Optimization Framework For Phylogenetic Compression

Comité de Suivi Individuelle (1st year)

Tam Truong, *Dominique Lavenier, Pierre Peterlongo, Karel Břinda*

25 June 2025

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

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

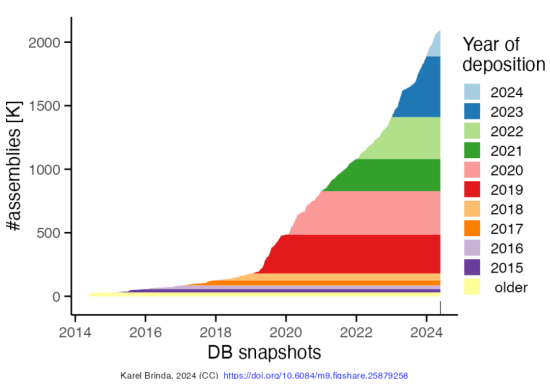
at the

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

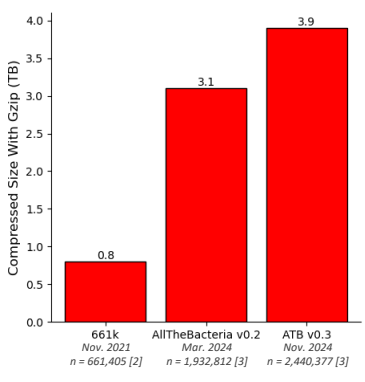
My PhD Thesis

Motivation: Rapidly Growing Bacteria Genome Data & Collections

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



Multi-Terabytes Microbial Genome Collections

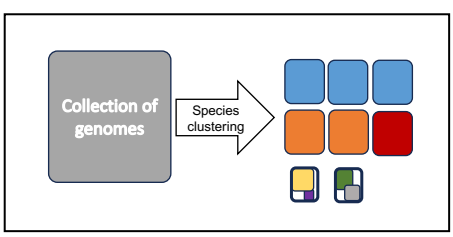


18 Jun 2025: ATB v0.4 with 330k new genomes
Next decade: Even larger collections (n = ~10^7), higher diversity, …

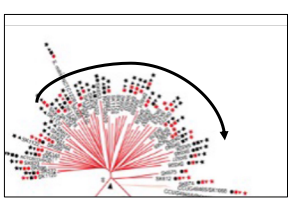
Challenging: Efficient storage & analysis (indexing, searching)
Standard compression protocol is not sufficient

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

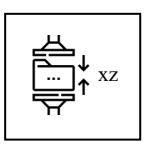
Phylogenetic Compression: Key Steps – Implemented in **MiniPhy**



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



Step 2 : Phylogenetic Reordering Per Batch

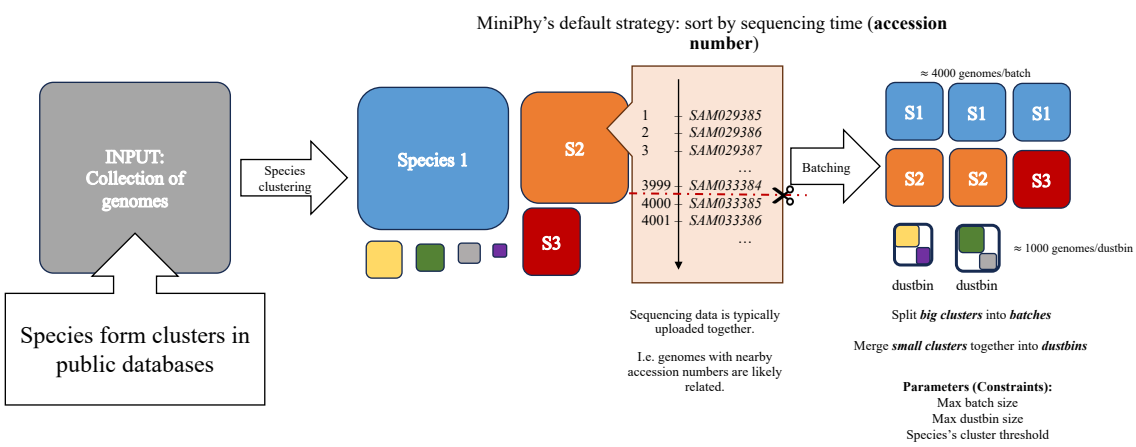


Step 3 : Compression

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

Key Concepts

Batching In Phylogenetic Compression (MiniPhy)



