# HyperLogLog-Based Load Balancing and Bin Packing for Efficient Compression and Querying of Bacterial Genomes

## otivation

eria genomes is growing rapidly

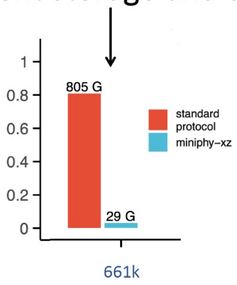
**Increasing Availability of Larger Bacterial Genome Collections**

661k (Blackwell et al., PLOS Biology, 2021) n = 661,405

AllTheBacteria (Hunt et al., bioRxiv, 2024) n = 2,440,377

FUTURE : collection of ten of millions bacteria genomes

fficient storage and query on bacteria collections

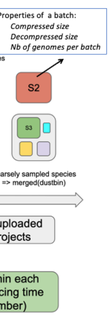


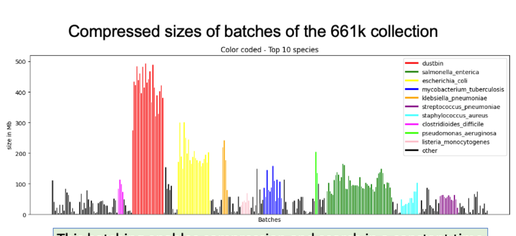

High compression ratio
Compressed using MiniPhy:
*https://github.com/karel-brinda/miniphy*

**Phylign:** tool for search of across the compressed 661 collection on a standard laptop.

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

c compression: clustering/batching




This batching enables compression and search in a constant time (parallelization) or linear time (using a single machine)

This batching is not adapted for different hardware with different memory constraint
The compressed size of the batch is not balanced

straint (i.e. memory) while minimizing the number of batches
mber of of genomes in the batches)

Bin Packing Optimization Problem

**Instance**:

Finite set I of items, a size s(i) ∈ Z + for each i ∈ I.

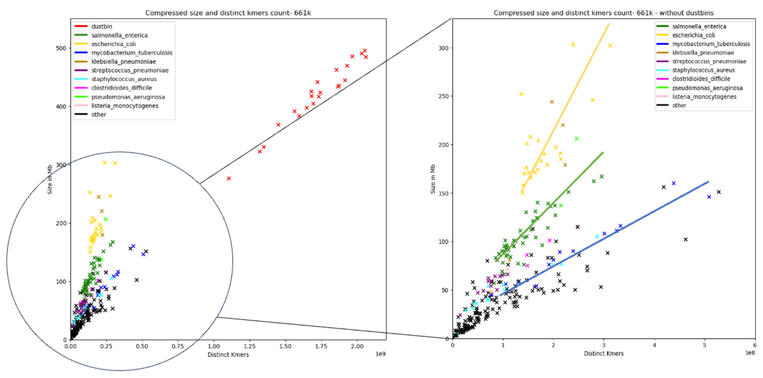
Given a capacity K for each batch

**Objective**: Partition the items in set I into batches and minimize the number of batches used

blem are extensively studied
t there are efficient heuristic algorithms
pt these problem for DNA data

## Method

Compressed size of batches can be estimated using biological property




k-mers: substring of length k of DNA sequences

Correlation between kmer and compressed size of the batches

Different correlation for different species

➔ **Distinct kmers count can be used to set the capacity of batches**

Fast distinct kmer estimation using probabilistic counting – HyperLogLog sketching

**Sketching:** generating an approximate, compact summary of data (a sketch)

TCGATCGATCGATCGA
TCGAT
CGATC
GATCG
...
ATCG
TCGA

**Probabilistic counting/approximate counting:**

Sketches
Hash(item_1) = 0000... => Prob = 1 / $2^4$
Hash(item_2) = 0001...
Hash(item_3) = 0010...
Hash(item_4) = 0011...
Hash(item_5) = 0100...
Hash(item_6) = 0101...
...

