Bin Packing and Load Balancing for Efficient Compression of Large Bacterial Genomes Collection

Tam TRUONG, Dominique LAVENIER, Pierre PETERLONGO, Karel BRINDA

IRISA

Inria

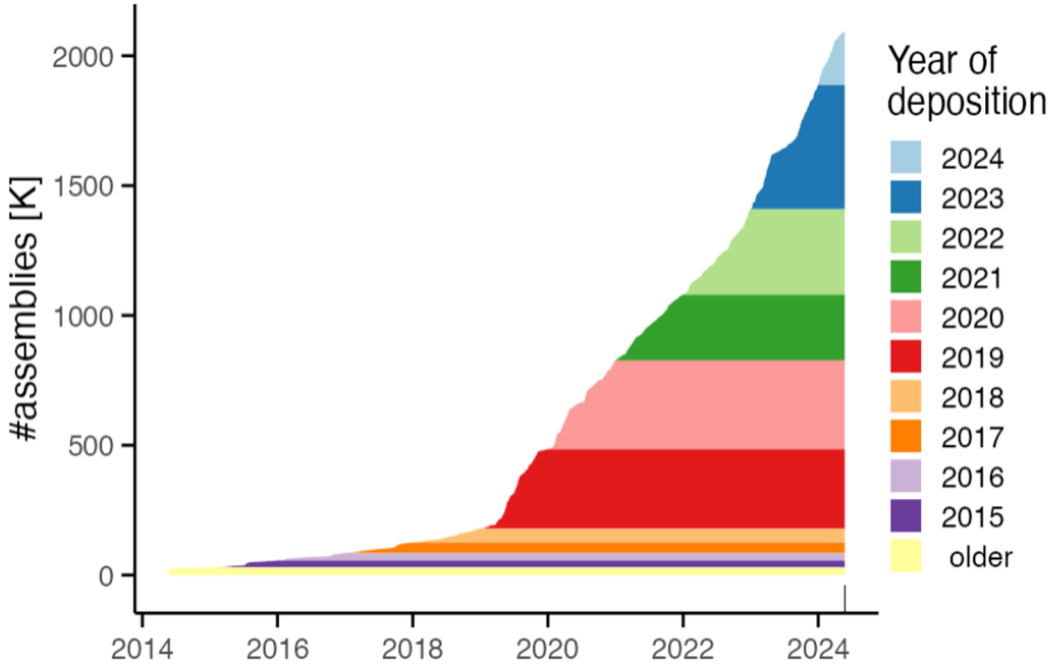
GENSCALE

Université de Rennes

Introduction

Motivation: Rapidly Growing Bacterial Genome Data

Fast growth of bacterial genomes data[1]



Increasing Availability of Larger Bacterial Genome Collections

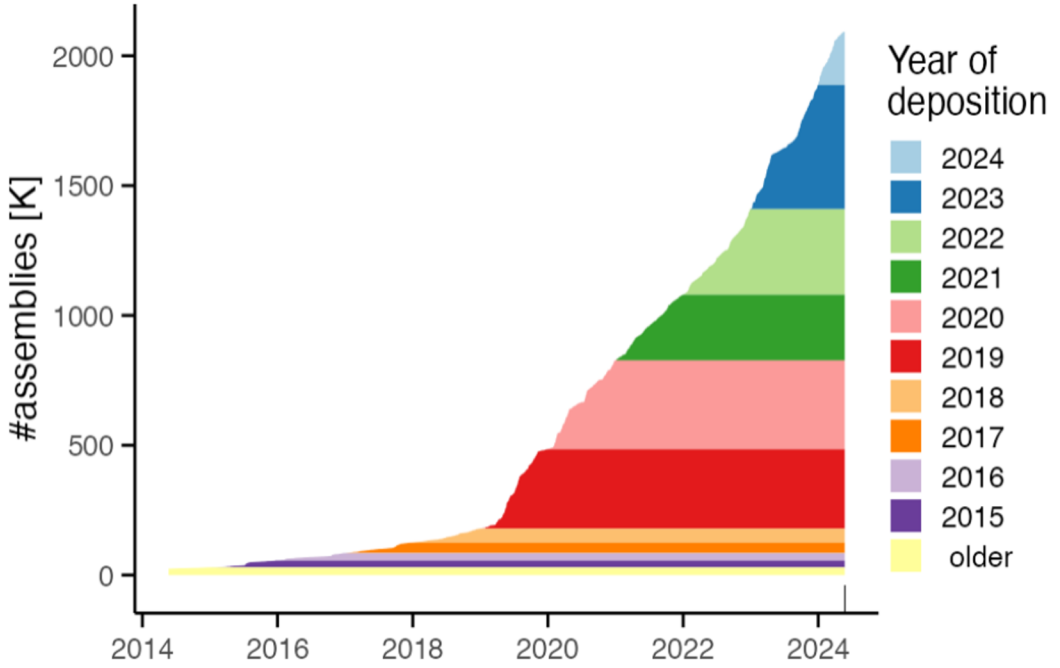
2021	- 661k Collection[2],	n = 661,405
03/2024	- AllTheBacteria[3] v0.1,	n = 1,932,812
11/2024	- AllTheBacteria[3] v0.2,	n = 2,440,377
Future	- Collections,	$n \geq 10^7$

Collections will have higher diversity, more metagenomes,…

[1] Břinda et al., Efficient and Robust Search of Microbial Genomes via Phylogenetic Compression. To be appeared in *Nature Methods*. 2025
[2] Blackwell et al., Exploring bacterial diversity via a curated and searchable snapshot of archived DNA sequences. *PLOS Biology* 19, 11. 2021
[3] Hunt et a.l,. AllTheBacteria - all bacterial genomes assembled, available and searchable. *bioRxiv*. 2024

Introduction: Rapidly Growing Bacterial Genome Data

Fast growth of bacterial genomes data[1]



Increasing Availability of Larger Bacterial Genome Collections

2021	- 661k Collection[2],	n = 661,405
03/2024	- AllTheBacteria[3] v0.1,	n = 1,932,812
11/2024	- AllTheBacteria[3] v0.2,	n = 2,440,377
Future	- Collections,	$n \geq 10^7$

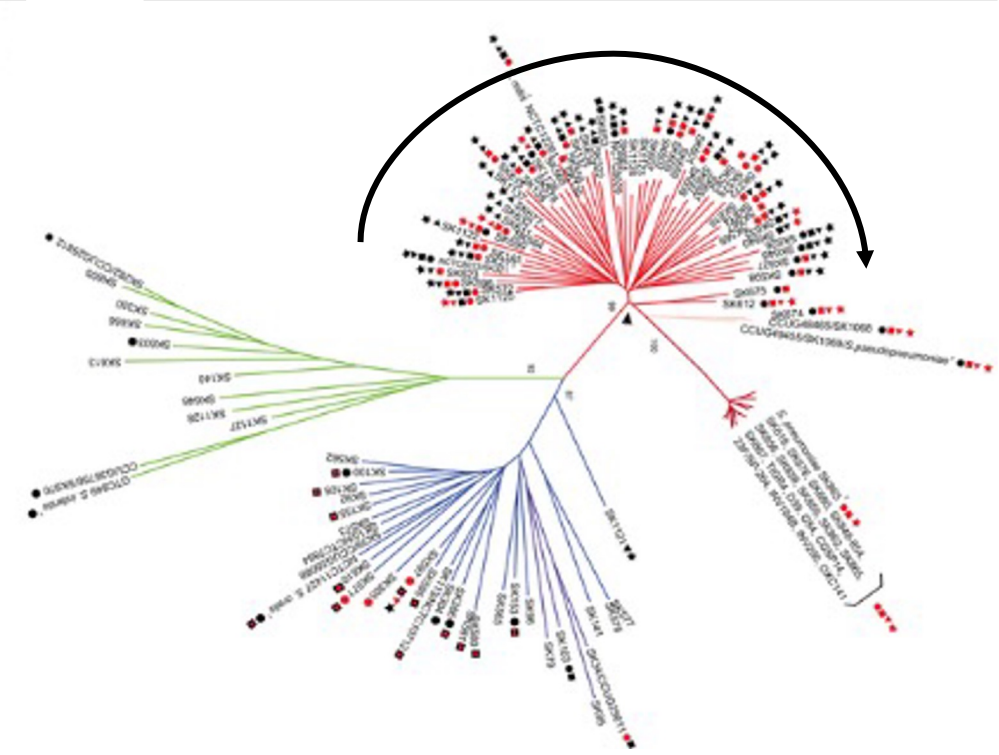
Collections will have higher diversity, more metagenomes,…

