Optimization Framework For Phylogenetic Compression

Comité de Suivi Individuelle (1st year)

Tam Truong, Dominique Lavenier, Pierre Peterlongo, Karel Břinda

25 June 2025

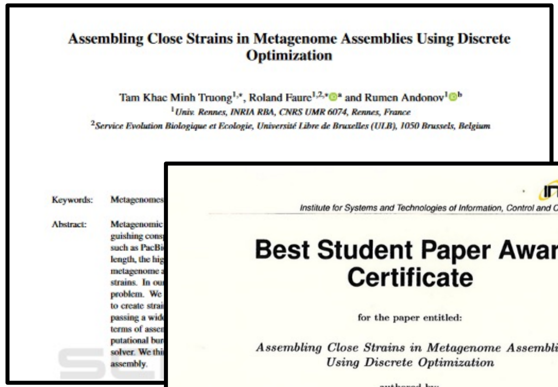
Presentation Outline:

My Backround: International & Multidisciplinary

- 2018-2021: Licence MIAGE – Rennes University
- 2021: Full-stack developer – Enedis, Paris
- 2021-2023: Dbl-deg Master Data Science & Business: Rennes & Aalto Univ (Finland)
- 2022 & 2023: 2 Internships at INRIA: Optimization for strains seperation - Supervisor: R. Andonov
- Start: Nov. 2024 PhD: Optimization for Phylogenetic Compression

Multidisciplinary competences

Big data
Method development & researching
Master's internships in bioinformatics

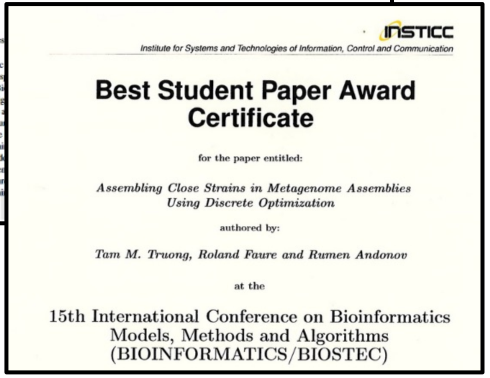


Assembling Close Strains in Metagenome Assemblies Using Discrete Optimization

Tam Khac Minh Truong[1,*], Roland Faure[1,2,*,a] and Rumen Andonov[1,b]
[1]*Univ. Rennes, INRIA RBA, CNRS UMR 6074, Rennes, France*
[2]*Service Evolution Biologique et Ecologie, Université Libre de Bruxelles (ULB), 1050 Brussels, Belgium*

Keywords: Metagenomes

Abstract: Metagenomic
guishing cons
such as PacBi
length, the hig
metagenome a
strains. In ou
problem. We
to create strai
passing a wid
terms of assen
putational bur
solver. We thi
assembly.



INSTICC
Institute for Systems and Technologies of Information, Control and Communication

Best Student Paper Award Certificate

for the paper entitled:

Assembling Close Strains in Metagenome Assemblies Using Discrete Optimization

authored by:

Tam M. Truong, Roland Faure and Rumen Andonov

at the

15th International Conference on Bioinformatics Models, Methods and Algorithms (BIOINFORMATICS/BIOSTEC)

Training Programs

First Year

- Tranversial training (50h)
  - Starthèse: Entrepreneurship
  - French language B1
- Scientific training
  - Conferences: SeqBim
  - Various programmation training: Git, Good practices

Plan: Second Year

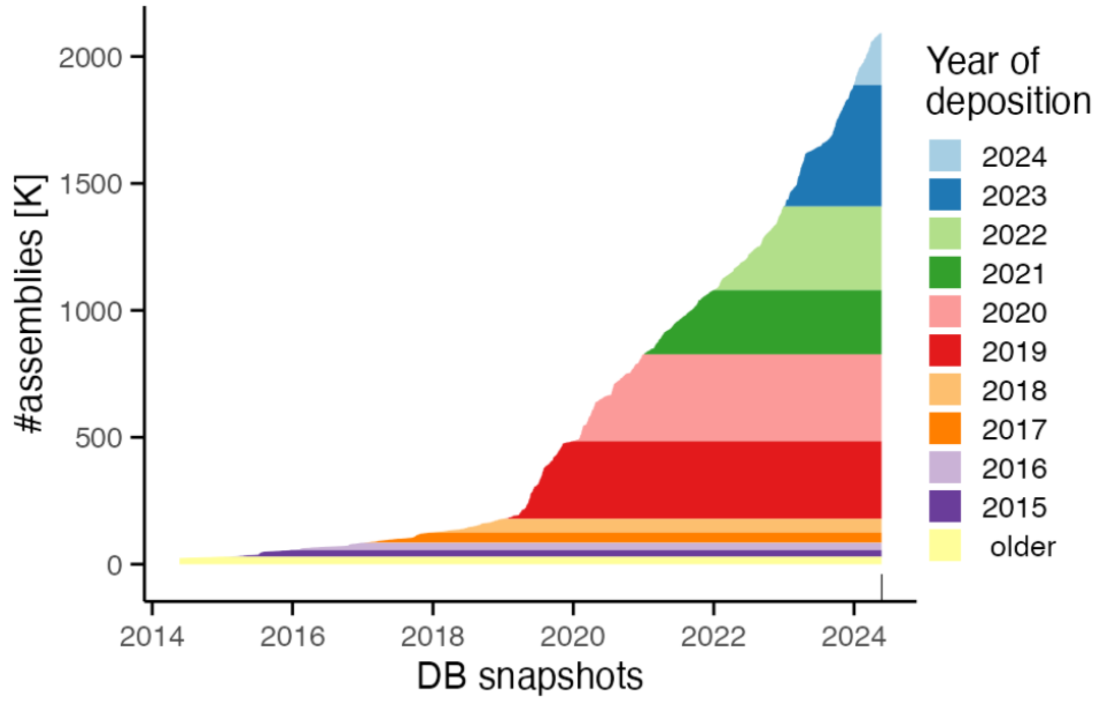
- Tranversial training
  - Ethic for Science
- Scientific training
  - Scientific writing
  - Teaching

My PhD Thesis

Motivation & State Of The Art

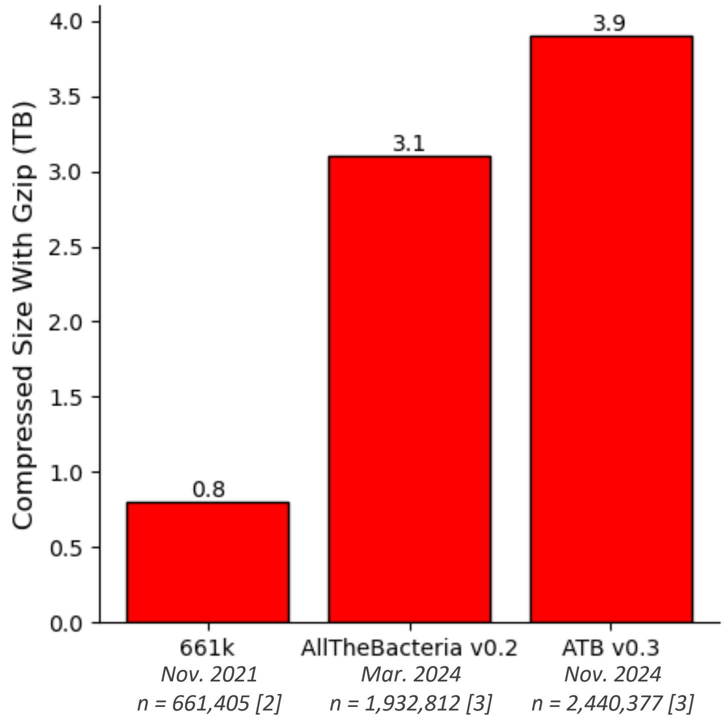
Motivation: Rapidly Growing Bacteria Genome Data & Collections

Fast Growth Of Bacterial Genomes Data[1] (NCBI)



