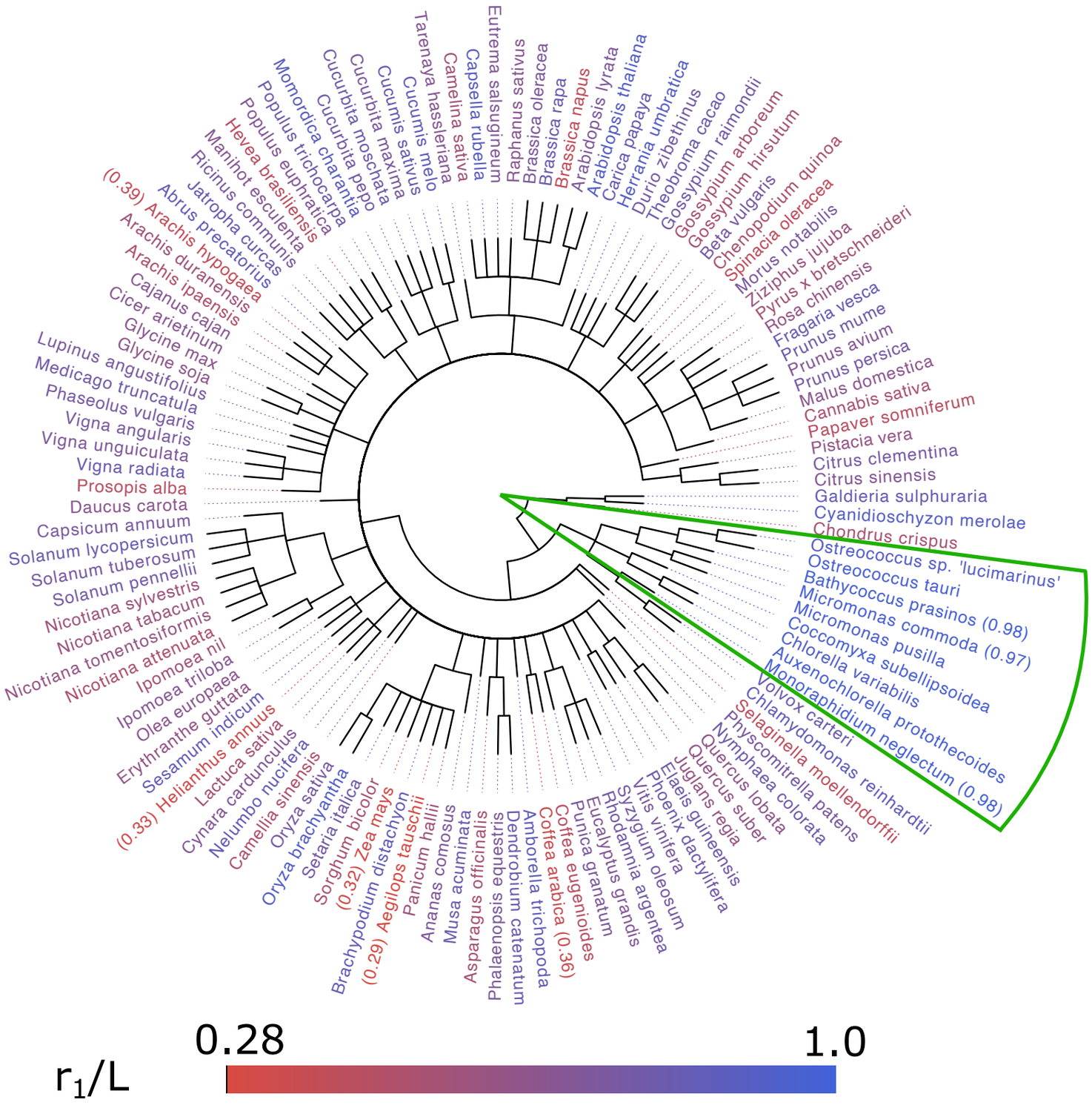
- 622 RefSeq genomes analyzed
- Ratio of unique k-mers, r_1/L, computed for each genome



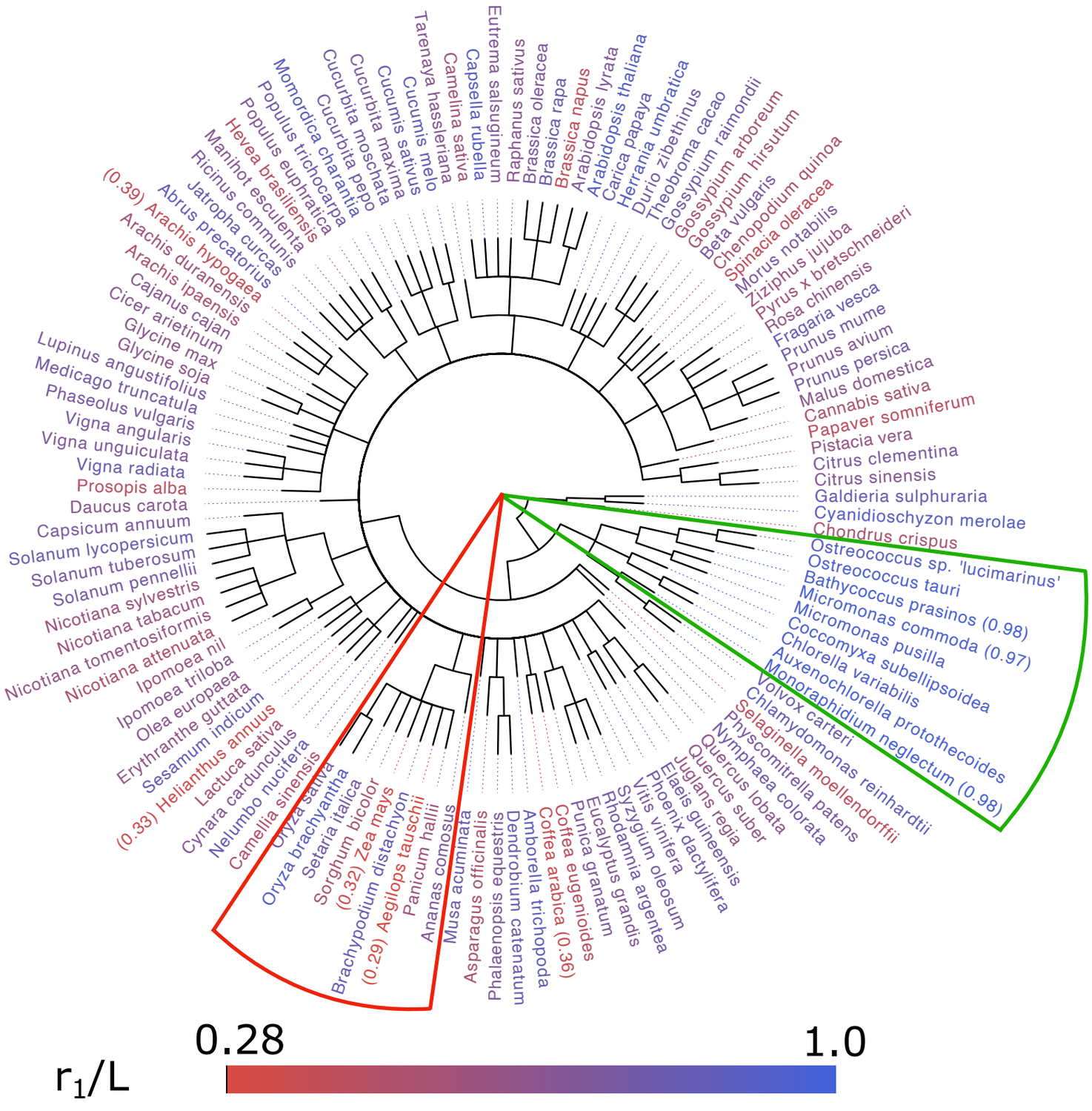
Repetitiveness across taxonomy



Repetitiveness across taxonomy



Repetitiveness across taxonomy



Probabilistic model for k-mer counts

K-mer repeat spectra

Expected k-mer counts

$$\mathbf{m} = \mathbf{Pr} + \mathbf{1}_{h=1} E$$

Error term: $E = L\lambda(1-(1-\epsilon)^k)$

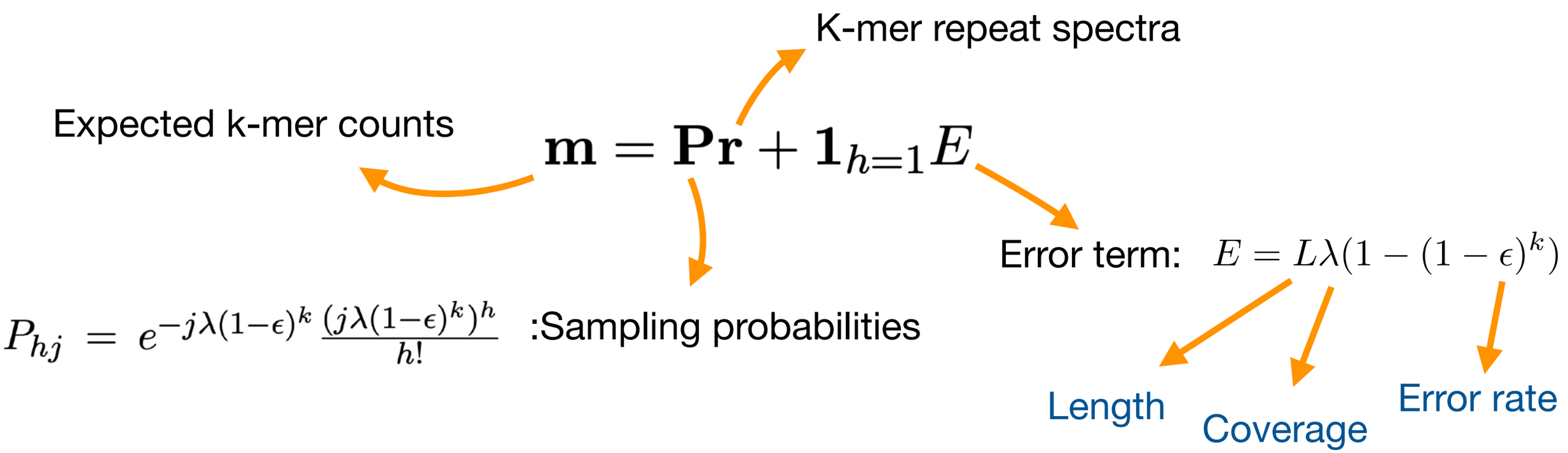
$P_{hj} = e^{-j\lambda(1-\epsilon)^k} \frac{(j\lambda(1-\epsilon)^k)^h}{h!}$:Sampling probabilities

Length

Coverage

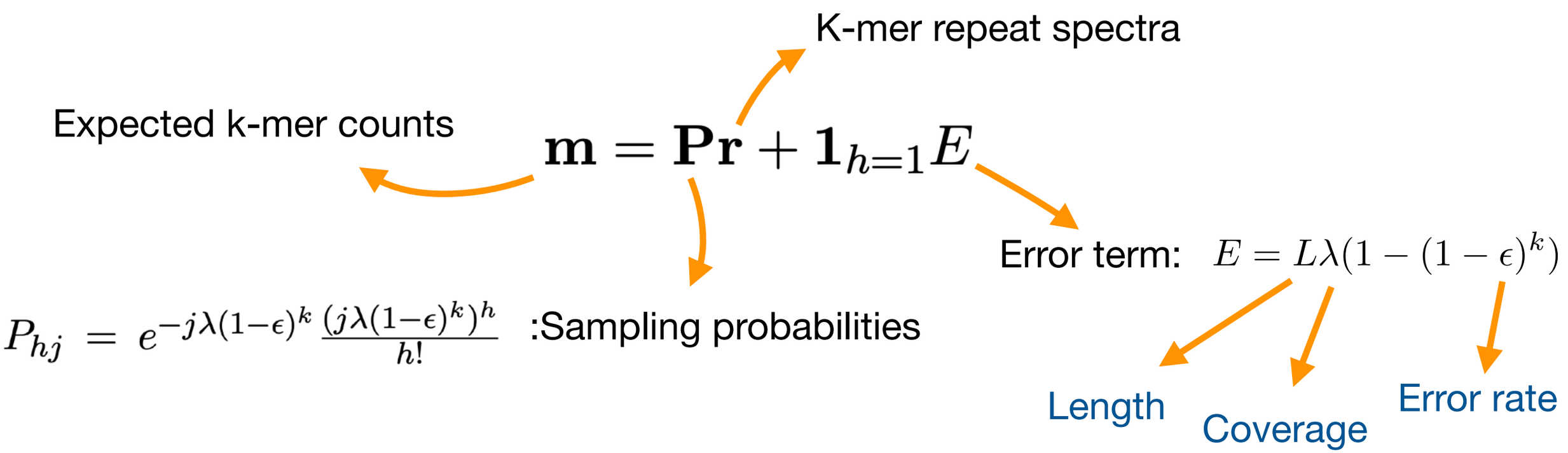
Error rate

Probabilistic model for k-mer counts



- Iterative procedure to estimate unknowns
  - Assuming **P**, solve for **r** using **o** = **Pr** + E
  - Compute λ and ϵ, knowing **r**.

Probabilistic model for k-mer counts



- Iterative procedure to estimate unknowns
  - Assuming **P**, solve for **r** using **o** = **Pr** + E
  - Compute λ and ϵ, knowing **r**.
- This is not enough; some more algorithmic tricks needed

RESPECT

https://github.com/shahab-sarmashghi/RESPECT

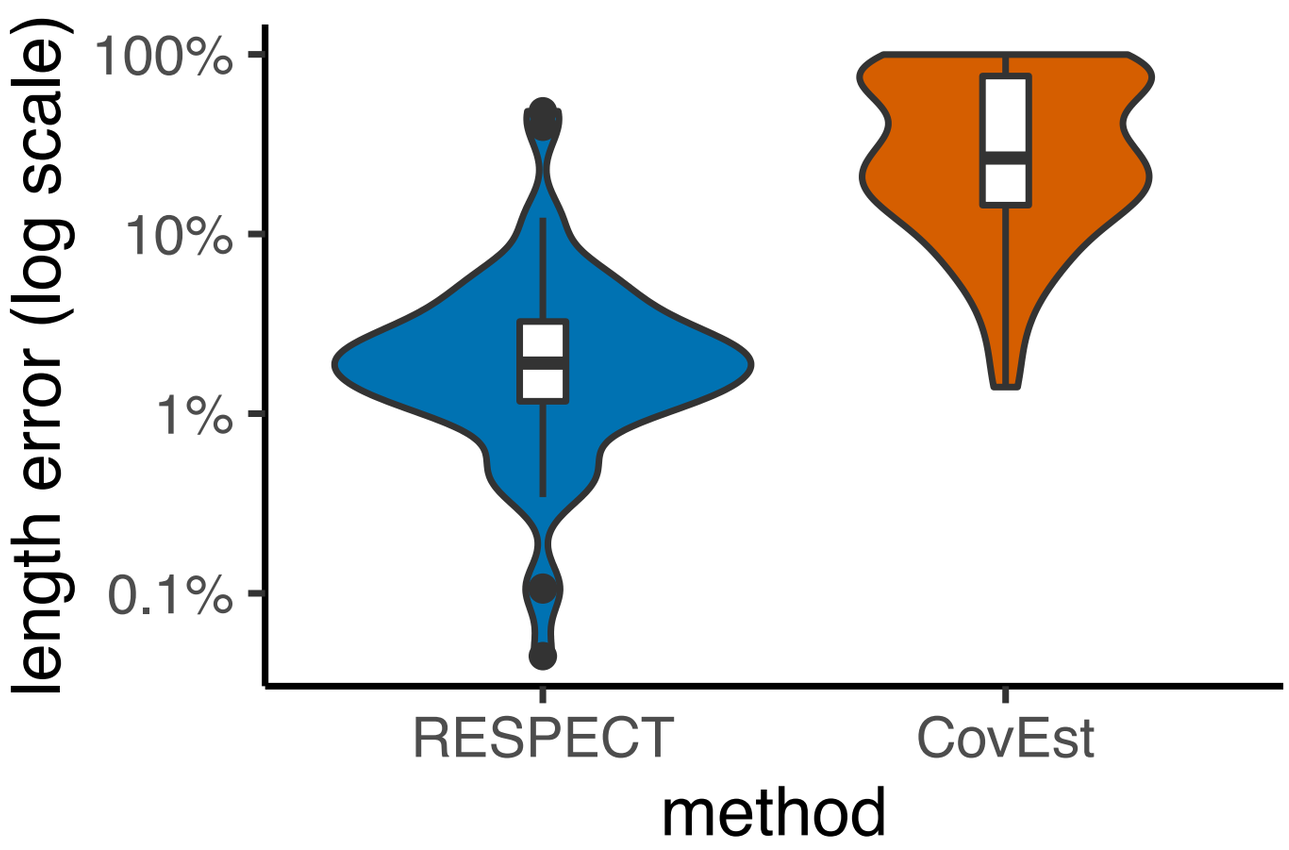
https://doi.org/10.1101/2021.01.28.428636