Goal: efficient compression and search within those collections

[1] Břinda et al., Efficient and Robust Search of Microbial Genomes via Phylogenetic Compression. To be appeared in *Nature Methods*. 2025
[2] Blackwell et al., Exploring bacterial diversity via a curated and searchable snapshot of archived DNA sequences. *PLOS Biology* 19, 11. 2021
[3] Hunt et a.l,. AllTheBacteria - all bacterial genomes assembled, available and searchable. *bioRxiv*. 2024

Recent Innovation: Phylogenetic Compression

Key Idea: improves compressibility via reordering according to the evolutionary history[1]

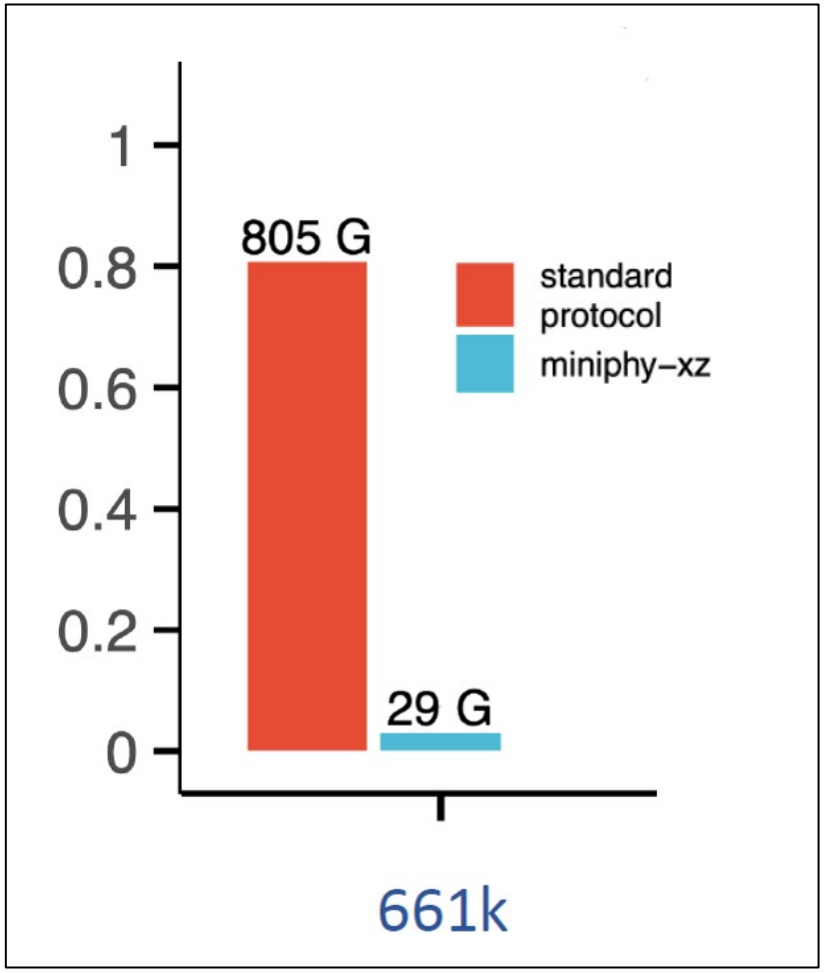


Individual genomes are **not highly compressible but collections of related genomes** are **extremely compressible**.[4]

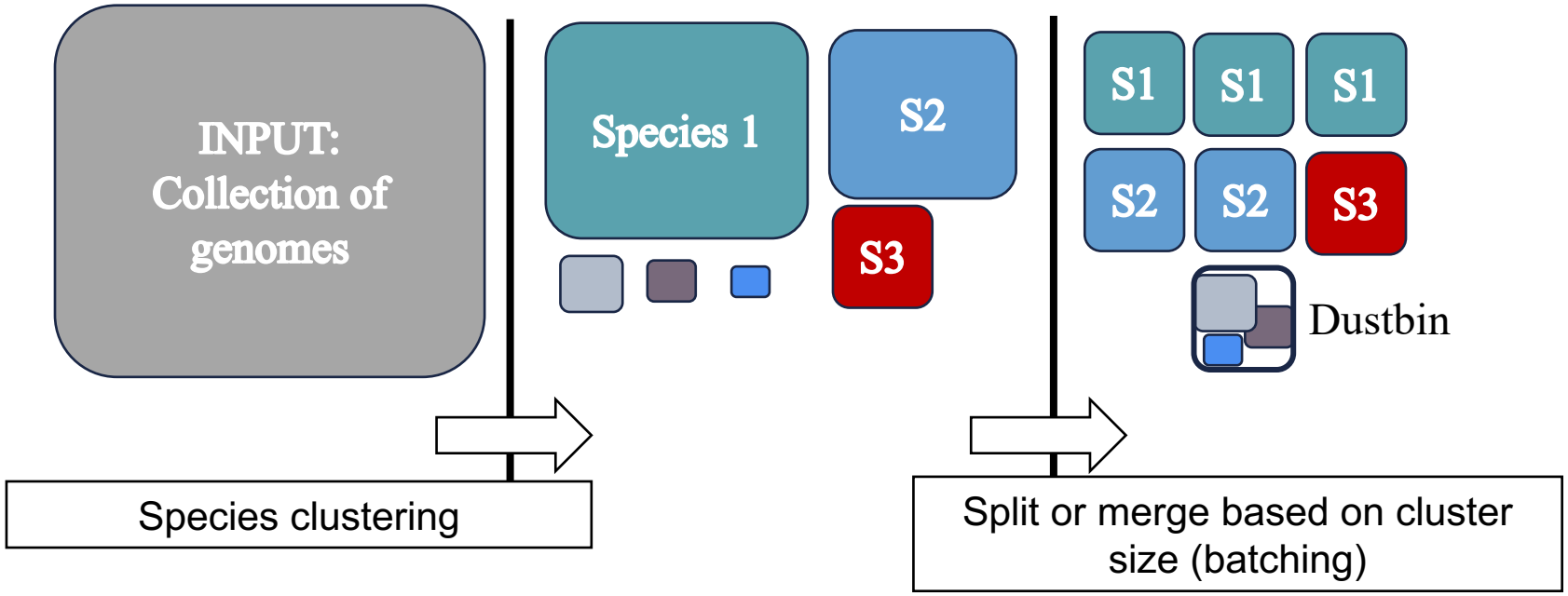
[1] Břinda et al., Efficient and Robust Search of Microbial Genomes via Phylogenetic Compression. To be appeared in *Nature Methods*. 2025
[4] Loh, PR., Baym, M. & Berger, B. Compressive genomics. *Nat Biotechnol.* 2012