Karel Brinda, 2024 (CC) https://doi.org/10.6084/m9.figshare.25879258

Multi-Terabytes Microbial Genome Collections

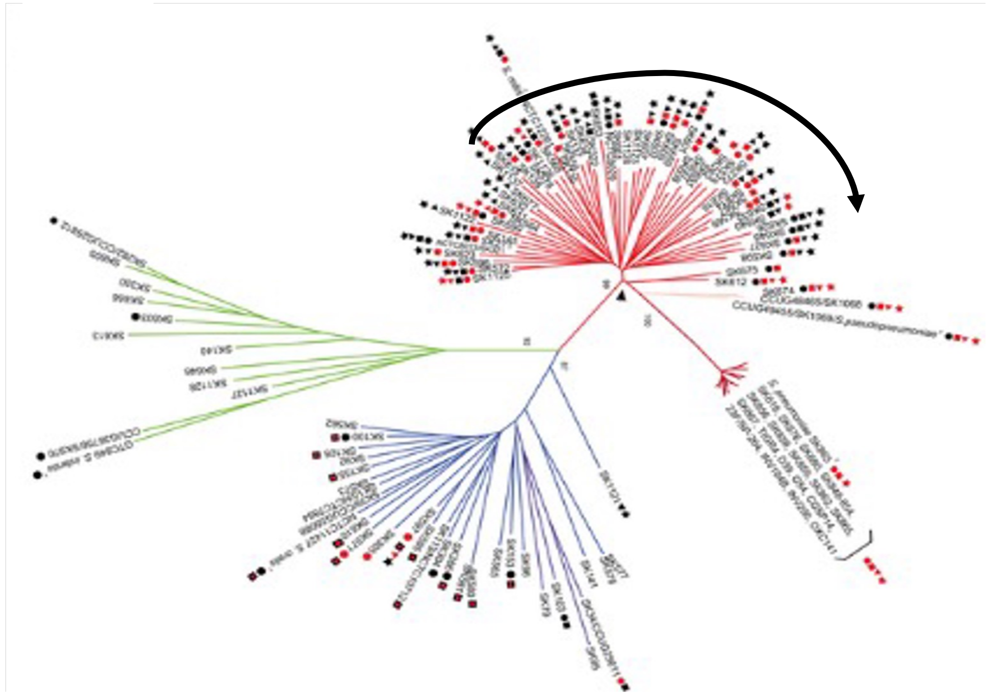


18 Jun 2025: ATB v0.4 with 330k new genomes
Next decade: Even larger collections (n = ~10^7), higher diversity, …

Challenging: Efficient storage & analysis (indexing, searching)
Standard compression protocol is not sufficient

[1] Břinda et al. 2025, [2] Blackwell et al. 2021, [3] Hunt et al. 2024

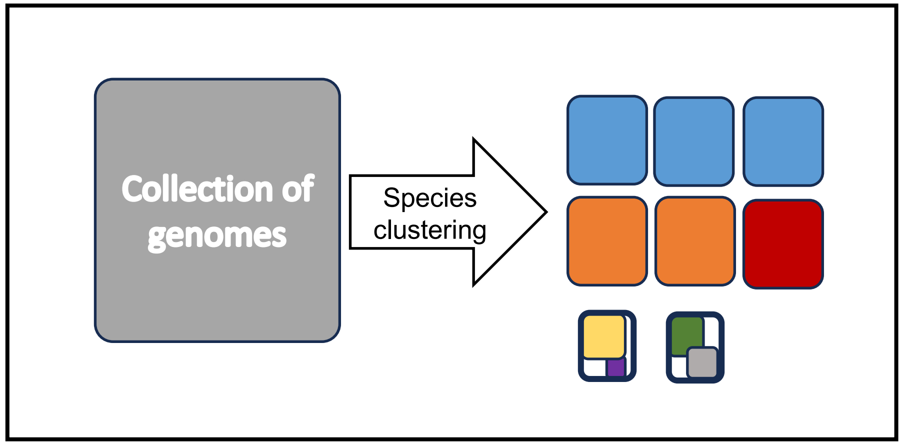
Phylogenetic Compression: New Method For Microbial Genomes Compression



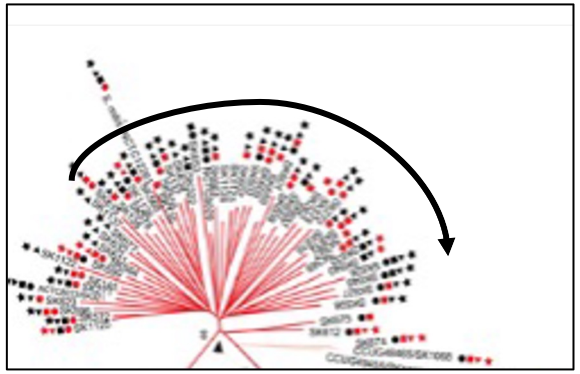
Difficulty: Compression of genomes is challenging due to the widespread redundancy in the data.

Key Idea: Reordering genomes based on evolutionary history enhances local compressibility[1]

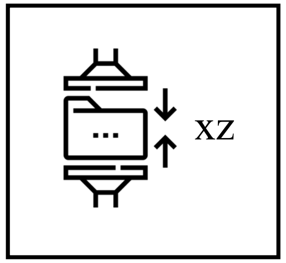
Phylogenetic Compression: Key Steps – Implemented in **MiniPhy**



Step 1 : Phylogenetic Species-based Pre-ordering & Batching



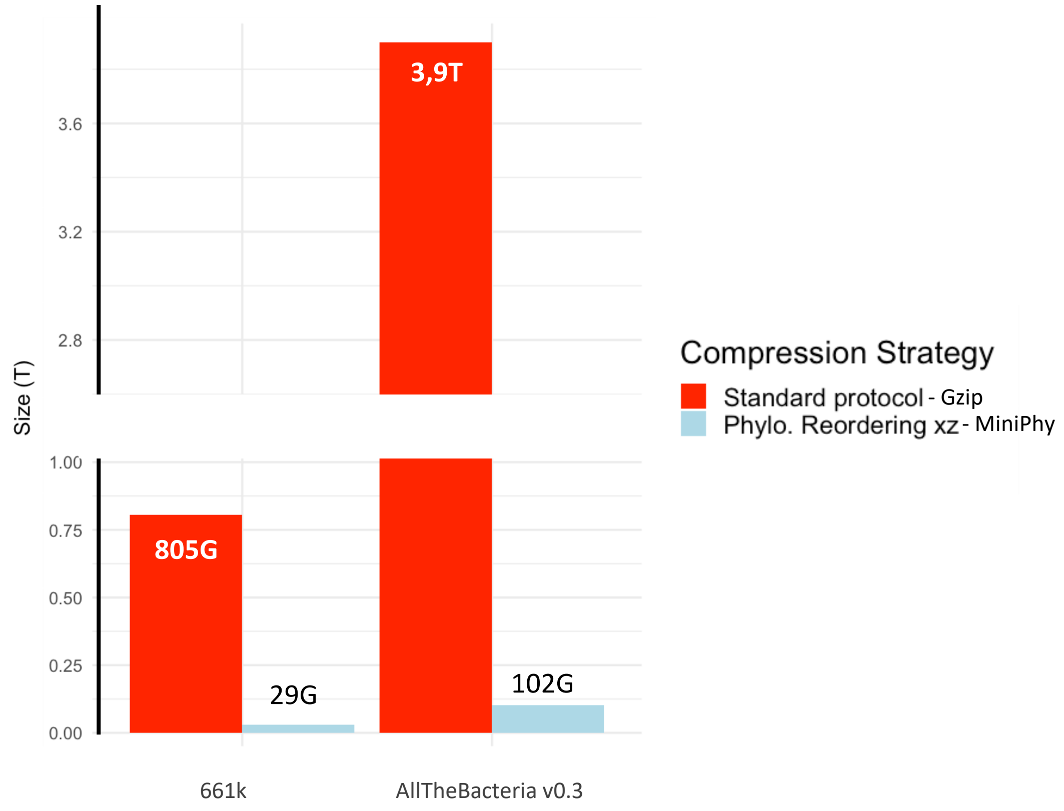
Step 2 : Phylogenetic Reordering Per Batch



Step 3 : Compression

Břinda et al. 2025, Blackwell et al. 2021, Hunt et al. 2024

Phylogenetic Compression: Applications On Large Genomes Collections – 1-2 Orders Of Magnitude Reduction In Size

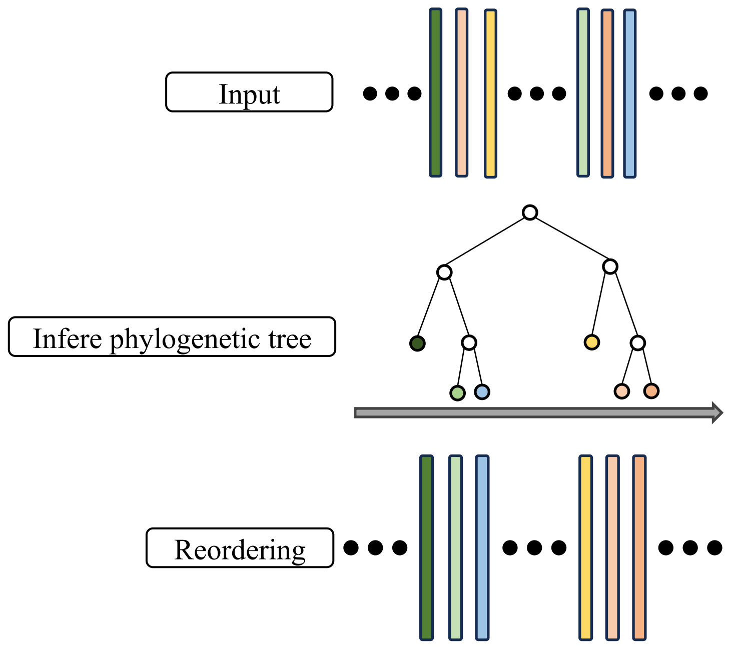


➔ Core compression technique for ATB, currently the largest microbial genomes collection

