Towards subquadratic

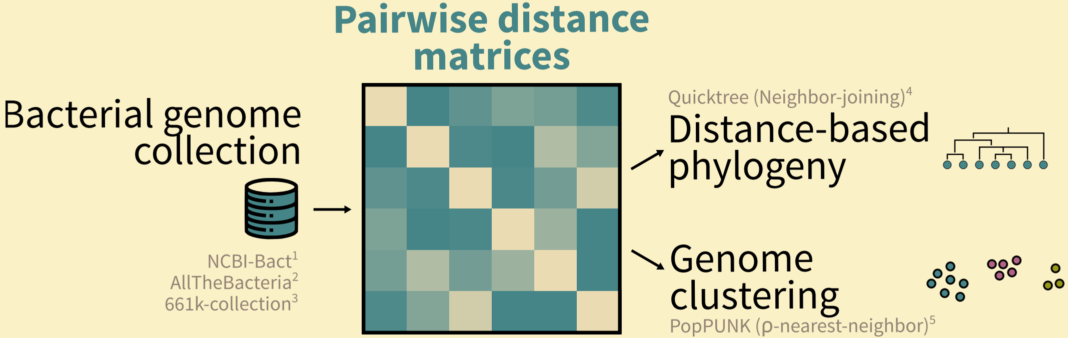
data structures for large genome-distance matrices with quick retrieval

Léo Ackermann[1], Pierre Peterlongo[1], Karel Břinda[1]

[1] Inria, Genscale, Rennes

DSB Workshop, 5th March 2025

Importance of pairwise distance matrices



Pairwise distance matrices

Bacterial genome collection

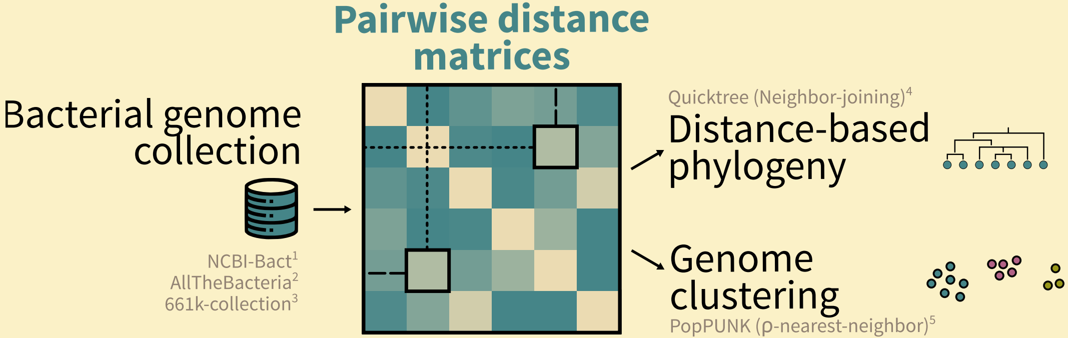
NCBI-Bact[1]
AllTheBacteria[2]
661k-collection[3]

Quicktree (Neighbor-joining)[4]
Distance-based phylogeny

Genome clustering
PopPUNK (ρ-nearest-neighbor)[5]

[1] National Center for Biotechnology Information (*https://ncbi.nlm.nih.gov*) [2] *Hunt et. al.* BioRxiv, 2024. [3] *Blackwell et. al.* PLOS Biology, 2021. [4] *Howe et. al.* Bioinformatics, 2002. [5] *Lees et. al.* Genome Research, 2019. [6] *Ondov et. al.* Genome Biology, 2016. [7] *Baker et. al.* Genome Biology, 2019.

Importance of pairwise distance matrices



Pairwise distance matrices

Bacterial genome collection

NCBI-Bact[1]
AllTheBacteria[2]
661k-collection[3]

Quicktree (Neighbor-joining)[4]

Distance-based phylogeny

Genome clustering

PopPUNK (ρ-nearest-neighbor)[5]

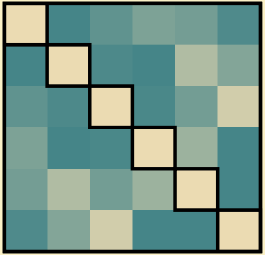
[1] National Center for Biotechnology Information (*https://ncbi.nlm.nih.gov*) [2] *Hunt et. al.* BioRxiv, 2024. [3] *Blackwell et. al.* PLOS Biology, 2021. [4] *Howe et. al.* Bioinformatics, 2002. [5] *Lees et. al.* Genome Research, 2019. [6] *Ondov et. al.* Genome Biology, 2016. [7] *Baker et. al.* Genome Biology, 2019.

Importance of pairwise distance matrices

Pairwise distance matrices

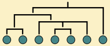
Bacterial genome collection

NCBI-Bact[1]
AllTheBacteria[2]
661k-collection[3]



Quicktree (Neighbor-joining)[4]

Distance-based phylogeny



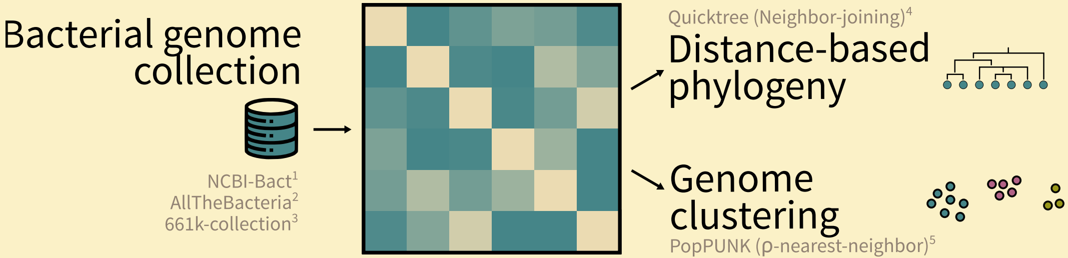
Genome clustering

PopPUNK (ρ-nearest-neighbor)[5]

[1] National Center for Biotechnology Information (*https://ncbi.nlm.nih.gov*) [2] *Hunt et. al.* BioRxiv, 2024. [3] *Blackwell et. al.* PLOS Biology, 2021. [4] *Howe et. al.* Bioinformatics, 2002. [5] *Lees et. al.* Genome Research, 2019. [6] *Ondov et. al.* Genome Biology, 2016. [7] *Baker et. al.* Genome Biology, 2019.

Importance of pairwise distance matrices

Pairwise distance matrices



Bacterial genome collection

NCBI-Bact[1]
AllTheBacteria[2]
661k-collection[3]

Quicktree (Neighbor-joining)[4]

Distance-based phylogeny

Genome clustering

PopPUNK (ρ-nearest-neighbor)[5]

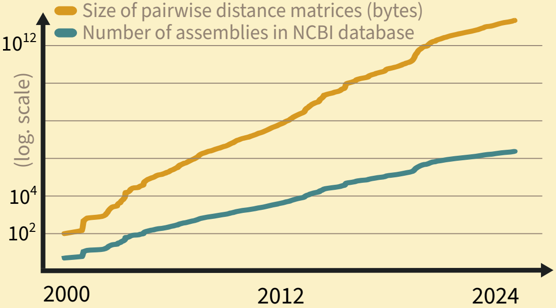
Efficient computation of distance matrices

Sketching (e.g., Mash[6], Dashing[7]) and **parallel computing** make it tractable

[1] National Center for Biotechnology Information (*https://ncbi.nlm.nih.gov*) [2] *Hunt et. al.* BioRxiv, 2024. [3] *Blackwell et. al.* PLOS Biology, 2021. [4] *Howe et. al.* Bioinformatics, 2002. [5] *Lees et. al.* Genome Research, 2019. [6] *Ondov et. al.* Genome Biology, 2016. [7] *Baker et. al.* Genome Biology, 2019.

Storage of genome distances is challenging

Size of bacterial collections increases exponentially



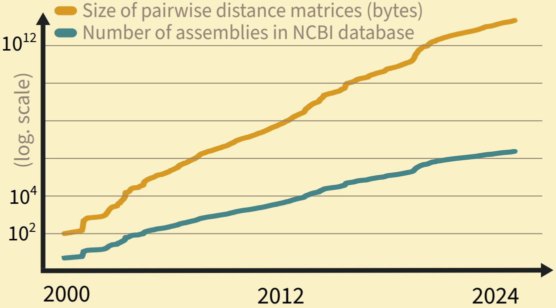
NCBI-bact[1]: 2.4M genomes

- ▶ $2.8 \cdot 10^{12}$ distances, 11 TeraBytes

[1] National Center for Biotechnology Information (*https://ncbi.nlm.nih.gov*) [2] *Hunt et. al.* BioRxiv, 2024. [3] *Blackwell et. al.* PLOS Biology, 2021.