Length estimation on simulated genome-skims

- Each test genome is skimmed at 1X sequencing depth with 1% sequencing error

Length estimation on simulated genome-skims

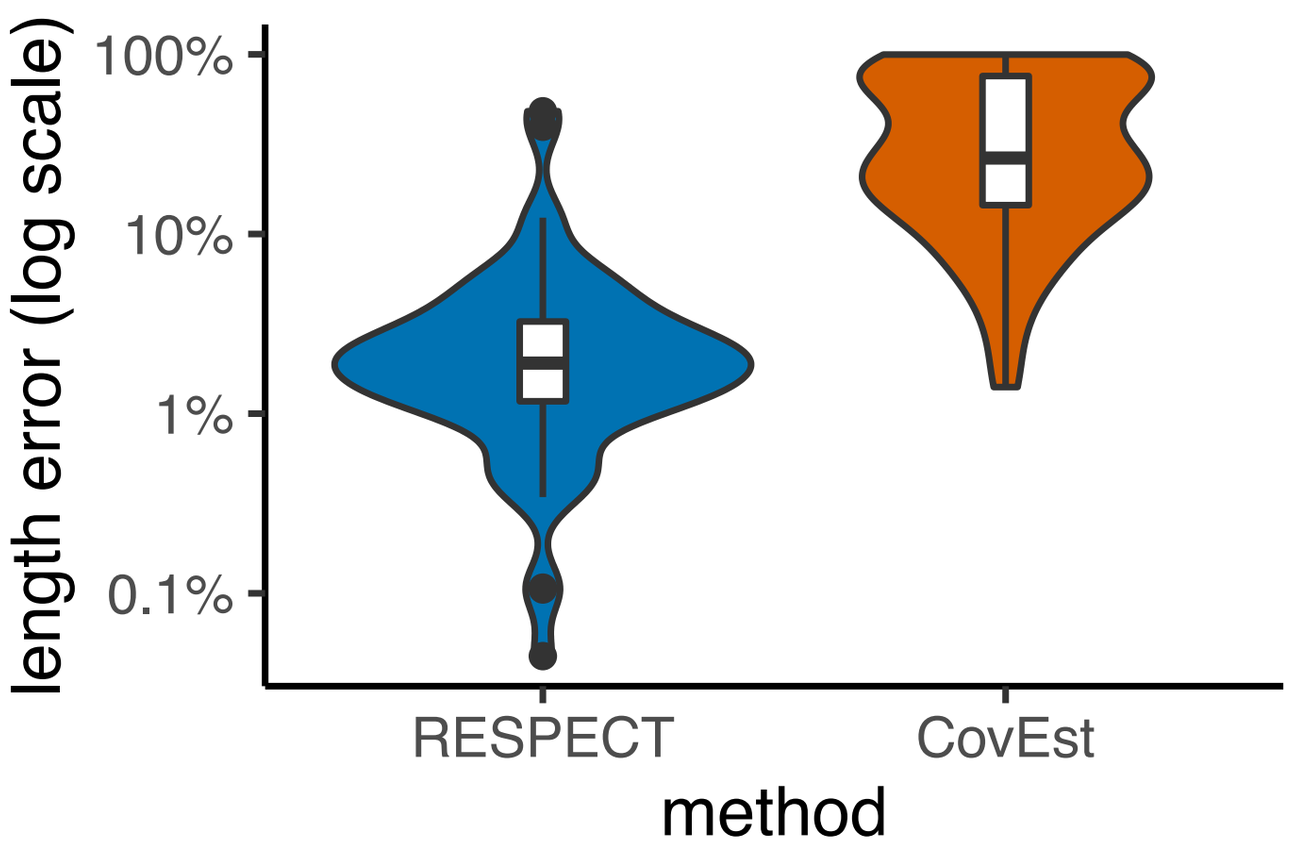
- Each test genome is skimmed at 1X sequencing depth with 1% sequencing error



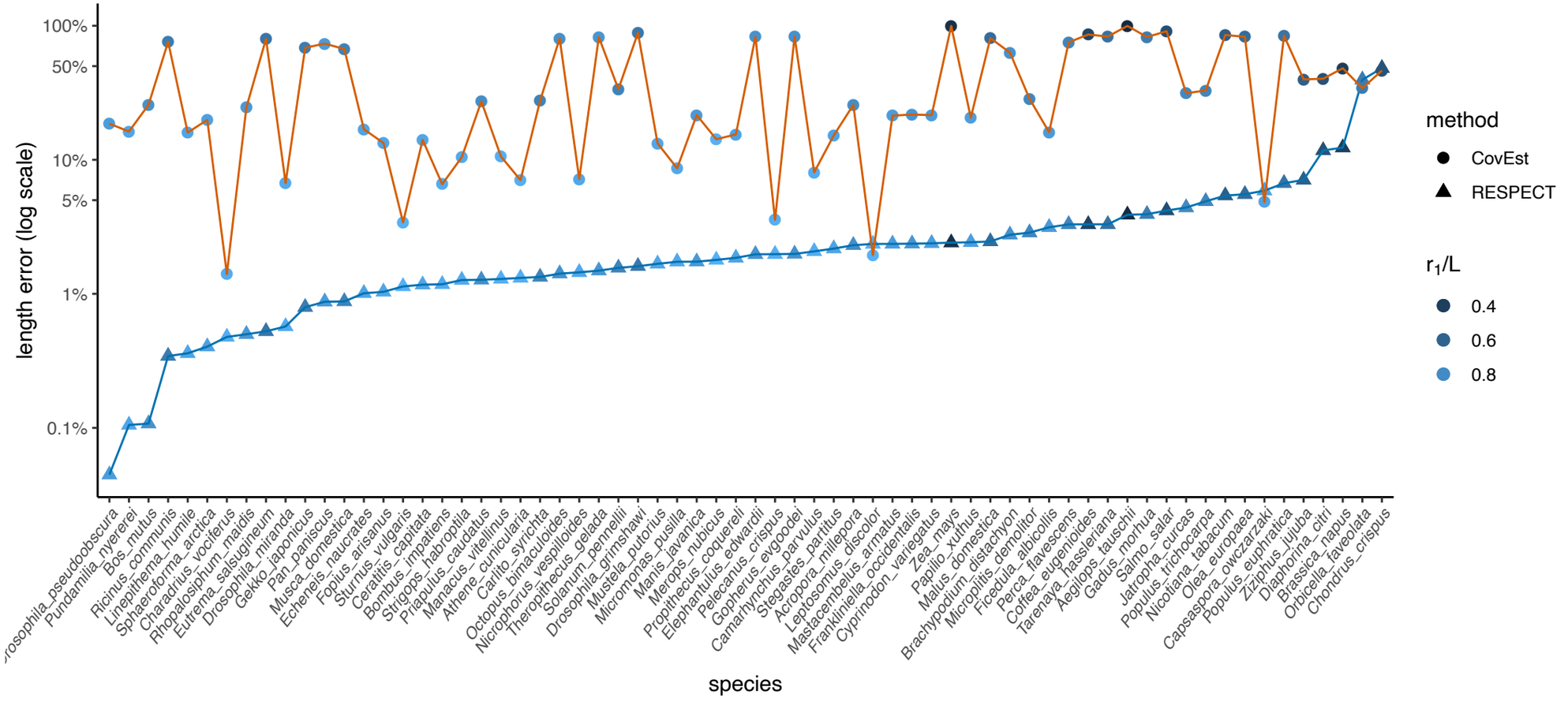
Length estimation on simulated genome-skims

- Each test genome is skimmed at 1X sequencing depth with 1% sequencing error

Average error: **4%.** vs. **40%**

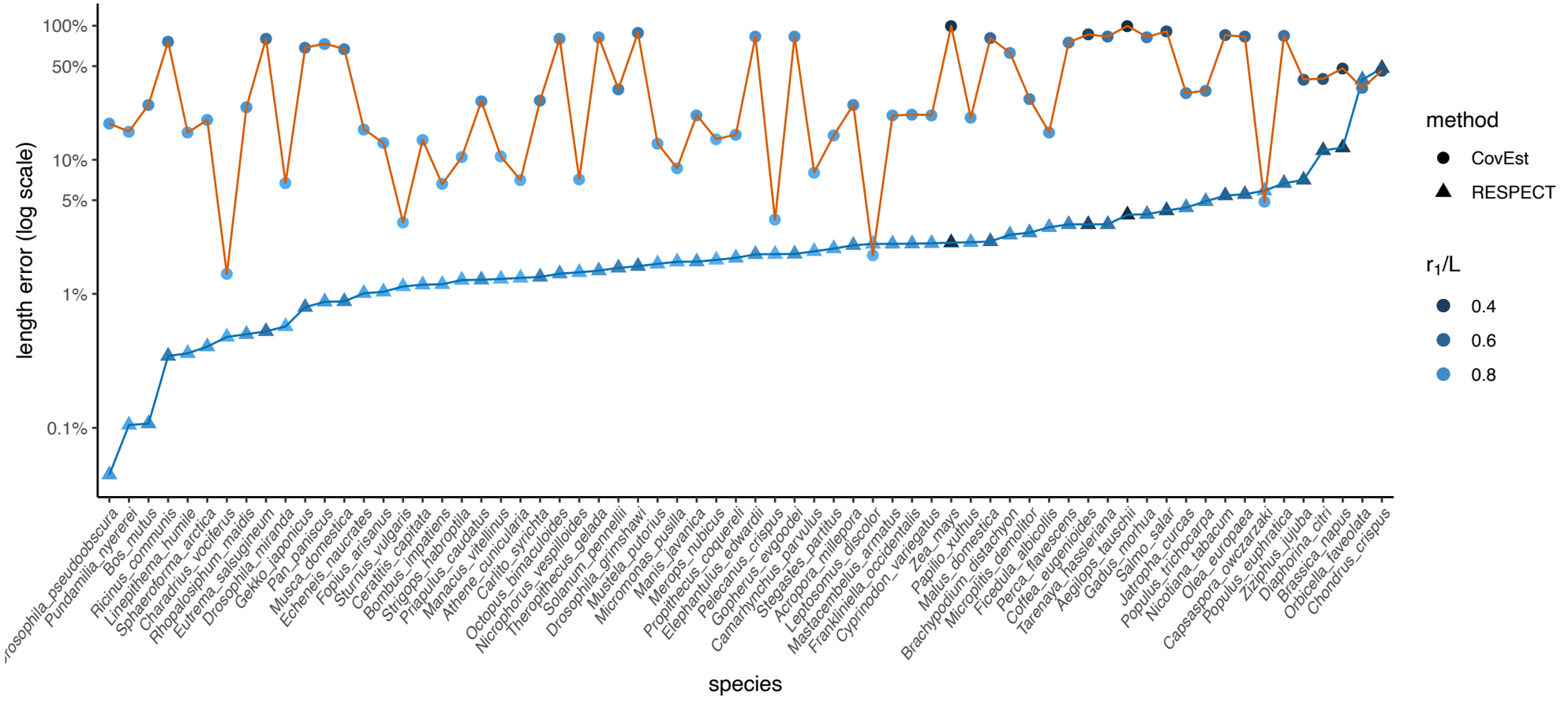


Simulated genome-skims



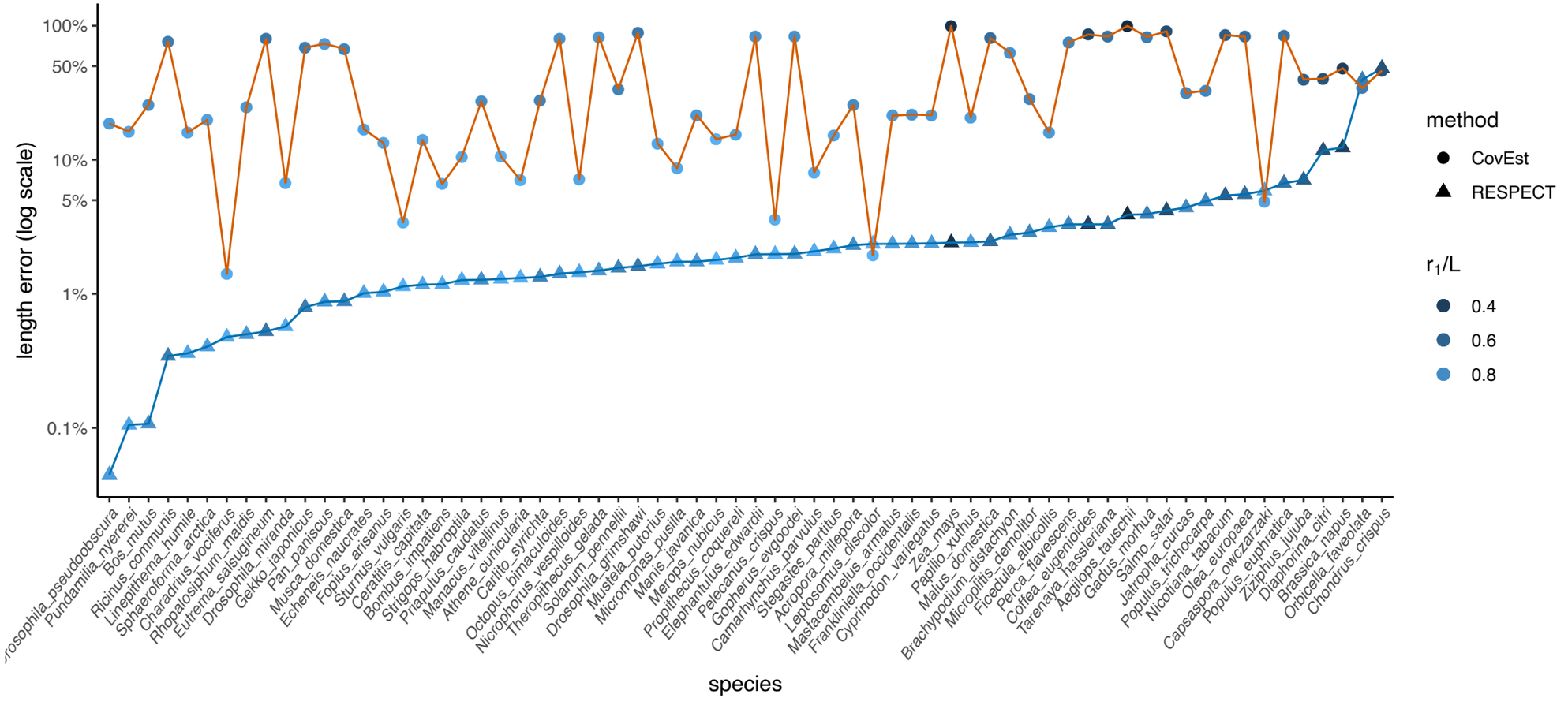
Simulated genome-skims

- RESPECT: less than 5% error in 90% of cases

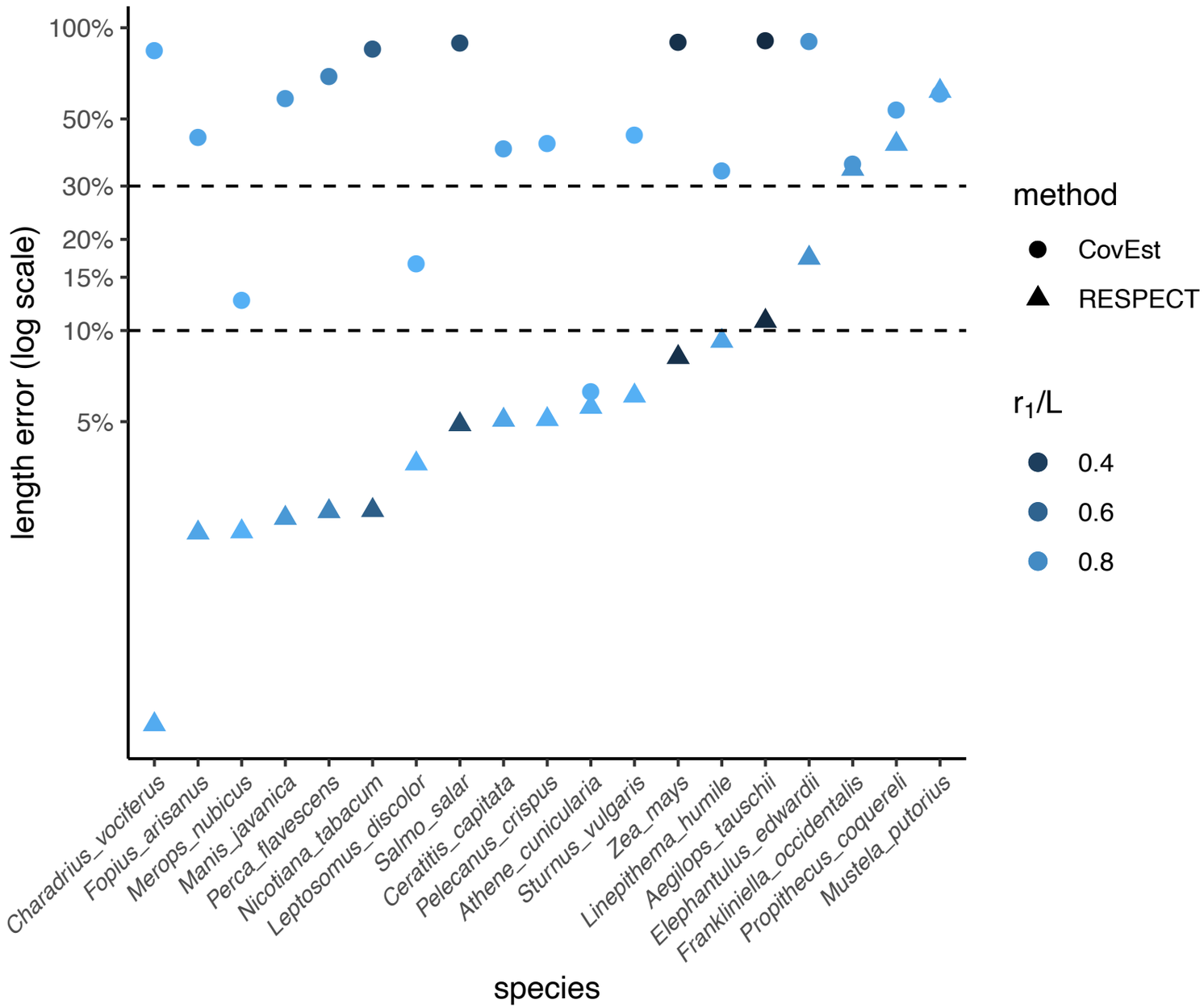


Simulated genome-skims

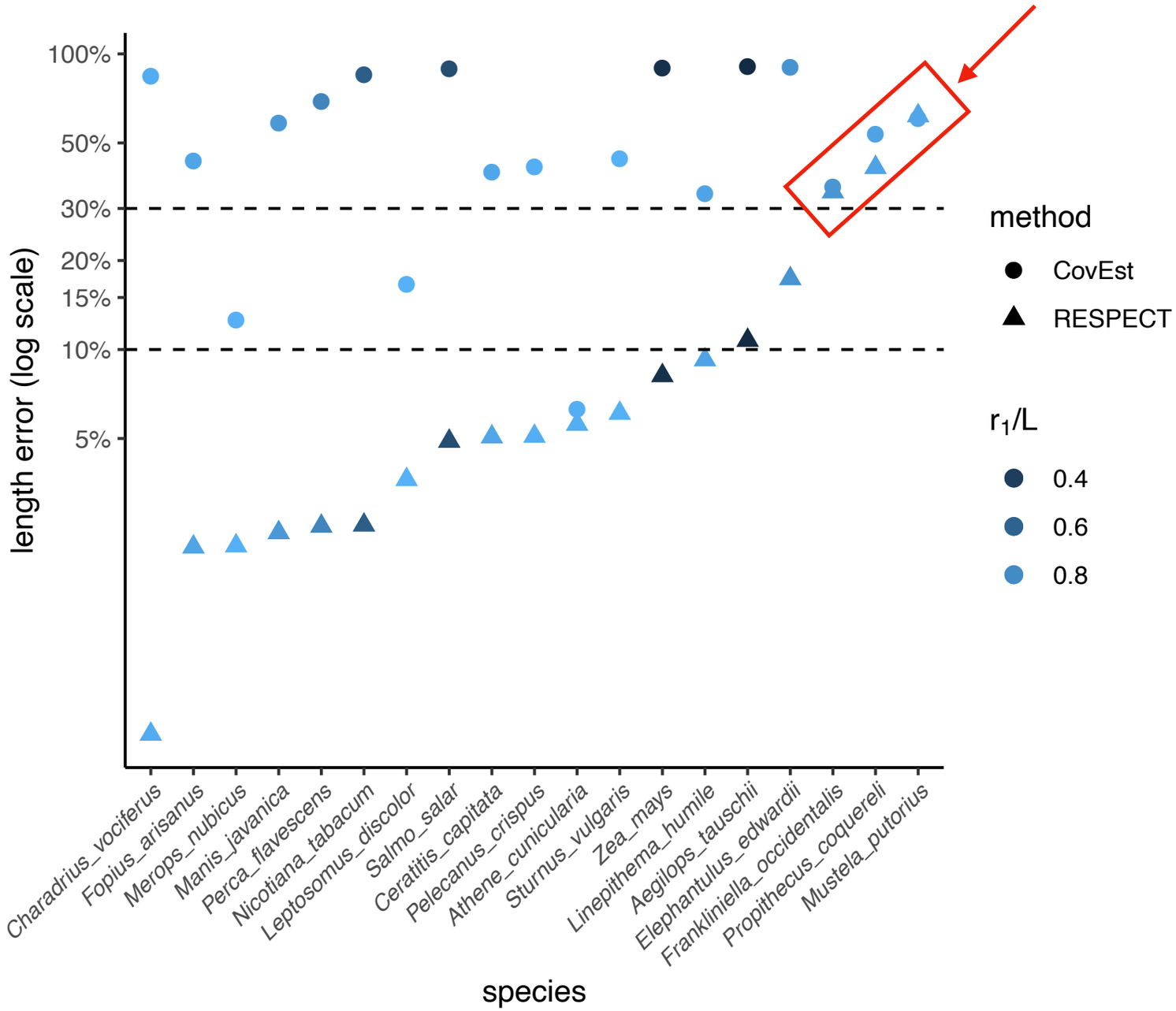
- RESPECT: less than 5% error in 90% of cases
- CovEst: more than 50% error in 30% of cases