Břinda et al. 2025, Blackwell et al. 2021, Hunt et al. 2024
MiniPhy: github.com/karel-brinda/miniphy

Key Concepts

Ordering In Phylogenetic Compression



Redundancies are predictable because they result from evolution and sampling

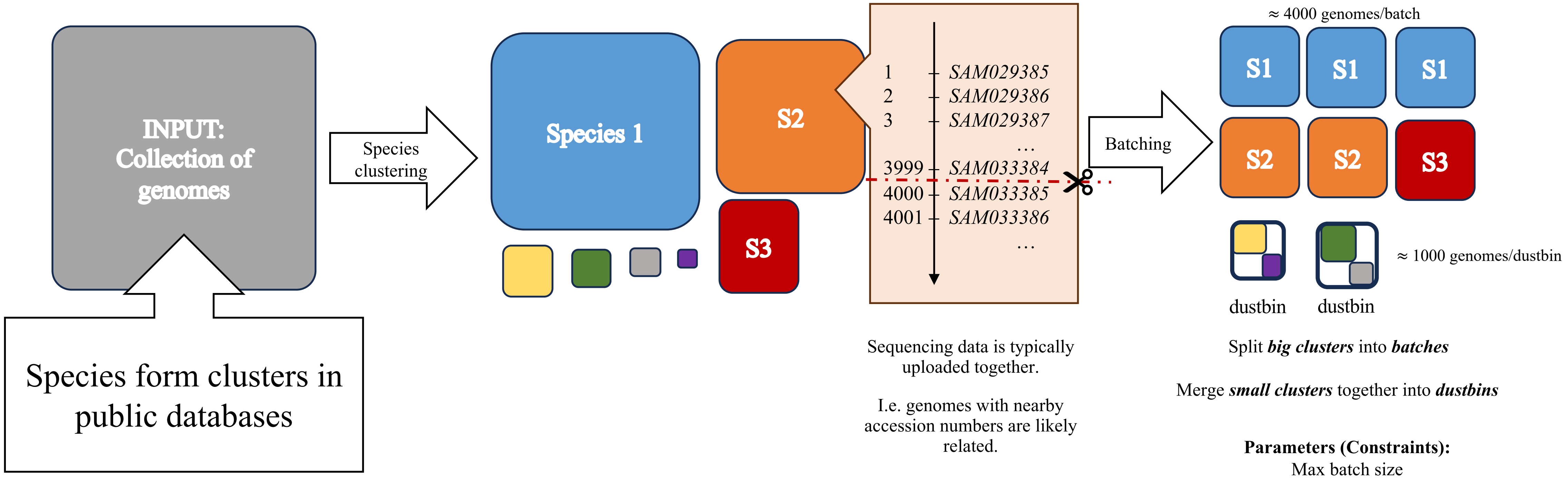
Ordering puts similar genomes closer to each other ➔ Improve local compressibility

Not scalable to million of genomes: inferring phylogenetic tree has quadratic complexity [2]

[1] Břinda et al. 2025; [2] Howe, Bateman, and Durbin 2002

Batching In Phylogenetic Compression (MiniPhy)

MiniPhy's default strategy: sort by sequencing time (**accession number**)



Sequencing data is typically uploaded together.

I.e. genomes with nearby accession numbers are likely related.

Split ***big clusters*** into ***batches***

Merge ***small clusters*** together into ***dustbins***

Parameters (Constraints):
Max batch size
Max dustbin size
Species's cluster threshold

Limitations

Lack Of Suitable Methods For The Phylogenetic Batching Step

No Formalization Of Batching As An Optimization Problem.

Order-awareness
Constraints on:
Uncompressed sizes
Compressed sizes
Number of genomes
Bounds on size of the used search indexes

Batching Are Not Suitable For Hardware Specific Target Application.

Search on GPUs or for processing-in-memory (PIM) architectures

Heavily Dependent On Metadata For An Approximative Input Order

Accession number
Species labels

Optimization Formulation For Phylogenetic Compression

Optimization Problem Formulation For Phylogenetic Compression

Inputs:

- $G = \{g_1, g_2, \dots, g_n\}$*: set of genomes*
- *We want to split G into m ordered batches,* $m \leq n$
- $B = \{b_1, b_2, \dots, b_m\}$*: set of* ***ordered batches***

Parameters:

- u : *bound on uncompressed size*
- c : *bound on compressed size*
- e : *bound on number of genomes*

Decision Variables:

- $x_{ij} \in \{0,1\}$: 1 *if genome* g_i *is in batch* b_j, 0 *otherwise*
- $y_j \in \{0,1\}$: 1 *if batch* b_j *is used*, 0 *otherwise*

Functions:

- $U(b_j) = f_{uncompressed}(ordered\ \{g_i : x_{ij} = 1\})$
  - *Exp: disk size of a batch*
- $C(b_j) = f_{compressed}(ordered\ \{g_i : x_{ij} = 1\})$
  - *Exp: xz compression of a batch (param: level 9 compression, …)*

Objective function:

- *Minimize total compressed batch sizes :* $\boldsymbol{min \sum_{j=1}^{m} C(b_j) \cdot y_j}$
  - *Find the order to achieve the best compression*
- *Minimize the number of batches :* $\boldsymbol{min \sum_{j=1}^{m} y_j}$
  - *Fewer batches ➔ better compression*
- *Combined:*

$$\boldsymbol{min \sum_{j=1}^{n} C(b_j) \cdot y_j + \sum_{j=1}^{n} y_j}$$

Subjects to (possible) constraints:

1) $\sum_{j=1}^{m} x_{ij} = 1 \quad \forall i \in \{1, \dots, n\}$: *a genome must be assigned*
2) $x_{ij} \leq y_j \quad \forall i, j$: *no genomes in unselected batches*
3) $U(b_j) \leq u \cdot y_j \quad \forall j \in \{1, \dots, m\}$: *bound on uncompressed size*
4) $C(b_j) \leq c \cdot y_j \quad \forall j \in \{1, \dots, m\}$: *bound on compressed size*
5) $\sum_{i=1}^{n} x_{ij} \leq e \cdot y_j \quad \forall j \in \{1, \dots, m\}$: *bound on genomes count per batch*

Example Scenarios For Optimization

Internet Transmission & Memory Constraints Platforms

Objective function – xz compressor:

$$\min \sum_{j=1}^{n} C_{xz}(b_j) \cdot y_j + \sum_{j=1}^{n} y_j$$

Subjects to:

1) $\sum_{j=1}^{m} x_{ij} = 1 \quad \forall i \in \{1, \dots, n\}$
2) $x_{ij} \leq y_j \quad \forall i, j$
3) *Each compressed batch must be less than 100 MB*

$$C(b_j) \leq 100(MB) \cdot y_j \quad \forall j \in \{1, \dots, m\}$$

Computation on many CPUs with shared memory

Objective function:

$$\min \sum_{j=1}^{n} C_{RLE}(b_j) \cdot y_j + \sum_{j=1}^{n} y_j$$

Subjects to:

1) $\sum_{j=1}^{m} x_{ij} = 1 \quad \forall i \in \{1, \dots, n\}$
2) $x_{ij} \leq y_j \quad \forall i, j$
3) $Search\_index(b_j) \leq 8(GB) \cdot y_j \quad \forall j \in \{1, \dots, m\}$

The Two Tracks Of My Work: Preordering & Partitionning

OBJECTIVE:

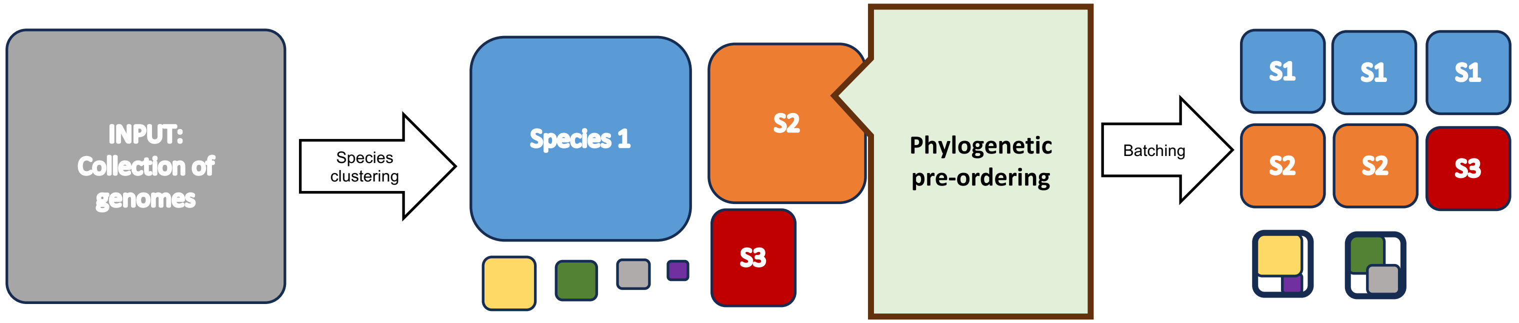
$$\min \boxed{\sum_{j=1}^{n} C(b_j) \cdot y_j} + \boxed{\sum_{j=1}^{n} y_j}$$

