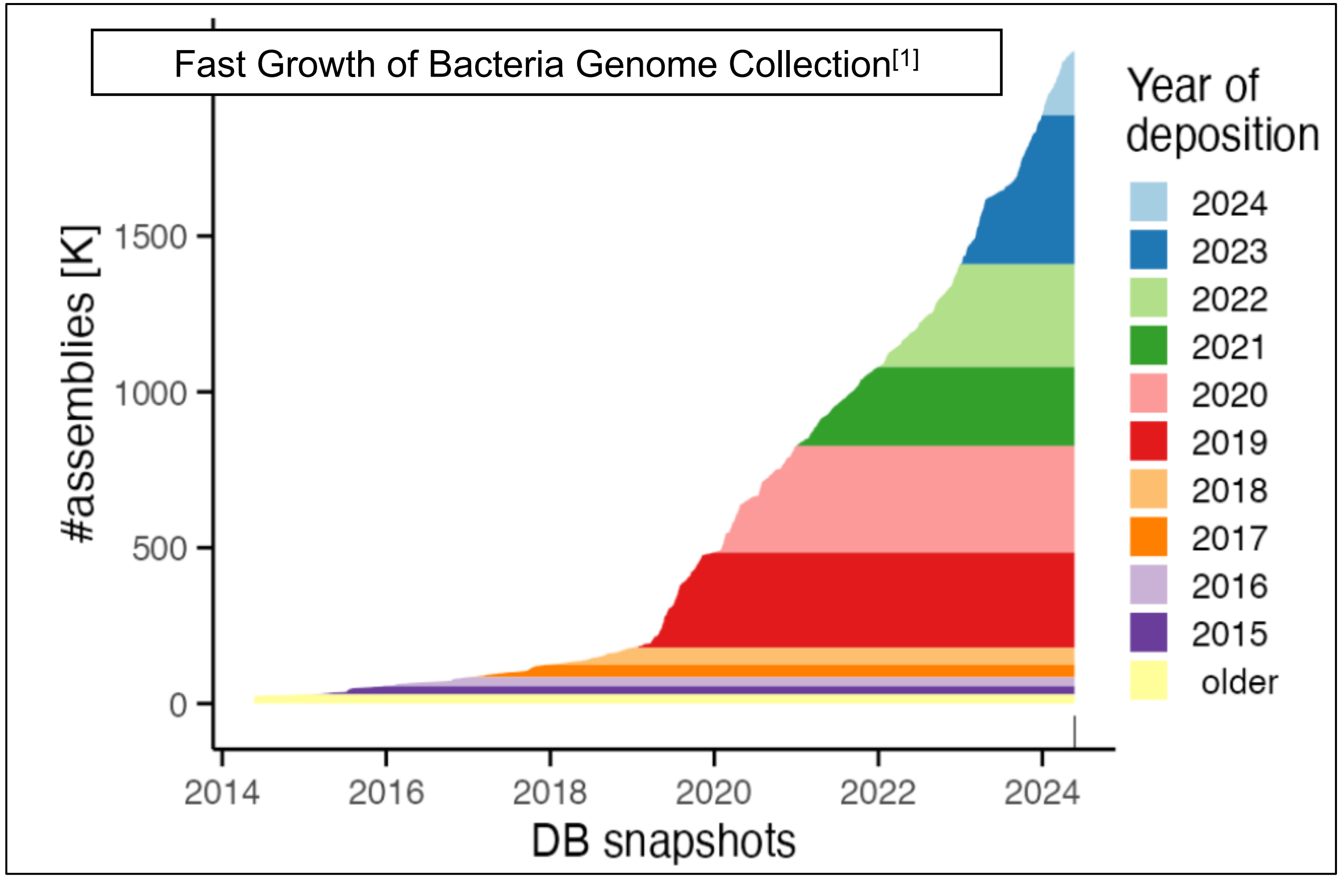
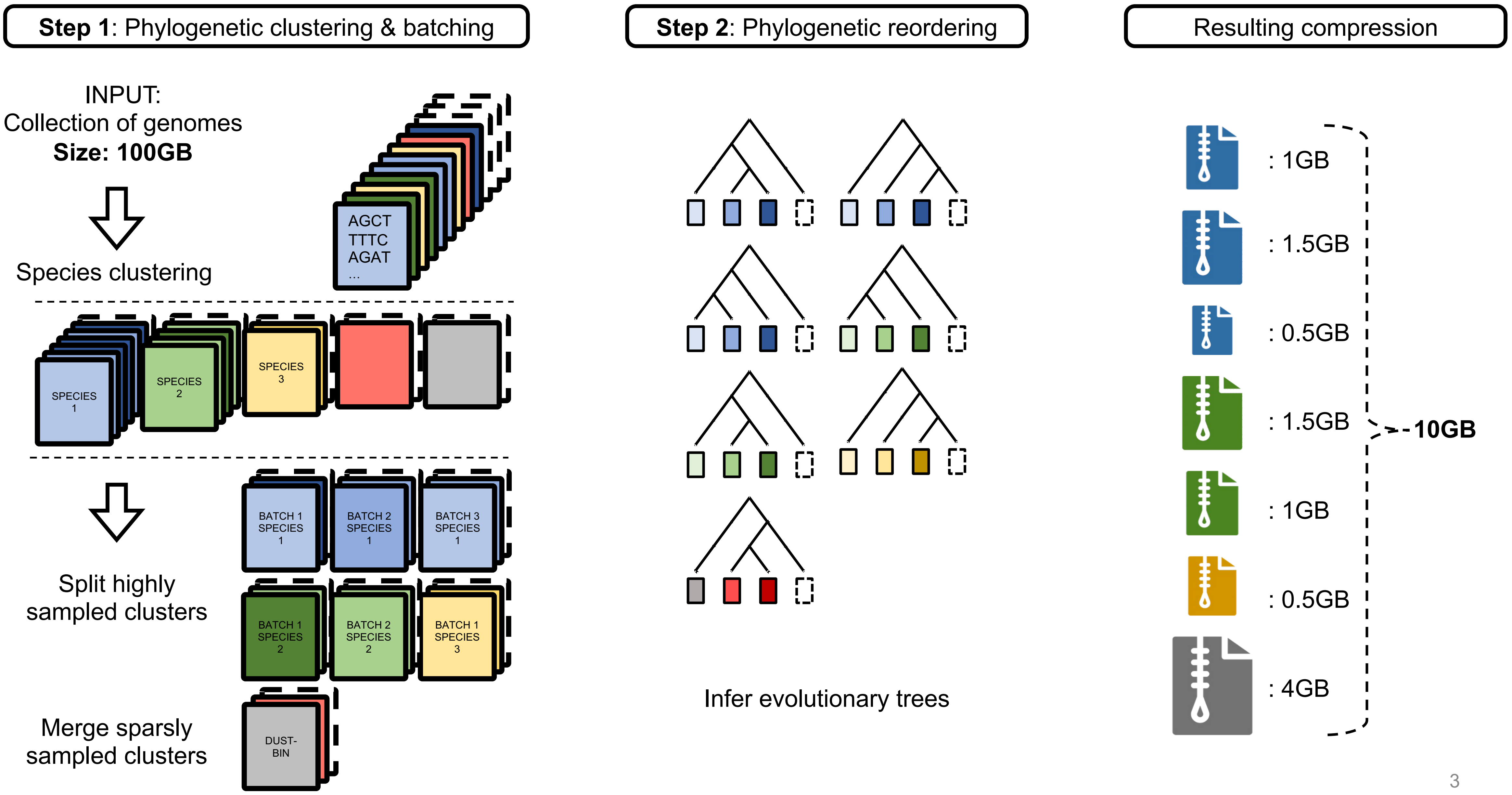
Optimization For Efficient Compression Of Large Bacterial Genome Collections