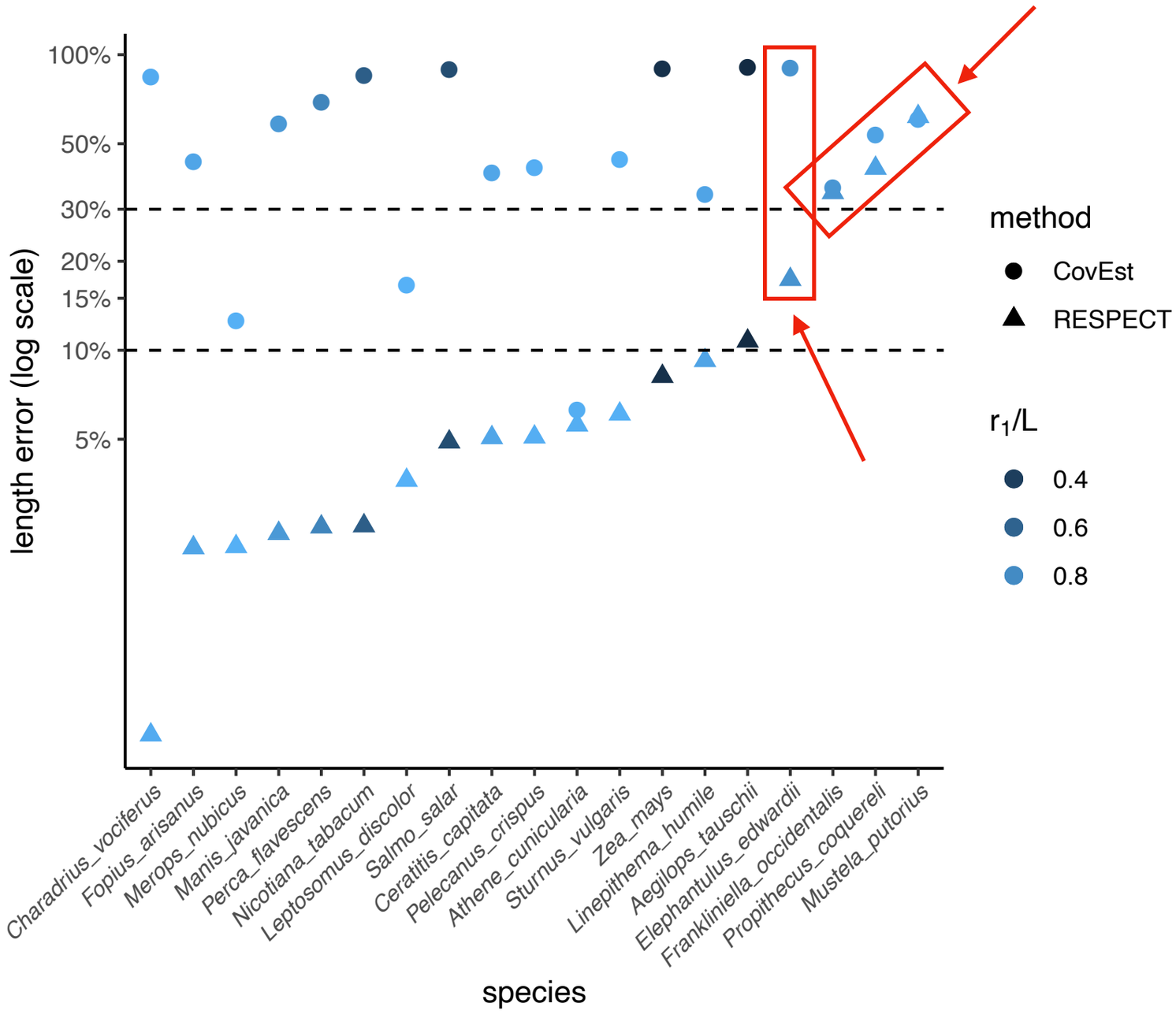
Real genome skimming data



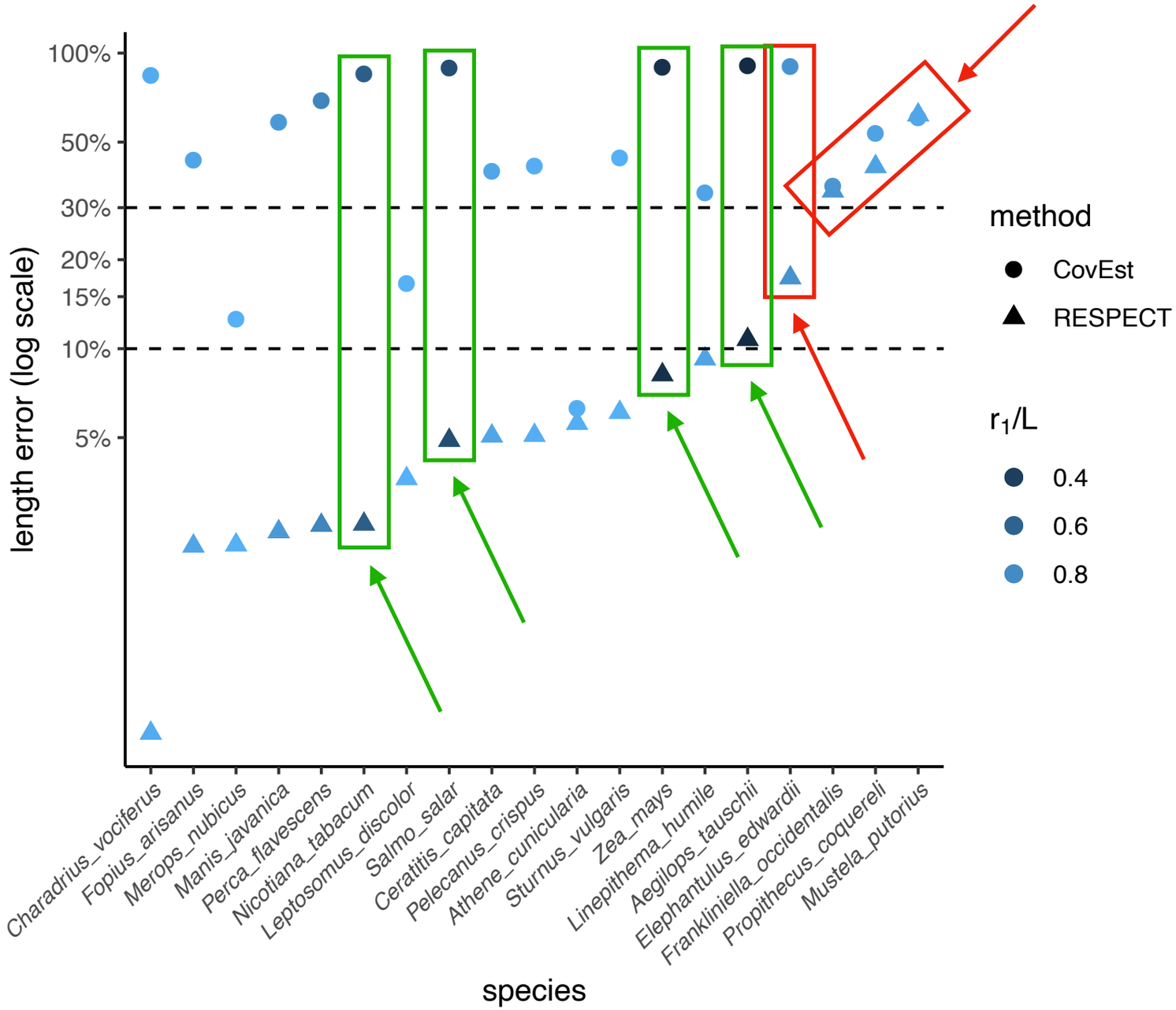
Real genome skimming data



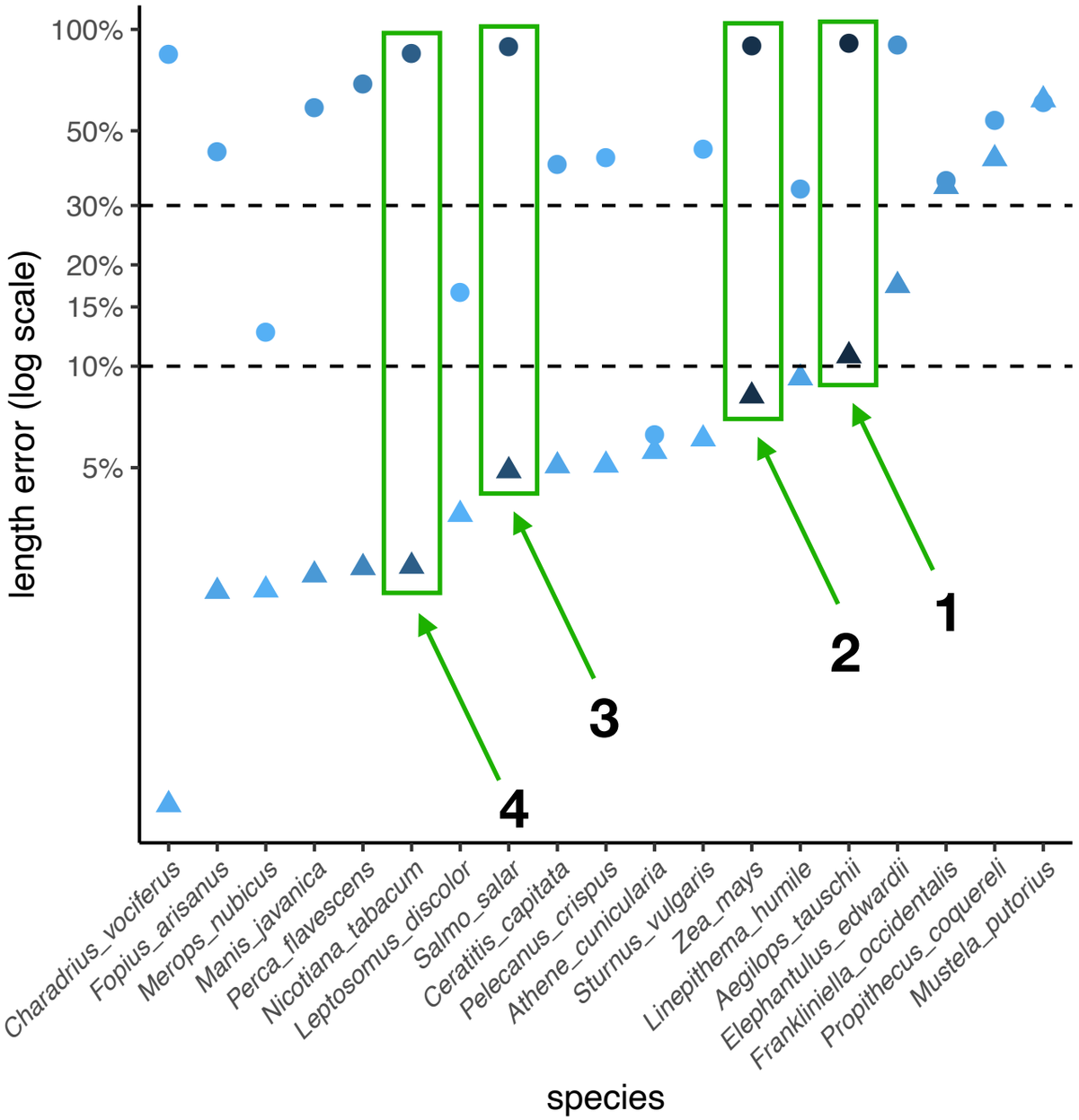
Real genome skimming data



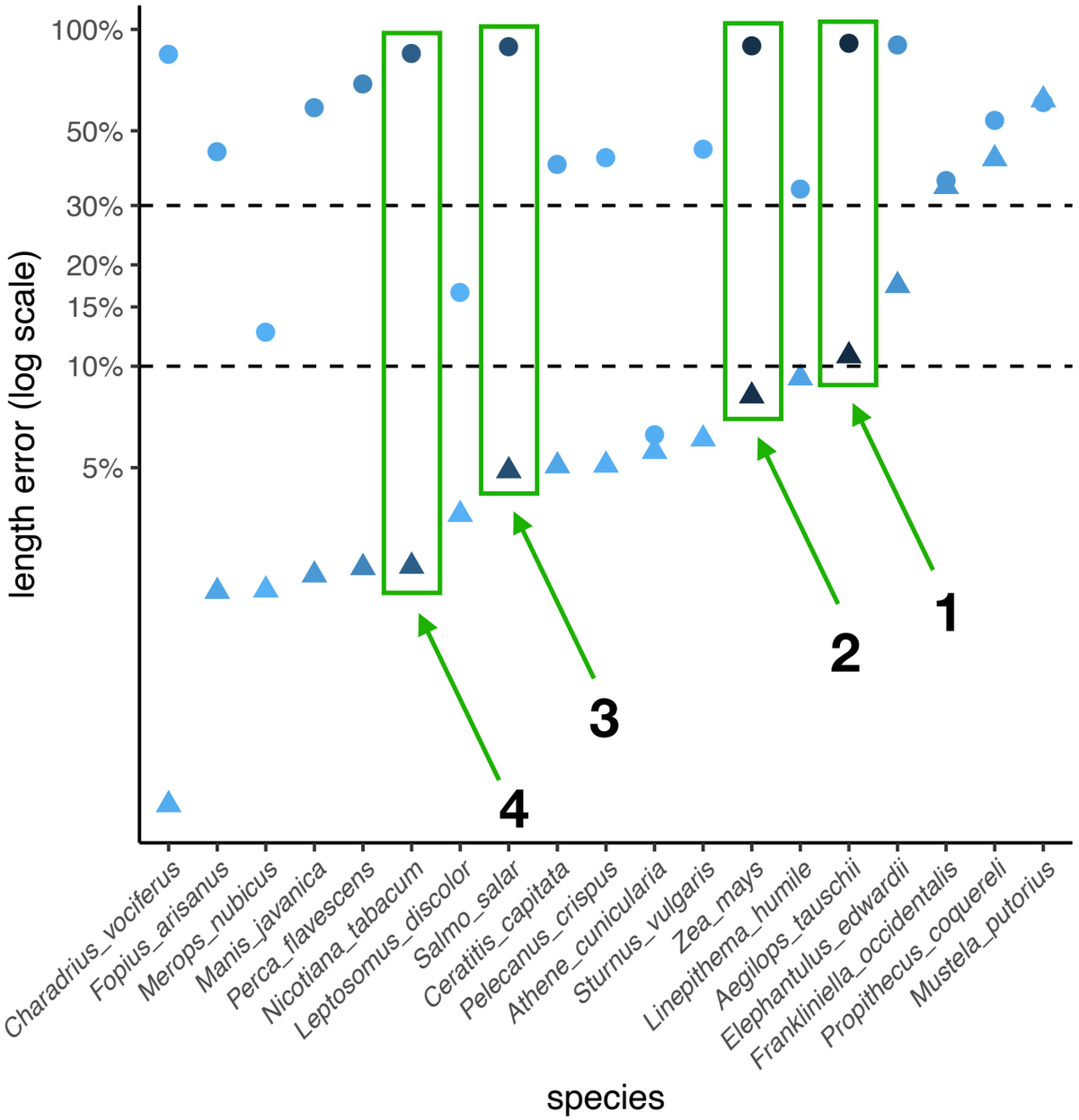
Real genome skimming data



SRA data

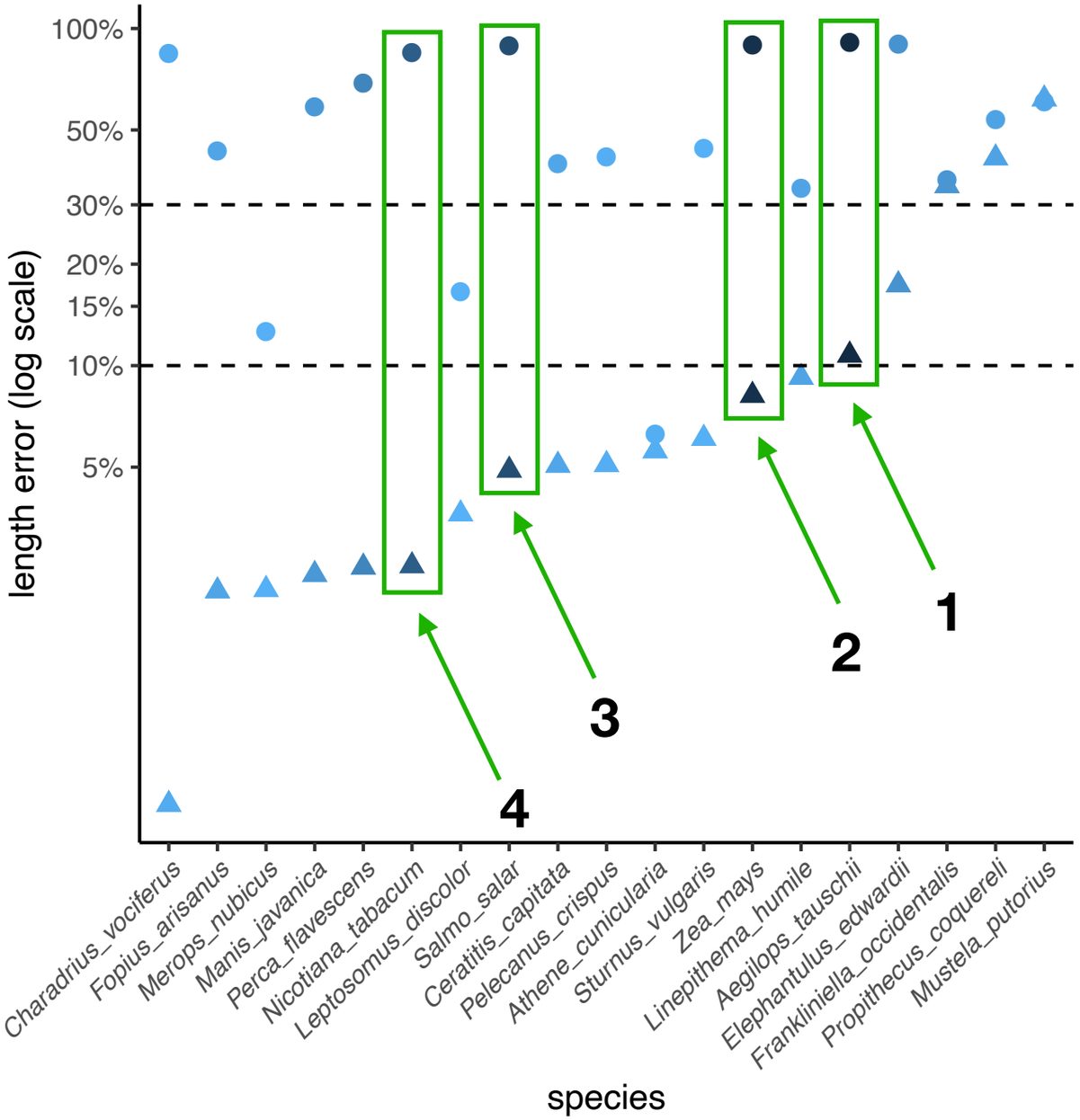

100%
50%
30%
20%
15%
10%
5%
length error (log scale)
1
2
3
4
Charadrius_vociferus
Fopius_arisanus
Merops_nubicus
Manis_javanica
Perca_flavescens
Nicotiana_tabacum
Leptosomus_discolor
Salmo_salar
Ceratitis_capitata
Pelecanus_crispus
Athene_cunicularia
Sturnus_vulgaris
Zea_mays
Linepithema_humile
Aegilops_tauschii
Elephantulus_edwardii
Frankliniella_occidentalis
Propithecus_coquereli
Mustela_putorius
species

SRA data