Storage of genome distances is challenging

Size of bacterial collections increases exponentially



NCBI-bact[1]: 2.4M genomes

▶ $2.8 \cdot 10^{12}$ distances, 11 TeraBytes

Other collections.

– AllTheBacteria[2]: 2.4M genomes

▶ $2.8 \cdot 10^{12}$ distances, 11 TeraBytes

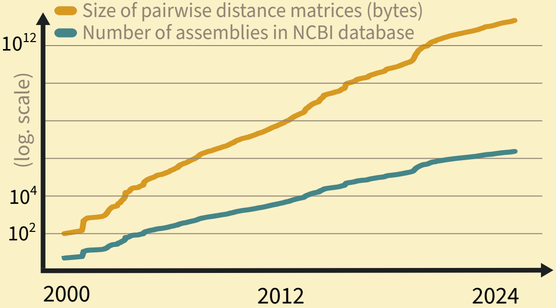
– 661k-collection[3]: 661k genomes

▶ $2.2 \cdot 10^{11}$ distances, 880 GigaBytes

[1] National Center for Biotechnology Information (*https://ncbi.nlm.nih.gov*) [2] *Hunt et. al.* BioRxiv, 2024. [3] *Blackwell et. al.* PLOS Biology, 2021.

Storage of genome distances is challenging

Size of bacterial collections increases exponentially



NCBI-bact[1]: 2.4M genomes

▶ $2.8 \cdot 10^{12}$ distances, 11 TeraBytes

Other collections.

– AllTheBacteria[2]: 2.4M genomes

▶ $2.8 \cdot 10^{12}$ distances, 11 TeraBytes

– 661k-collection[3]: 661k genomes

▶ $2.2 \cdot 10^{11}$ distances, 880 GigaBytes

Generic matrix compression techniques

Matrix-specific compression techniques are restricted to sparse and low-rank matrices, and are **not directly applicable**

[1] National Center for Biotechnology Information (*https://ncbi.nlm.nih.gov*) [2] *Hunt et. al.* BioRxiv, 2024. [3] *Blackwell et. al.* PLOS Biology, 2021.

Problem formulation

Many variants can be framed

- **Operability.** A **set of operations** to interact with the data structure, with constraints
  e.g., random access, sequencial access, nothing, …

Problem formulation

Many variants can be framed

- **Operability.** A **set of operations** to interact with the data structure, with constraints
  e.g., random access, sequencial access, nothing, . . .
- **Accuracy.** Whether the structure stores **exact or approximate** distances

Problem formulation

Many variants can be framed

- **Operability.** A **set of operations** to interact with the data structure, with constraints
  e.g., random access, sequencial access, nothing, …
- **Accuracy.** Whether the structure stores **exact or approximate** distances
- **Dynamicity.** Whether the structure **can(not) be updated** without recomputing everything

Problem formulation

Many variants can be framed

- **Operability.** A **set of operations** to interact with the data structure, with constraints
  e.g., random access, sequencial access, nothing, …
- **Accuracy.** Whether the structure stores **exact or approximate** distances
- **Dynamicity.** Whether the structure **can(not) be updated** without recomputing everything

◎ Focus of this presentation

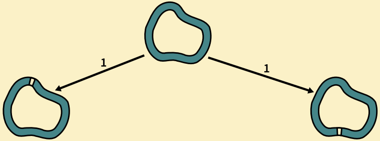
STATIC COMPRESSION OF PAIRWISE DISTANCE MATRICES OF **SINGLE SPECIES** COLLECTIONS, WITH **CONSTANT-TIME RANDOM ACCESS**

Method for the infinite sites model

The infinite sites model[1] of evolution

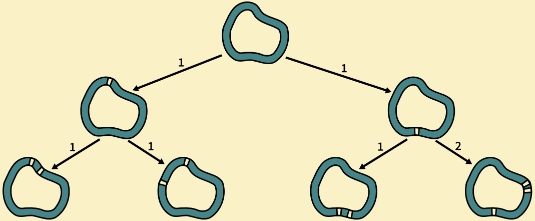
[1] *Kimura*. Genomics, 1969. [2] *Saitou et. al.* Molecular Biology and Evolution, 1987.

The infinite sites model[1] of evolution



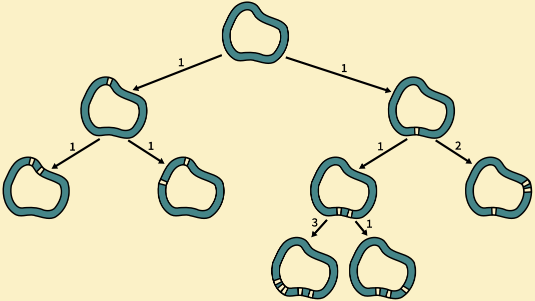
[1] *Kimura*. Genomics, 1969. [2] *Saitou et. al.* Molecular Biology and Evolution, 1987.

The infinite sites model[1] of evolution



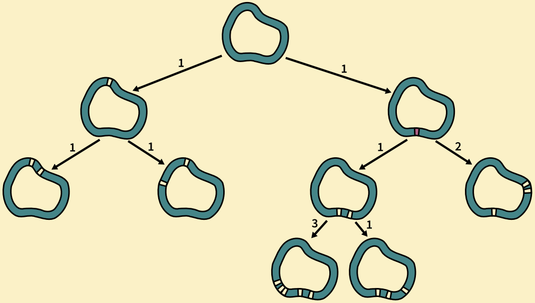
[1] *Kimura*. Genomics, 1969. [2] *Saitou et. al.* Molecular Biology and Evolution, 1987.

The infinite sites model[1] of evolution



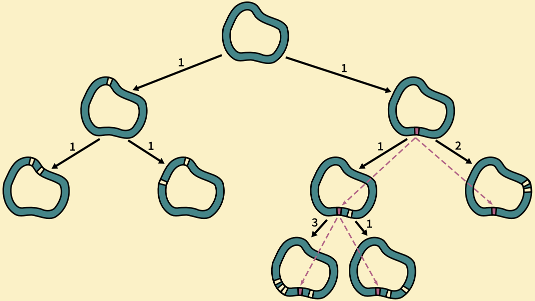
[1] *Kimura*. Genomics, 1969. [2] *Saitou et. al.* Molecular Biology and Evolution, 1987.

The infinite sites model[1] of evolution



[1] *Kimura*. Genomics, 1969. [2] *Saitou et. al.* Molecular Biology and Evolution, 1987.

The infinite sites model[1] of evolution

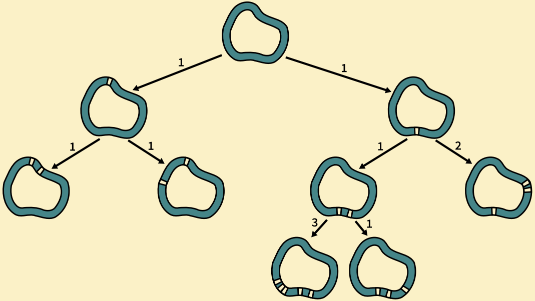


[1] *Kimura*. Genomics, 1969. [2] *Saitou et. al.* Molecular Biology and Evolution, 1987.

The infinite sites model[1] of evolution

Model. There is **no horizontal gene transfer** and genomes are of **infinite size**.

- Mutations always occur at a different genome location (hence not reversible!)
- Good model at very small time scale (eg. clinical outbreak)

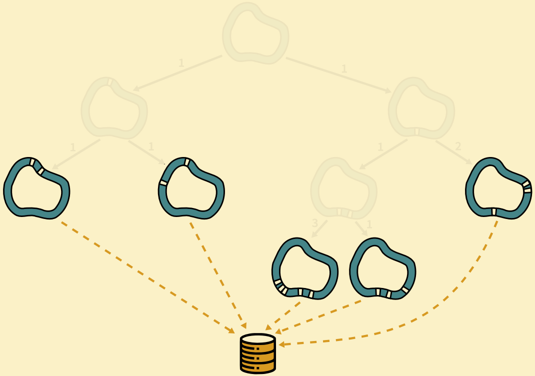


[1] *Kimura*. Genomics, 1969. [2] *Saitou et. al.* Molecular Biology and Evolution, 1987.

The infinite sites model[1] of evolution

Model. There is **no horizontal gene transfer** and genomes are of **infinite size**.