MOTIVATION: Larger And Higher Diversity Genome Collections Are Growing Rapidly

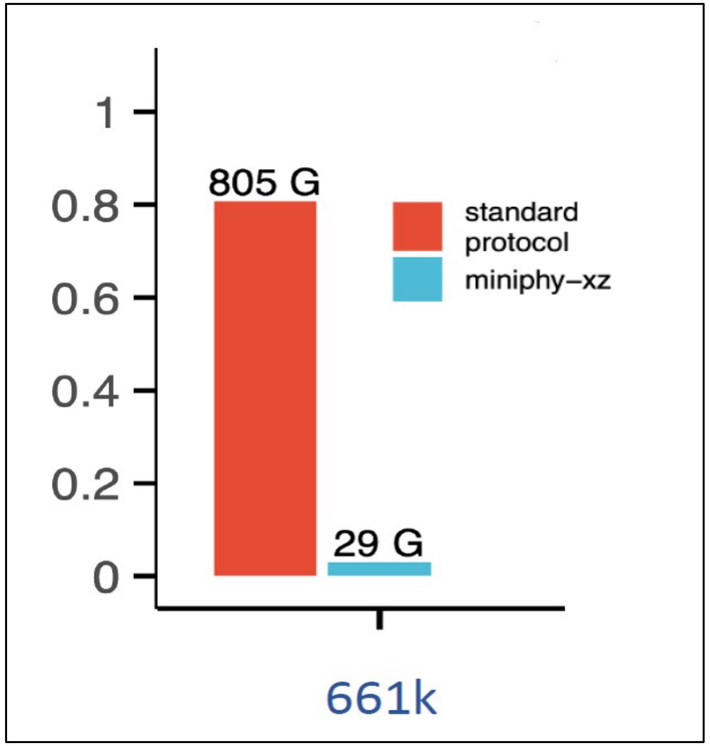


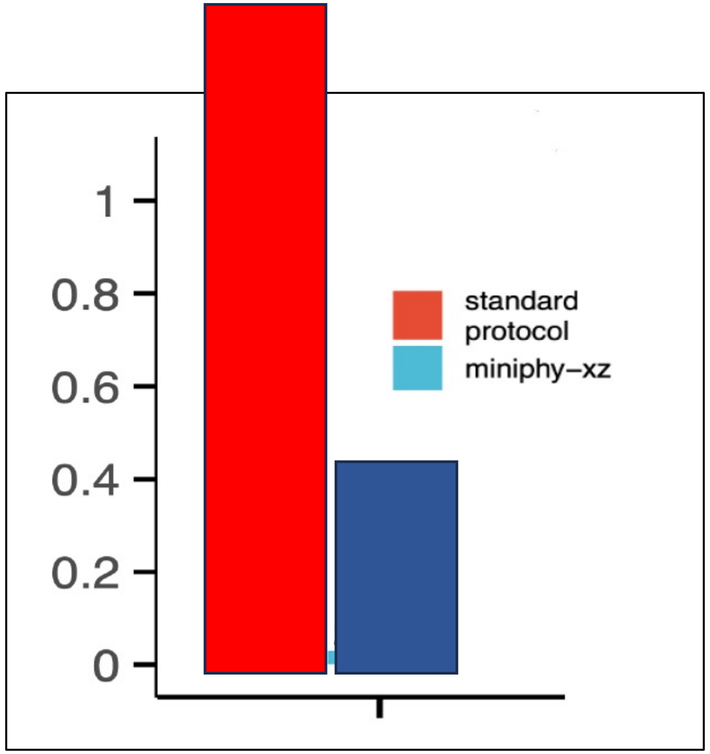
Large Bacterial Genome Collections:	
661k collection[2] (2021)	n = 661,405
AllTheBacteria[3] (2024)	n = 2,440,377
Future	n > 10^7

Phylogenetic Compression[1] Improves Compressibility Via Reordering According To The Evolutionary History



This Strategy Allows Lossless Compression Of 1-3 Orders Of Magnitude Across Different Genome Collections