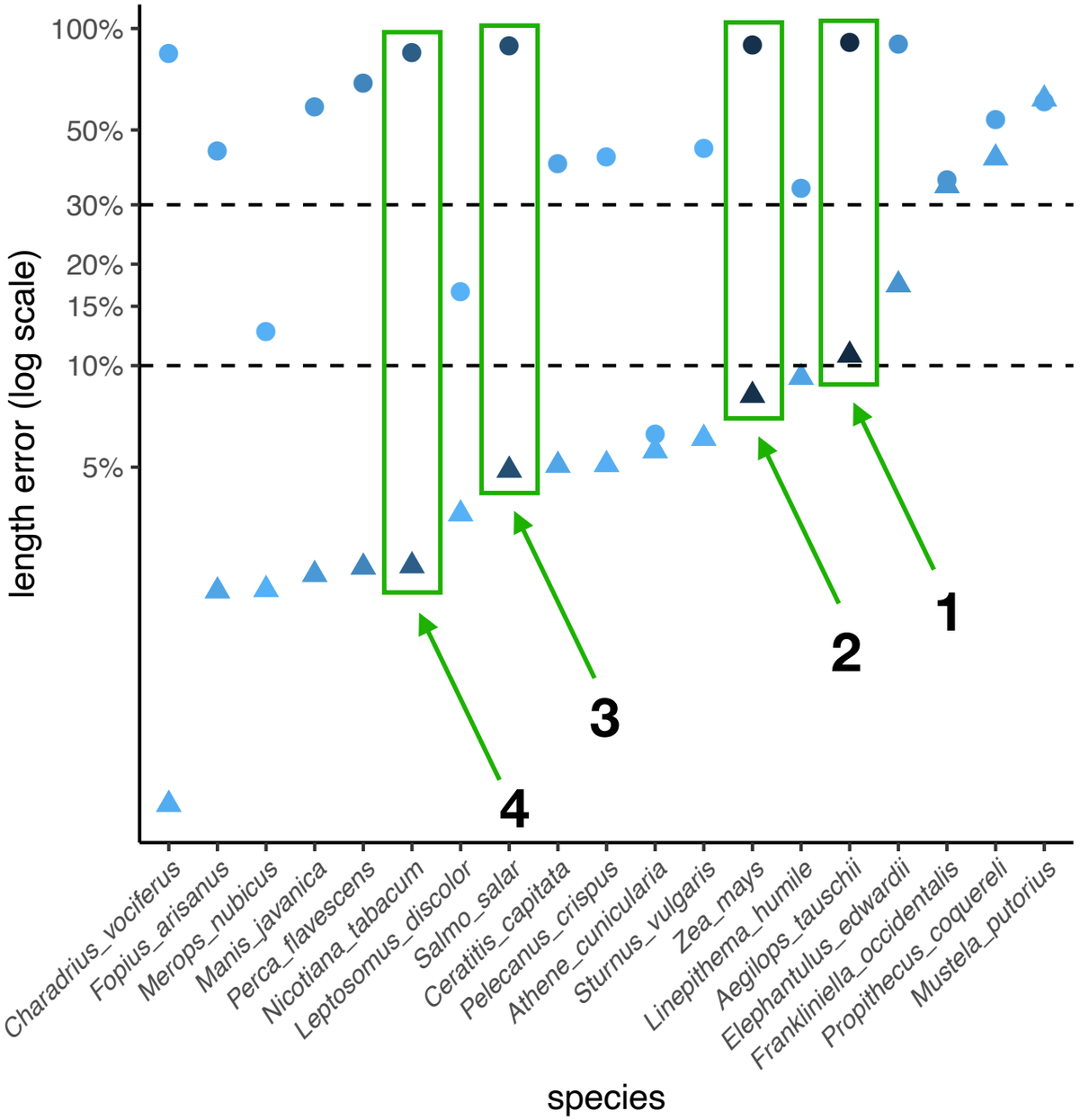
Species	r_1/L	Assembly length	RESPECT	CovEst
1	0.29	4.3Gb	3.9Gb	0.4Gb
2	0.32	2.1Gb	2Gb	0.2Gb
3	0.48	3Gb	2.8Gb	0.3Gb
4	0.57	3.6Gb	3.7Gb	0.5Gb

SRA data



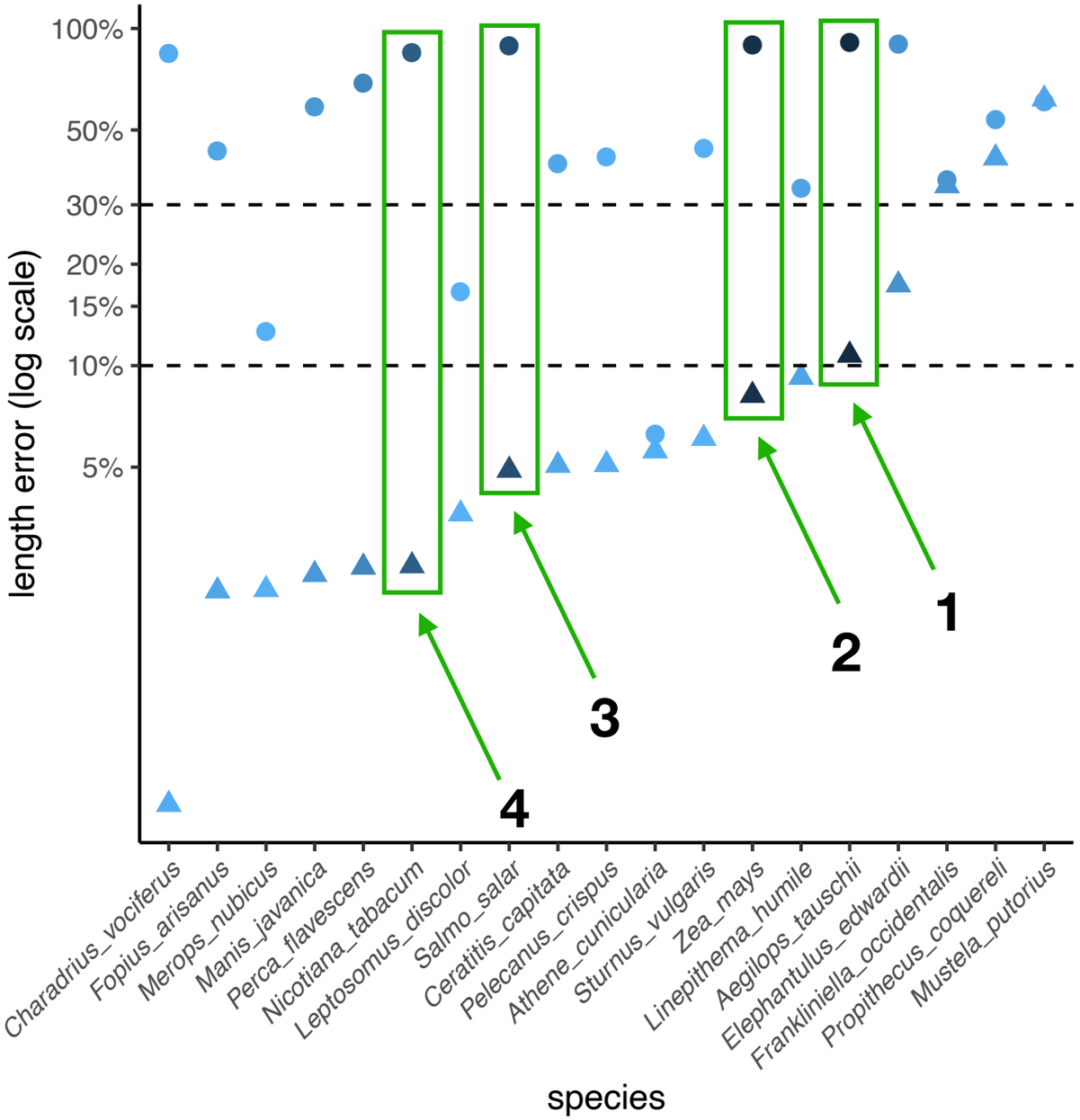
Species	r_1/L	Assembly length	RESPECT	CovEst
1	0.29	4.3Gb	3.9Gb	0.4Gb
2	0.32	2.1Gb	2Gb	0.2Gb
3	0.48	3Gb	2.8Gb	0.3Gb
4	0.57	3.6Gb	3.7Gb	0.5Gb

SRA data



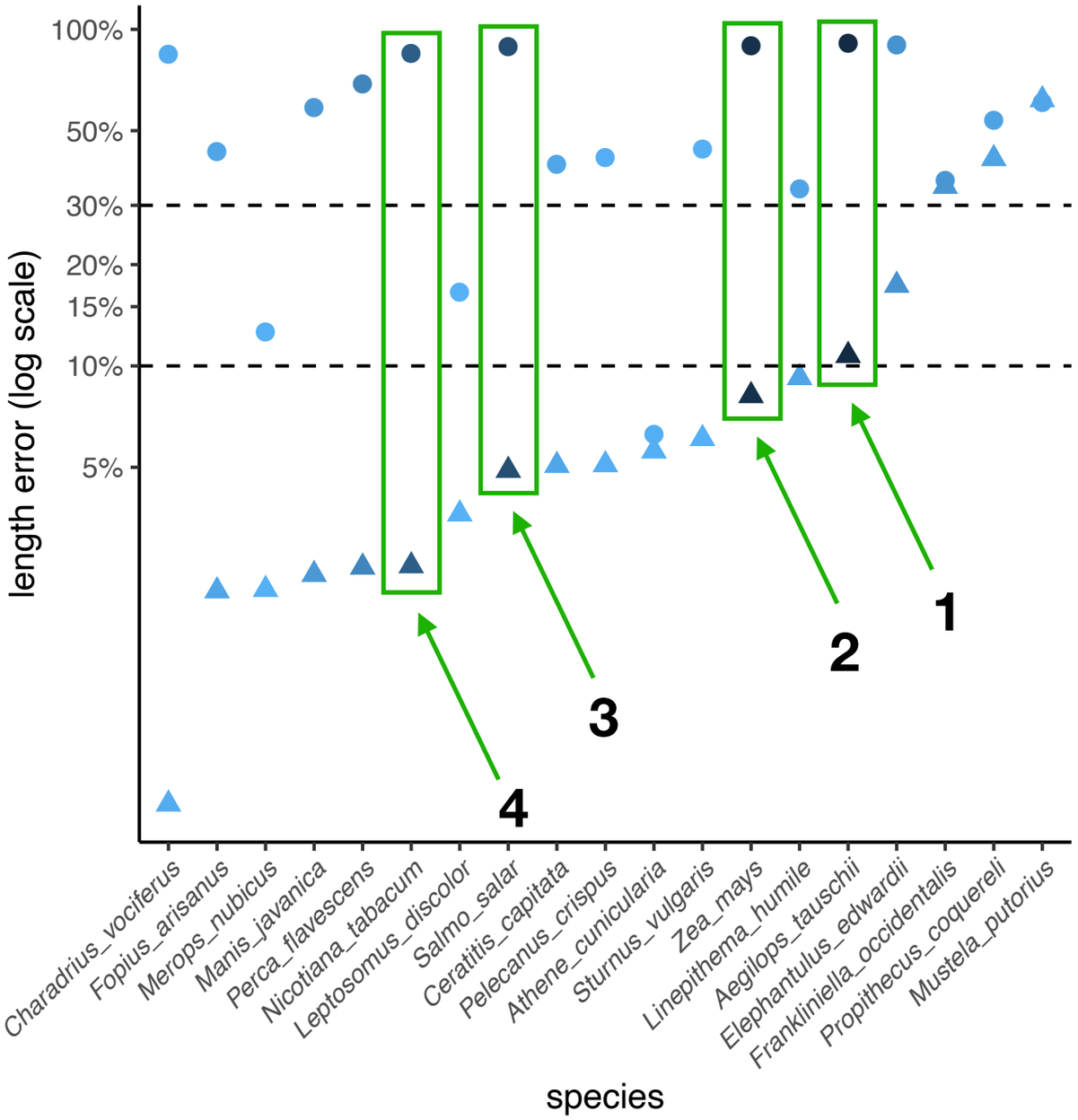
Species	r_1/L	Assembly length	RESPECT	CovEst
1	0.29	4.3Gb	3.9Gb	0.4Gb
2	0.32	2.1Gb	2Gb	0.2Gb
3	0.48	3Gb	2.8Gb	0.3Gb
4	0.57	3.6Gb	3.7Gb	0.5Gb

SRA data



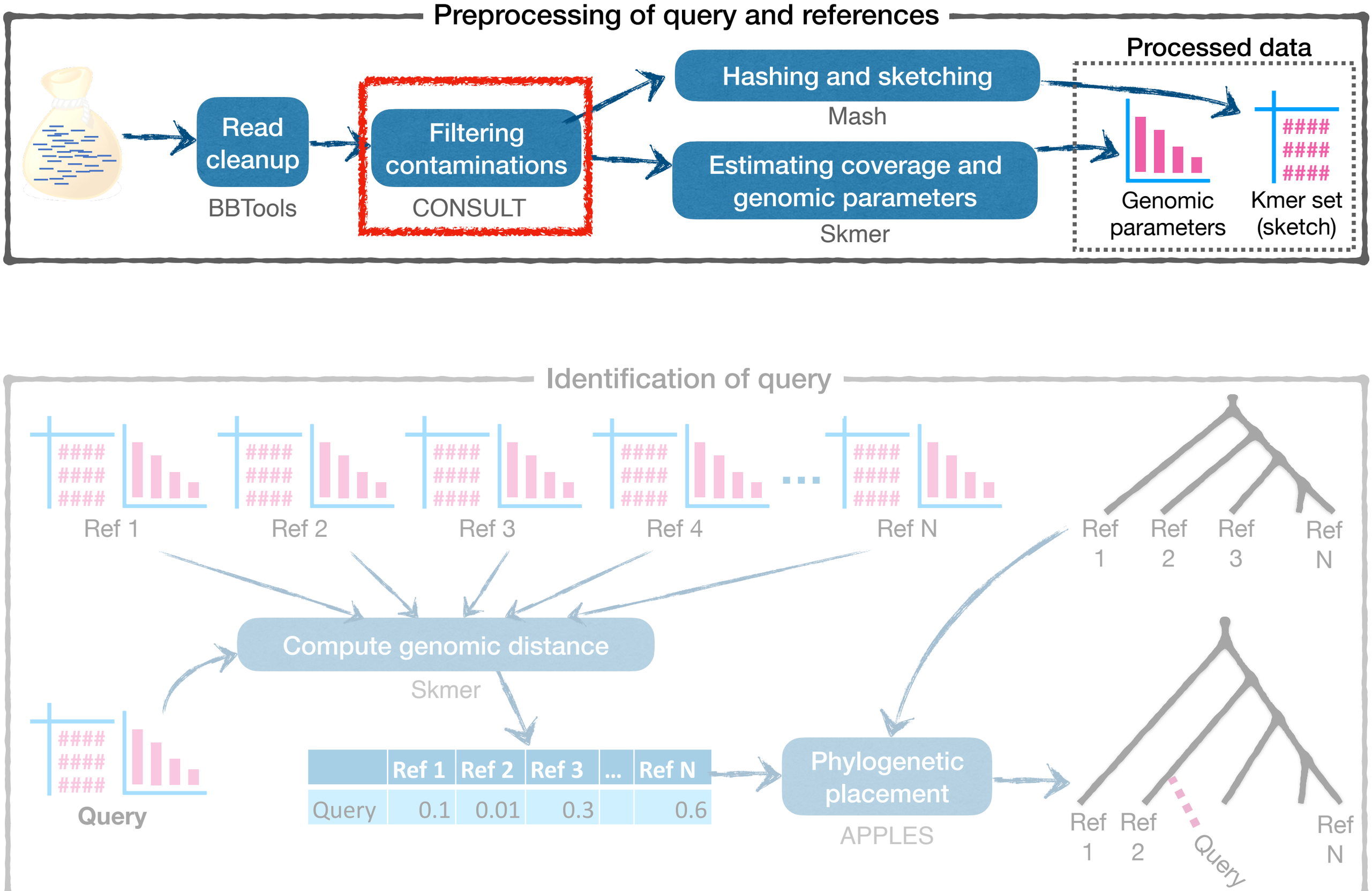
Species	r_1/L	Assembly length	RESPECT	CovEst
1	0.29	4.3Gb	3.9Gb	0.4Gb
2	0.32	2.1Gb	2Gb	0.2Gb
3	0.48	3Gb	2.8Gb	0.3Gb
4	0.57	3.6Gb	3.7Gb	0.5Gb