- Mutations always occur at a different genome location (hence not reversible!)
- Good model at very small time scale (eg. clinical outbreak)

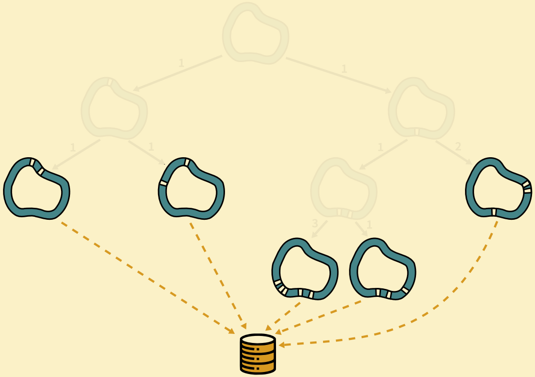


[1] *Kimura*. Genomics, 1969. [2] *Saitou et. al.* Molecular Biology and Evolution, 1987.

The infinite sites model[1] of evolution

Model. There is **no horizontal gene transfer** and genomes are of **infinite size**.

- Mutations always occur at a different genome location (hence not reversible!)
- Good model at very small time scale (eg. clinical outbreak)



High level idea

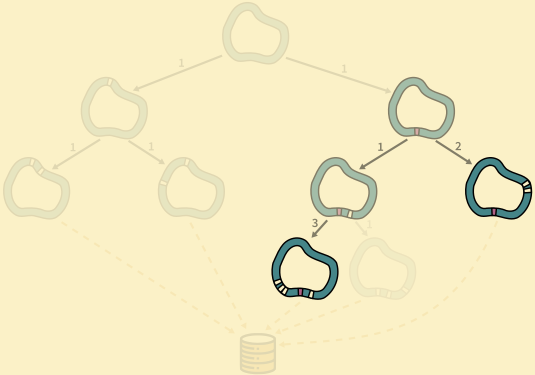
(1) Recover the phylogenetic tree,
(2) Compute pairwise distances from it

[1] *Kimura*. Genomics, 1969. [2] *Saitou et. al.* Molecular Biology and Evolution, 1987.

The infinite sites model[1] of evolution

Model. There is **no horizontal gene transfer** and genomes are of **infinite size**.

- Mutations always occur at a different genome location (hence not reversible!)
- Good model at very small time scale (eg. clinical outbreak)



💡 High level idea

(1) Recover the phylogenetic tree,
(2) Compute pairwise distances from it

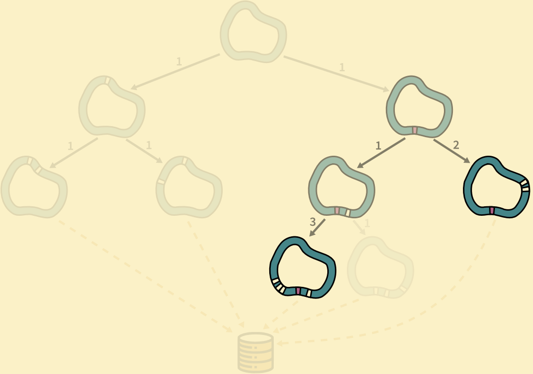
Step 1. We observe that

[1] *Kimura*. Genomics, 1969. [2] *Saitou et. al.* Molecular Biology and Evolution, 1987.

The infinite sites model[1] of evolution

Model. There is **no horizontal gene transfer** and genomes are of **infinite size**.

- Mutations always occur at a different genome location (hence not reversible!)
- Good model at very small time scale (eg. clinical outbreak)



High level idea

(1) Recover the phylogenetic tree,
(2) Compute pairwise distances from it

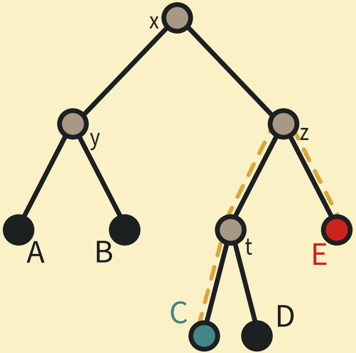
Step 1. We observe that

$$\mathcal{H}(G, G') = \sum_{(x,y)\in\mathcal{T}(G\to G')} \mathcal{H}(x,y) = \delta_{\mathcal{T}}(G, G'),$$

In these conditions, the **Neighbor-joining**[2] algorithm will exactly **retrieve the tree** from the leave pairwise distances

[1] *Kimura*. Genomics, 1969. [2] *Saitou et. al.* Molecular Biology and Evolution, 1987.

Computing leave distance in constant-time

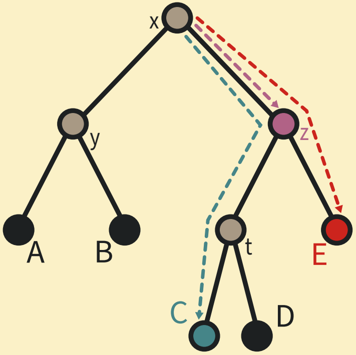


Objective. Compute $\delta(C, E)$

- Naive algorithm in time $O(depth)$

[1] Genome-Scale Algorithm Design (2nd edition). *Mäkinen et. al.* 2023.

Computing leave distance in constant-time



Objective. Compute $\delta(C, E)$

- ▶ Naive algorithm in time $O(depth)$

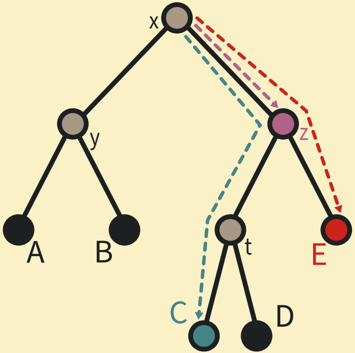
1. Expressing $\delta(C, E)$ with root-to-node distances

$$\delta_T(C, E) = \text{rtn}(C) + \text{rtn}(E) - 2 * \text{rtn}(\text{lca}(C, E))$$

- ▶ Storing root-to-node distances requires linear space

[1] Genome-Scale Algorithm Design (2nd edition). *Mäkinen et. al.* 2023.

Computing leave distance in constant-time



Objective. Compute $\delta(C, E)$

- ▶ Naive algorithm in time $O(depth)$

1. Expressing $\delta(C, E)$ with root-to-node distances

$$\delta_T(C, E) = \mathrm{rtn}(C) + \mathrm{rtn}(E) - 2 * \mathrm{rtn}(\mathrm{lca}(C, E))$$

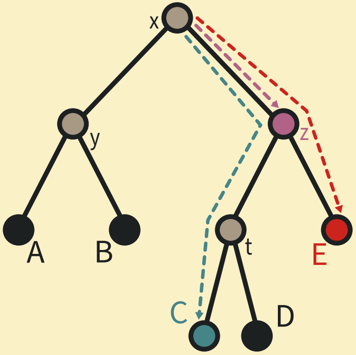
- ▶ Storing root-to-node distances requires linear space

2. Recover Lowest Common Ancestor in constant time

This takes constant time at the cost of extra linear space[1]

[1] Genome-Scale Algorithm Design (2nd edition). *Mäkinen et. al.* 2023.

Computing leave distance in constant-time



Objective. Compute $\delta(C, E)$

- ▶ Naive algorithm in time $O(depth)$

1. Expressing $\delta(C, E)$ with root-to-node distances

$$\delta_T(C, E) = \mathrm{rtn}(C) + \mathrm{rtn}(E) - 2 * \mathrm{rtn}(\mathrm{lca}(C, E))$$

- ▶ Storing root-to-node distances requires linear space

2. Recover Lowest Common Ancestor in constant time

This takes constant time at the cost of extra linear space[1]

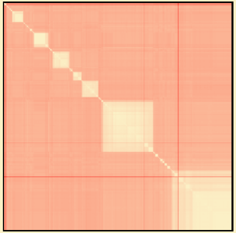
Overall. The pairwise **Hamming distance** between genomes following the **infinite sites model** can be stored in **linear space** with **constant-time random access**, after a linear time preprocessing, without any loss

[1] Genome-Scale Algorithm Design (2nd edition). *Mäkinen et. al.* 2023.