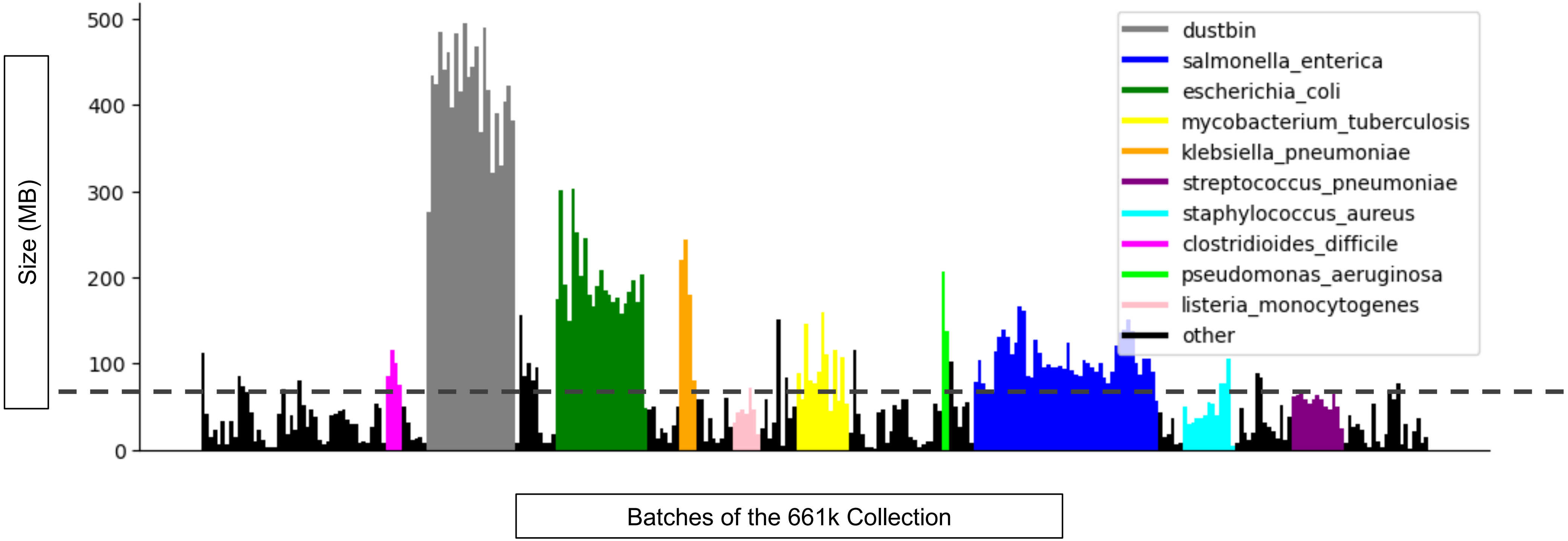




AllTheBacteria - TBA

RESULTING COMPRESSION

Current Limitation: Batching Results In Non-uniform Post-compression Sizes



Current Limitation: Batching Results In Non-uniform Post-compression Sizes

Size (MB)