SRA data



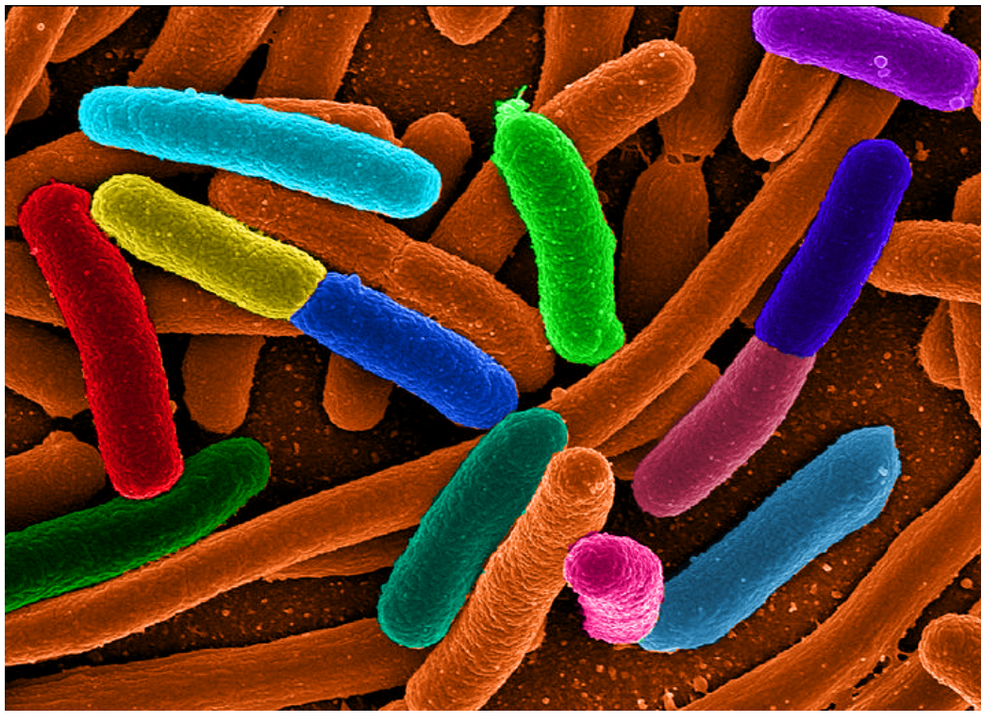
Species	r_1/L	Assembly length	RESPECT	CovEst
1	0.29	4.3Gb	3.9Gb	0.4Gb
2	0.32	2.1Gb	2Gb	0.2Gb
3	0.48	3Gb	2.8Gb	0.3Gb
4	0.57	3.6Gb	3.7Gb	0.5Gb

Tutorial


Preprocessing of query and references
Read cleanup
BBTools
Filtering contaminations
CONSULT
Hashing and sketching
Mash
Estimating coverage and genomic parameters
Skmer
Processed data
Genomic parameters
Kmer set (sketch)
Identification of query
Ref 1
Ref 2
Ref 3
Ref 4
Ref N
Compute genomic distance
Skmer
Query
Ref 1 Ref 2 Ref 3 ... Ref N
Query 0.1 0.01 0.3 0.6
Phylogenetic placement
APPLES
Ref 1
Ref 2
Ref 3
Ref N
Ref 1
Ref 2
Query
Ref N

Contamination removal

- We rarely get single-species genome-skims



Removing contaminates

- Human contamination: easy to filter (blast, bowtie, etc.)
    - unless you are working with primates

Removing contaminates

- Human contamination: easy to filter (blast, bowtie, etc.)
  - unless you are working with primates
- If we have a close reference genome ($d < 5\,\%$) to our genome skim:
  - Easy to handle: map reads to the reference and only keep reads that map

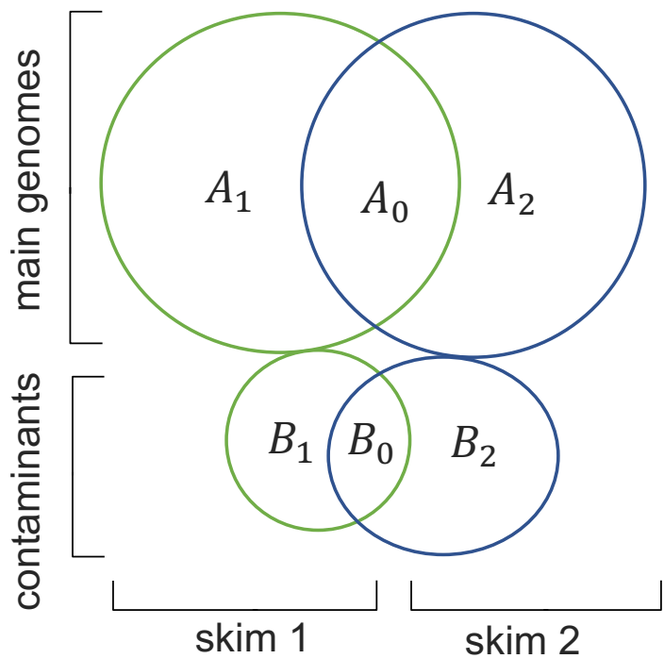
Removing contaminates

- Human contamination: easy to filter (blast, bowtie, etc.)
  - unless you are working with primates
- If we have a close reference genome ($d < 5\,\%$) to our genome skim:
  - Easy to handle: map reads to the reference and only keep reads that map
- Otherwise, what do you do? Remove contaminants?

Challenges in removing contaminant reads

- May not have a closely related reference for the contaminating species
  - Contaminant may not have complete assemblies
- Alignment to the available assemblies of contaminants would be slow
  - Using BLAST for example

Impact of contaminants on Skmer accuracy?

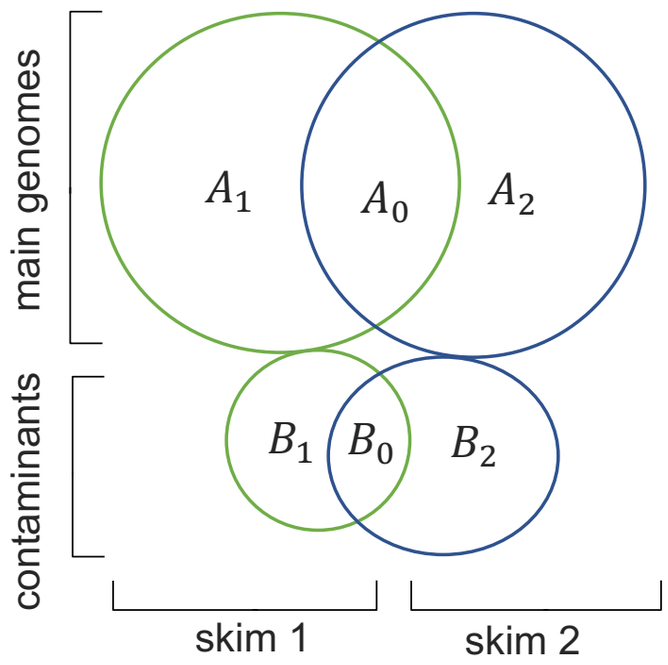


$$observed\ J = \frac{A_0 + B_0}{A_0 + A_1 + A_2 + B_0 + B_1 + B_2}$$

$$true\ J = \frac{A_0}{A_0 + A_1 + A_2}$$

$$H = \frac{B_0}{B_0 + B_1 + B_2}$$

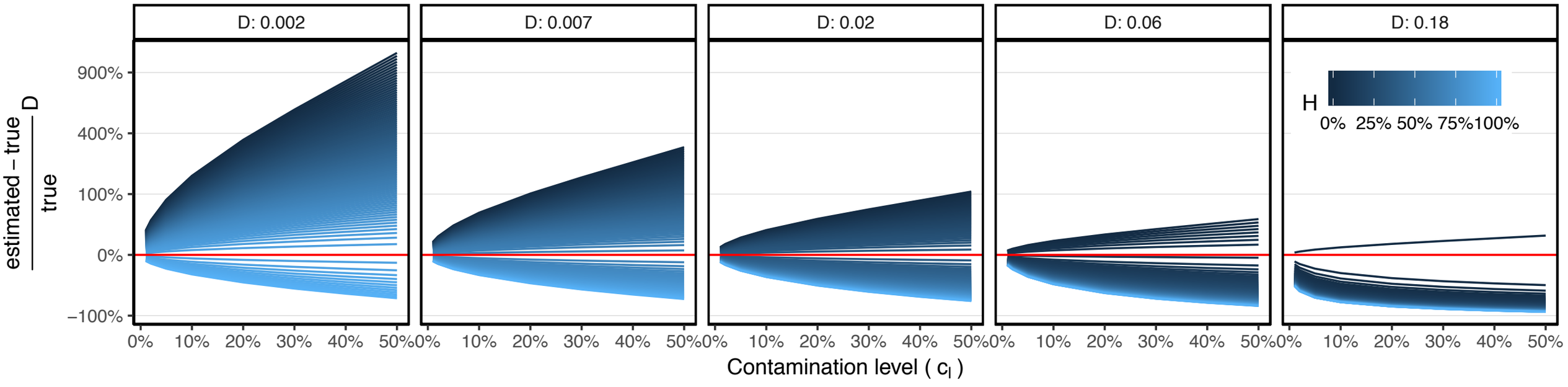
Impact of contaminants on Skmer accuracy?



$$observed\ J = \frac{A_0 + B_0}{A_0 + A_1 + A_2 + B_0 + B_1 + B_2}$$

$$true\ J = \frac{A_0}{A_0 + A_1 + A_2}$$

$$H = \frac{B_0}{B_0 + B_1 + B_2}$$



How about *k*-mer search methods

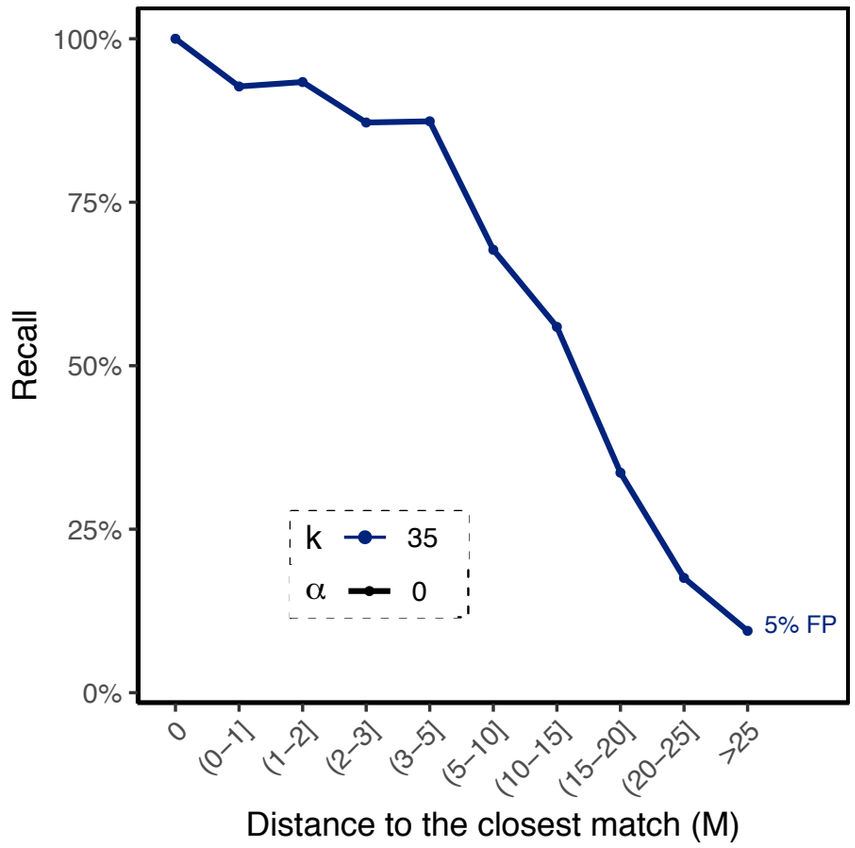
- Kraken [1,2] can search a reference database of many genomes
  - 100 seconds with 1 core or 10 seconds with 24 cores per 1Gb
  - Finds *k*-mer matches between each read and the reference set and calls a match if > α *k*-mer match

1. Wood and Salzberg, Genome Biology (2014).
2. Wood, Lu, Langmead (2019).

How about *k*-mer search methods

- Kraken [1,2] can search a reference database of many genomes
  - 100 seconds with 1 core or 10 seconds with 24 cores per 1Gb
  - Finds *k*-mer matches between each read and the reference set and calls a match if > α *k*-mer match
- Does it work?
  - Tested on a custom library controlling distance to the closest match

    [Rachtman *et al.*, 2019, MRE]

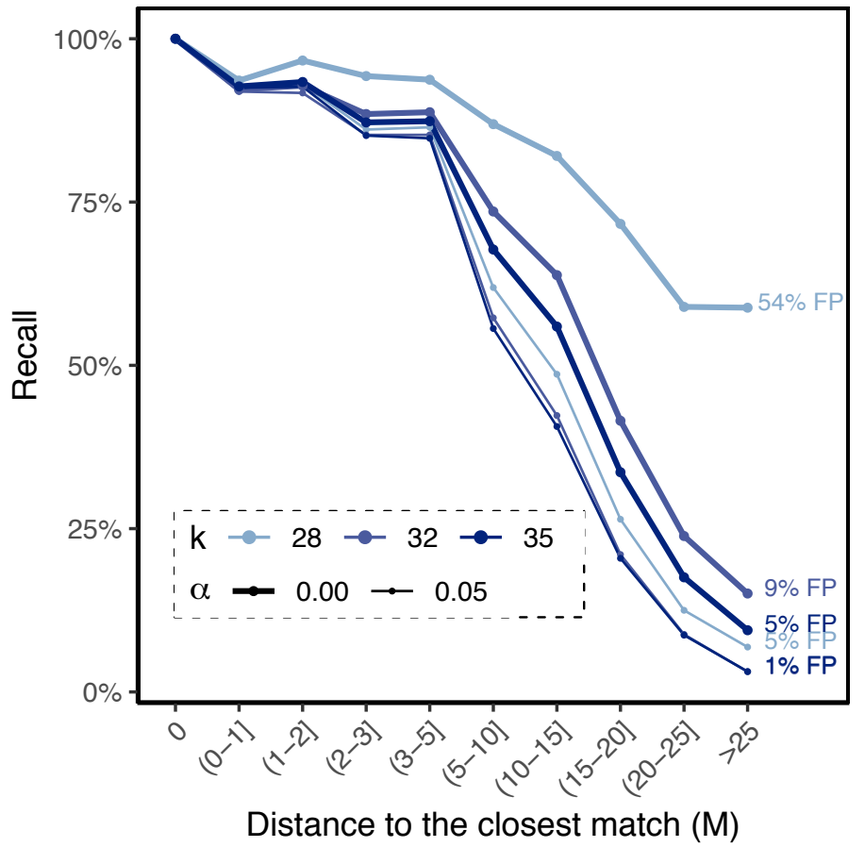


1. Wood and Salzberg, Genome Biology (2014).
2. Wood, Lu, Langmead (2019).

How about *k*-mer search methods

- Kraken [1,2] can search a reference database of many genomes
  - 100 seconds with 1 core or 10 seconds with 24 cores per 1Gb
  - Finds *k*-mer matches between each read and the reference set and calls a match if > α *k*-mer match
- Does it work?
  - Tested on a custom library controlling distance to the closest match