Methods for real data

Tree decomposition of distance matrices

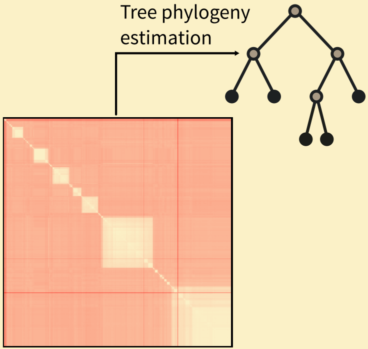
Challenge. Real genomes collections doesn’t follow the infinite sites model



Pairwise distance matrix

Tree decomposition of distance matrices

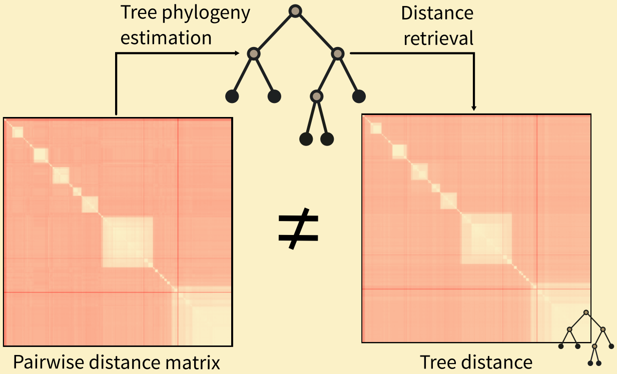
Challenge. Real genomes collections doesn't follow the infinite sites model



Pairwise distance matrix

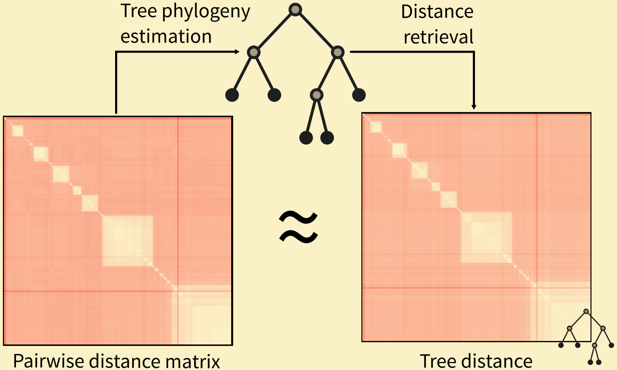
Tree decomposition of distance matrices

Challenge. Real genomes collections doesn't follow the infinite sites model



Tree decomposition of distance matrices

Challenge. Real genomes collections doesn’t follow the infinite sites model

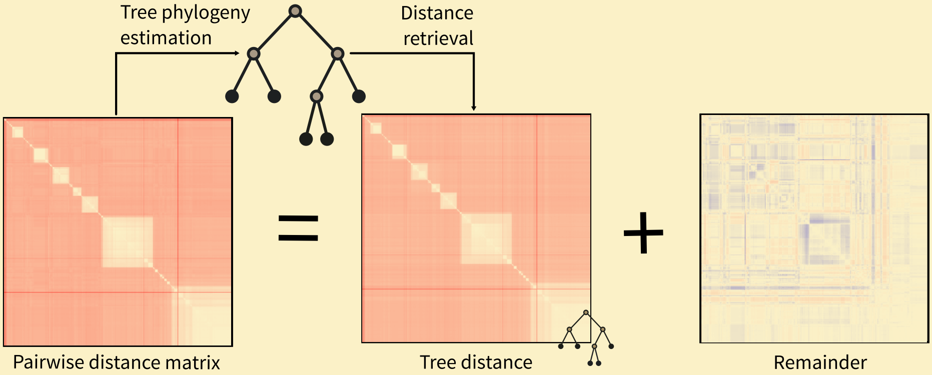


Pairwise distance matrix

Tree distance

Tree decomposition of distance matrices

Challenge. Real genomes collections doesn't follow the infinite sites model



Tree phylogeny estimation

Distance retrieval

Pairwise distance matrix

=

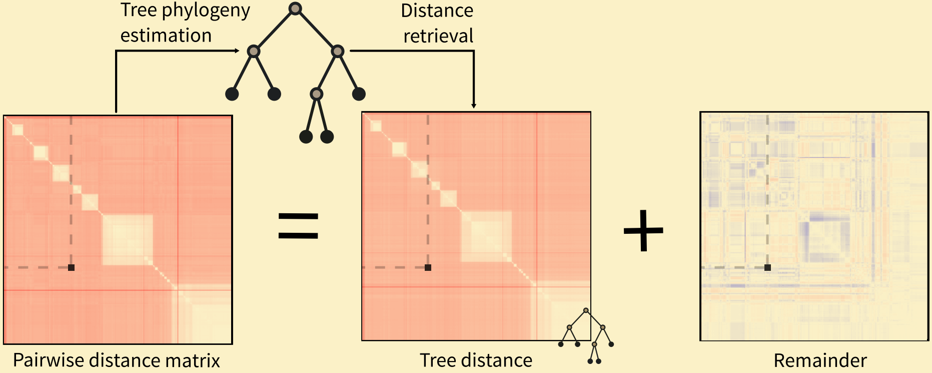
Tree distance

+

Remainder

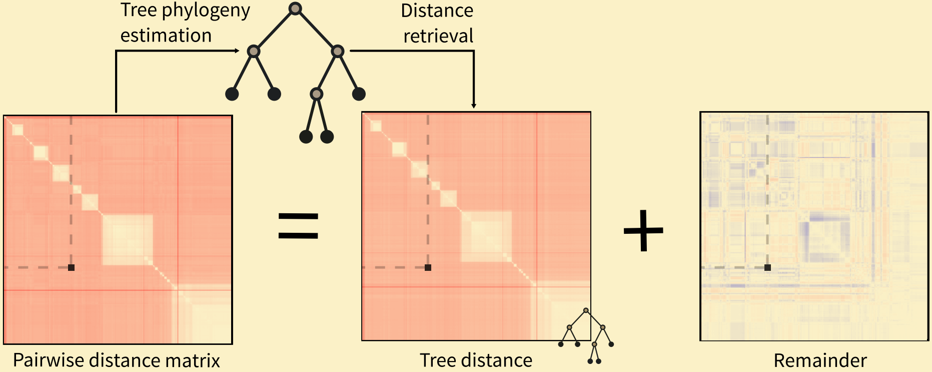
Tree decomposition of distance matrices

Challenge. Real genomes collections doesn't follow the infinite sites model



Tree decomposition of distance matrices

Challenge. Real genomes collections doesn't follow the infinite sites model



- ▶ The tree distance can be stored in linear space while providing $O(1)$ random access
- ▶ **Problem.** How to store the remainder?

A phylogenetic distance estimator: Mash[1]

Mash distance computation

[1] *Ondov et. al.* Genome Biology, 2016. [2] *Broder.* Compression and complexity of sequences, 1997.

A phylogenetic distance estimator: Mash[1]

Mash distance computation

1. compute the k-mer sets K_A and K_B of the genomes A and B

[1] *Ondov et. al.* Genome Biology, 2016. [2] *Broder.* Compression and complexity of sequences, 1997.

A phylogenetic distance estimator: Mash[1]

Mash distance computation

1. compute the k-mer sets K_A and K_B of the genomes A and B
2. get an (unbiased) **estimate** $\widehat{j}$ of the **Jaccard index** $j = \frac{|K_A \cap K_B|}{|K_A \cup K_B|}$, using **MinHash**[2]
   - Much faster than computing the Jaccard distance extensively

[1] *Ondov et. al.* Genome Biology, 2016. [2] *Broder.* Compression and complexity of sequences, 1997.

A phylogenetic distance estimator: Mash[1]

Mash distance computation

1. compute the k-mer sets K_A and K_B of the genomes A and B
2. get an (unbiased) **estimate** $\widehat{j}$ of the **Jaccard index** $j = \frac{|K_A \cap K_B|}{|K_A \cup K_B|}$, using **MinHash**[2]
   - ▶ Much faster than computing the Jaccard distance extensively
3. convert it into the evolutionary distance $d(A, B) = -1/k \cdot \log\left(\frac{2 \cdot \widehat{j}}{\widehat{j}+1}\right)$
   - ▶ Morally, $d(A, B)$ is the SNP evolution rate mapping K_A to K_B in one epoch

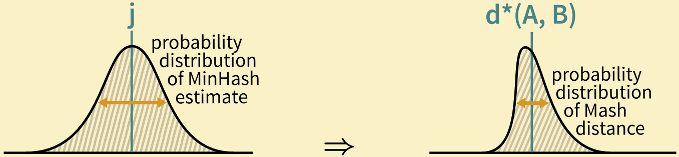
[1] *Ondov et. al.* Genome Biology, 2016. [2] *Broder.* Compression and complexity of sequences, 1997.