1. Phylogenetic Pre-ordering Of Genomes

2. Order-based Genome Batching

Axis 1: Phylogenetic Pre-ordering Of Genomes

Track's Objective: Achieving Global Phylogenetic Pre-order At The Million-genome Scale



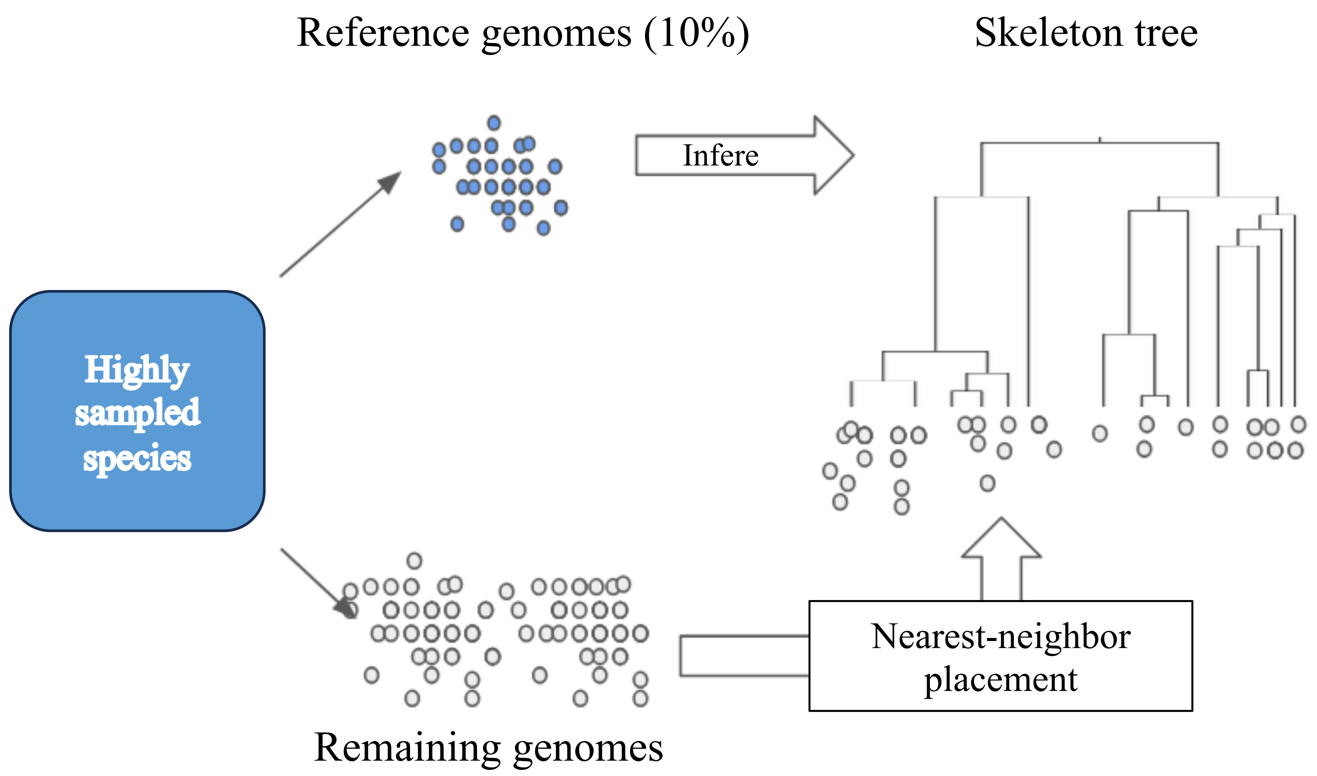
Infere phylogenetic tree for each species : Distance estimation (Mash) + Neighbor joining tree (Quicktree)

Challenging for highly sampled species (>50k genomes)

Infeasible for large modern collection (million-genome scale)

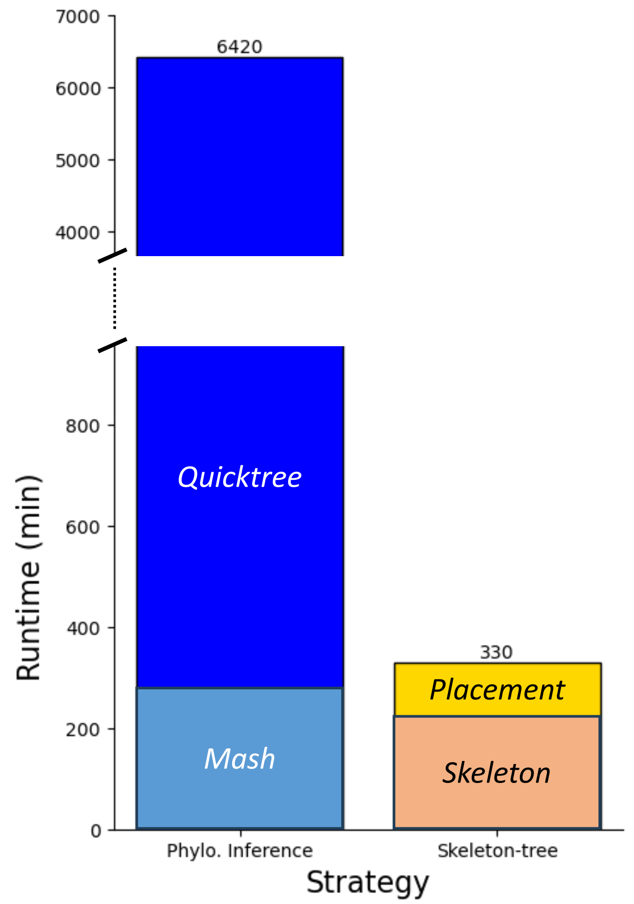
Ondov et al. 2016, Howe, Bateman, and Durbin 2002
Attotree: github.com/karel-brinda/attotree

Skeleton-tree Based Pre ordering – For Highly Sampled Species



Ondov et al. 2016, Howe, Bateman, and Durbin 2002
Attotree: github.com/karel-brinda/attotree

Run Time Comparison: >10x Improvement Compared To Standard Strategy



Comparing execution time of 2 pre-ordering strategies:

- ***Phylogenetic inference*** vs ***Skeleton-tree based***
- ***Test dataset:*** ~90k genomes of E. coli from the 661k collection

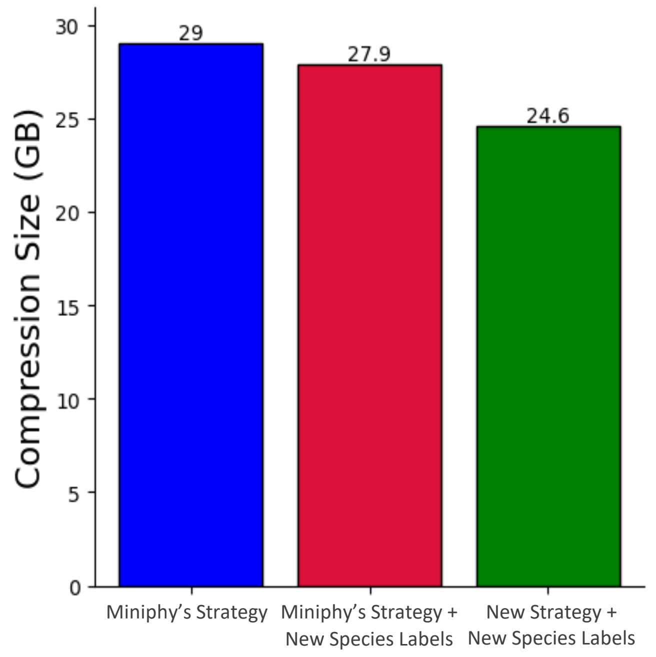
Compression result:

- Cut into batches of 4000 genomes then compress with xz

	Phylo. Inf.	**Ske. Tree**
Comp. size	3.03 G	3.05G

Significantly faster and produce similar compression result

Result (661k) Species-wise: 18% Size Reduction (With New Species Labels)