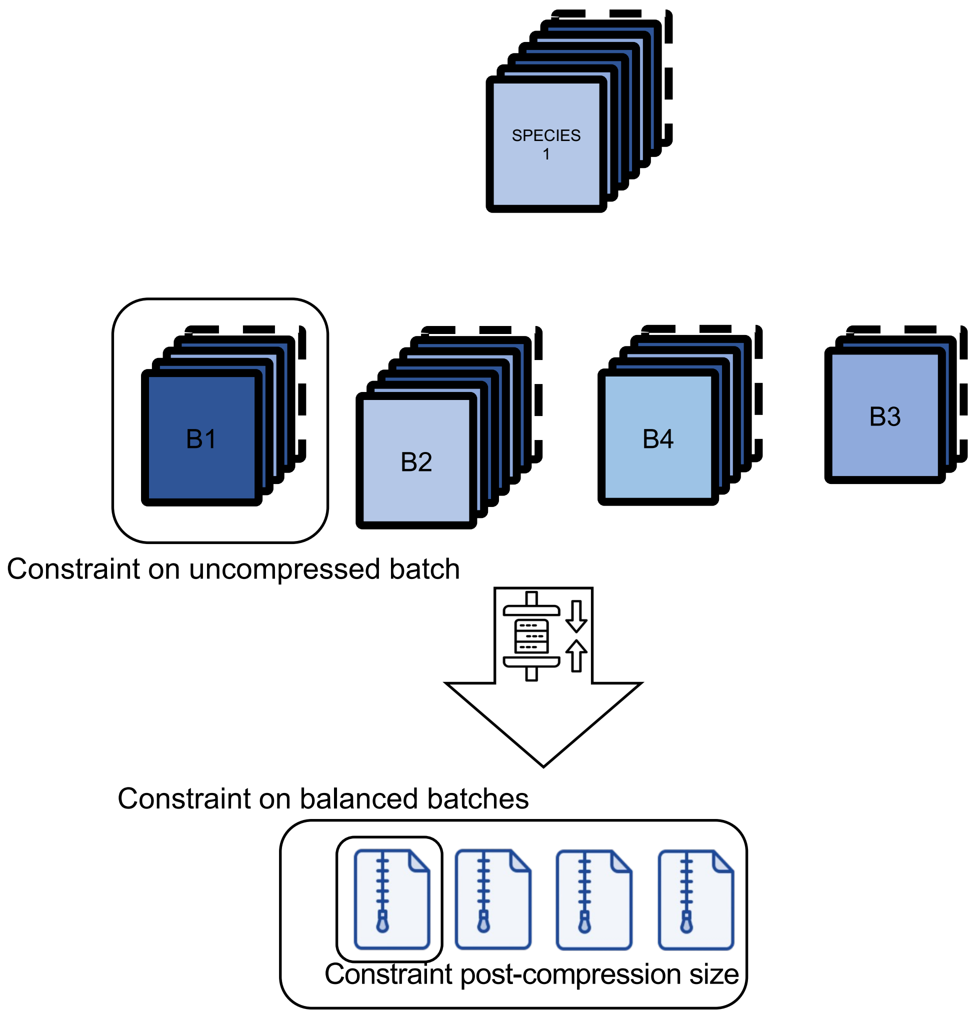
AllTheBacteria Batches here

Batches of the 661k Collection

Consequences: Negative Impact on Downstream Analysis

- Unbalanced Workloads
- Hinder Parallelization
- Inconsistent Query Times
- Memory Overuse
- Inefficient Transmission

Challenge: Find Develop An Optimized Batching Strategy For Various Use Cases



Batching Problem:

Given a set of genomes.
Partition it into a set of batches.
User-input parameters for each batch i.e. number of genomes N, uncompressed size U, post-compression size C.
Maximize the compression ratio of the set and in such a way that some constraints are satisfied.

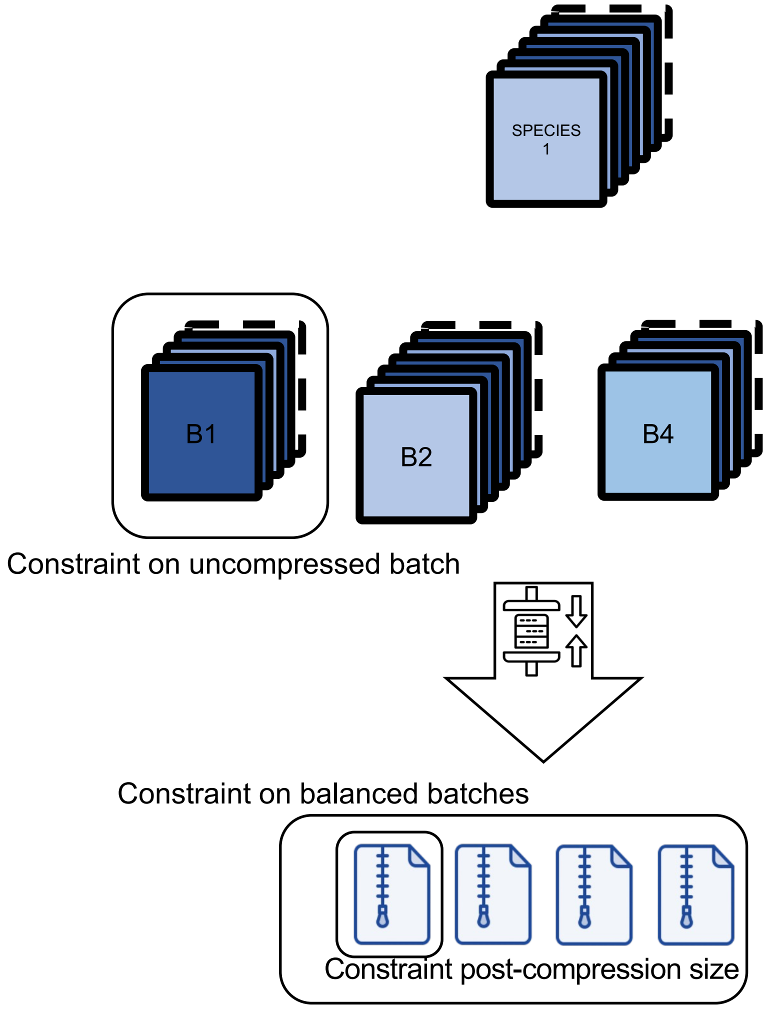
OBJECTIVE:

$$min \sum_{i}^{Batches} \mathrm{PostCompressionSize}(\mathrm{b}_i)$$

Subjects to:
For for all batches:

$$\mathrm{Cardinality}(\mathrm{b}_i) \leq \mathrm{N}$$
$$\mathrm{UncompressedSize}(\mathrm{b}_i) \leq \mathrm{U}$$
$$\mathrm{PostCompressionSize}(\mathrm{b}_i) \leq \mathrm{C}$$
$$\mathrm{PostCompressionSize}(\mathrm{b}_i) - \mathrm{PostCompressionSize}(\mathrm{b}_j) \leq \varepsilon$$

Challenge: Find Develop An Optimized Batching Strategy For Various Use Cases



Batching Problem:

Given a <u>set of genomes</u>.
Partition it into <u>a set of batches</u>.
User-input parameters for each batch i.e. number of genomes N, uncompressed size U, post-compression size C.
<u>Maximize</u> the compression ratio of the set and in such a way that some <u>constraints</u> are satisfied.

OBJECTIVE:

$$min \sum_{i}^{Batches} \mathrm{PostCompressionSize}(\mathrm{b}_i)$$

Subjects to:
For for all batches:

$$\mathbf{Cardinality}(\mathbf{b_i}) \leq \mathbf{N}$$
$$\mathbf{UncompressedSize}(\mathbf{b_i}) \leq \mathbf{U}$$
$$\mathrm{PostCompressionSize}(\mathbf{b_i}) \leq \mathbf{C}$$
$$\mathrm{PostCompressionSize}(\mathbf{b_i}) - \mathrm{PostCompressionSize}(\mathbf{b_j}) \leq \boldsymbol{\varepsilon}$$

Predicting Post-compression Size Is Non-trivial

Batching of Genomes Collection: A Familiar yet Novel Problem

Assumption:
For simplicity, we stop considering genome compression

OBJECTIVE:

$$min \sum_{i}^{Batches} \text{PostCompressionSize}(\text{b}_i)$$

Subjects to:
For for all batches:

$$\text{Cardinality}(\text{b}_\text{i}) \leq \text{N}$$
$$\text{UncompressedSize}(\text{b}_\text{i}) \leq \text{U}$$
$$\text{PostCompressionSize}(\text{b}_\text{i}) \leq \text{C}$$
$$\text{PostCompressionSize}(\text{b}_\text{i}) - \text{PostCompressionSize}(\text{b}_\text{j}) \leq \varepsilon$$