A phylogenetic distance estimator: Mash[1]

Mash distance computation

1. compute the k-mer sets K_A and K_B of the genomes A and B
2. get an (unbiased) **estimate** $\widehat{j}$ of the **Jaccard index** $j = \frac{|K_A \cap K_B|}{|K_A \cup K_B|}$, using **MinHash**[2]
   - ▶ Much faster than computing the Jaccard distance extensively
3. convert it into the evolutionary distance $d(A,B) = -1/k \cdot \log\left(\frac{2 \cdot \widehat{j}}{\widehat{j}+1}\right)$
   - ▶ Morally, $d(A,B)$ is the SNP evolution rate mapping K_A to K_B in one epoch

Lemma. Mash is an estimator of $d^*(A,B) = -1/k \cdot \log\left(\frac{2 \cdot j}{j+1}\right)$, hence can be associated to a standard error



j

probability distribution of MinHash estimate

⇒

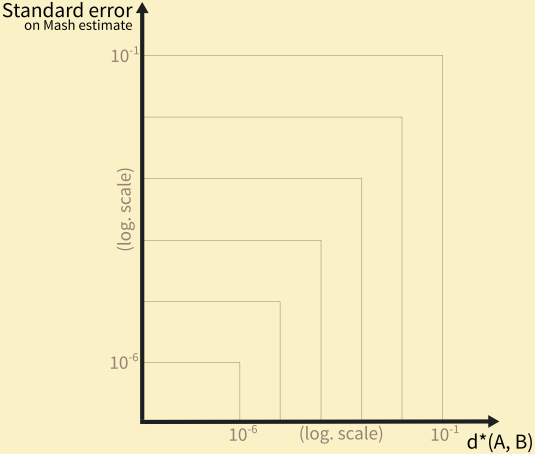
d*(A, B)

probability distribution of Mash distance

[1] *Ondov et. al.* Genome Biology, 2016. [2] *Broder.* Compression and complexity of sequences, 1997.

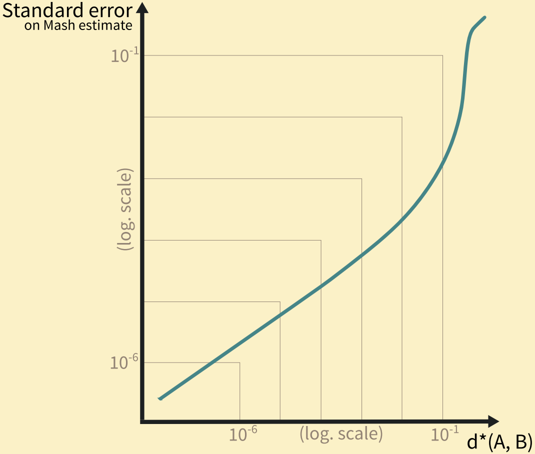
Synchronising float and Mash precision

For fixed Mash parameters *k* and *s*,



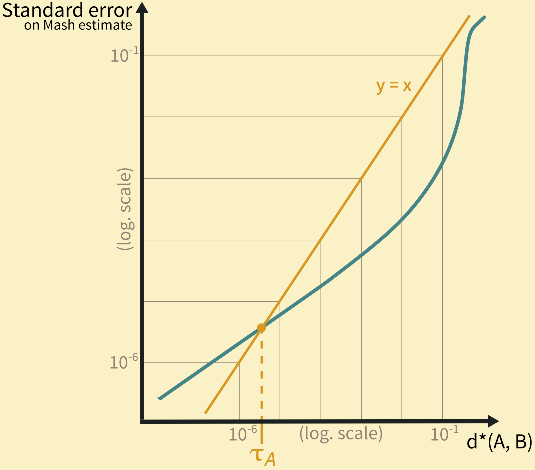
Synchronising float and Mash precision

For fixed Mash parameters *k* and *s*,



Synchronising float and Mash precision

For fixed Mash parameters k and s,

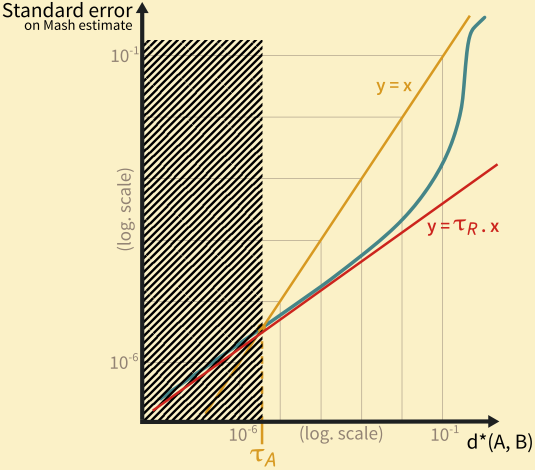


Absolute error. If $d^*(A, B) \leq \tau_A$, the biological signal is completely masked by the standard error of the estimator

- Any signal smaller than τ_A can be ignored

Synchronising float and Mash precision

For fixed Mash parameters *k* and *s*,



Absolute error. If $d^*(A, B) \leq \tau_A$, the biological signal is completely masked by the standard error of the estimator

- Any signal smaller than τ_A can be ignored

Relative error. For any d^*, the relative error made by the estimator is bigger than τ_R

- Relative errors smaller than τ_R do not perturb the signal d^*

Trick 1. “Lossless” truncation and quantization of floats

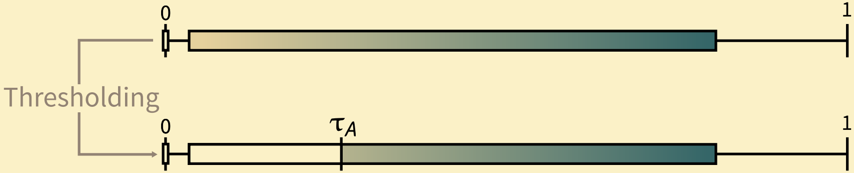
(1) Any signal smaller than τ_A can be ignored **(2)** Relative errors smaller than τ_R do not perturb the signal d^*

Trick 1. “Lossless” truncation and quantization of floats

(1) Any signal smaller than τ_A can be ignored **(2)** Relative errors smaller than τ_R do not perturb the signal d^*

Trick 1. "Lossless" truncation and quantization of floats

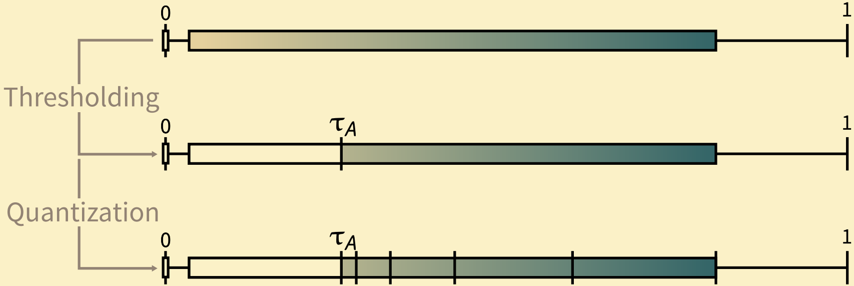
(1) Any signal smaller than τ_A can be ignored **(2)** Relative errors smaller than τ_R do not perturb the signal d^*



Thresholding. Map all values smaller than τ_A to 0

Trick 1. "Lossless" truncation and quantization of floats

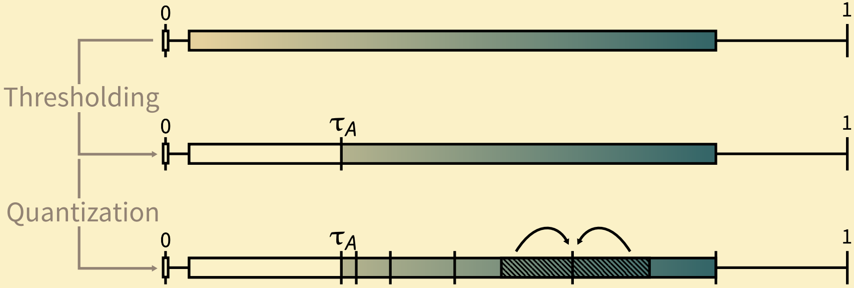
(1) Any signal smaller than τ_A can be ignored **(2)** Relative errors smaller than τ_R do not perturb the signal d^*



Thresholding. Map all values smaller than τ_A to 0

Trick 1. "Lossless" truncation and quantization of floats

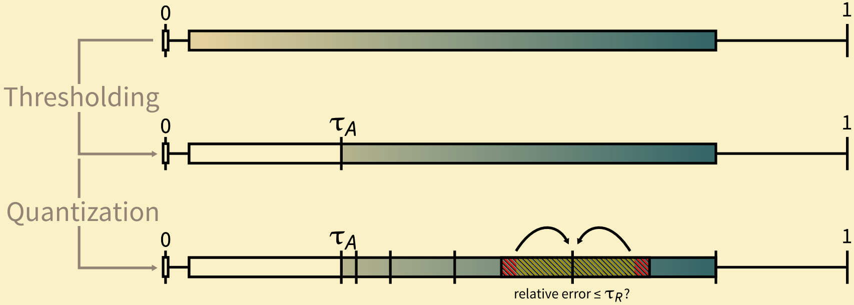
(1) Any signal smaller than τ_A can be ignored **(2)** Relative errors smaller than τ_R do not perturb the signal d^*