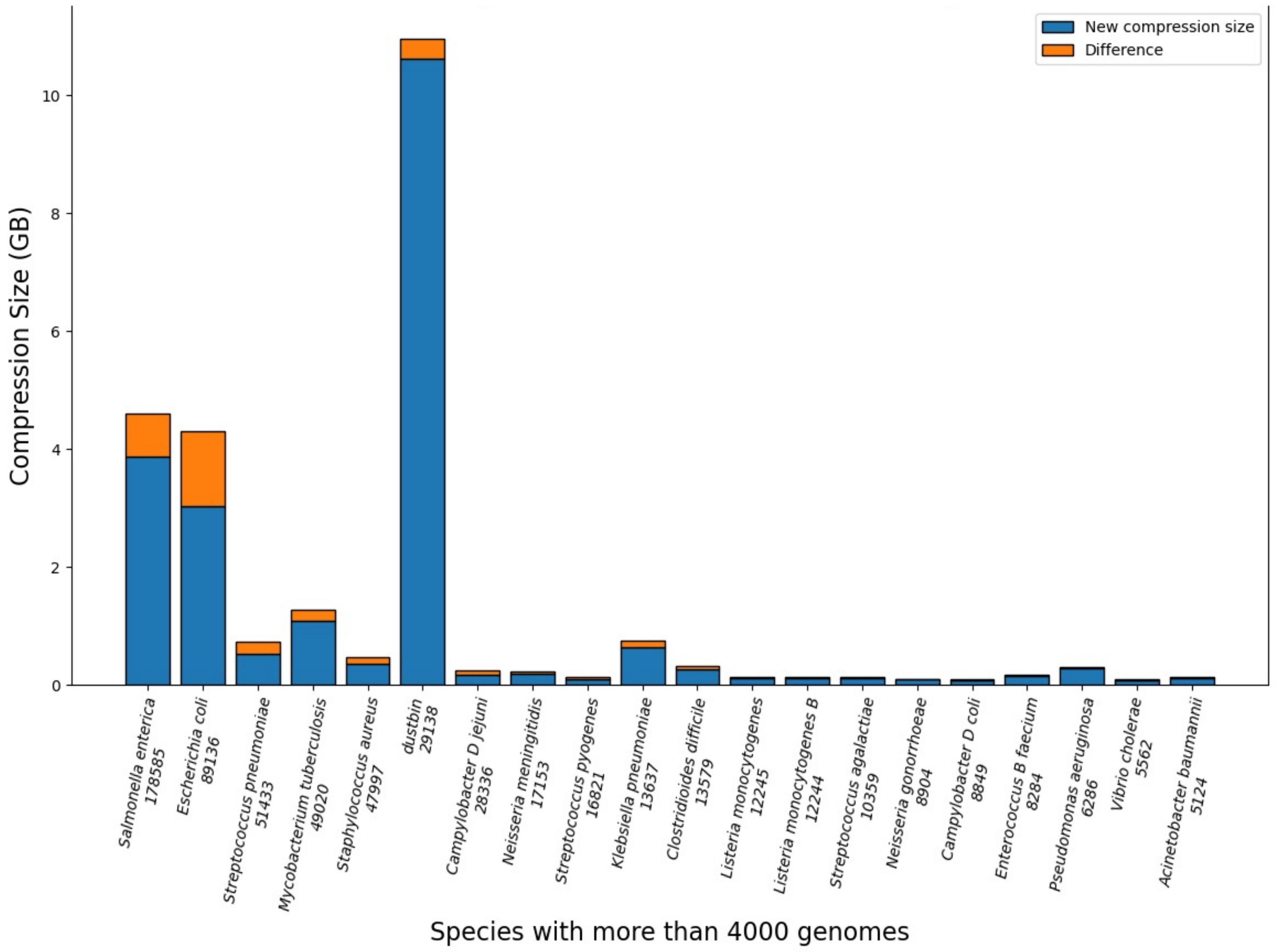


Compression size of
3 most highly sampled species & dustbin [GB]

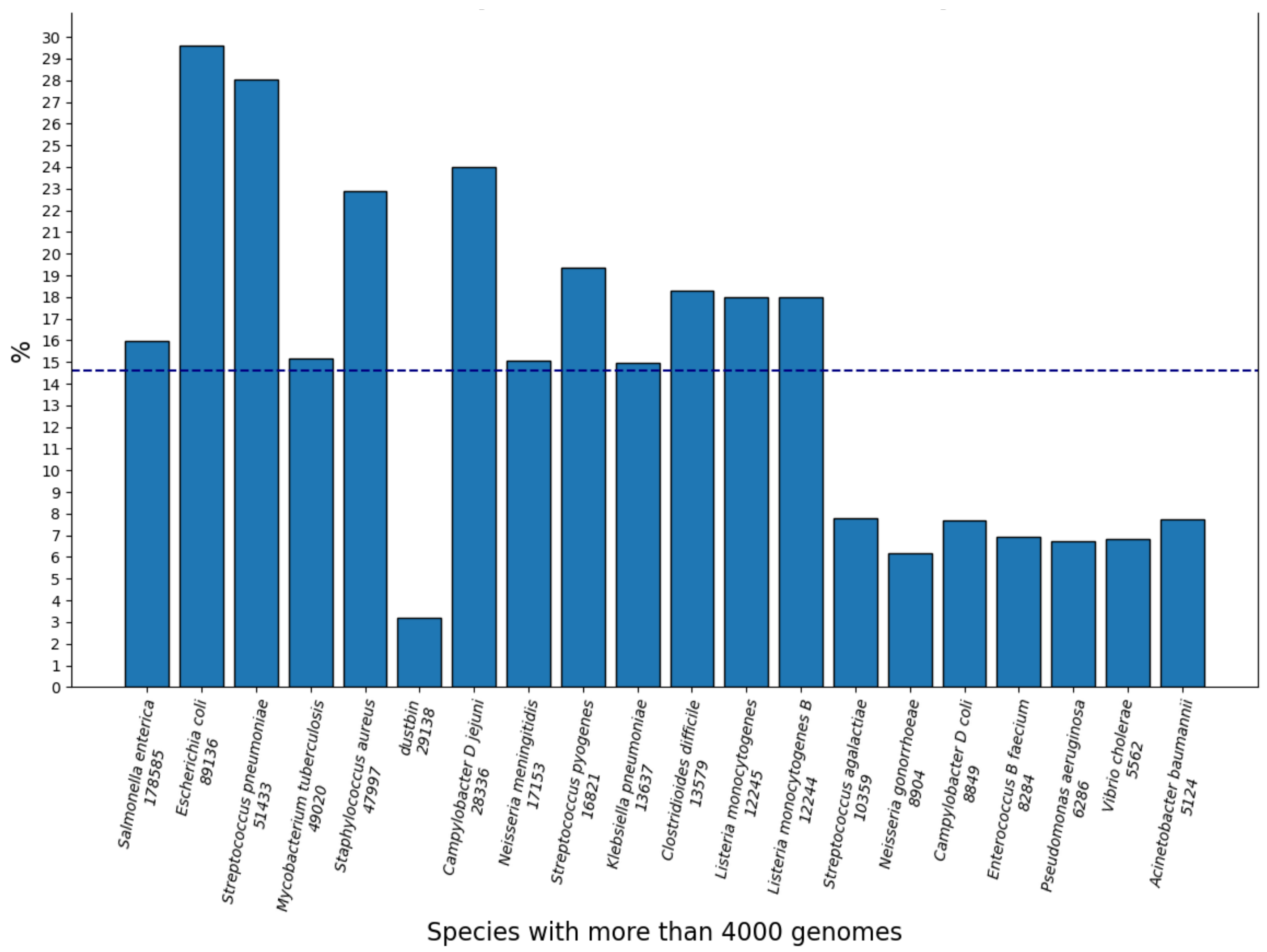
	Original	**New**
S. enterica	4.6	3.8
E. coli	4.3	3.0
S. pneumoniae	0.7	0.5
Dustbin	10.9	10.6

(Fenske et al. 2024)