Resulting Compression

Lossless compression of 1-3 orders of magnitude

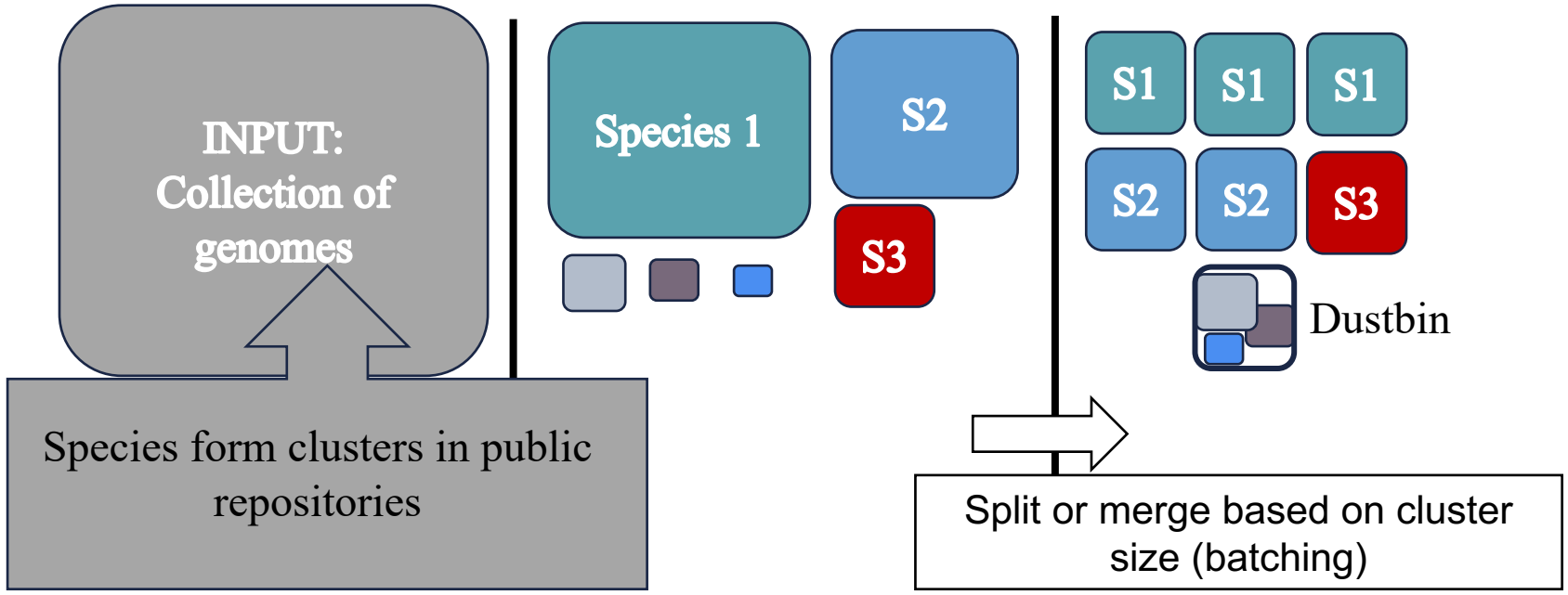


Batching of Genomes is a crucial step in the Phylogenetic Compression pipeline (MiniPhy*)



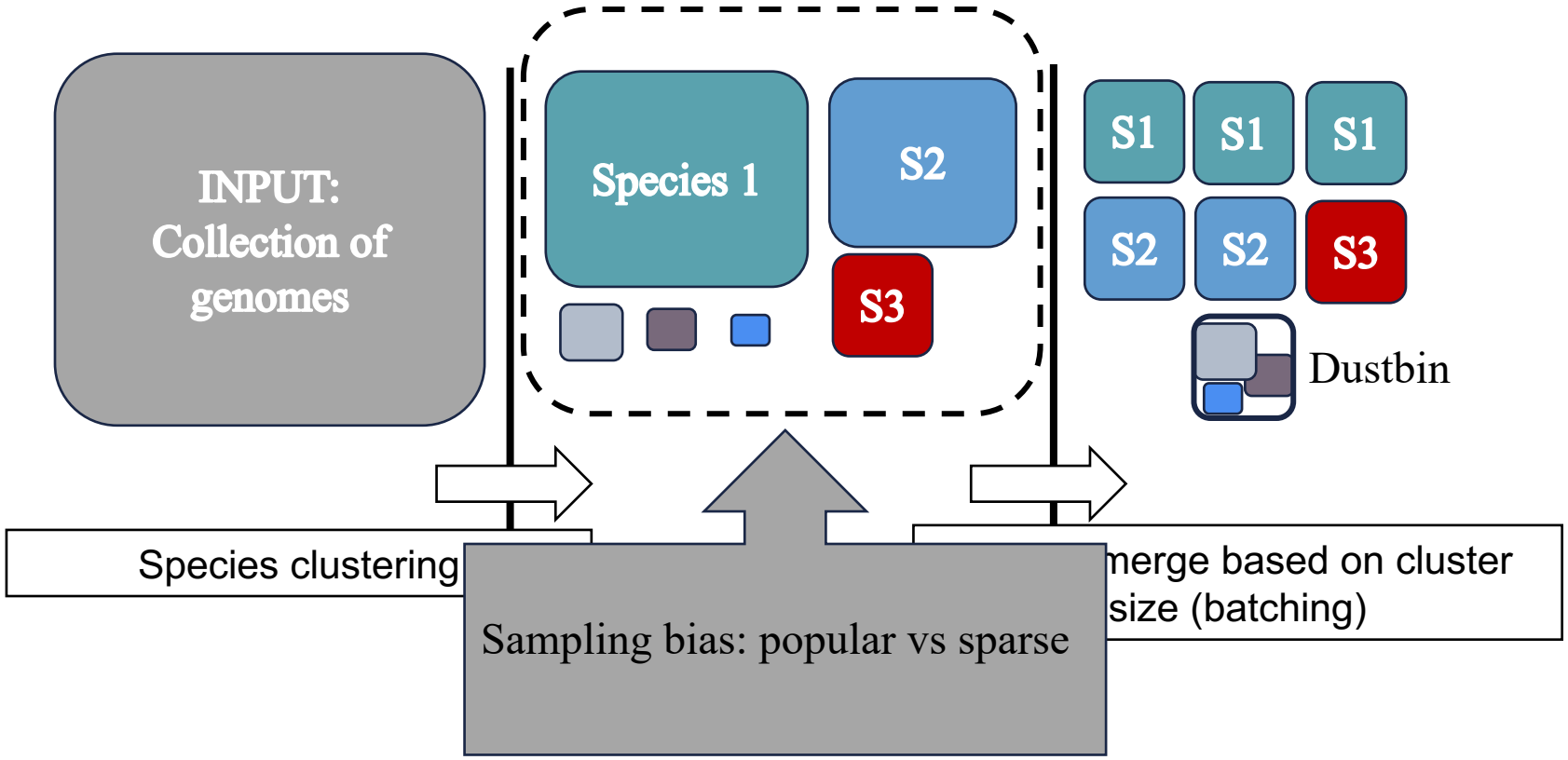
https://github.com/karel-brinda/miniphy

Batching of Genomes is a crucial step in the Phylogenetic Compression pipeline (MiniPhy*)