Thresholding. Map all values smaller than τ_A to 0

Trick 1. “Lossless” truncation and quantization of floats

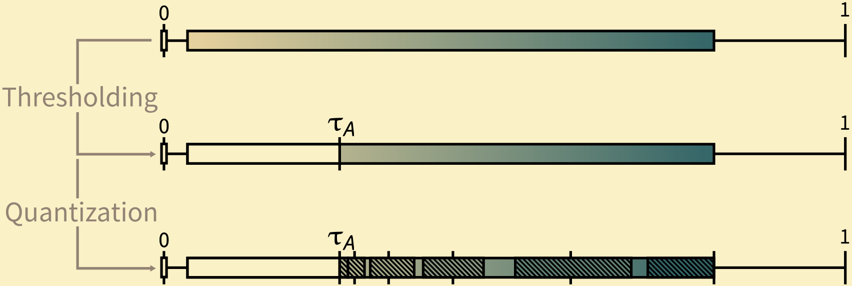
(1) Any signal smaller than τ_A can be ignored **(2)** Relative errors smaller than τ_R do not perturb the signal d^*



Thresholding. Map all values smaller than τ_A to 0

Trick 1. “Lossless” truncation and quantization of floats

(1) Any signal smaller than τ_A can be ignored **(2)** Relative errors smaller than τ_R do not perturb the signal d^*

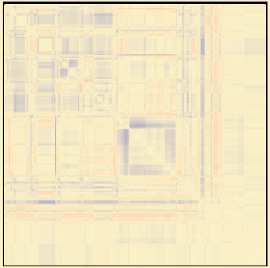


Thresholding. Map all values smaller than τ_A to 0

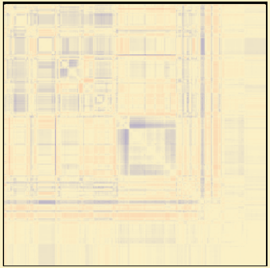
Quantization. Map x to $\text{repr}(x)$ if the induced relative error is smaller than τ_R

- Only store $\text{index}(\text{repr}(x)) \in \mathbb{N}$ to **gain space**
- **Tradeoff** between the size of non-quantized intervals and the size of indexes to store

Trick 2. A *the-smaller-the-lighter* float format



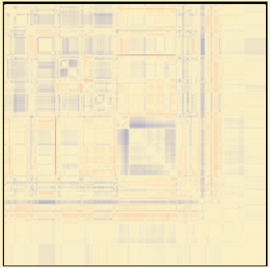
Trick 2. A *the-smaller-the-lighter* float format



Observation. The **values** of the remainder **are much smaller** than in the original distance matrix

- About 2 orders of magnitude smaller

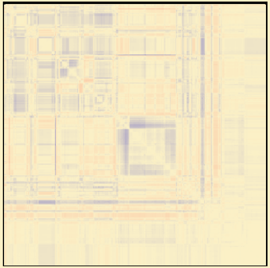
Trick 2. A *the-smaller-the-lighter* float format



Observation. The **values** of the remainder **are much smaller** than in the original distance matrix

- ▶ About 2 orders of magnitude smaller
- ▶ Combined with absolute thresholding, fewer non-zero digits

Trick 2. A *the-smaller-the-lighter* float format



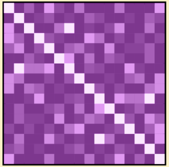
Observation. The **values** of the remainder **are much smaller** than in the original distance matrix

- ▶ About 2 orders of magnitude smaller
- ▶ Combined with absolute thresholding, fewer non-zero digits

$$\Downarrow$$

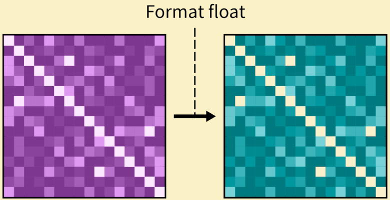
$$\underbrace{\texttt{-0.005699721}}_{\text{12 chars}} \xrightarrow{\tau_A\text{-thresholding}} \underbrace{\texttt{-0.005600}}_{\text{9 chars}} \xrightarrow{\text{scientific notation}} \underbrace{\texttt{-56-3}}_{\text{5 chars}}$$

METHOD. "Lossless" compression of pairwise genome distances



Mash distance matrix

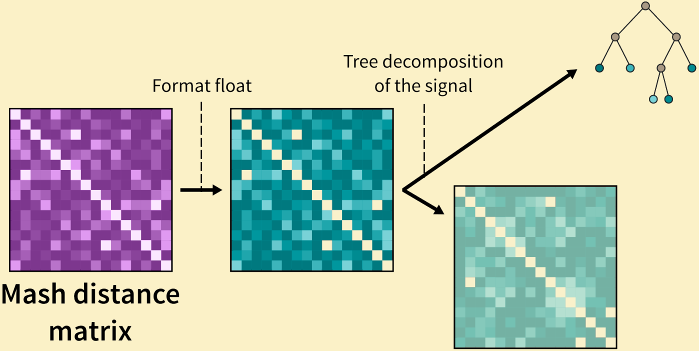
METHOD. "Lossless" compression of pairwise genome distances



Format float

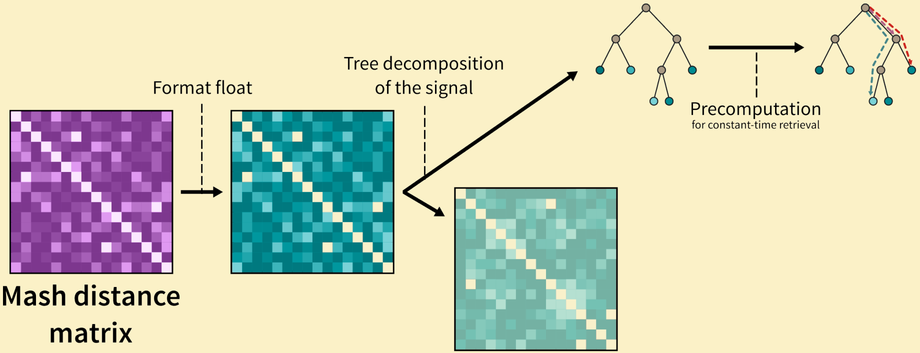
Mash distance matrix

METHOD. "Lossless" compression of pairwise genome distances

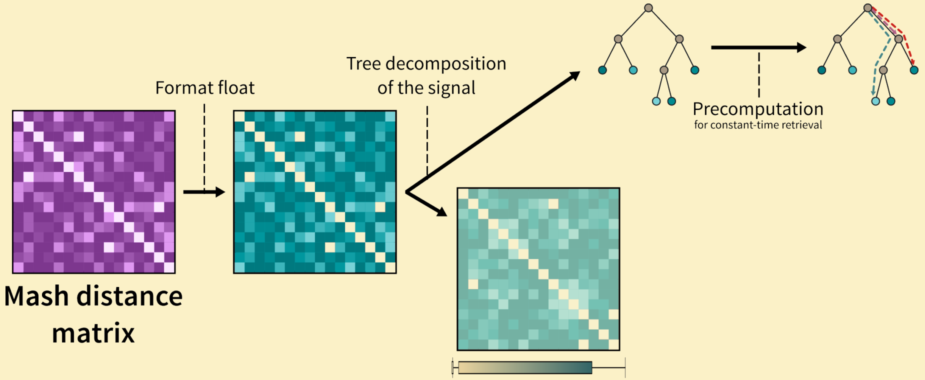


Mash distance matrix

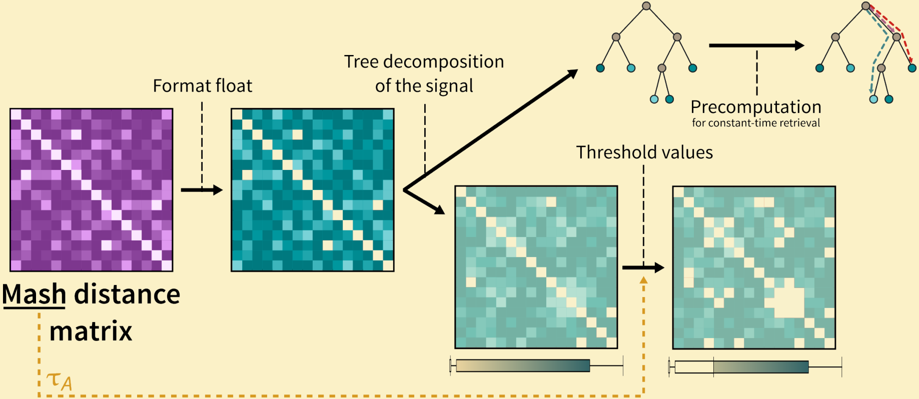
METHOD. “Lossless” compression of pairwise genome distances