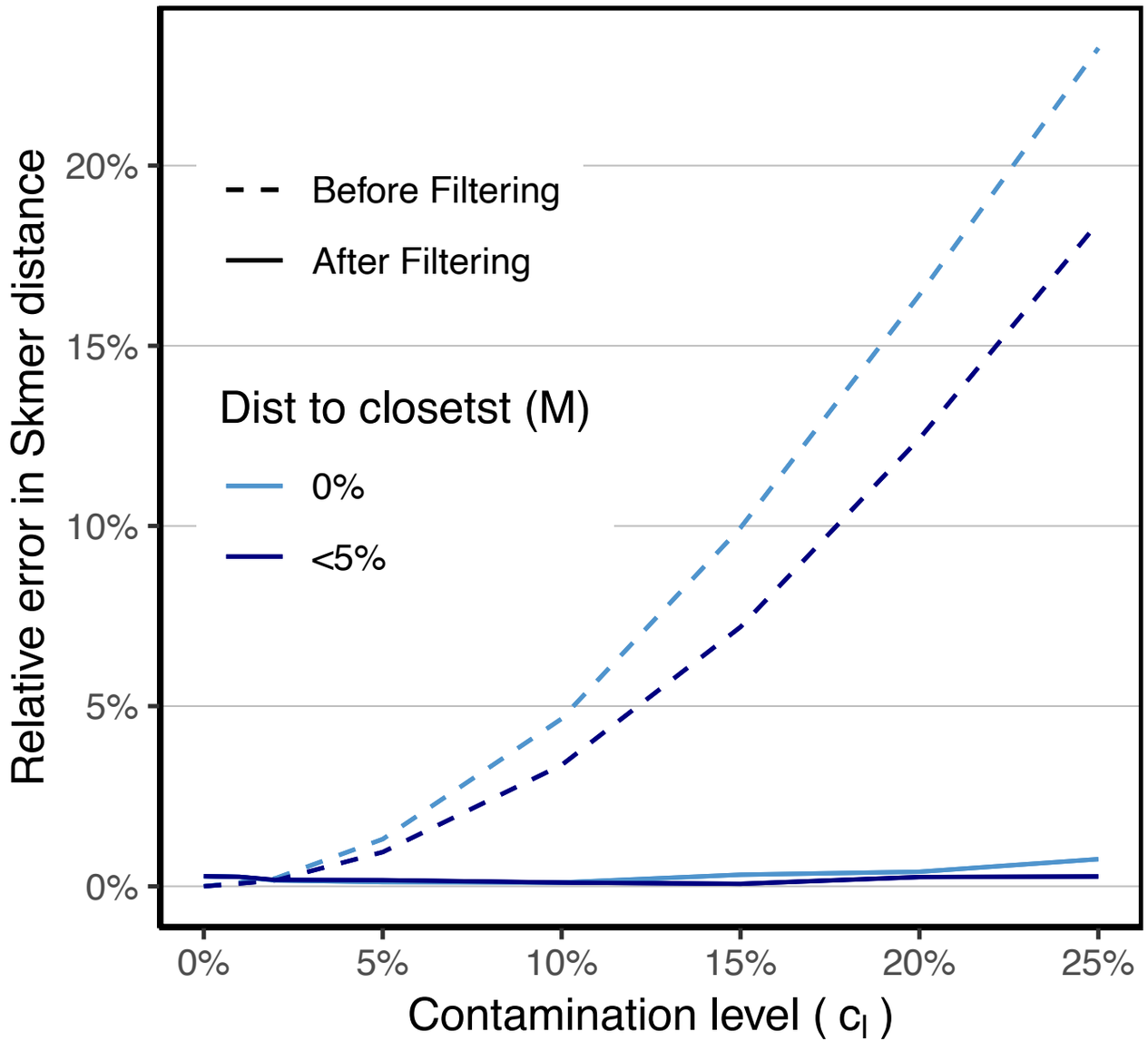
    [Rachtman *et al.*, 2019, MRE]



1. Wood and Salzberg, Genome Biology (2014).
2. Wood, Lu, Langmead (2019).

Impact on Skmer accuracy

- Simulated presence of microbial contamination for Drosophila genome skims (100 Mb)
- Measured Skmer error before and after filtering

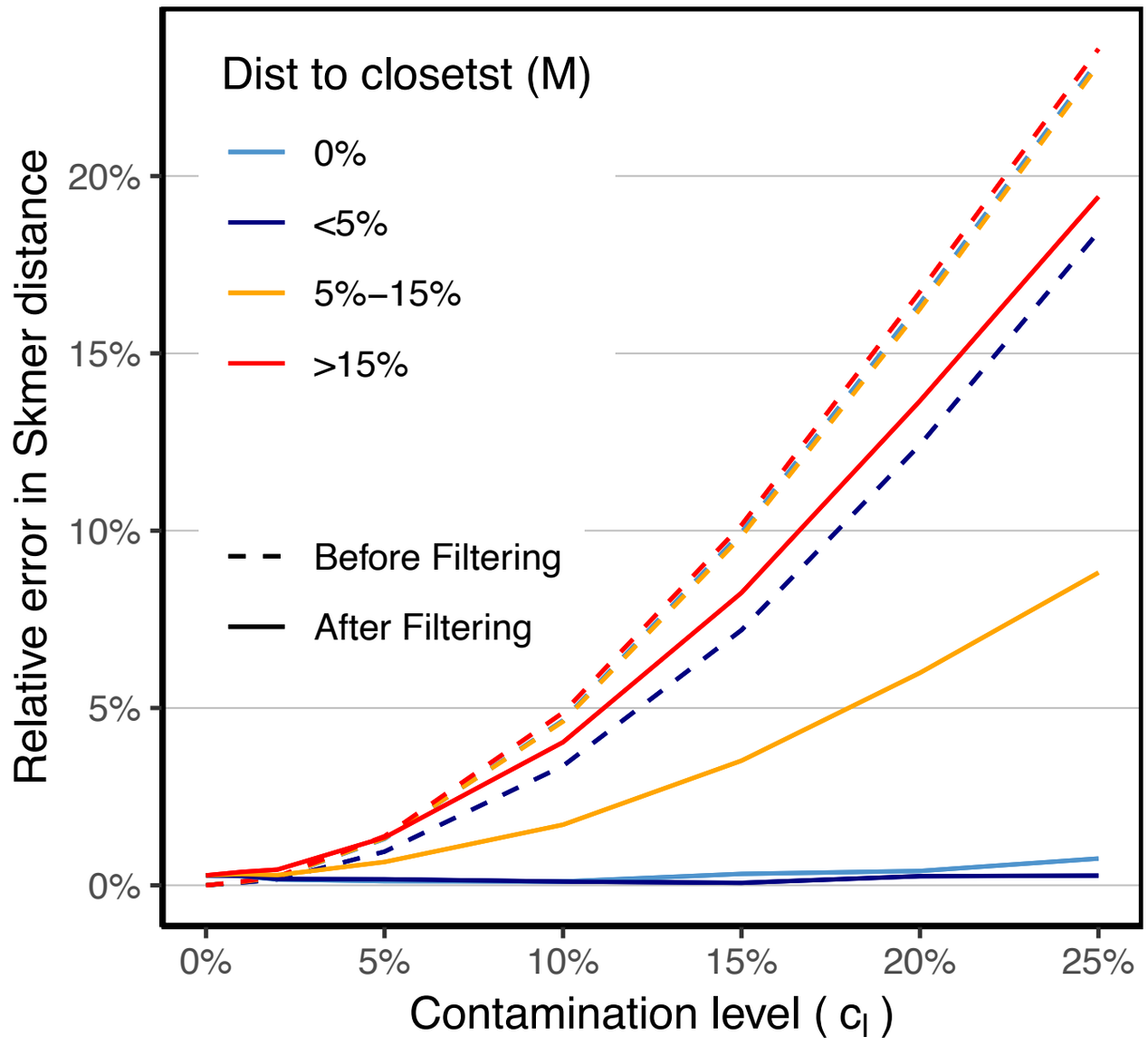


[Rachtman *et al.*, 2019, MRE]

Impact on Skmer accuracy

- Simulated presence of microbial contamination for Drosophila genome skims (100 Mb)

- Measured Skmer error before and after filtering



[Rachtman *et al.*, 2019, MRE]

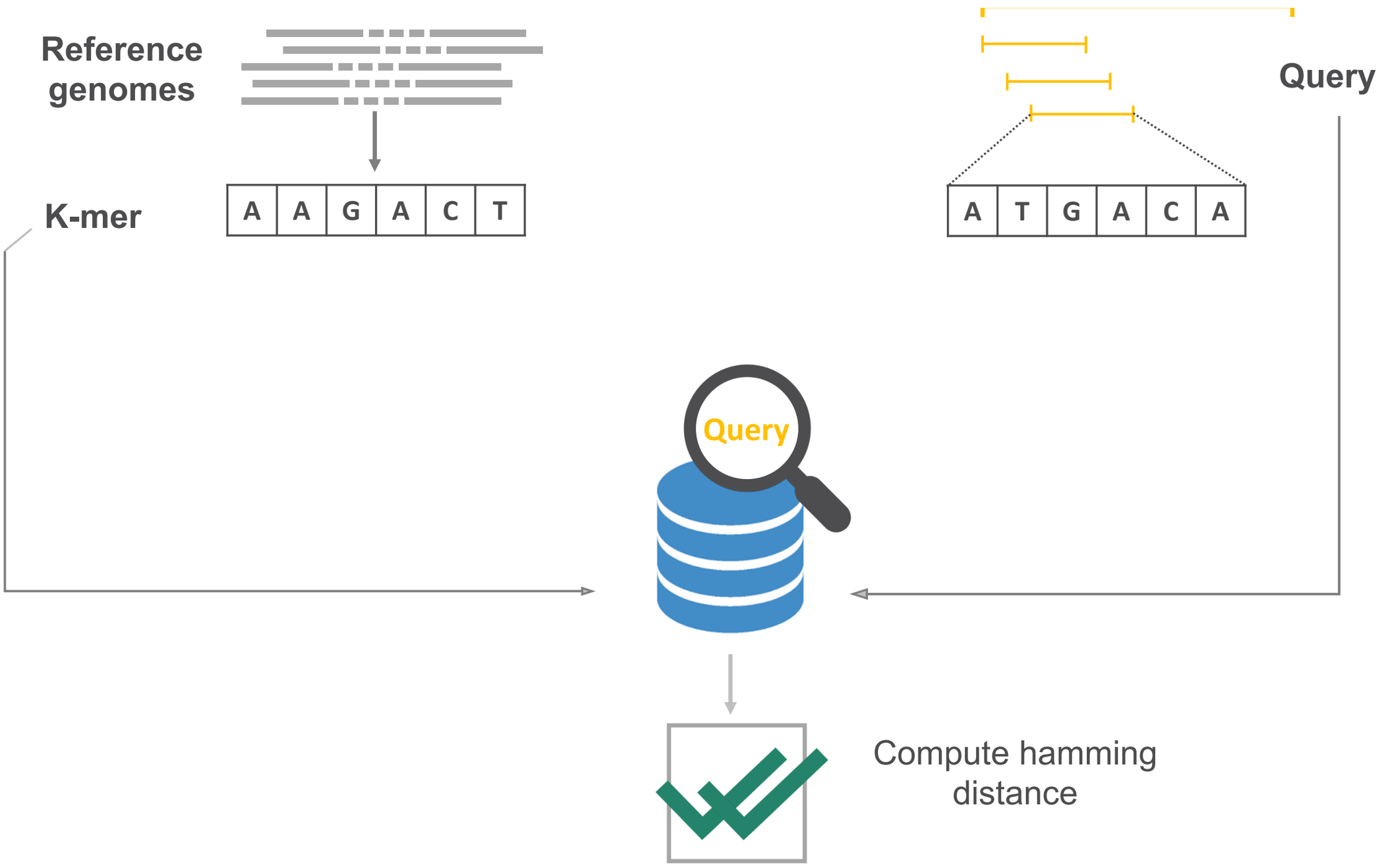
Can we do any better?

- Increase sensitivity of read search when the closest match is relatively far away (e.g., 5-15% distance).
    - Needed because reference libraries *will* remain incomplete
    - Requires inexact k-mer matches between reference and a query sequence

Can we do any better?

- Increase sensitivity of read search when the closest match is relatively far away (e.g., 5-15% distance).
    - Needed because reference libraries *will* remain incomplete
        - Requires inexact k-mer matches between reference and a query sequence
- Formulation:
    - Call a read a match if it has enough (e.g., 1) k-mer(s) that are within a hamming distance (e.g., 3/32) from some k-mer in the reference set

Conceptually …



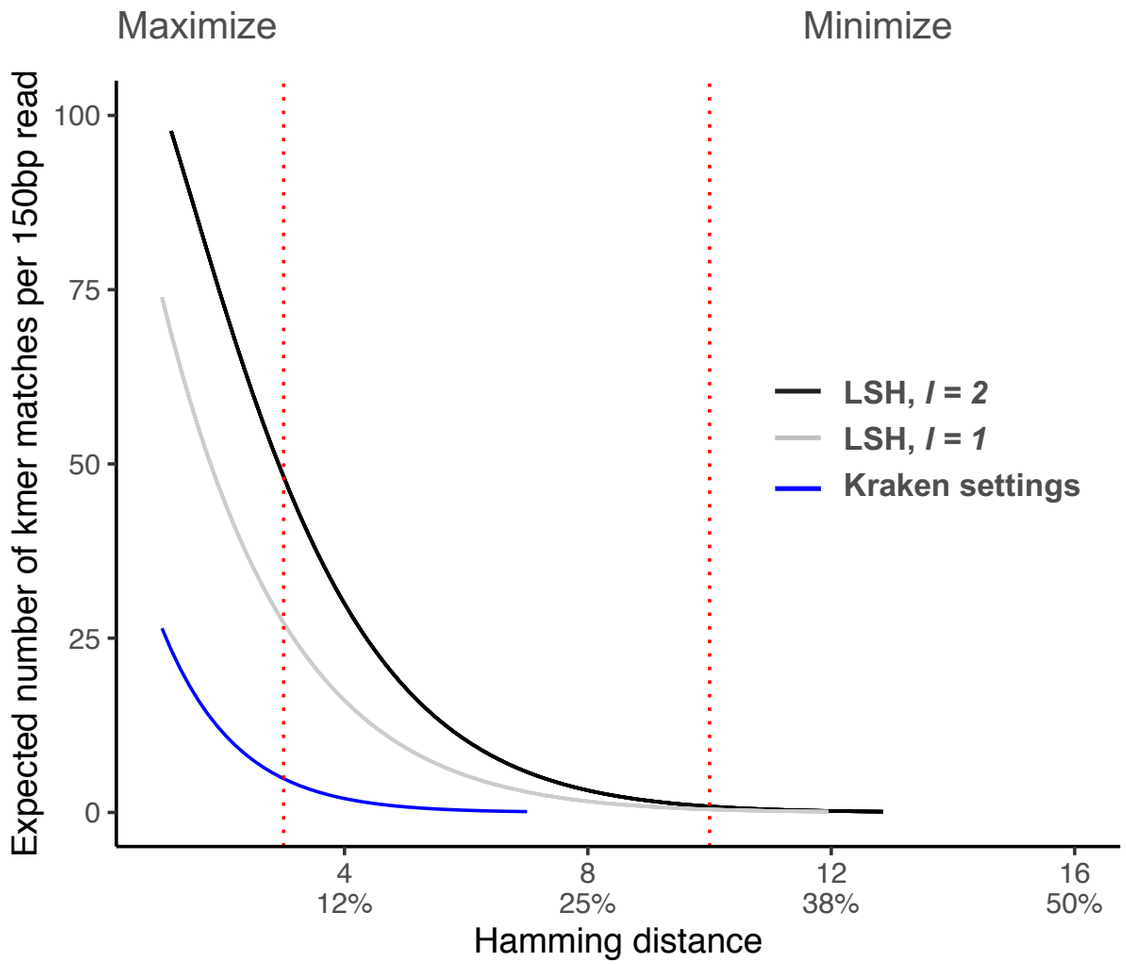
Too slow?

- Use a technique called Locality sensitive hashing (LSH) to compare each k-mer to only a few k-mers (e.g., 200)
    - Guaranteed to find matches with low hamming distance with high probability.
- However, there is a catch!
    - Our approach requires 120GB of memory to keep 8 billion k-mers.

Probability of match

$$Pr\ of\ \ match = 1 - \left(1 - \left(1 - \frac{d}{k}\right)^{h}\right)^{l}$$

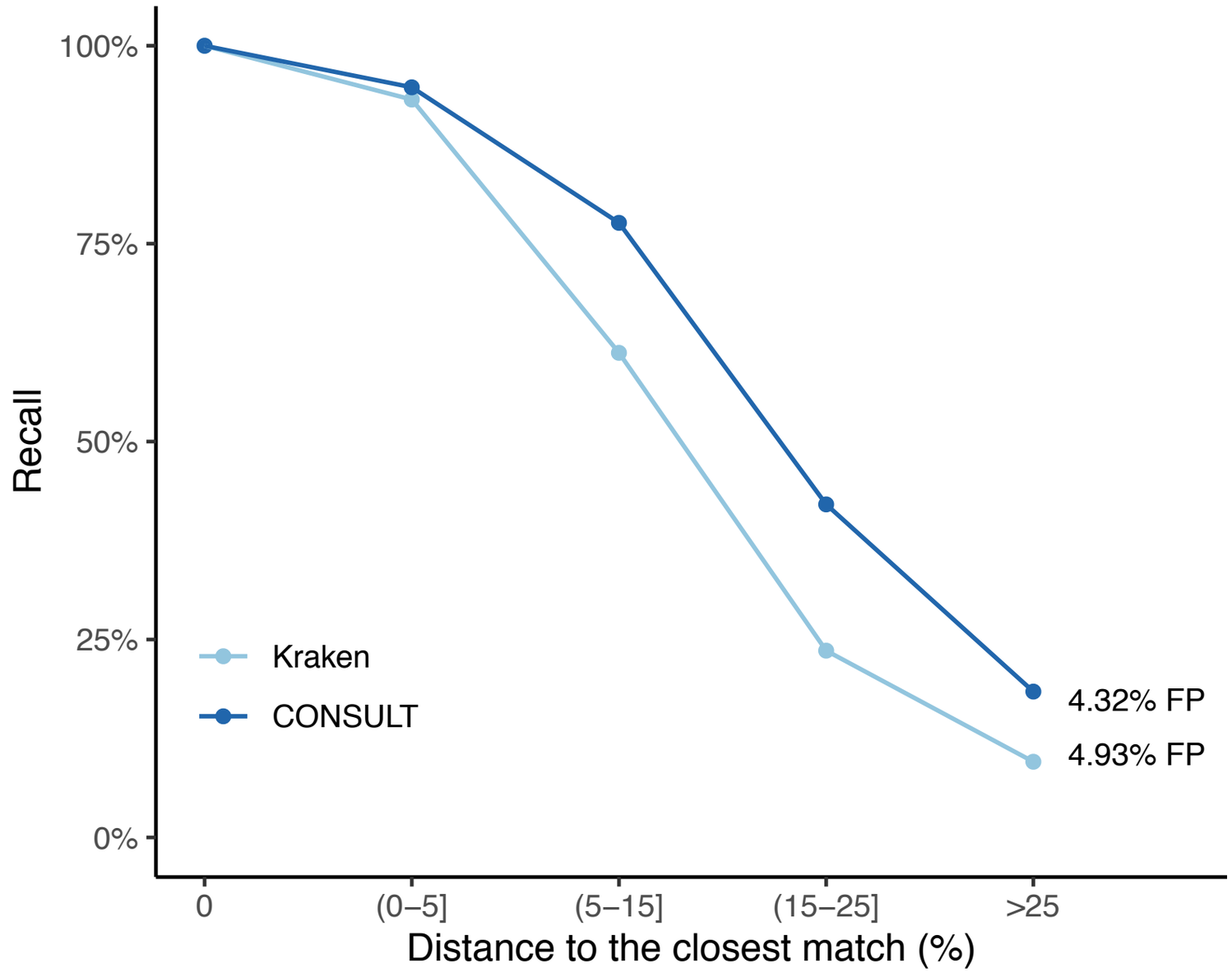
d: distance between k – mers
k: k – mer length
h: number of sampled positions
l: number of hash functions