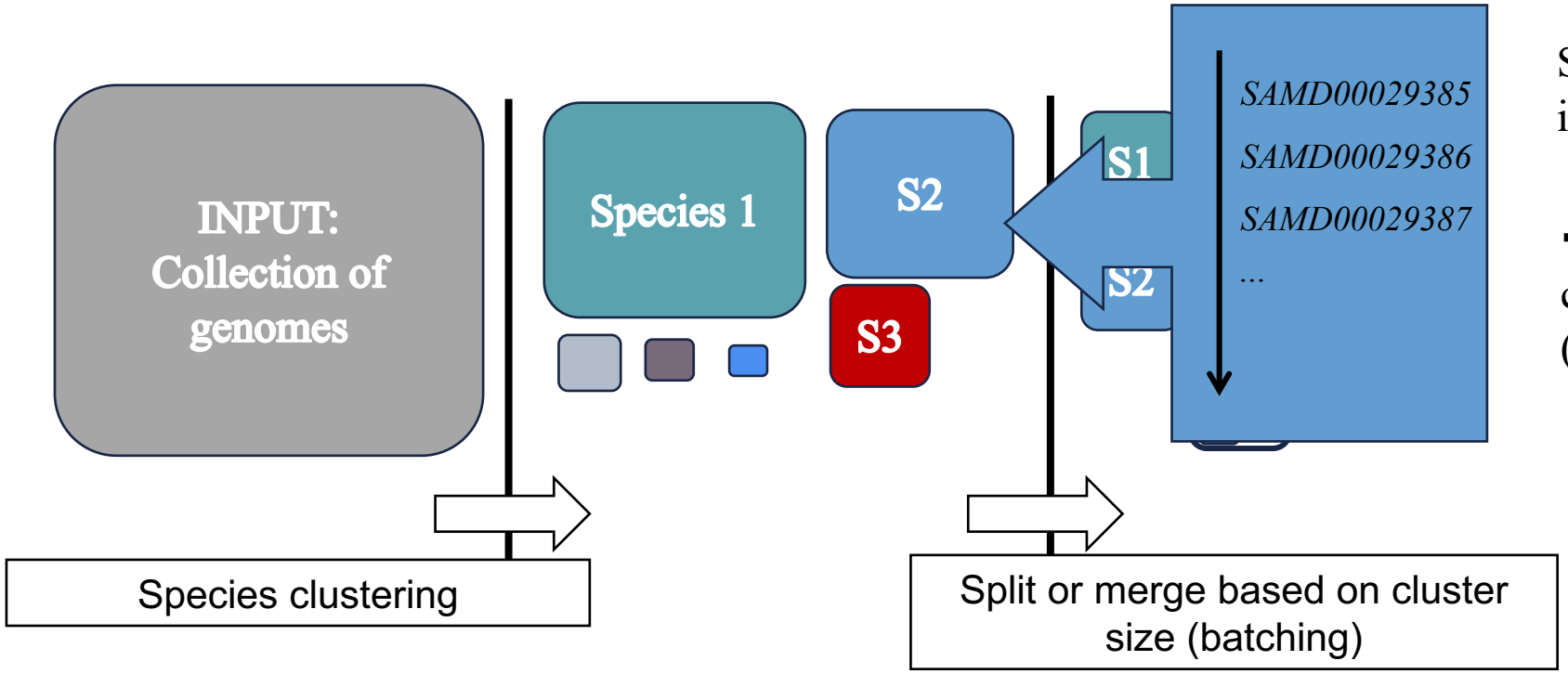
https://github.com/karel-brinda/miniphy

Batching of Genomes is a crucial step in the Phylogenetic Compression pipeline (MiniPhy*)



https://github.com/karel-brinda/miniphy

Batching of Genomes is a crucial step in the Phylogenetic Compression pipeline (MiniPhy*)

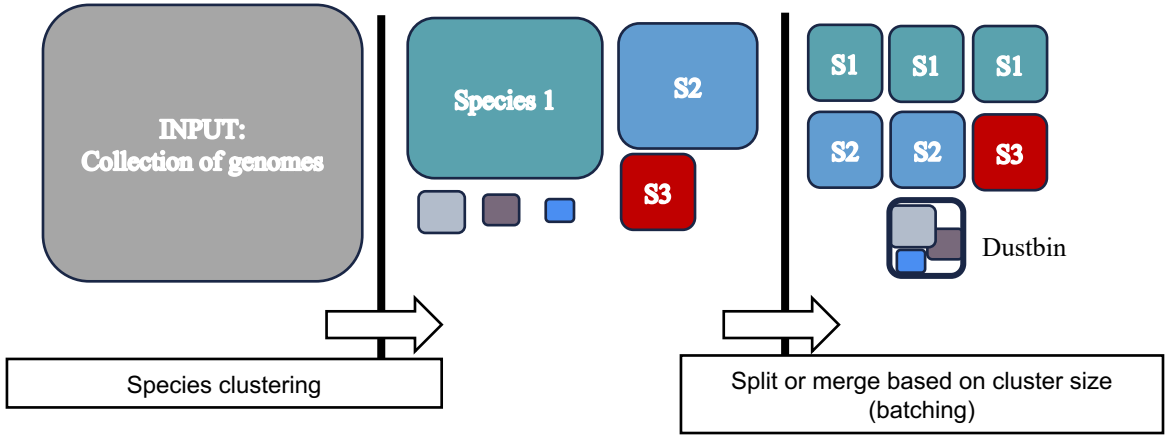


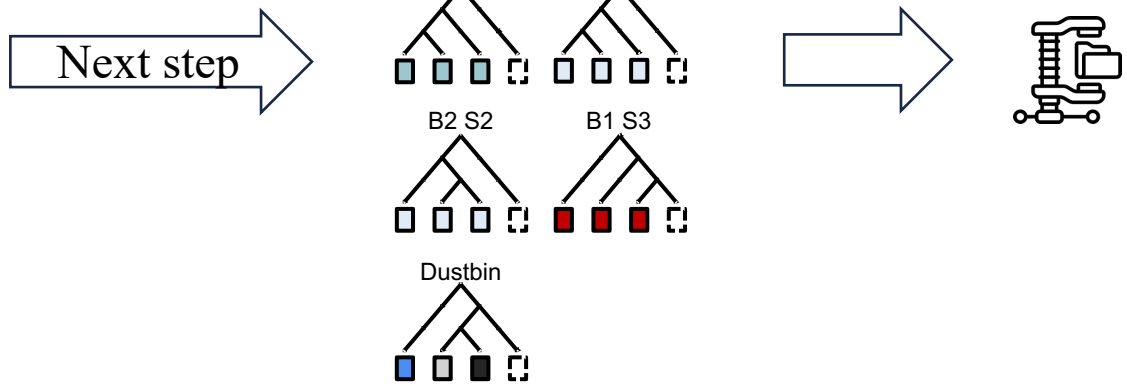
Sequencing data uploaded per individual projects

➔ Sort genome within each cluster by sequencing time (accession number)

https://github.com/karel-brinda/miniphy

Batching of Genomes is a crucial step in the Phylogenetic Compression pipeline (MiniPhy*)