OBJECTIVE:

$$min \sum_{i}^{Batches} \text{b}_i$$

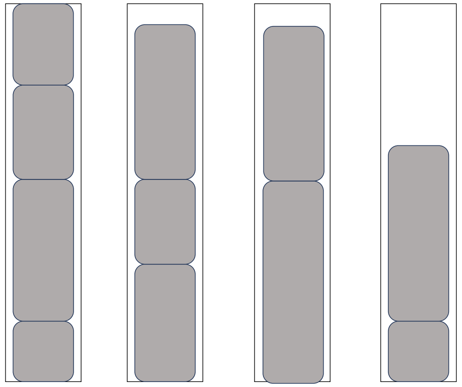
Subjects to:
For for all batches:

$$\text{Cardinality}(\text{b}_\text{i}) \leq \text{N}$$
$$\text{UncompressedSize}(\text{b}_\text{i}) \leq \text{U}$$

This becomes an instance of the classic Bin Packing Problem

Bin Packing Problem Is One Of The First Studied Combinatorial Optimization Problem

Bin Packing Problem:
Given a list of item i = 1, . . . , n, each having a size ci ∈ R+, 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.



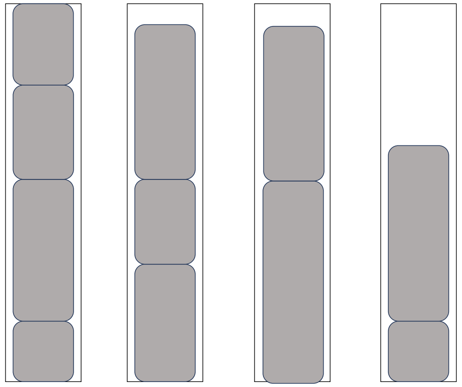
The problem is NP-complete

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.

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

Bin Packing Problem Is One Of The First Studied Combinatorial Optimization Problem

Bin Packing Problem:
Given a list of item i = 1, . . . , n, each having a size ci ∈ R+, 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.