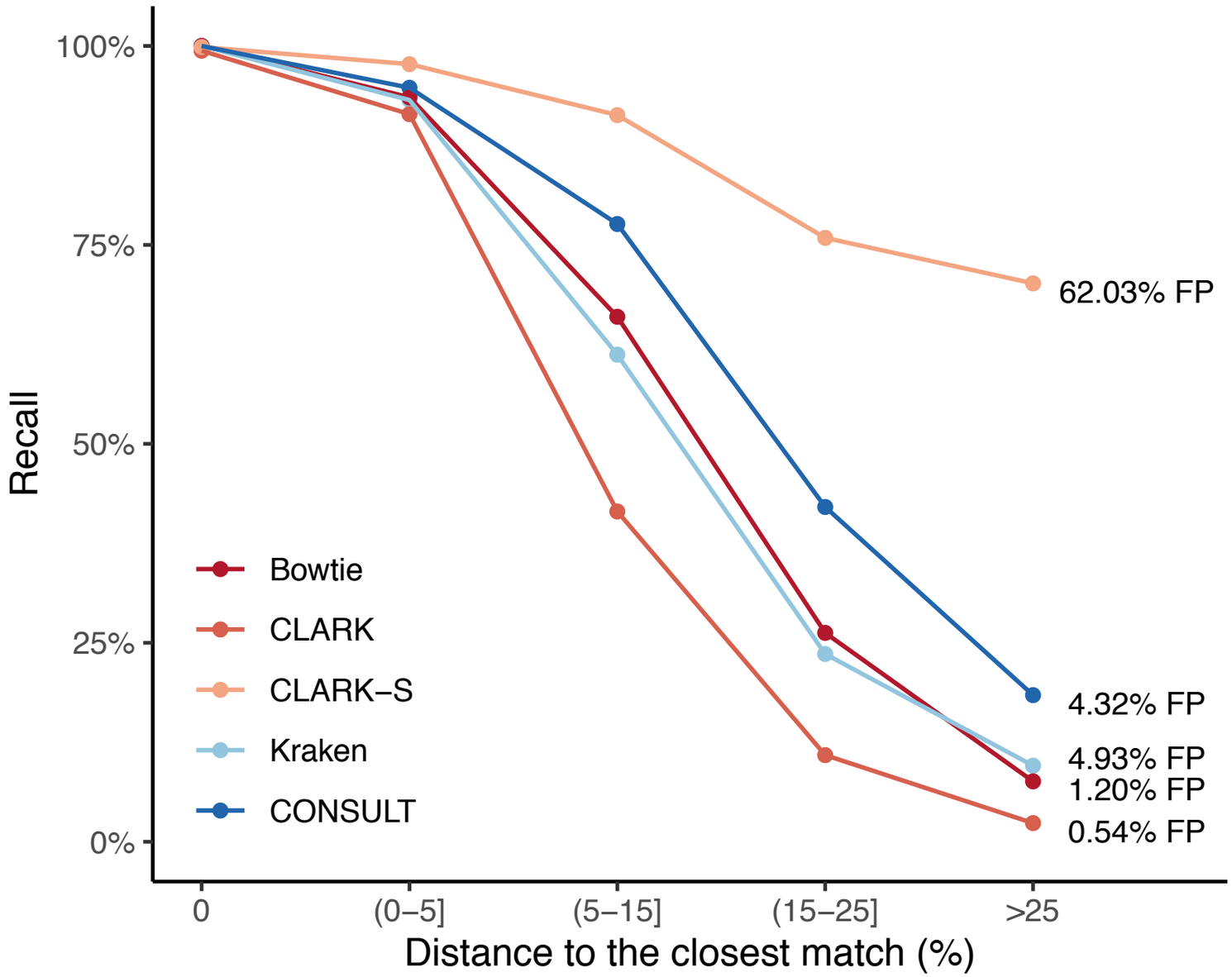
CONSULT

- https://github.com/noraracht/CONSULT
- Paper:
    - Eleonora Rachtman, Vineet Bafna, Siavash Mirarab. NAR Genomics and Bioinformatics, Volume 3, Issue 3, September 2021, https://doi.org/10.1093/nargab/lqab071

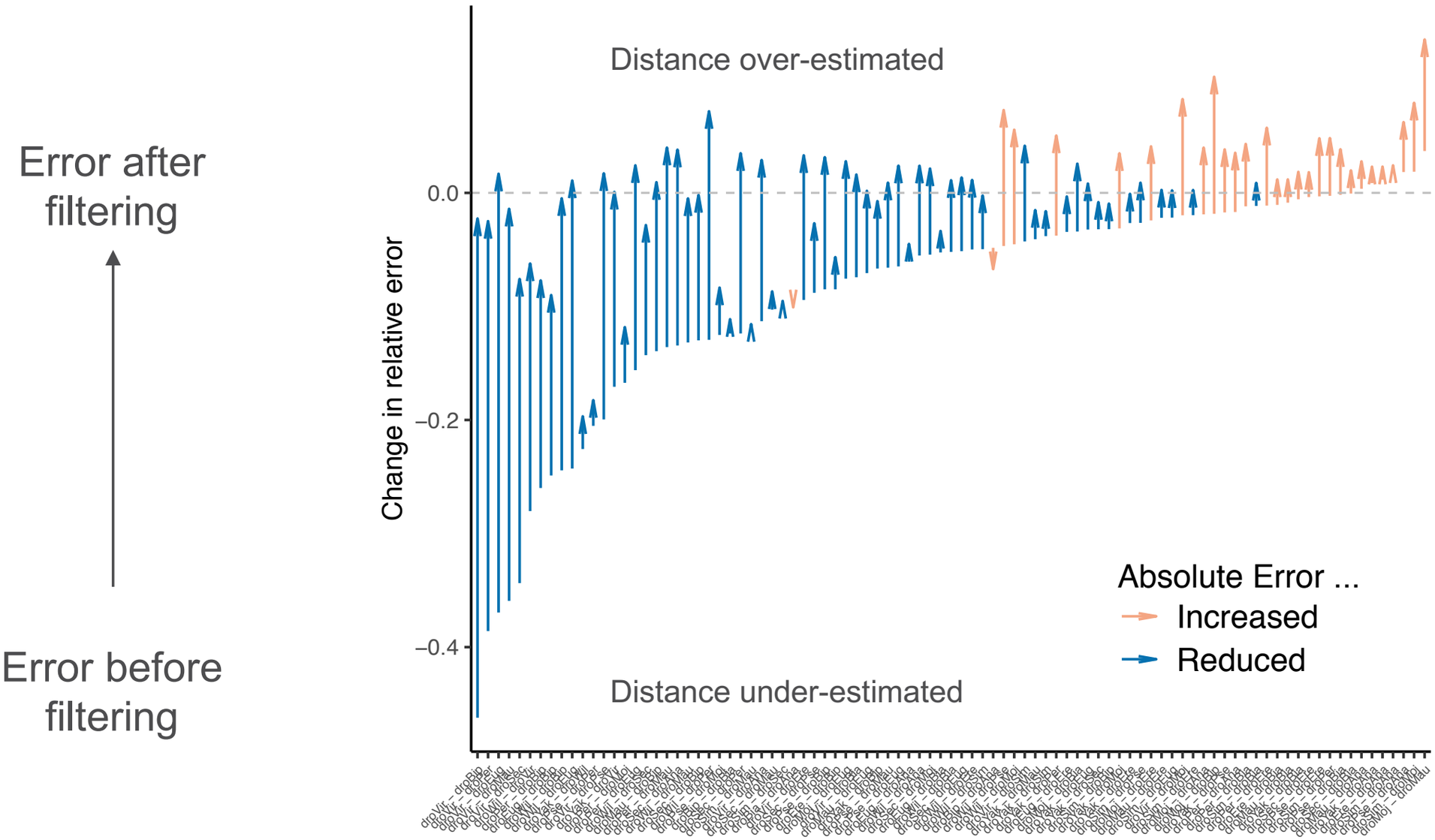
Improves on Kraken-II



And other tools



Contamination removal from a real Drosophila dataset



$$Error = \frac{\widehat{D}_{est} - D_{true}}{D_{true}}$$

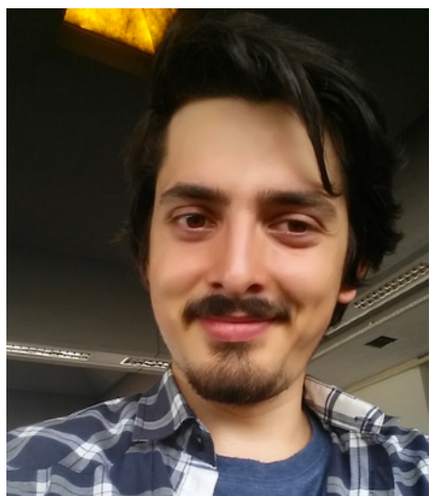
Miller, D. E., Staber, C., Zeitlinger, J., Hawley, R. S. (2018) *G3: Genes, Genomes, Genetics,* 8(10), 3131-3141.

Shahab Sarmashghi

Vineet Bafna

Kristine Bohmann

Tom Gilbert

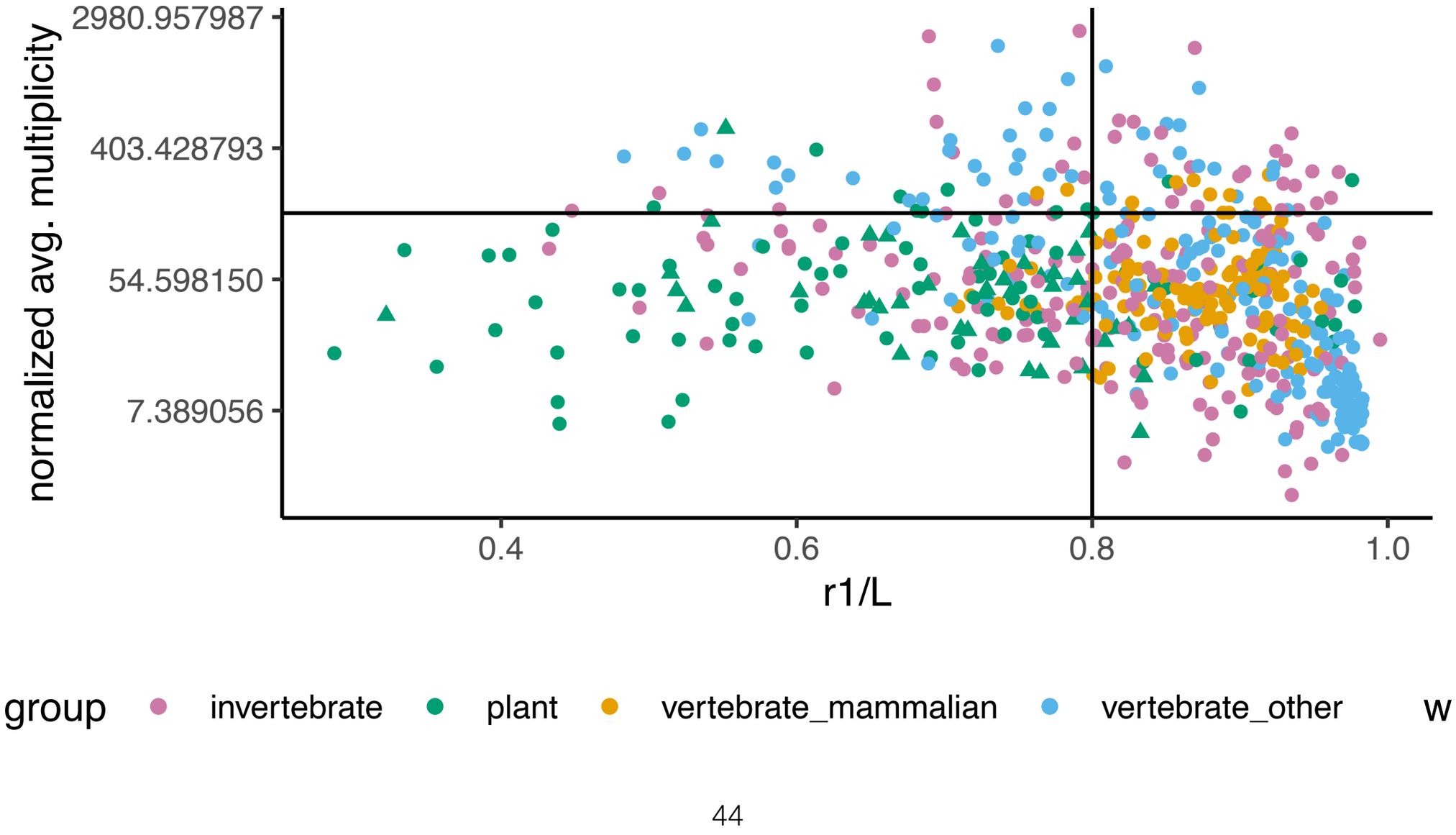


Metin Balaban

Eleonora (Nora) Rachtman

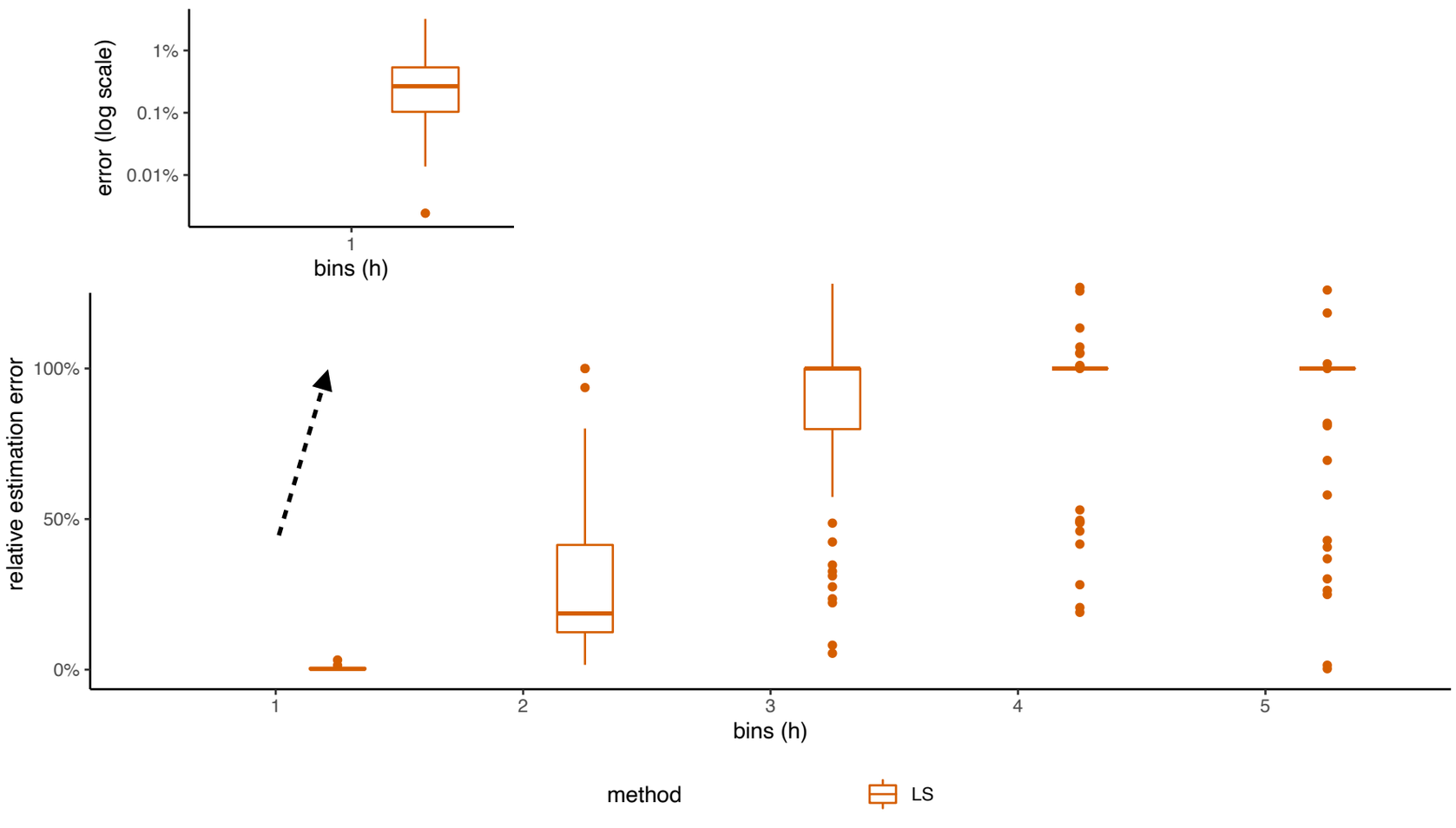
1815485 and 1565862

Putative candidate genomes for WGD events



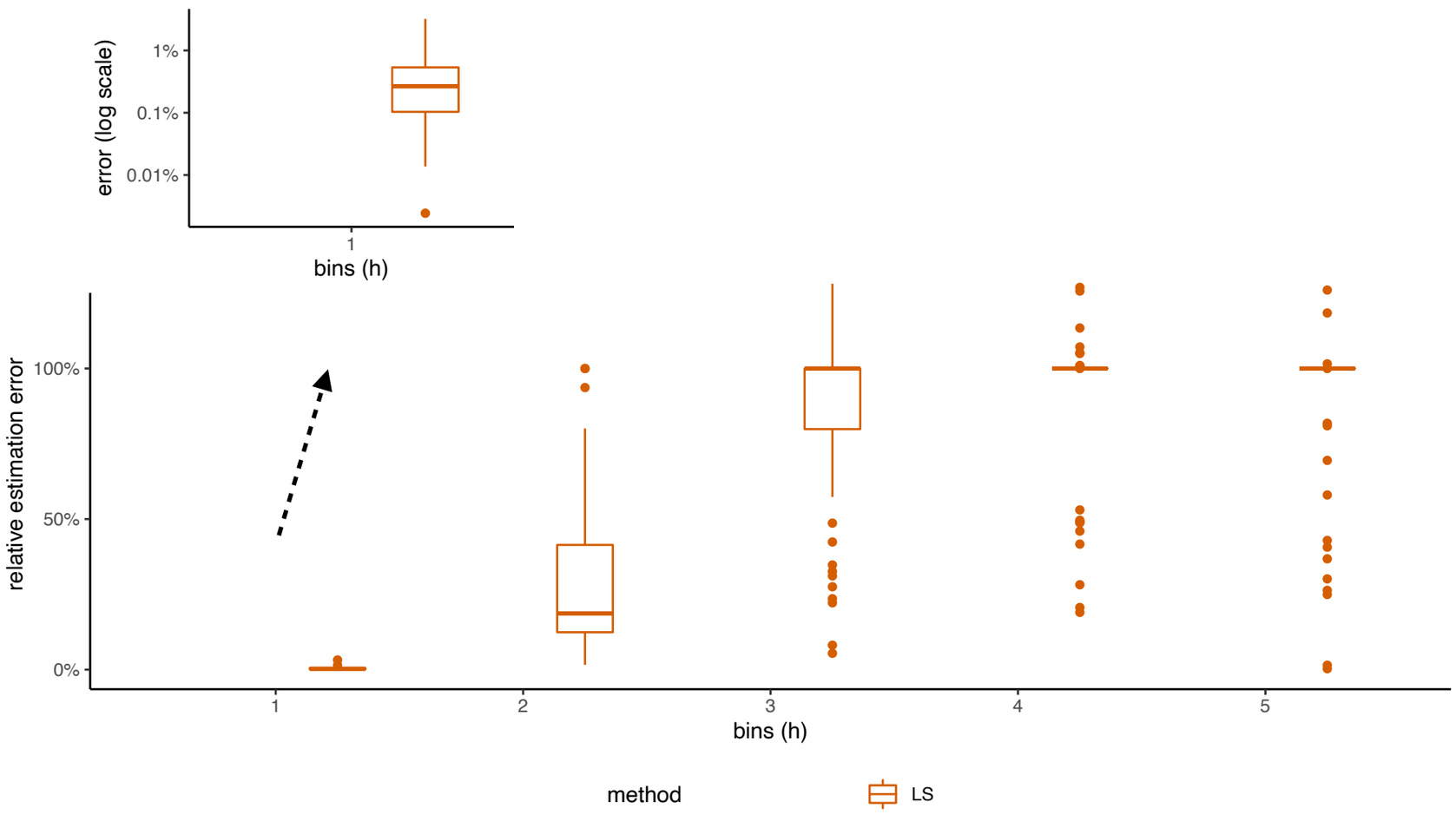
Repeat spectra estimation

- We want to estimate repeat spectra from $m = Pr$



Repeat spectra estimation

- We want to estimate repeat spectra from $m = Pr$
- A simple least-squares (LS) method doesn't work