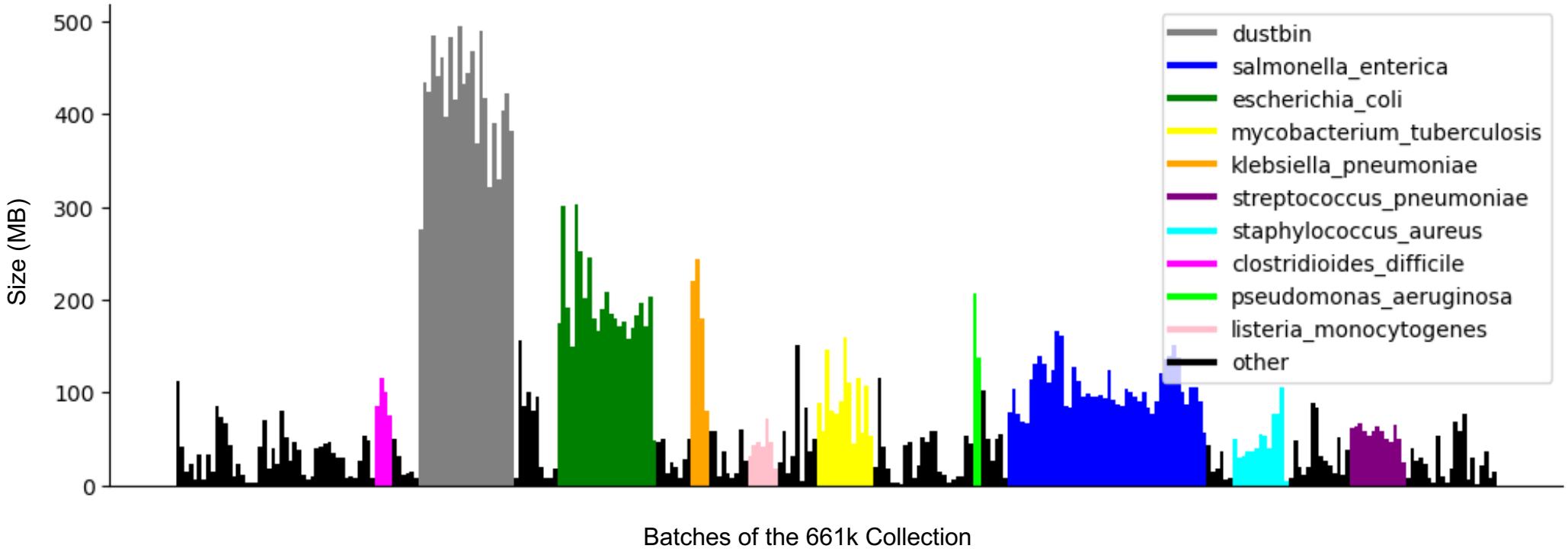




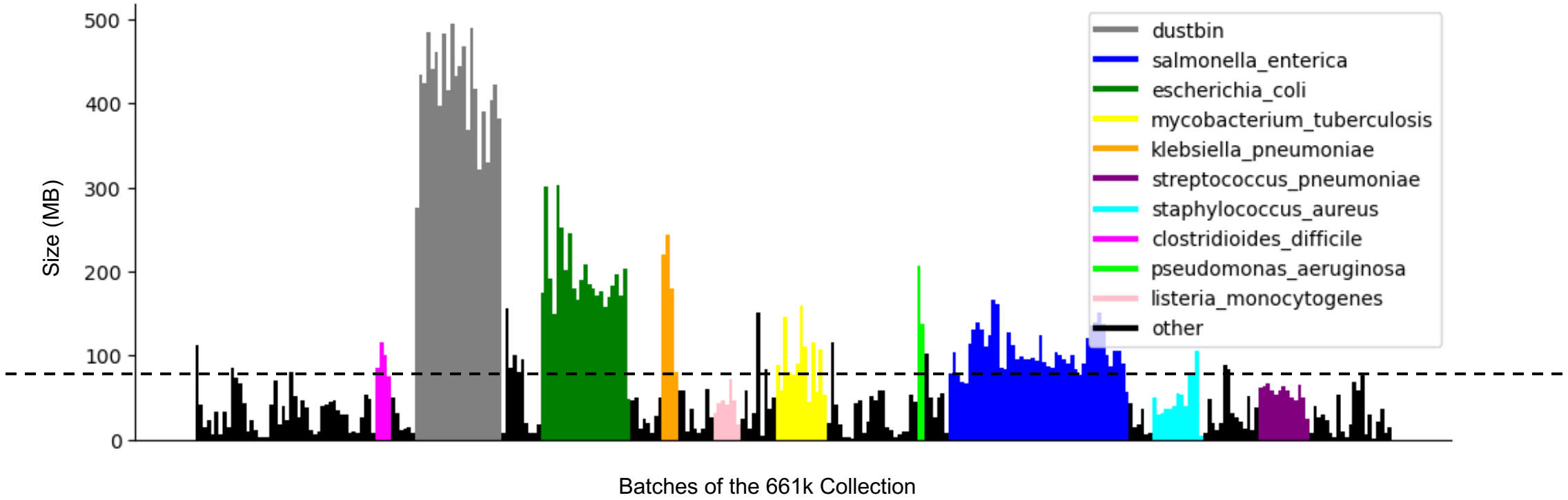
Reordering of genomes in each batch using an estimated evolutionary tree

https://github.com/karel-brinda/miniphy

Current Limitation: **Non-uniform** post-compression sizes



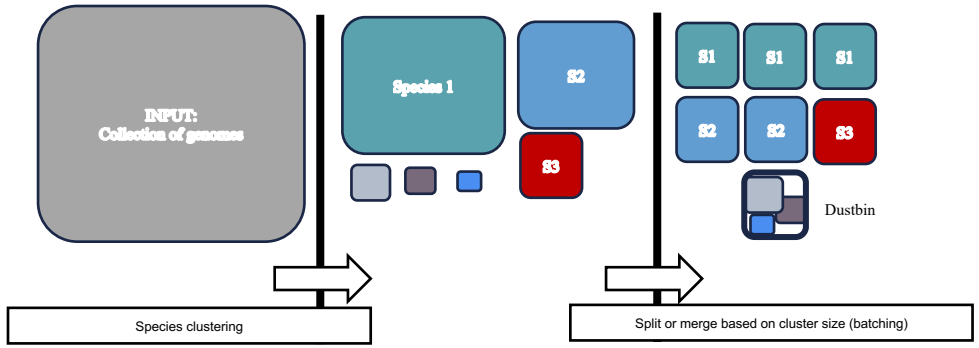
Current Limitation: **Non-uniform** post-compression sizes

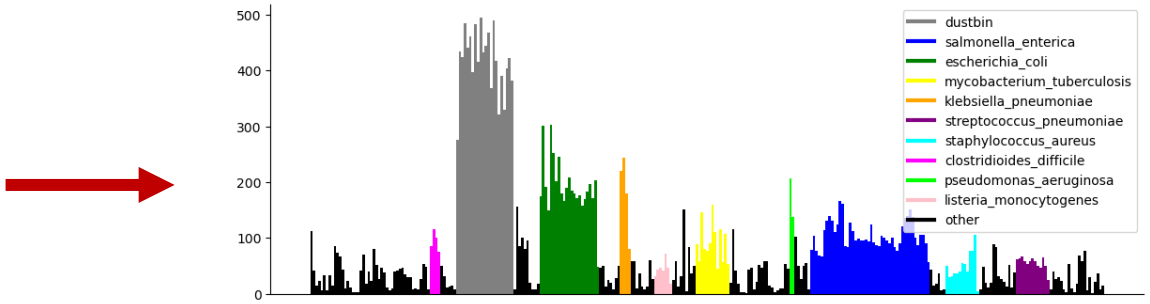


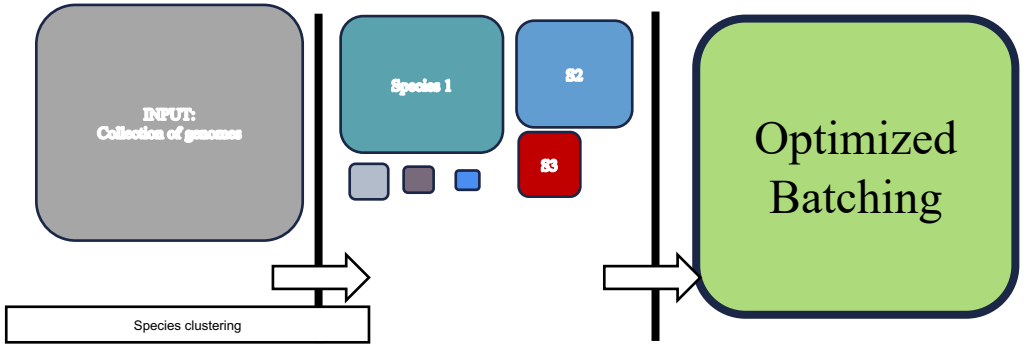
CONSEQUENCES: Unbalanced Workloads Hinder Parallelization Inconsistent Query Time

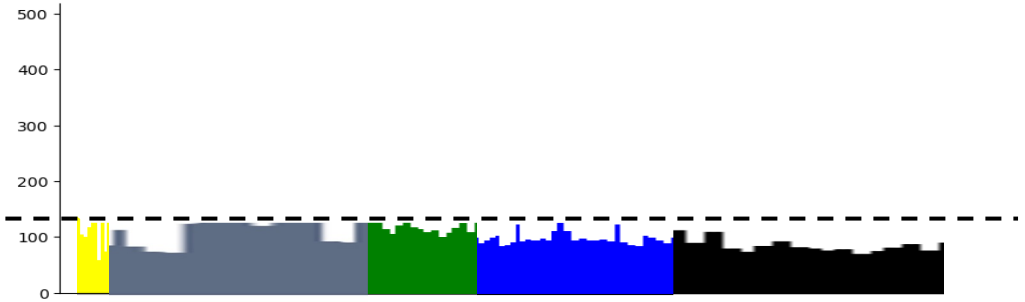
Memory Overuse Inefficient Transmission

Our goal: Design A Balancing Batching Strategy









Toward the first optimization model

Problem Formulation 1: Without Considering Compression

- Let G={g1,g2,g3,…,gn} be the set of genomes.
- B={b1,b2,b3,…,bm} be the set of batches, where bi⊆G. All genomes need to be assigned, one gen in one batch

$$b_j = \begin{cases} 1 & \text{if the } j \text{ batch is used} \\ 0 & \text{otherwise} \end{cases} \qquad x_{ij} = \begin{cases} 1 & \text{if genome } i \text{ is assigned to batch } j \\ 0 & \text{otherwise} \end{cases}$$

- The size of each batch must be less than or equal to a size parameter A (balancing):

$$|b_j| \leq A, \forall j \in \{1, \ldots, m\}$$

Objective: $\text{Minimize} \sum_{j=1}^{m} b_j$

Problem Formulation 1: Without Considering Compression

- Let G={g1,g2,g3,…,gn} be the set of genomes.
- B={b1,b2,b3,…,bm} be the set of batches, where bi⊆G. All genomes need to be assigned, one gen in one batch

$$b_j = \begin{cases} 1 & \text{if the } j \text{ batch is used} \\ 0 & \text{otherwise} \end{cases} \qquad x_{ij} = \begin{cases} 1 & \text{if genome } i \text{ is assigned to batch } j \\ 0 & \text{otherwise} \end{cases}$$

- The size of each batch must be less than or equal to a certain size parameter A:

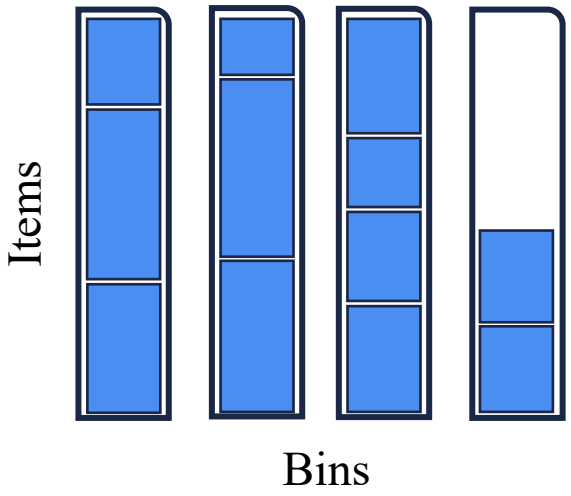
$$|b_j| \leq A, \forall j \in \{1, \ldots, m\}$$

Objective: $\text{Minimize} \sum_{j=1}^{m} b_j$

Bin Packing Problem

Bin Packing Problem: Definition

Bin Packing Problem:
Given a list of items i = 1, . . . , n, each having a size ci ∈ Z+, and an integer value CAPACITY. Find the minimum number of bin to pack all items in such a way that the sum of the item sizes in one bin is always smaller than CAPACITY.



A clasic combinatorial optimization problem.
The problem is NP-complete

Classical heuristics are ordered-based algorithms.

Initially, an empty bin is created. At each step, the next item is selected and packed in a bin. A new bin may be created at each step.

- First-fit: choose the first possible bin
- Best-fit: choose largest remaining CAPACITY bin
- Worst-fit: choose smallest remaining CAPACITY bin

Continue to be a trending research topics (presented at ROADEF 2024)

Problem Formulation 2: Taking Into Account Compression

- Taking into account the compression step:

$$|b_j| \leq A, \forall j \in \{1, \ldots, m\} \quad \text{becomes} \quad |\text{compressor}(b_j)| \leq A, \forall j \in \{1, \ldots, m\}$$

➔ Getting the compression size is non-trivial

- xz compresses 1 genome per sec ➔ 1h20m to compress a batch with n = 5000

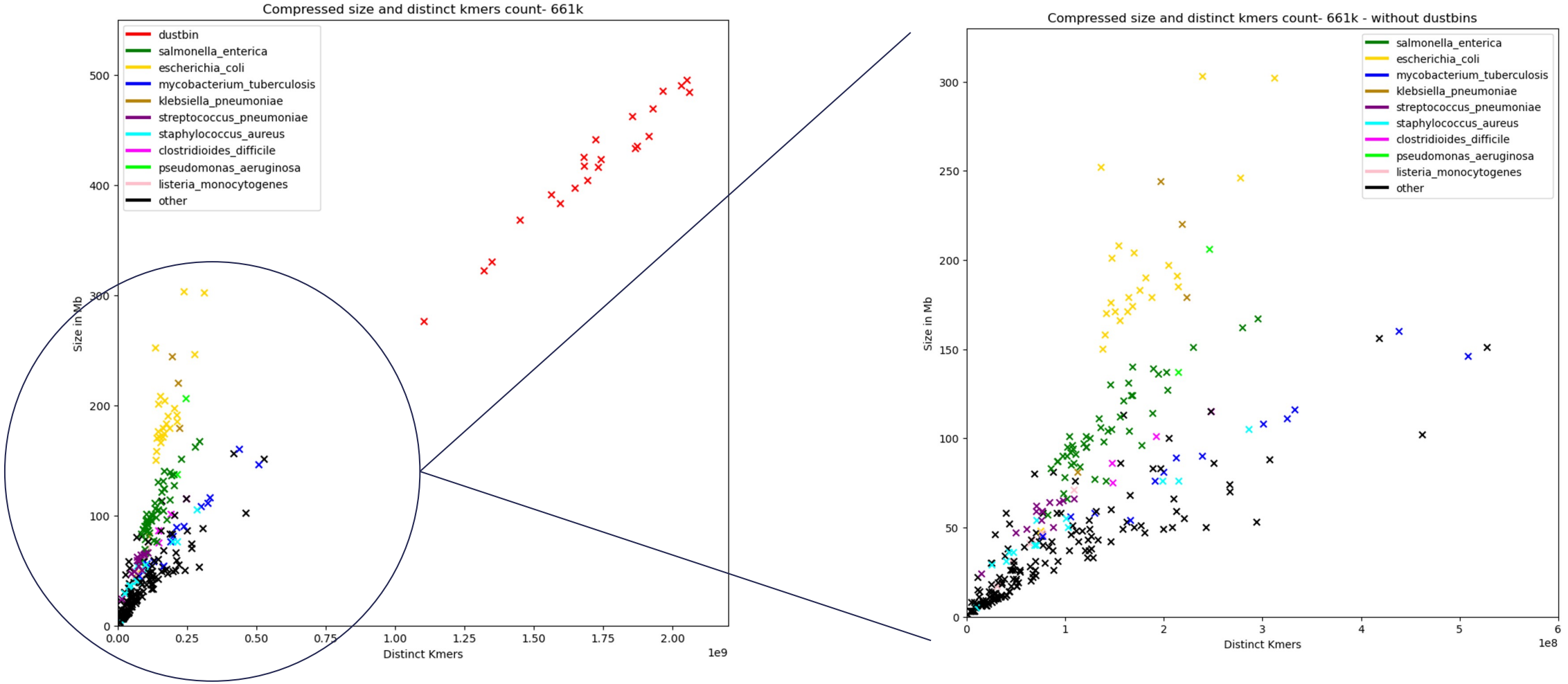
Challenge: Fast estimation of post-compression size for blancing batches.

Methods

Balancing xz Compression Batches:

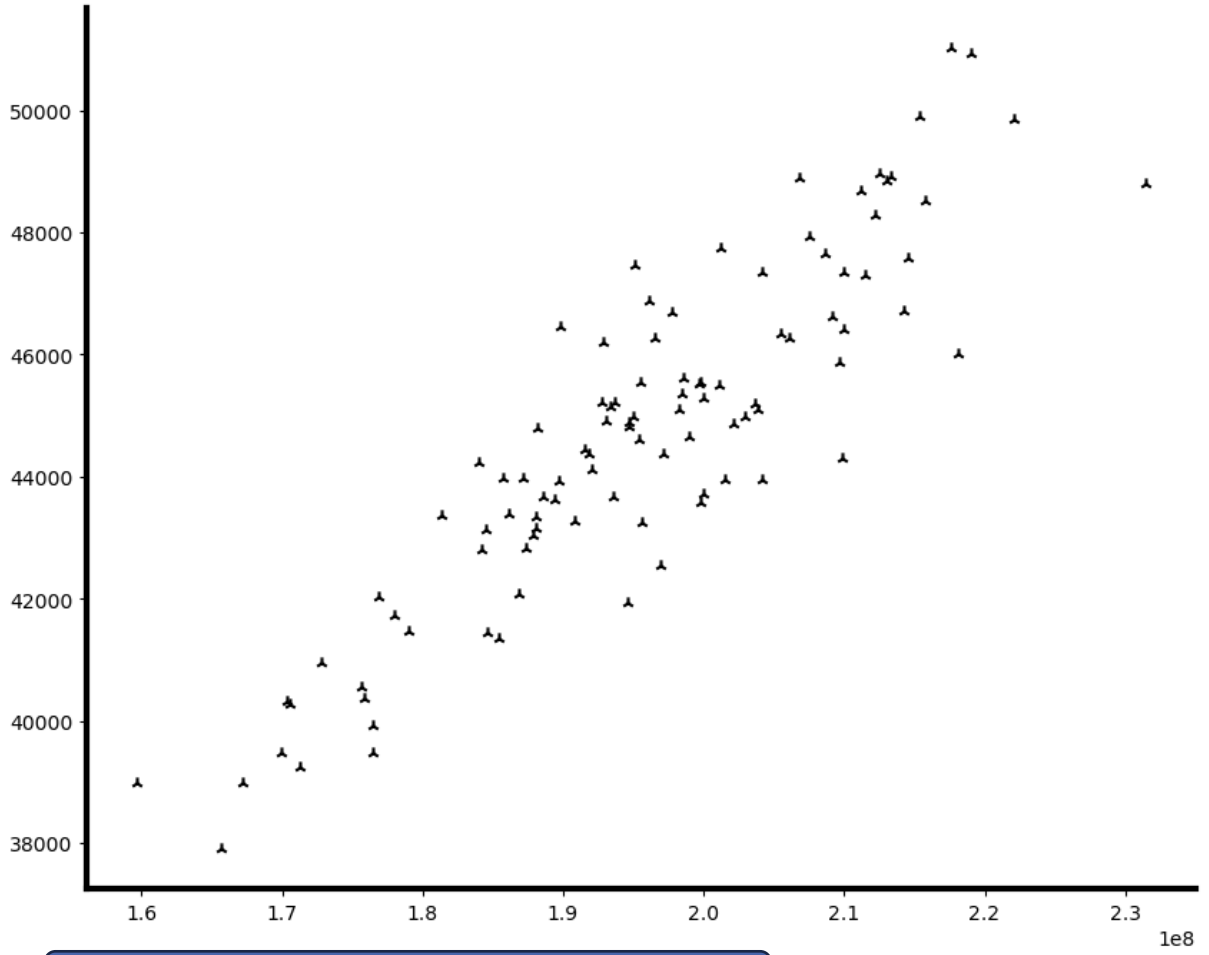
- Ingredient 1: Approximation of Post-compression sizes via distinct Kmers count.
- Ingredient 2: Fast estimation of Distinct Kmers count using HyperLogLog sketching.
- Ingredient 3: Bin Packing and Load Balancing