METHOD. "Lossless" compression of pairwise genome distances



METHOD. "Lossless" compression of pairwise genome distances



Format float

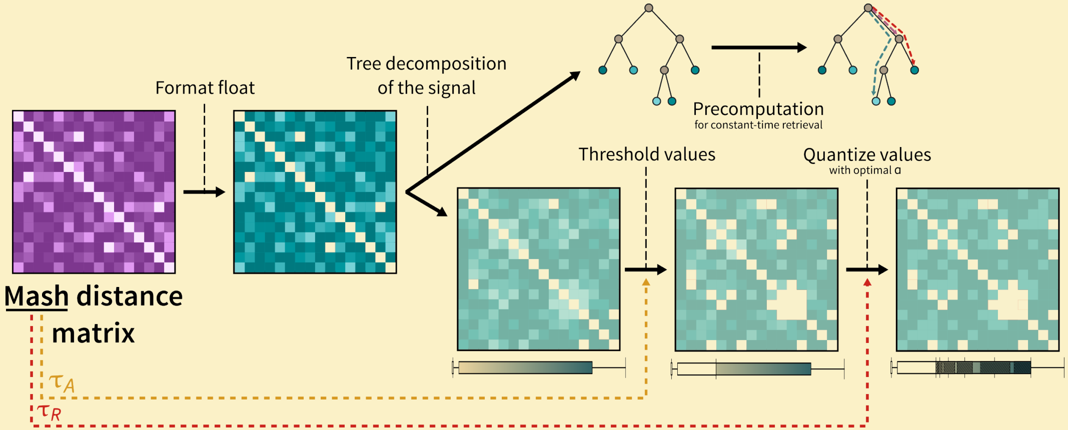
Tree decomposition of the signal

Precomputation
for constant-time retrieval

Threshold values

Mash distance matrix

METHOD. “Lossless” compression of pairwise genome distances



Format float

Tree decomposition of the signal

Precomputation
for constant-time retrieval

Threshold values

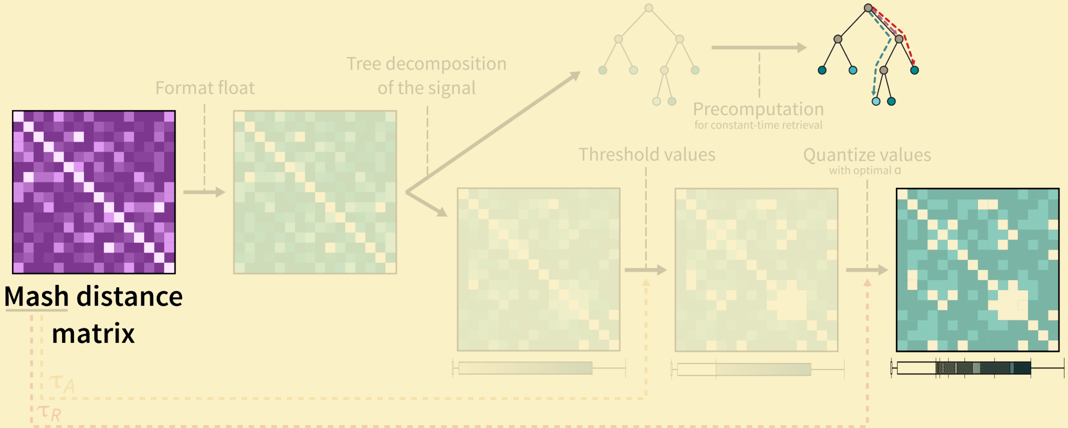
Quantize values
with optimal α

Mash distance matrix

τ_A

τ_R

METHOD. "Lossless" compression of pairwise genome distances



Trick 3. Lossy biology-informed thresholding

For similar enough genomes, **taxonomy can be defined with distance thresholds**[1]

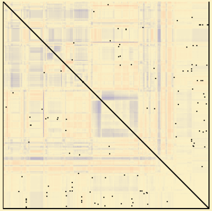
- e.g., Species ≡ >90% ANI ≡ <0.05 Mash distance
  Strain ≡ >99.99% ANI ≡ <0.0001 Mash distance

[1] *Rodriguez et. al.* mBio, 2024

Trick 3. Lossy biology-informed thresholding

For similar enough genomes, **taxonomy can be defined with distance thresholds**[1]

- e.g., Species $\equiv$ >90% ANI $\equiv$ <0.05 Mash distance
  Strain $\equiv$ >99.99% ANI $\equiv$ <0.0001 Mash distance



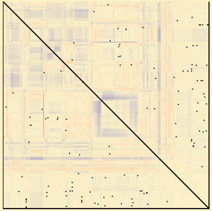
No threshold

[1] *Rodriguez et. al.* mBio, 2024

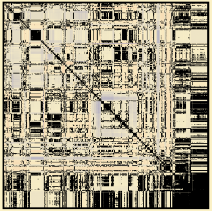
Trick 3. Lossy biology-informed thresholding

For similar enough genomes, **taxonomy can be defined with distance thresholds**[1]

- e.g., Species ≡ >90% ANI ≡ <0.05 Mash distance
  Strain ≡ >99.99% ANI ≡ <0.0001 Mash distance



No threshold



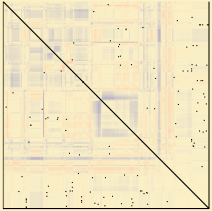
Beyond-strain threshold (<10^{-4})

[1] *Rodriguez et. al.* mBio, 2024

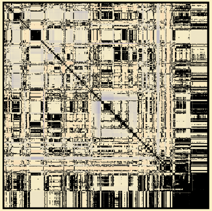
Trick 3. Lossy biology-informed thresholding

For similar enough genomes, **taxonomy can be defined with distance thresholds**[1]

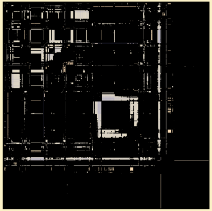
- e.g., Species $\equiv$ >90% ANI $\equiv$ <0.05 Mash distance
  Strain $\equiv$ >99.99% ANI $\equiv$ <0.0001 Mash distance



No threshold



Beyond-strain threshold ($<10^{-4}$)



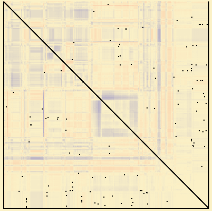
Beyond-genomovar threshold ($<5.10^{-3}$)

[1] *Rodriguez et. al.* mBio, 2024

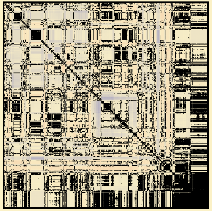
Trick 3. Lossy biology-informed thresholding

For similar enough genomes, **taxonomy can be defined with distance thresholds**[1]

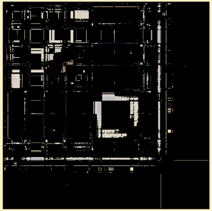
- e.g., Species ≡ >90% ANI ≡ <0.05 Mash distance
  Strain ≡ >99.99% ANI ≡ <0.0001 Mash distance



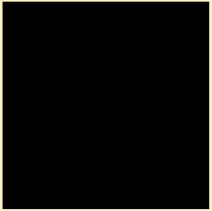
No threshold



Beyond-strain threshold ($<10^{-4}$)



Beyond-genomovar threshold ($<5.10^{-3}$)



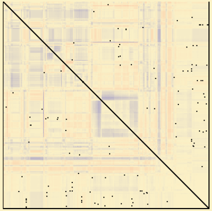
Beyond-species threshold ($<10^{-2}$)

[1] *Rodriguez et. al.* mBio, 2024

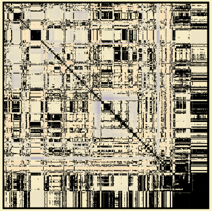
Trick 3. Lossy biology-informed thresholding

For similar enough genomes, **taxonomy can be defined with distance thresholds**[1]

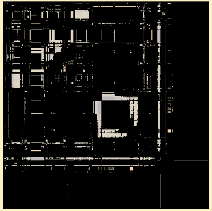
- e.g., Species $\equiv$ >90% ANI $\equiv$ <0.05 Mash distance
  Strain $\equiv$ >99.99% ANI $\equiv$ <0.0001 Mash distance