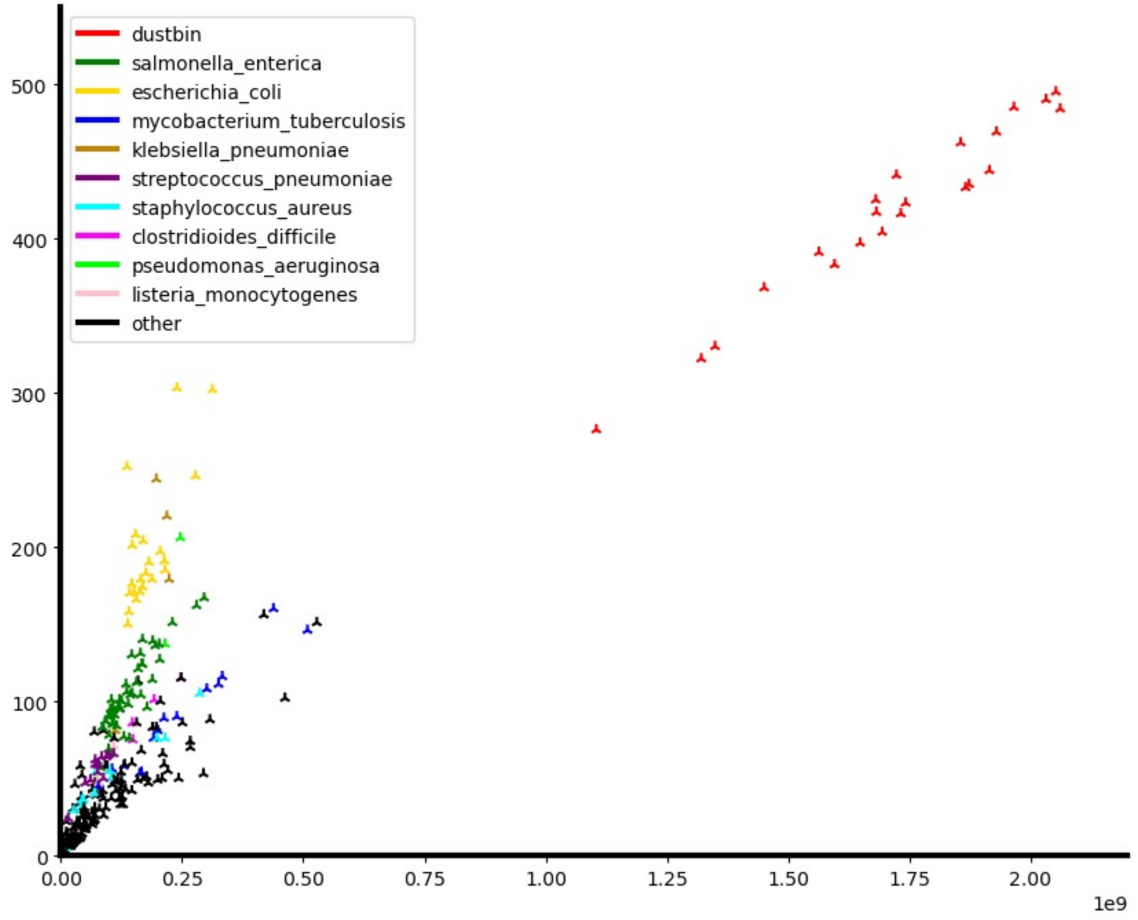
The problem is NP-complete

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.

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

Now that we have a strategy to dynamically pack bins based on different parameters, can we estimate the post-compression size without actually compressing the data?

Observation: xz Post-compression 661k Batch Sizes Correlate With Their Distinct Kmers Count



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.

HyperLogLog Bin Packing Strategy For Genomes Batching

Pseudocode of Strategy 1

HyperLogLog Load Balancing Strategy For Genomes Batching

Pseudocode of Strategy 2

Recap of the Three Batching Strategies

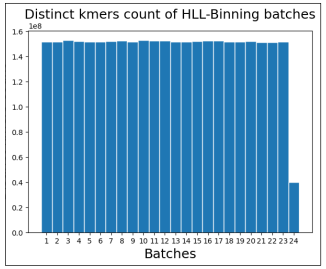
Comparisons of the batching strategíe

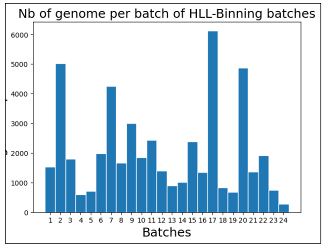
STRATEGY 1: HLL-Binning

STRATEGY 2: HLL-Balancing

DATA : Genomes of *Mycobacterium tuberculosis* from the 661k Collection[2] , B = 24

Batches Obtained From Strat. 1

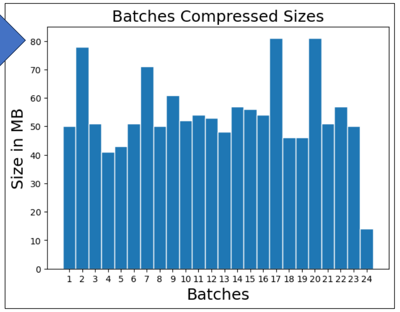




PHYLOGENETIC COMPRESSION

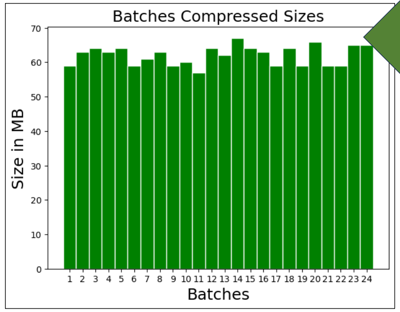
Batch capacity :
C = 152,000,000
(C obtained by linear regression)

Number of genome per batch varies



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.

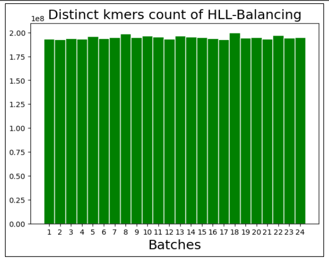


All Batches are well balanced
(between 59-67MB, max size 67MB)

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

PHYLOGENETIC COMPRESSION

Batches Obtained From Strat. 2



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