Ingredient 1: The compressed size of batches is related to its distinct kmers count



Ingedient 1: random samples, 100 batches and 50 genomes in each batch

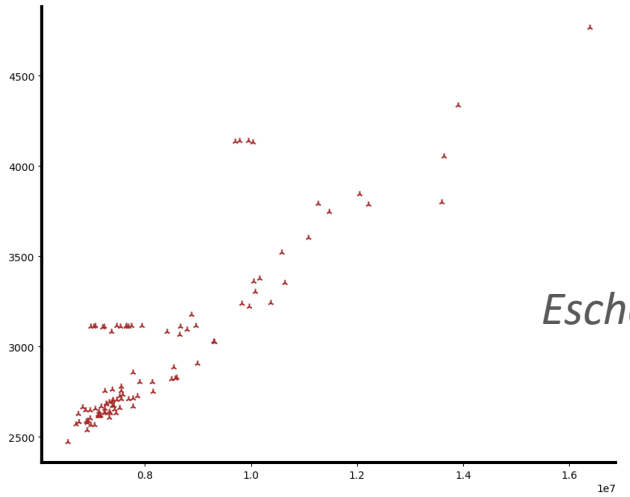
- 100 mixed-species genome batches.
- Dataset:
- Sample 5000 genomes from species with >5000 genomes in 661k.
- Shuffle and distribute into 100 batches, avoiding duplicates per species.
- ⇒ Overall, batch cardinality shows a clear linear correlation with post-compression size.



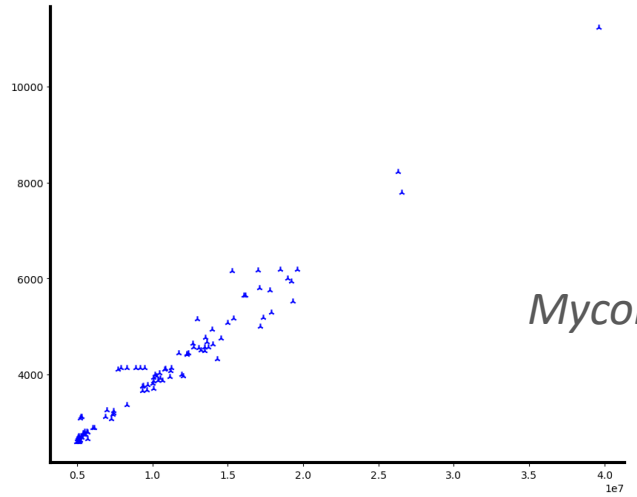
Note to self: varies the number of genomes per batch

Note to self: change the axis to origin 0

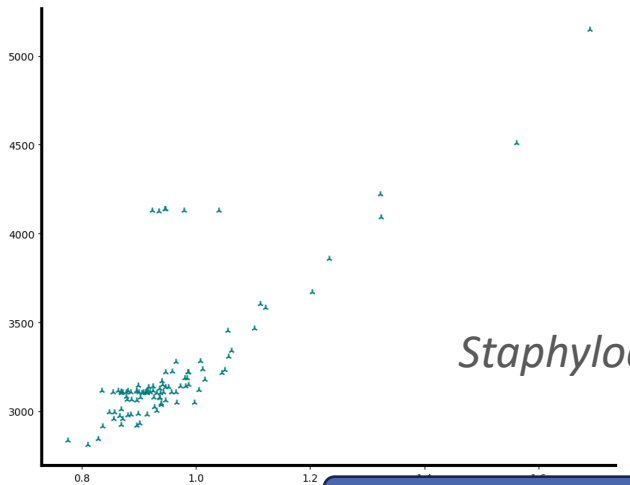
Ingredient 1: The same correlation is observed for the popular species



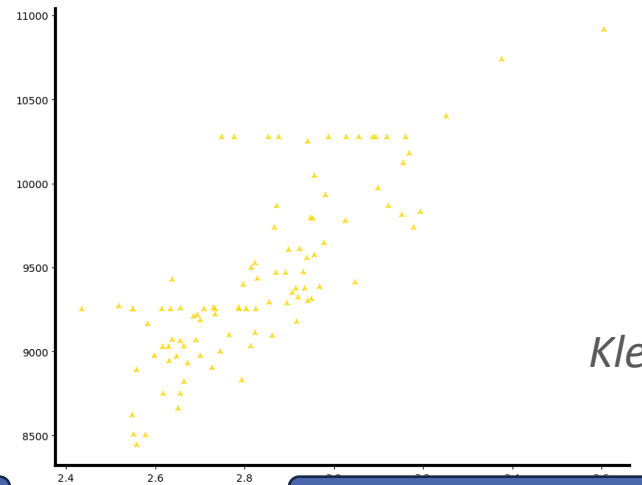
Escherichia coli



Mycobacterium tuberculosis



Staphylococcus aureus



Klebsiella pneumoniae

Note to self: varies the number of genomes per batch

Note to self: change the axis to origin 0

Ingredient 2: Cardinality estimation using HyperLogLog sketching

- Sketches : approximate data structures.
- HyperLogLog sketches for cardinality est.: bit patterns,
- i.e. hash(ATGCG) □ 00010100, hash(CGTAC) □ 00000010.
- Fast and efficient UNION operation for sketches.
- Is implemented in Dashing[5]

[5] Baker, D.N., Langmead, B. Dashing: fast and accurate genomic distances with HyperLogLog. *Genome Biol* 20, 265. 2019.
[6] Bonnie et al., DandD: Efficient measurement of sequence growth and similarity. *iScience* 27, 3. 2024

Ingredient 3: Bin Packing and Load Balancing

Preliminary : Given m genomes, put genomes into batches :

STRATEGY 1 : given unlimited batches with capacity C

$$\boldsymbol{Minimize\ nb\ of\ batch\ B}$$

s.t. $$distinct\_kmers(bj) < C, \text{ for } (\mathrm{j} = 1, \ldots, \mathrm{n})$$