MiniPhy: github.com/karel-brinda/miniphy

Lim

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

Opt

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

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

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*

The Two Tracks Of My Work: Preordering & Partitionning

OBJECTIVE:

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

1. Phylogenetic Pre-ordering Of Genomes
2. Order-based Genome Batching

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

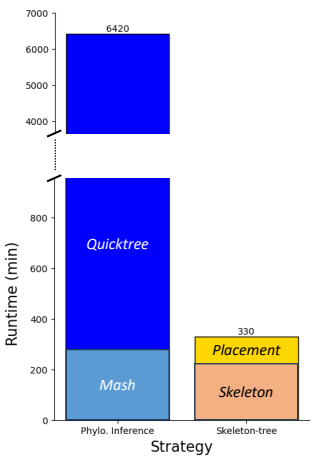
Infere phylogenetic tree for each species : Distance estimation (Mash) + Neighbor joining tree (Quicktree)

Challenging for highly sampled species (>50k genomes)

Infeasiable for large modern collection (million-genome scale)

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

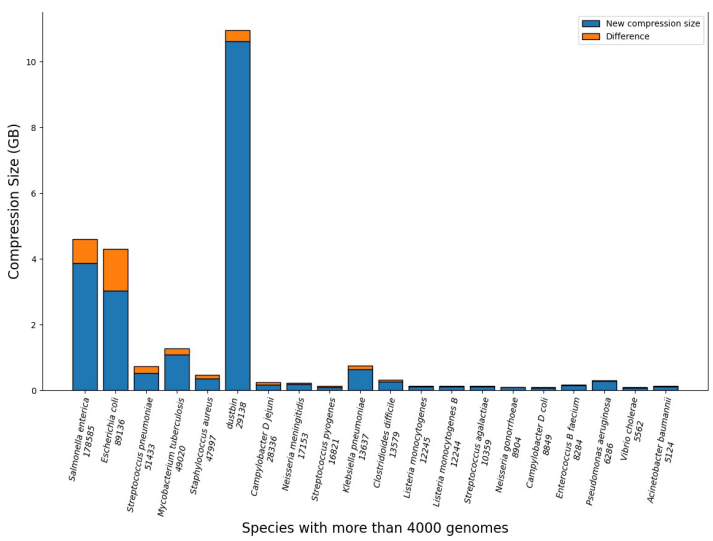
Compression result:

- Cut into batches of 4000 genomes then compress with xz

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

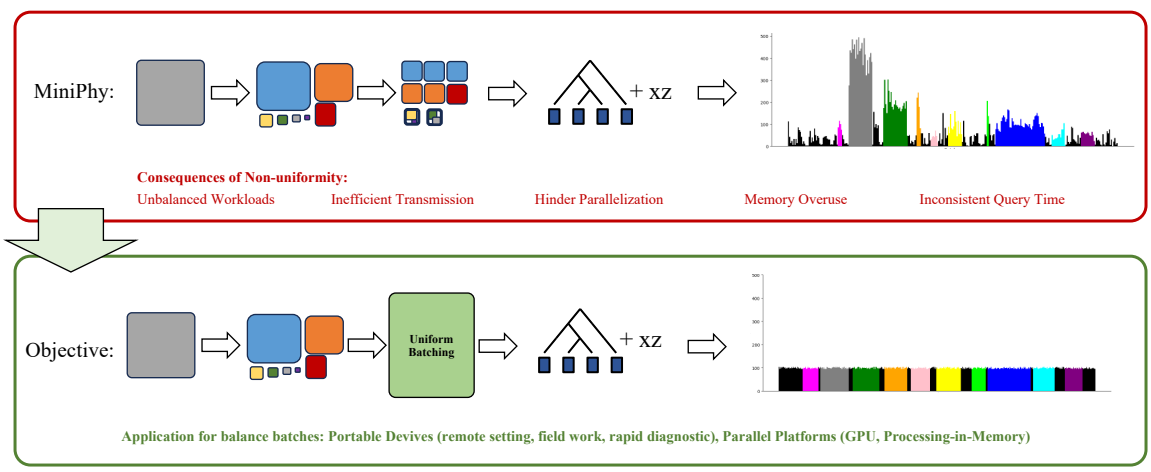
Significantly faster and produce similar compression result