Longest prefix of leading zeros in the hash values is **4**
Estimated cardinality ≈ **$2^4$**

**Set cardinality estimation with Dashing:**

Hyperloglog is a sketching algorithm based on advanced approximate counting.

Baker, D.N., Langmead, B. implemented it in the tool dashing.

*https://github.com/dnbaker/dashing*

## Result

### Hyperloglog based bin packing and load balancing for genomes

**HLL-binning:**
Put genomes into batches that fit within a user-defined memory constraint

**INPUT: GENOMES SKETCHES, CAPACITY (max_distinct_kmers)**
**Initialize:**
Sort the genomes based on their accession number.
**Place Items in BATCH:**
For each genome:
Try to place it in the first batch that can accommodate it (based on capacity).
If it fits, update the batch and move on to the next genome.
**Create New BATCHES if Necessary**:
If no batch can accommodate the current genome, create a new batch and place the genome there.

https://github.com/tam-km-truong/HLL-Binning

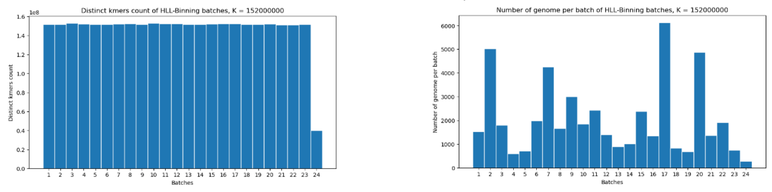
**HLL-balancing:**
Put genomes into balanced, user-defined number of batches

**INPUT: GENOMES SKETCHES, NUMBER OF BATCH b**
**Initialize:**
Sort the genomes based on their accession number.
**Initial Assignment:**
Assign the first b genomes directly to b batches.
**Greedy Assignment**:
For each remaining genomes, place it in the batches with the smallest number of distinct kmers.
Update the batch's contents and size.

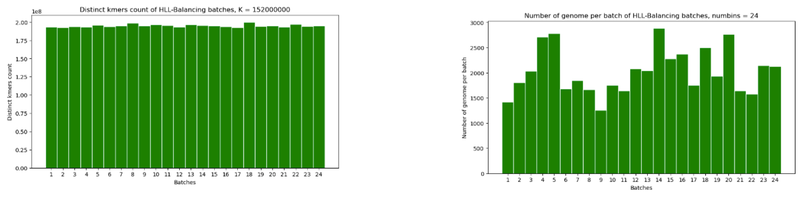
https://github.com/tam-km-truong/HLL-Balancing

### EXPERIMENT: batching of Mycobacterium Turberculosis

Max distinct kmers for batches: 153000000 kmers so that the compressed size stays under 64MB (caculated by using the correlation of compressed size and distinct kmers count)




Using the number off batches found by HLL-Binning, run HLL-Balancing with numbins b = 24




Eventhough the baches are balances, HLL-Balancing batches have higher distinct kmers count than the capacity set in HLL-Binning

## Conclusion and

**Conclusion**

- Genomic data benefits greatly from c
  when guided by evolutionary charact
- Clustering and batching of genome c
  compression ratio and facilitate paral
- The compressed size of batches (usin
  correlated with the distinct k-mer cou
- The distinct k-mer count can be effici
  HyperLogLog sketching (implemented
- The Bin Packing heuristic algorithm c
  defined capacity, i.e., distinct k-mer c
- The Load Balancing heuristic algorith
  balanced batches.

**Perspectives**:

- Combining HLL-Binning and HLL-Bala
- Studying the collection as a whole ins
- Extending the scope to include differ
  graph).