Why LS doesn't work

- $P = \Lambda VE$ is a highly ill-conditioned matrix

Why LS doesn't work

- $P = \Lambda VE$ is a highly ill-conditioned matrix
- Theorem. The condition number of P grows exponentially with the number of spectra n

Why LS doesn't work

- $P = \Lambda VE$ is a highly ill-conditioned matrix

- Theorem. The condition number of P grows exponentially with the number of spectra n

$$cond(\mathbf{P}) \geq \frac{\mathbf{2}^n}{n}$$

Why LS doesn't work

- $P = \Lambda VE$ is a highly ill-conditioned matrix

- Theorem. The condition number of P grows exponentially with the number of spectra n

$$cond(\mathbf{P}) \geq \frac{\mathbf{2}^n}{n}$$

Why LS doesn't work

- $P = \Lambda VE$ is a highly ill-conditioned matrix
- Theorem. The condition number of P grows exponentially with the number of spectra n

$$cond(\mathbf{P}) \geq \frac{2^n}{n}$$

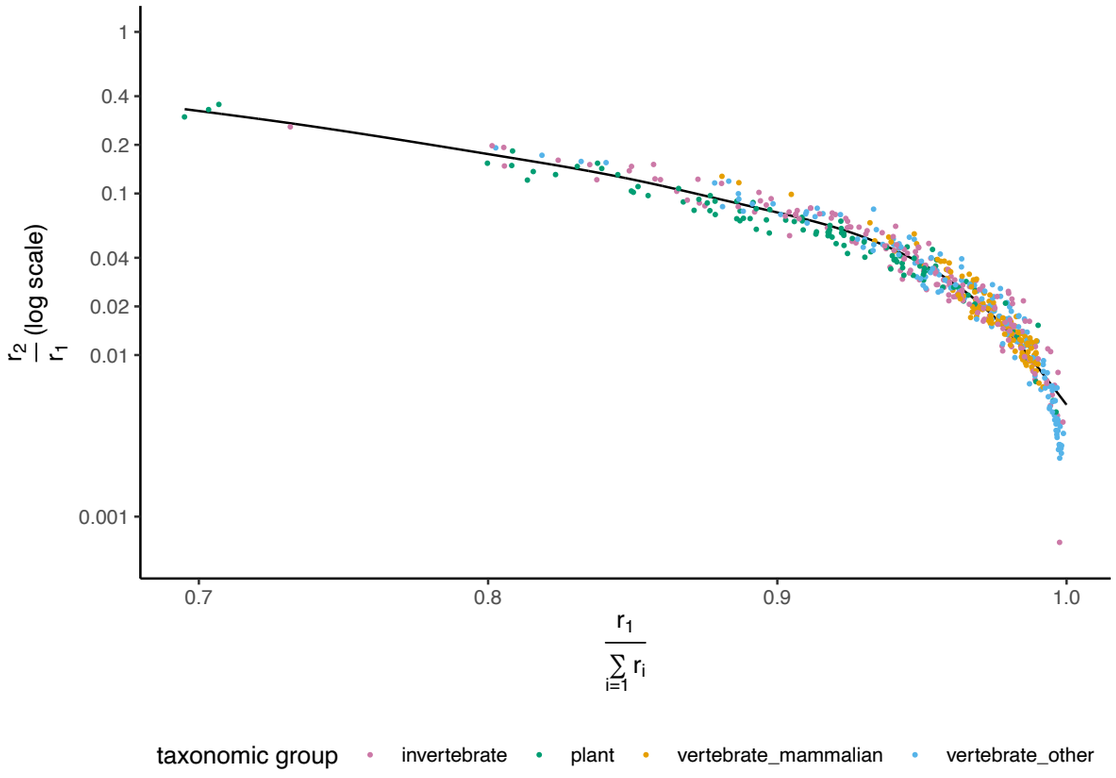
- Small change in $\mathbf{o}$ relative to $m = \mathbb{E}[\mathbf{o}] = \mathbf{Pr}$, results in large error in estimating $\mathbf{r}$

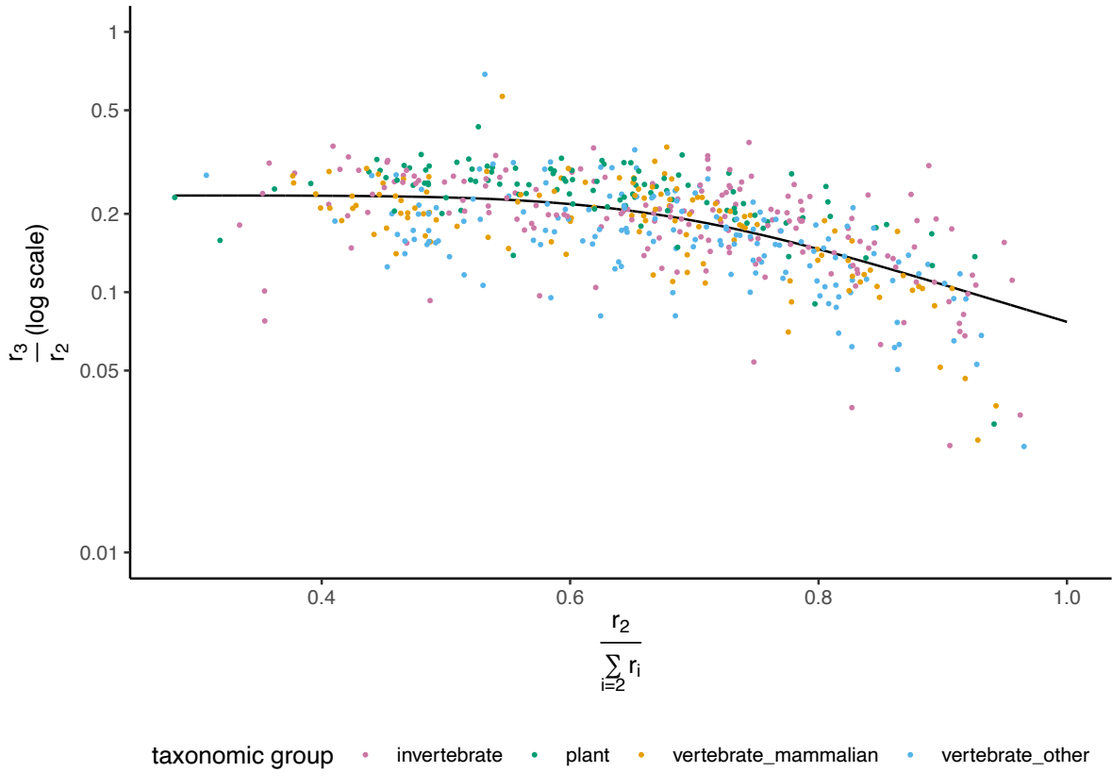
Repeat spectra estimation

- We want to estimate repeat spectra from $m = Pr$
- Learning spectral ratios to constraint the solution space

Repeat spectra estimation

- We want to estimate repeat spectra from $m = Pr$
- Learning spectral ratios to constraint the solution space





Repeat spectra estimation

- Learning spectral ratios to constraint the solution space

Repeat spectra estimation

- Learning spectral ratios to constraint the solution space
- Linear programming

Repeat spectra estimation

- Learning spectral ratios to constraint the solution space
- Linear programming

$$\min_{\mathbf{r}} \sum_{h=2}^{n} \left| o_h - \sum_{j=1}^{n} P_{hj} r_j \right| ,$$

$$\mathcal{L}_h \leq \frac{r_h}{r_{h+1}} \leq \mathcal{U}_h, \quad h = 1, 2, \cdots, n-1$$

Repeat spectra estimation

- Learning spectral ratios to constraint the solution space
- Linear programming

$$\min_{\mathbf{r}} \sum_{h=2}^{n} \left| o_h - \sum_{j=1}^{n} P_{hj} r_j \right| ,$$

$$\mathcal{L}_h \leq \frac{r_h}{r_{h+1}} \leq \mathcal{U}_h, \quad h = 1, 2, \cdots, n-1$$

Repeat spectra estimation

- Learning spectral ratios to constraint the solution space
- Linear programming

$$\min_{\mathbf{r}} \sum_{h=2}^{n} \left| o_h - \sum_{j=1}^{n} P_{hj} r_j \right| ,$$

$$\mathcal{L}_h \leq \frac{r_h}{r_{h+1}} \leq \mathcal{U}_h, \quad h = 1, 2, \cdots, n-1$$

Repeat spectra estimation

- Learning spectral ratios to constraint the solution space
- Linear programming

$$\min_{\mathbf{r}} \sum_{h=2}^{n} \left| o_h - \sum_{j=1}^{n} P_{hj} r_j \right| ,$$

$$\mathcal{L}_h \leq \frac{r_h}{r_{h+1}} \leq \mathcal{U}_h, \quad h = 1, 2, \cdots, n-1$$

- Spline linear programming

Repeat spectra estimation

- Learning spectral ratios to constraint the solution space

- Linear programming

$$\min_{\mathbf{r}} \sum_{h=2}^{n} \left| o_h - \sum_{j=1}^{n} P_{hj} r_j \right| ,$$

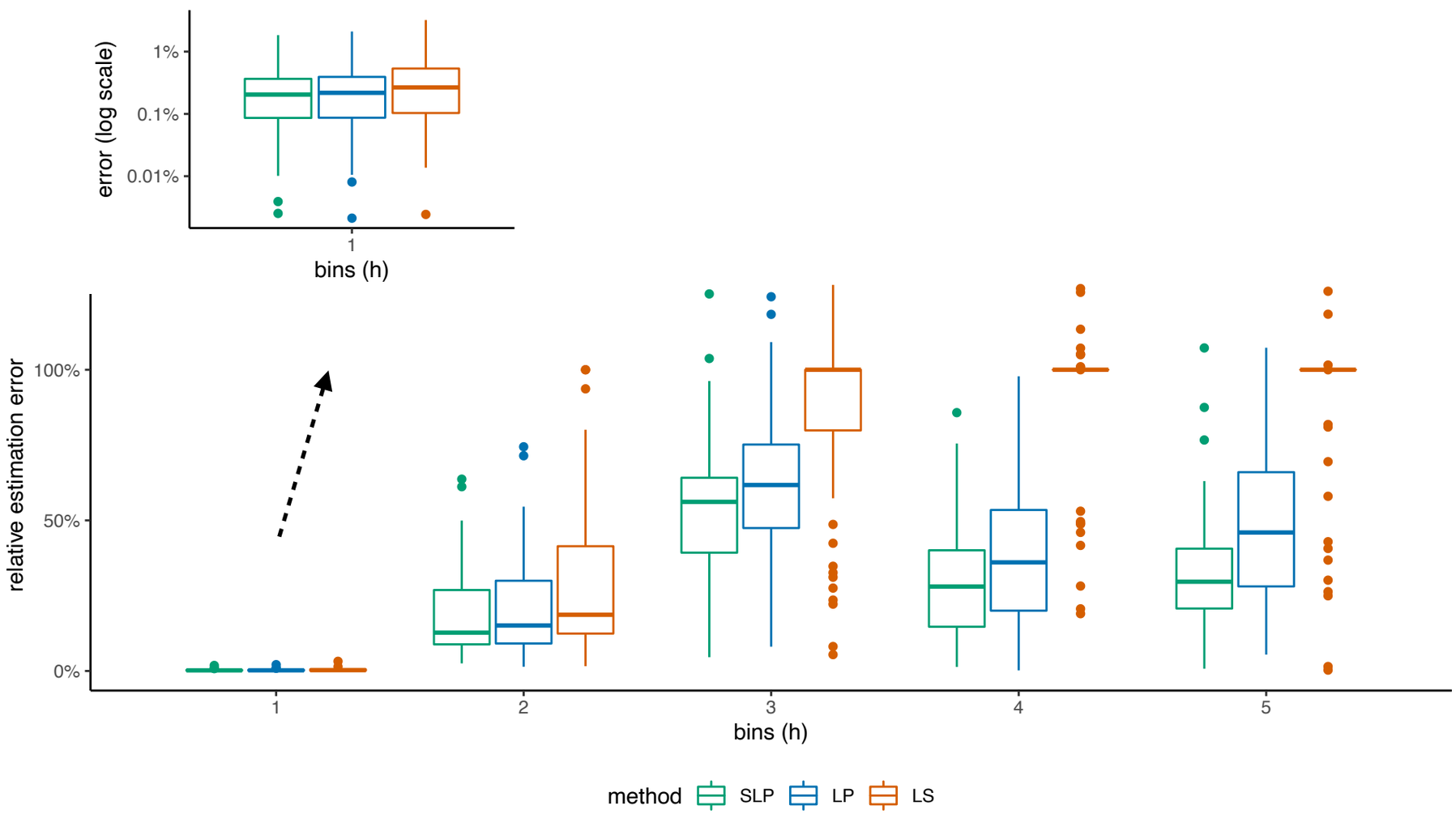
$$\mathcal{L}_h \leq \frac{r_h}{r_{h+1}} \leq \mathcal{U}_h, \quad h = 1, 2, \cdots, n-1$$

- Spline linear programming

$$r_j^{\mathrm{SLP}} = \begin{cases} r_j^{\mathrm{LP}} & j = 1 \ \text{ and } \ j > 6 \\ r_{j-1}^{\mathrm{SLP}} / y_{j-1}^{\mathrm{SLP}} & 2 \leq j \leq 6 \end{cases}$$

K-mer spectra estimation

- Improved accuracy of estimated repeat spectra



RESPECT algorithm

- Estimating parameters by minimizing error between expected and observed k-mer counts, using a simulated annealing scheme

RESPECT algorithm

- Estimating parameters by minimizing error between expected and observed k-mer counts, using a simulated annealing scheme

Algorithm 1: The RESPECT method

Start with $\lambda_{\text{ef}} = \lambda^{(0)}(1-\epsilon)^k = \frac{(h^*+1)o_{h^*+1}}{o_{h^*}}$, where $h^* = \arg\max_{h>1} o_h$;

Compute $\mathbf{P}^{(0)}$, $\mathcal{E}^{(0)} = \min_{\mathbf{r}} \mathcal{E}(\mathbf{P}^{(0)}, \mathbf{r}^{(0)}, \mathbf{o})$, and $\mathbf{r}^{(0)} = \arg\min_{\mathbf{r}} \mathcal{E}(\mathbf{P}^{(0)}, \mathbf{r}^{(0)}, \mathbf{o})$;

Find $E = o_1 - \sum_j P_{1j}^{(0)} r_j^{(0)}$;

Set $\lambda^{(0)} = e^{-\lambda_{\text{ef}}} \frac{\lambda_{\text{ef}}^{h^*}}{h^*!} \frac{o_1}{o_{h^*}} + \lambda_{\text{ef}}(1 - e^{-\lambda_{\text{ef}}})$, and compute ϵ from λ_{ef} and $\lambda^{(0)}$;

for $1 \le t \le N$ **do**

  $\lambda^{(t)} \leftarrow \mathcal{U}[\frac{1}{2} \cdot \lambda^{(t-1)}, 3 \cdot \lambda^{(t-1)}]$;

  Use $\lambda^{(t)}$ and ϵ to compute $\mathbf{P}^{(t)}$, $\mathbf{r}^{(t)} = \arg\min_{\mathbf{r}} \mathcal{E}(\mathbf{P}^{(t)}, \mathbf{r}^{(t)}, \mathbf{o})$, and $\mathcal{E}^{(t)} = \min_{\mathbf{r}} \mathcal{E}(\mathbf{P}^{(t)}, \mathbf{r}^{(0)}, \mathbf{o})$;

  Move to $\lambda^{(t)}$ with probability $\min\left\{1, \exp\left(\frac{\mathcal{E}^{(t-1)} - \mathcal{E}^{(t)}}{N-t+1}\right)\right\}$;

end

Output $c^{(N)} = \frac{\ell}{\ell-k+1}\lambda^{(N)}$, $L^{(N)} = B/c^{(N)}$, $\mathbf{r}^{(N)}$, and ϵ