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



Axis 2: Order-based Genome Batching

Consequences and Applications of Uniform Batches



Bin I

Inputs:

Parameter

Decision V

Functions:

Subjects to

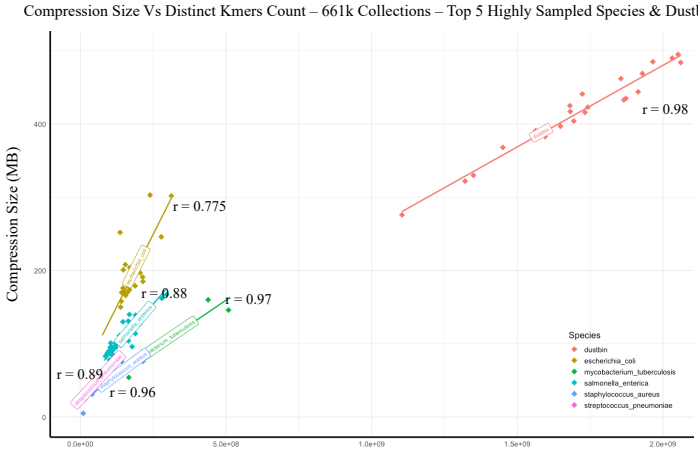
1)

2)

3)

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

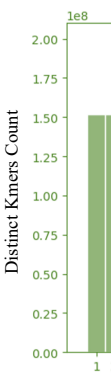
First-

Axis 2 Experimental Result:

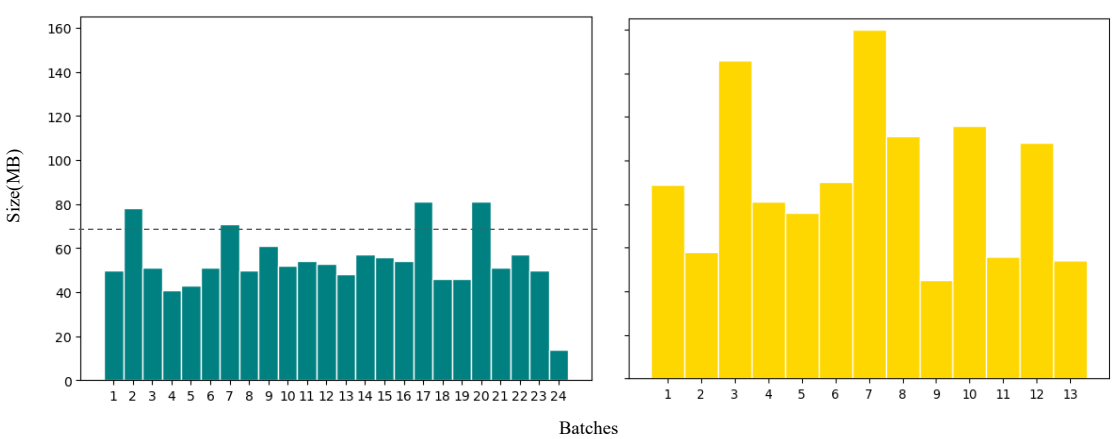
- Dataset: Assemblies of *Mycobacterium tuberculosis* from 661k collection
- Number of Genomes: around 49,000
- We want the compression size stay below 64MB (DRAM for PIM [1,2])
  - CAPACITY (distinct kmers count) of batches: 152,000,000

Ghose et al. 2019; Mutlu et al. 2019

Disti



Compression size per batch



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

Outc

- Pos
- Pres
- **Nex**
  -
  -
  -
- **Fut**
  -
  -
  -
  -
  -