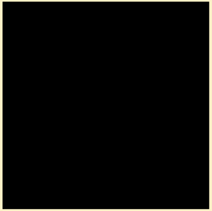
No threshold



Beyond-strain threshold ($<10^{-4}$)



Beyond-genomovar threshold ($<5.10^{-3}$)

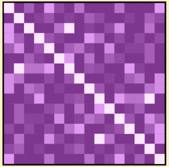


Beyond-species threshold ($<10^{-2}$)

Storing sparse matrices. Matrices can be represented in $O(\#(\text{non-zero-entries}))$ space

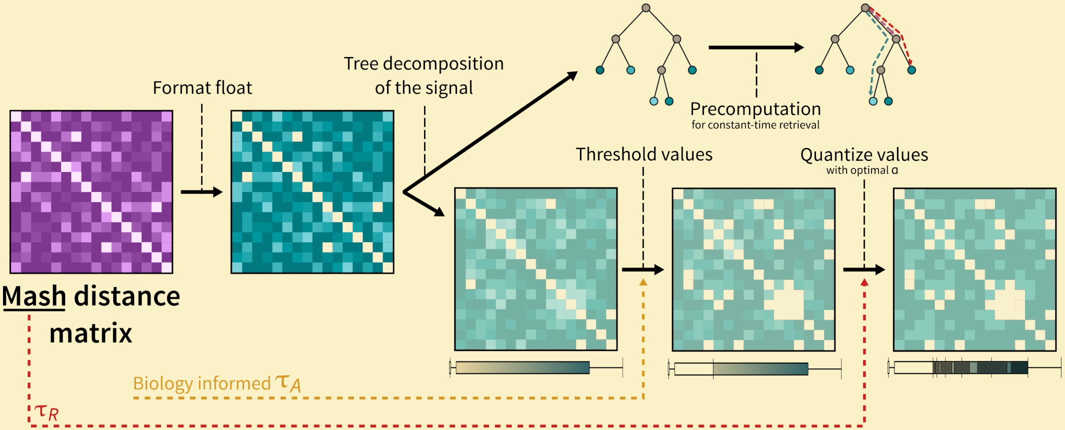
[1] *Rodriguez et. al.* mBio, 2024

METHOD. Lossy compression of pairwise genome distances



Mash distance matrix

METHOD. Lossy compression of pairwise genome distances



Format float

Tree decomposition of the signal

Precomputation for constant-time retrieval

Threshold values

Quantize values with optimal α

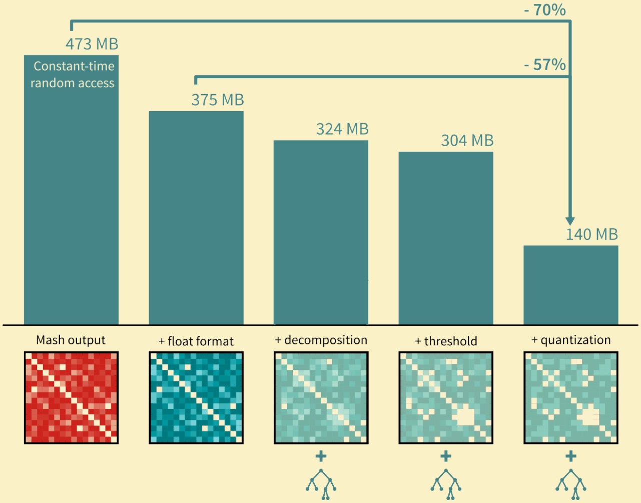
Mash distance matrix

Biology informed τ_A

τ_R

Results

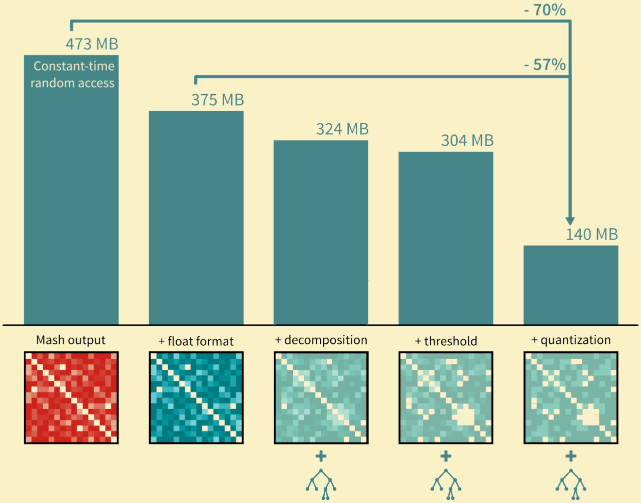
“Lossless” compression of pairwise distance matrices



Data.
10k *Streptococcus pneumoniae* genomes from the 661k collection[1].
Distances estimated using `Mash`[2] with $k = 21, s = 10^4$, which gives $\tau_A = 10^{-6}, \tau_R = 10^{-2}$

[1] *Blackwell et. al.* PLOS Biology, 2021. [2] *Ondov et. al.* Genome Biology, 2016. [3] *Shaw et. al.* Nature Methods, 2023. [4] *Baker et. al.* Genome Biology, 2019.
[5] `https://gitlab.inria.fr/lackerma/nwk2phy`

“Lossless” compression of pairwise distance matrices



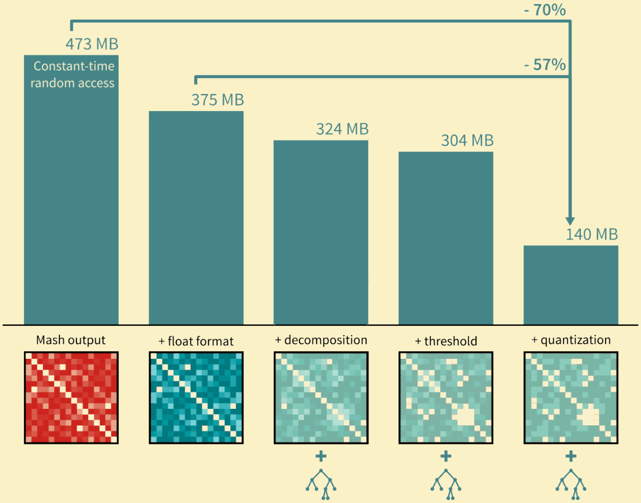
Data.
10k *Streptococcus pneumoniae* genomes from the 661k collection[1].
Distances estimated using `Mash`[2] with $k = 21, s = 10^4$, which gives $\tau_A = 10^{-6}, \tau_R = 10^{-2}$

We observed similar results on other species
- *10k Neisseria gonorrhoeae*
- *10k Escherichia coli*

and with other distance estimators
- *Skani*[3]
- *Dashing*[4]

[1] *Blackwell et. al.* PLOS Biology, 2021. [2] *Ondov et. al.* Genome Biology, 2016. [3] *Shaw et. al.* Nature Methods, 2023. [4] *Baker et. al.* Genome Biology, 2019.
[5] `https://gitlab.inria.fr/lackerma/nwk2phy`

“Lossless” compression of pairwise distance matrices



Data.
10k *Streptococcus pneumoniae* genomes from the 661k collection[1].
Distances estimated using `Mash`[2] with $k = 21, s = 10^4$, which gives $\tau_A = 10^{-6}, \tau_R = 10^{-2}$

We observed similar results on other species
- *10k Neisseria gonorrhoeae*
- *10k Escherichia coli*

and with other distance estimators
- *Skani*[3]
- *Dashing*[4]

Software.
The whole pipeline is implemeted in the (prototype) tool `phdcomp`
Several components are of independent interest (eg. `nwk2phy`[5])

[1] *Blackwell et. al.* PLOS Biology, 2021. [2] *Ondov et. al.* Genome Biology, 2016. [3] *Shaw et. al.* Nature Methods, 2023. [4] *Baker et. al.* Genome Biology, 2019.
[5] `https://gitlab.inria.fr/lackerma/nwk2phy`

“ Conclusion

Conclusion

Context. Many **downstream analyses** rely on **pairwise distance matrices**, that are already challenging to store due to their **quadratic size**

Approach. We aim to leverage the **specific structure** of genomic data, that can extensively be **explained by the underlying phylogeny**

First results.

- **Theory.** Pairwise matrices of genome collections following the *infinite sites model* can be stored in **linear space** supporting **constant-time** queries
- **Practice.** Lossless compression of *10k s.-pneumo.* pairwise matrices with constant-time random access **saves around 70% space**

What's next? Generalization to many-species collections, and larger scale experiments

- This is where we expect the subquadraticity to arise

Thank you for your attention!

Léo Ackermann
Pierre Peterlongo
Karel Břinda

Towards subquadratic

data structures for large genome-distance matrices with quick retrieval