Result (661k) Species-wise: Absolute Compressed Size Reduction P. Species



Result (661k) Species-wise: Relative Compressed Size Reduction Per Species

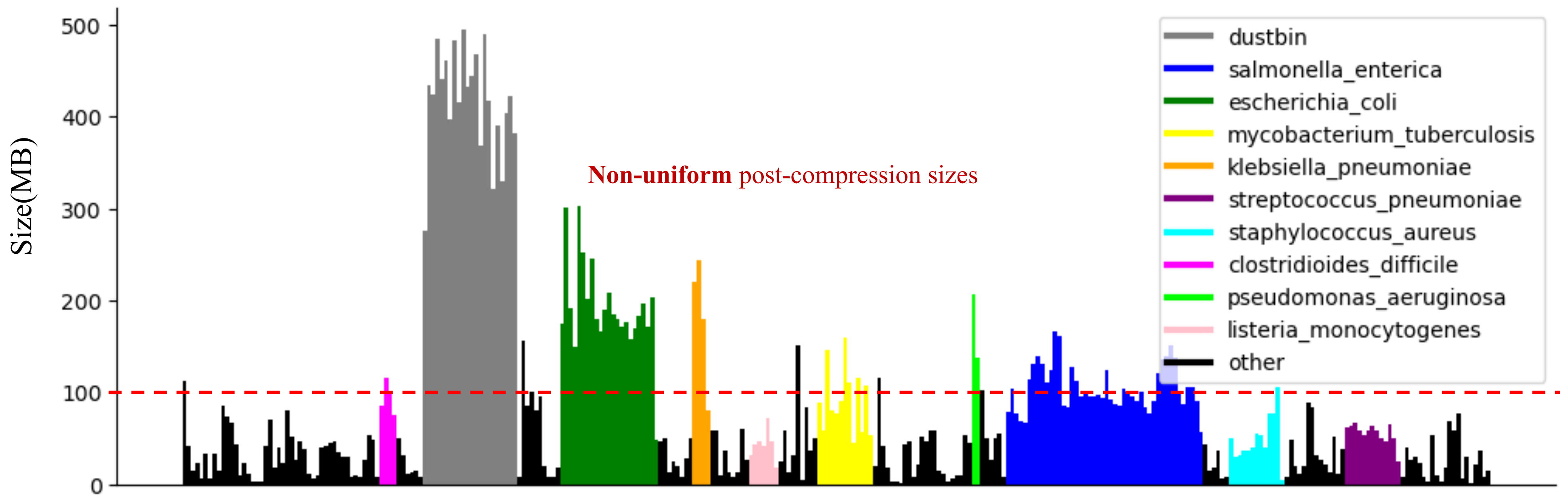


Axis 2: Order-based Genome Batching

Limitation Of The Current Batching Results

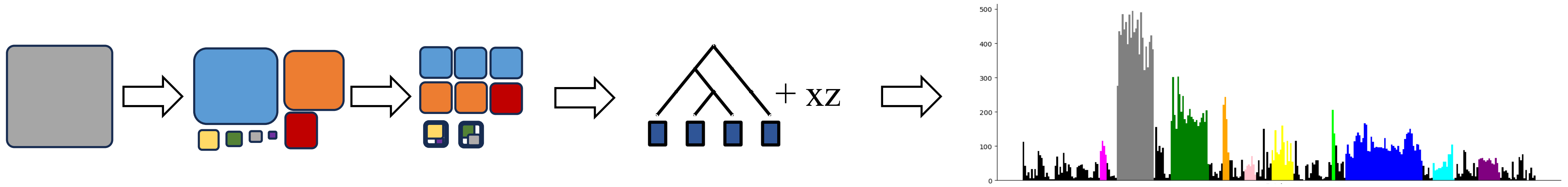
Reminder: After batching (step 1), batches are reordered (step 2) then compressed

Compression result of 661k Batches



Consequences and Applications of Uniform Batches

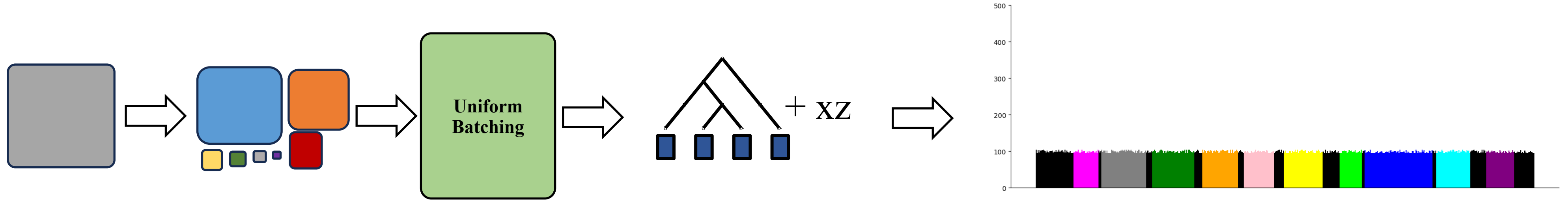
MiniPhy:



Consequences of Non-uniformity:

Unbalanced Workloads Inefficient Transmission Hinder Parallelization Memory Overuse Inconsistent Query Time

Objective:



Application for balance batches: Portable Devives (remote setting, field work, rapid diagnostic), Parallel Platforms (GPU, Processing-in-Memory)

Bin Packing For Microbial Genomes Compression

Inputs:

- $G = \{g_1, g_2, \dots, g_n\}$: ***ordered set*** *of genomes*
- *Species labels*
- *We want to split G into m ordered batches,* $m \le n$
- $B = \{b_1, b_2, \dots, b_m\}$: *set of* ***ordered batches***

Parameters:

- c : *bound on compressed size*

Decision Variables:

- $x_{ij} \in \{0,1\}$: *1 if genome g_i is in batch b_j, 0 otherwise*
- $y_j \in \{0,1\}$: *1 if batch* b_j *is used, 0 otherwise*

Functions:

- $C(b_j) = f_{compressed}(\text{ordered } \{g_i : x_{ij} = 1\})$

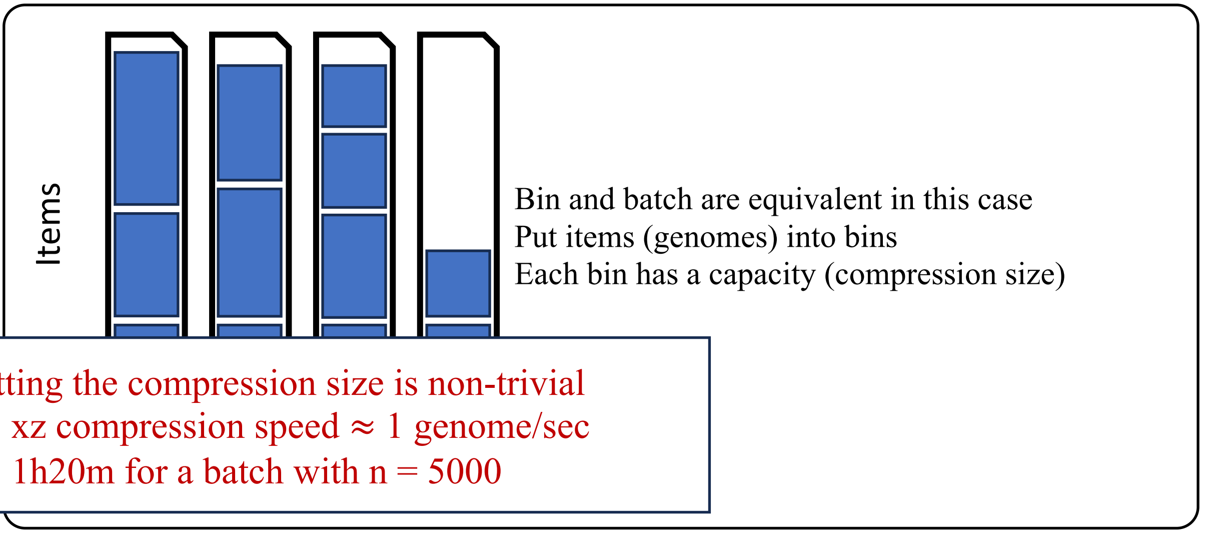
Subjects to :

1) $\sum_{j=1}^{m} x_{ij} = 1 \quad \forall i \in \{1, \dots, n\}$: *a genome must be assigned*
2) $x_{ij} \le y_j \quad \forall i, j$: *no genomes in unselected batches*
3) $C(b_j) \le c \cdot y_j \quad \forall j \in \{1, \dots, m\}$: *bound on compressed size*

Objective function:

$$\min \cancel{\sum_{j=1}^{n} C(b_j) \cdot y_j} + \sum_{j=1}^{n} y_j$$

Bin packing [1]