STRATEGY 2 : given a fixed number of batch n

$$\mathrm{T} \geq \boldsymbol{distinct\_kmers}(\boldsymbol{b_j}), \boldsymbol{for\ j} = \mathbf{1}, \ldots, \boldsymbol{n}$$

$$\boldsymbol{Minimize\ max}(\mathrm{T})$$

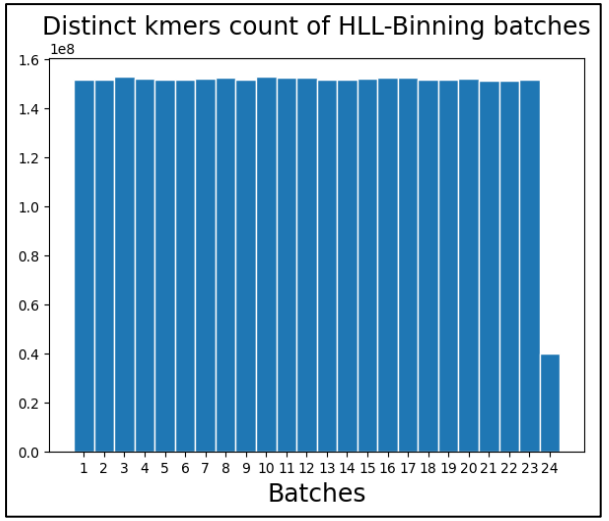
Result

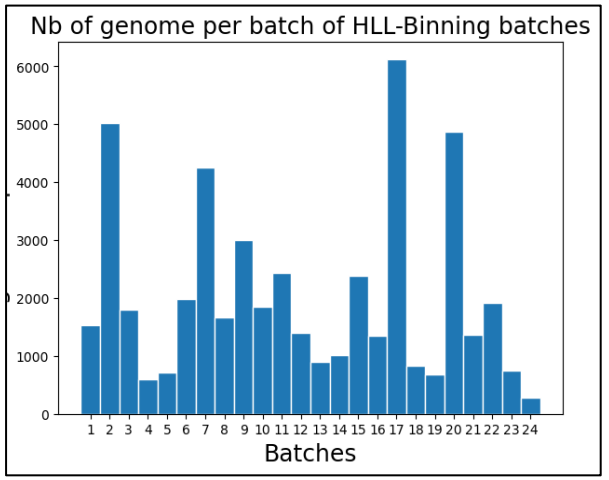
Strategy 1: Bin Packing HyperLogLog – Batching result

Number of Batches = 24

Batch capacity :
C = 152,000,000
(C obtained by linear regression)

Number of genome per batch varies

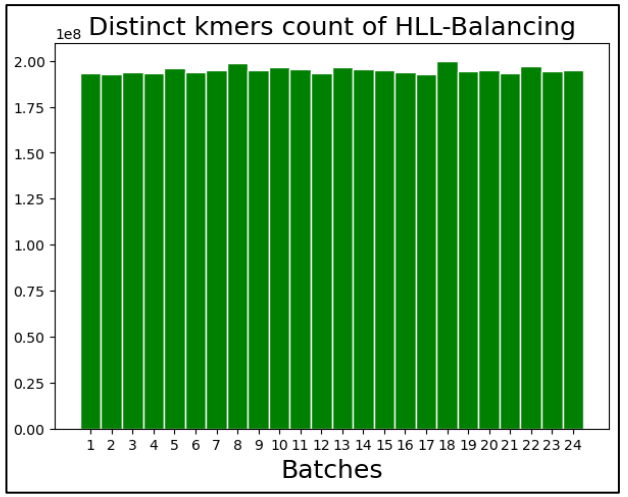


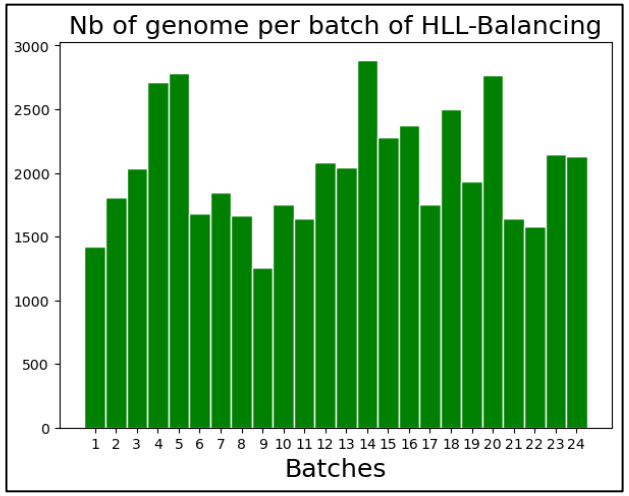


Strategy 2: Load Balancing HyperLogLog – Batching Result

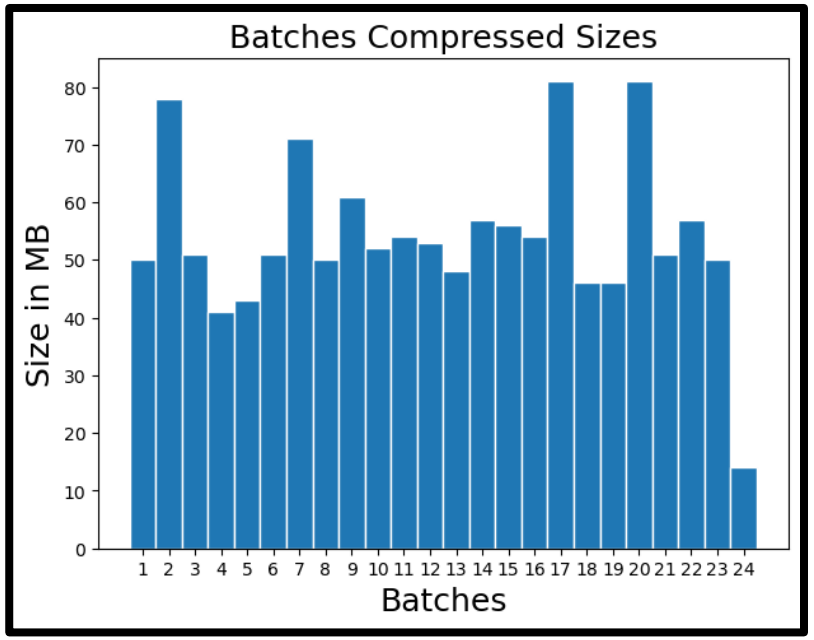
Number of Batches = 24.

Nb of genomes per batch varies but to a lesser extent compared to Strat. 1



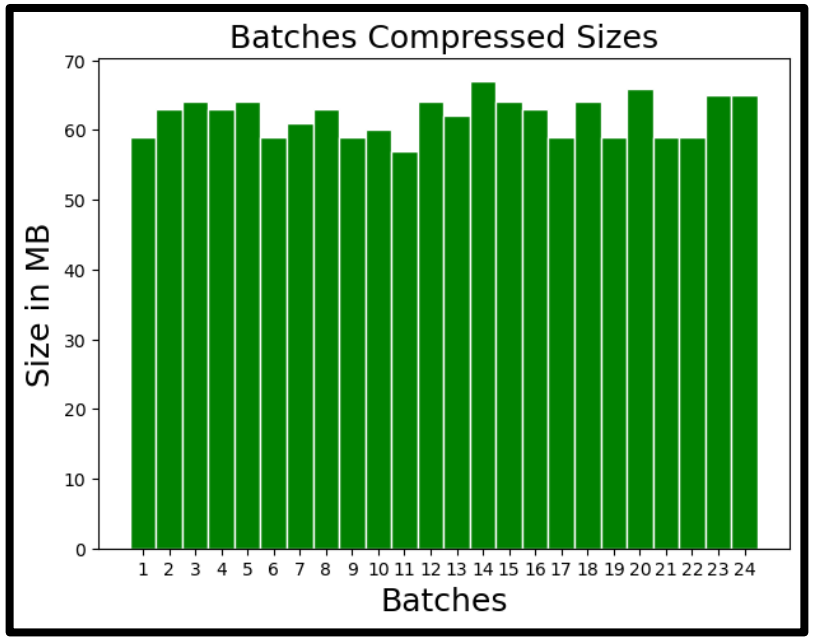


Comparison Of 2 Strategy



Most of the batches are balanced (between 40-50MB, max size 81MB)

Evaluation strat. 1:
Allowing a capacity on distinct kmers.
The result remains somewhat imbalanced.



All Batches are well balanced
(between 59-67MB, max size 67MB)

Evaluation strat. 2:
Producing more balanced batches.
No control over the maximum distinct k-mer count per batch.

Conclusion:

Batching by Predicting Compression Size Using HyperLogLog Distinct K-mer Estimation Improves balancing of the final compressed sizes *Mycobacterium tuberculosis*.

Current Goals:

Extending the results and methods to the whole 661k collection.

Enabling control over the number of genomes in each batch.

Scaling up to AllTheBacteria collection.

Applications in querying data structures such as Bloom filter, on PIM and GPU.