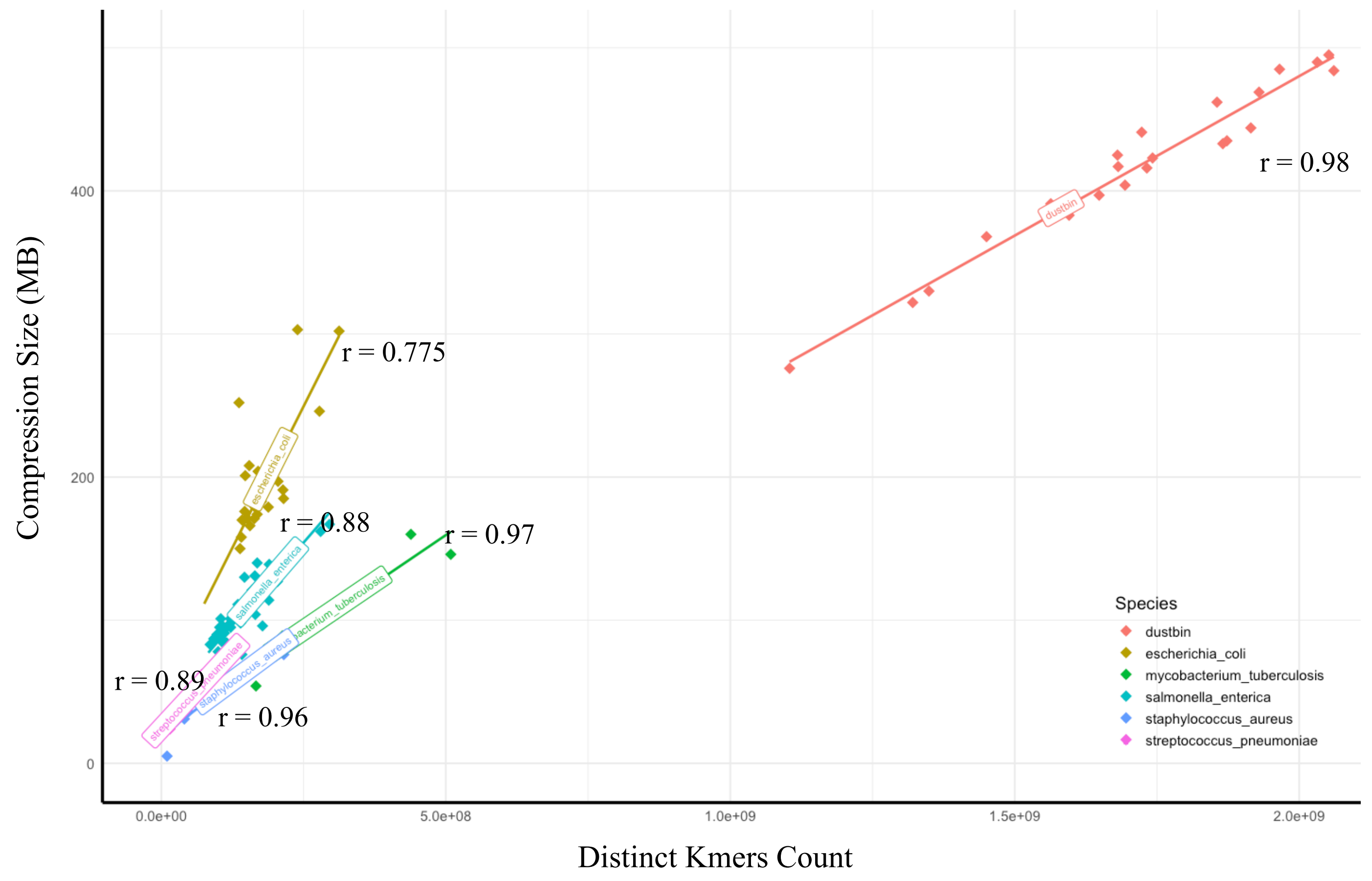


Bin and batch are equivalent in this case
Put items (genomes) into bins
Each bin has a capacity (compression size)

Getting the compression size is non-trivial
➔ xz compression speed ≈ 1 genome/sec
➔ 1h20m for a batch with n = 5000

Compression Size Estimation Using Proxy: Distinct K-mers Counts

Compression Size Vs Distinct Kmers Count – 661k Collections – Top 5 Highly Sampled Species & Dustbin

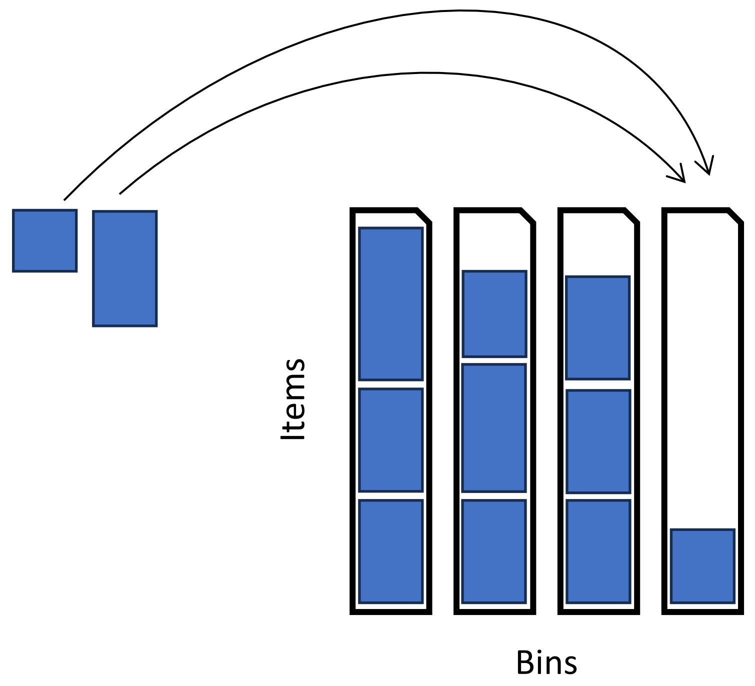


K-mers : length k substring of a genomes

K-mers estimation using HyperLogLog sketching [1,2]

[1] Baker and Langmead 2019; [2] Flajolet et al. 2007

First-fit Bin Packing Algorithm



Fit item in the first available bin

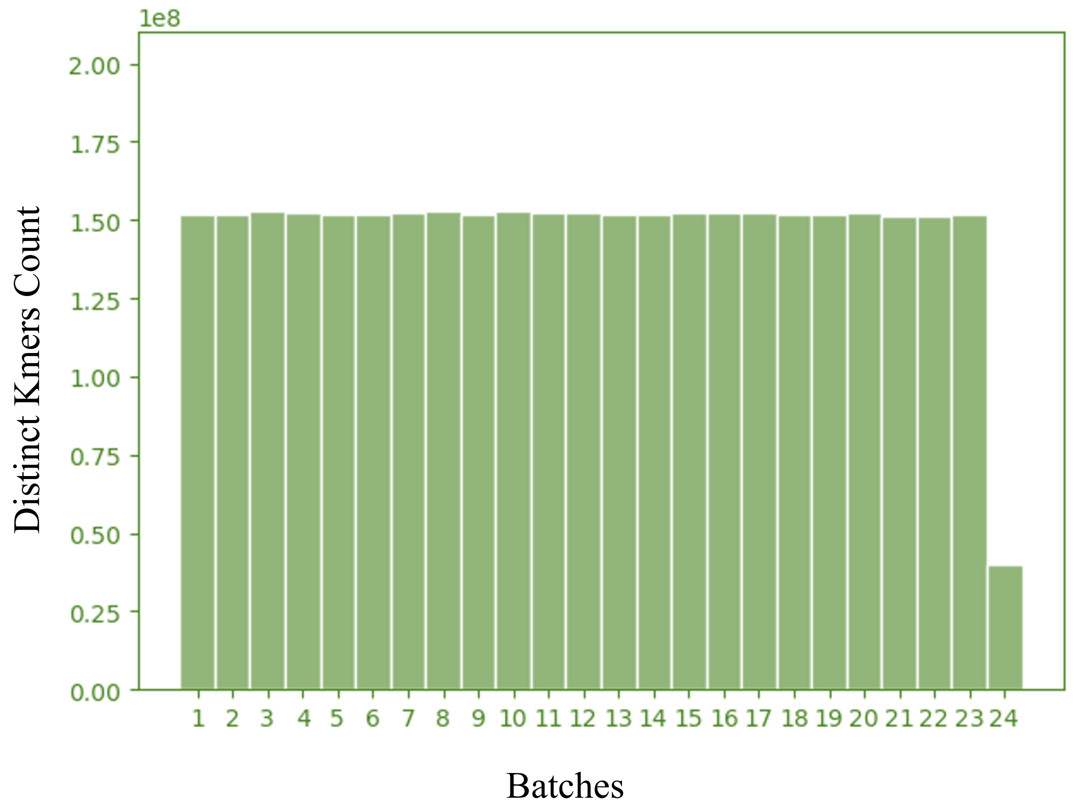
Preserve the order of the input items

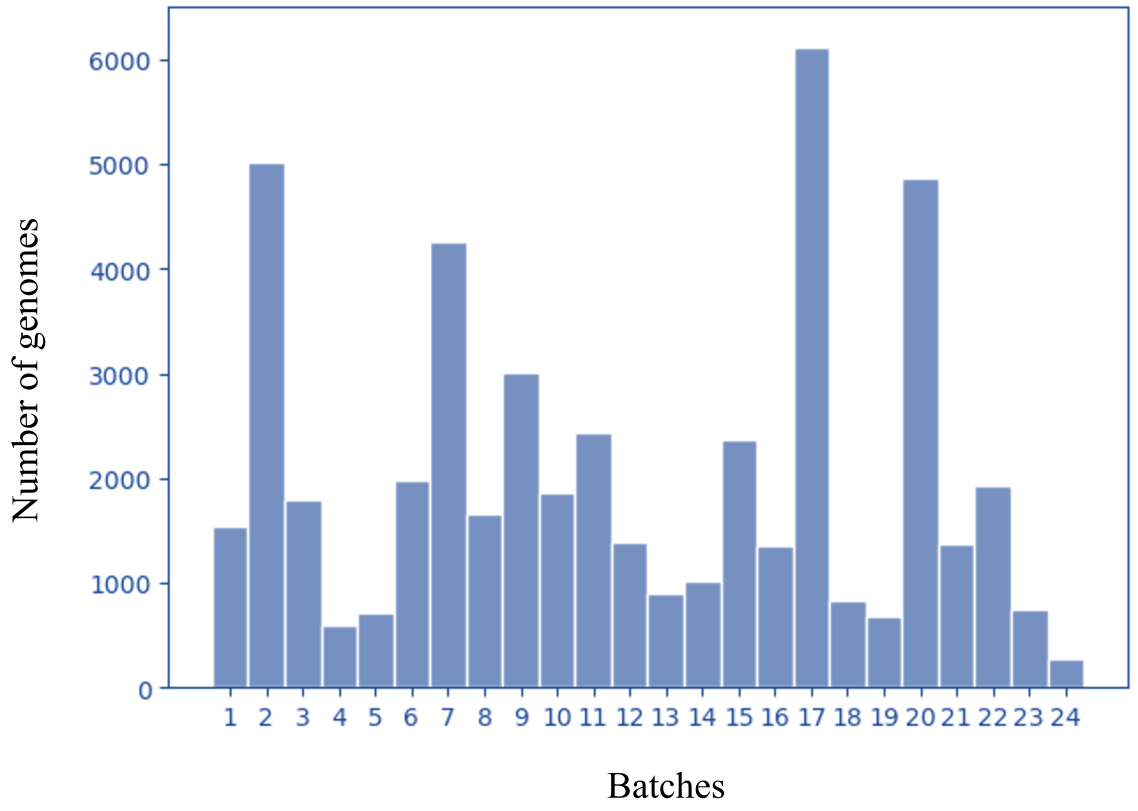
Axis 2 Experimental Result:

- Dataset: Assemblies of *Mycobacterium tuberculosis* from 661k collection
- Number of Genomes: around 49,000
- We want the compression size stay below 64MB (DRAM for PIM [1,2])
    - CAPACITY (distinct kmers count) of batches: 152,000,000

Ghose et al. 2019; Mutlu et al. 2019

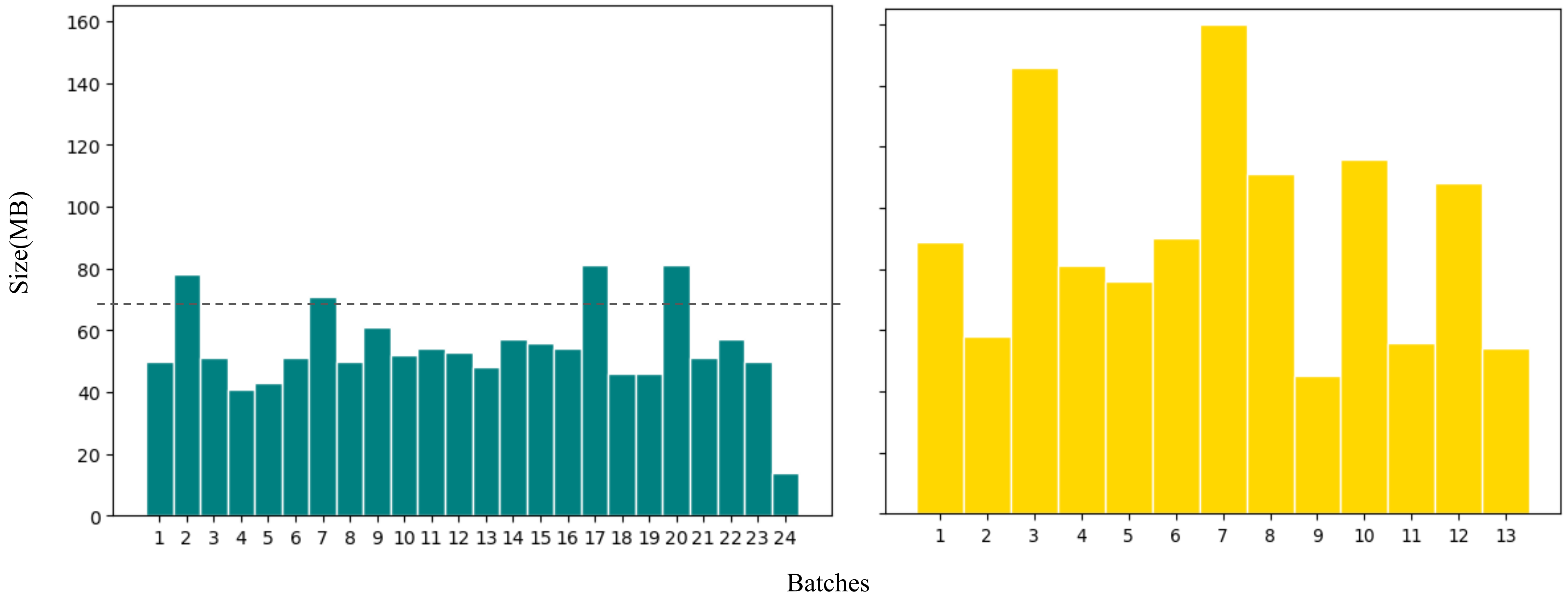
Distinct Kmers Count And Number Of Genomes Per Batches





Balanced distinct kmers count, varied number of genomes per batches

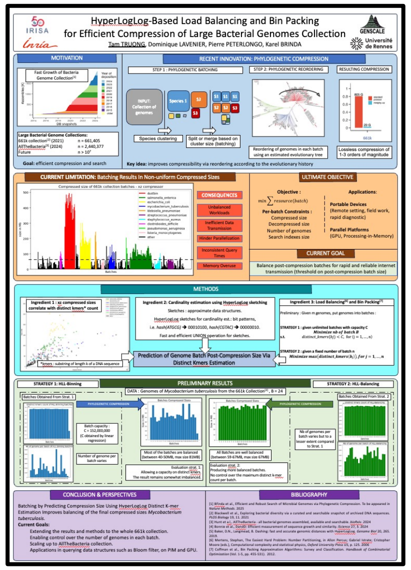
Compression size per batch

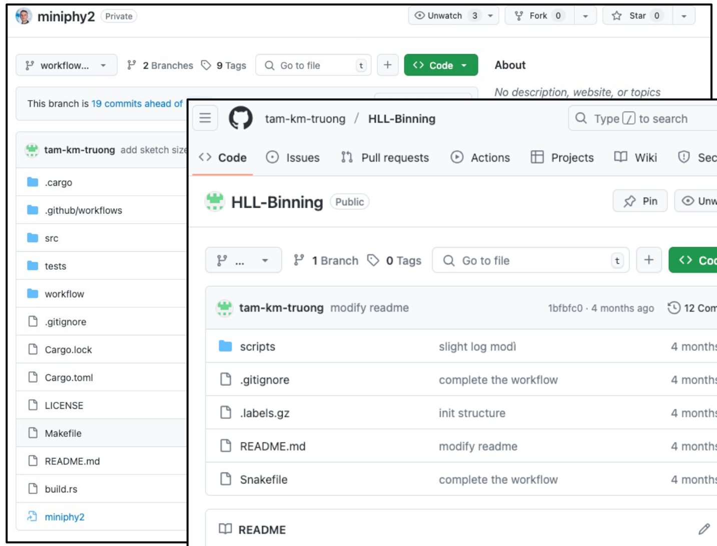


Most batches stay below 64MB and relatively balanced (except 4 batches)

Outcome and Next Steps

- Poster at the DKM department seminar
- Presentation an internal GenScale meeting
- **Next step:**
  - Species-oblivious pre-ordering
  - Improve prediction accuracy
  - Long term: extend to other genomic representation, such as indexes
- **Future deliverables:**
  - Presentation at upcoming SeqBim and DSB conferences
  - Tools:
    - Genome ordering and batching tool
    - Co-developing MiniPhy2 – released soon
  - Submit to conferences such as RECOMB and ISMB/ECCB
  - Eventual Publication in bioinformatics journals
  - Collaboration: incorporated into the next update of AllTheBacteria


HyperLogLog-Based Load Balancing and Bin Packing
for Efficient Compression of Large Bacterial Genomes Collection
Tam TRUONG, Dominique LAVENIER, Pierre PETERLONGO, Karel BRINDA


miniphy2 Private

workflow... 2 Branches 9 Tags Go to file Code

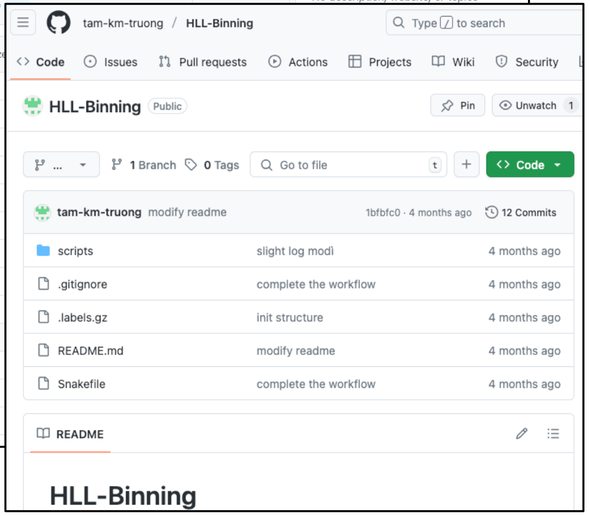
About

No description, website, or topics

This branch is 19 commits ahead of

tam-km-truong add sketch size

.cargo
.github/workflows
src
tests
workflow
.gitignore
Cargo.lock
Cargo.toml
LICENSE
Makefile
README.md
build.rs
miniphy2


tam-km-truong / HLL-Binning

HLL-Binning Public

1 Branch 0 Tags Go to file Code

tam-km-truong modify readme 1bfbfc0 · 4 months ago 12 Commits

scripts	slight log modi	4 months ago
.gitignore	complete the workflow	4 months ago
.labels.gz	init structure	4 months ago
README.md	modify readme	4 months ago
Snakefile	complete the workflow	4 months ago

README

HLL-Binning