## Bibliogra


Grace A. Blackwell, Martin Hunt, Kerri M. Malone, Leandro L
Thomson, and Zamin Iqbal. 2021. Exploring bacterial diversit
archived DNA sequences. *PLOS Biology* 19, 11 (November 20

Martin Hunt, Leandro Lima, Daniel Anderson, Jane Hawkey,
*AllTheBacteria - all bacterial genomes assembled, available*

Karel Břinda, Leandro Lima, Simone Pignotti, Natalia Quinon
Kucherov, Zamin Iqbal, and Michael Baym. 2024. *Efficient an*
*Phylogenetic Compression*. bioRxiv.

Po-Ru Loh, Michael Baym, and Bonnie Berger. 2012. Compre
2012)

Will P. M. Rowe. 2019. When the levee breaks: a practical gu
flood of genomic data. *Genome Biology* 20, 1 (September 20

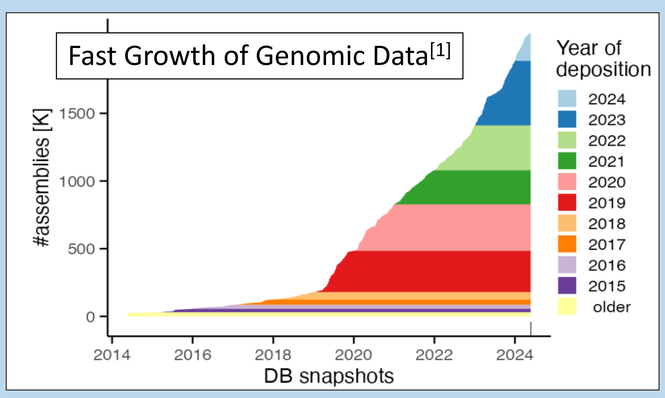
Jessica K. Bonnie, Omar Y. Ahmed, and Ben Langmead. 2024
growth and similarity. *iScience* 27, 3 (March 2024).

# HyperLogLog-Based Load Balancing and Bin Packing for Efficient Compression of Large Bacterial Genomes Collection

Tam TRUONG, Dominique LAVENIER, Pierre PETERLONGO, Karel BRINDA

IRISA Inria — GENSCALE Université de Rennes

## MOTIVATION




**Large Bacterial Genome Collections:**

661k collection[2] (2021) n = 661,405

AllTheBacteria[3] (2024) n = 2,440,377

Future collections could have 10 of millions genomes

## PHYLOGENETIC COMPRESSION: REORDERING STEPS

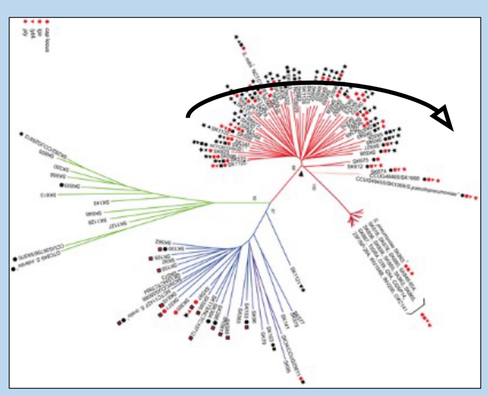
### STEP 1 : PHYLOGENETIC CLUSTERING/BATCHING


Species clustering

Split or merge based on cluster size (batching)

The collection is first partitioned into species clusters. These clusters are then split or merged into batches based on the number of genomes.

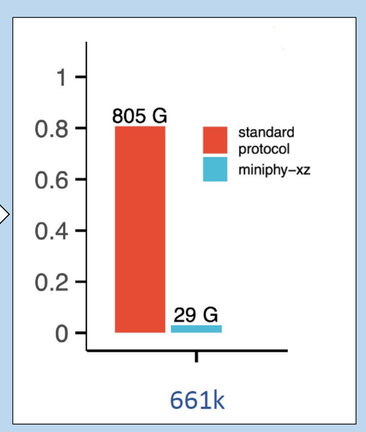
### STEP 2: COMPRESSIVE PHYLOGENY



Reordering of genomes in each batch using an estimated evolutionary tree

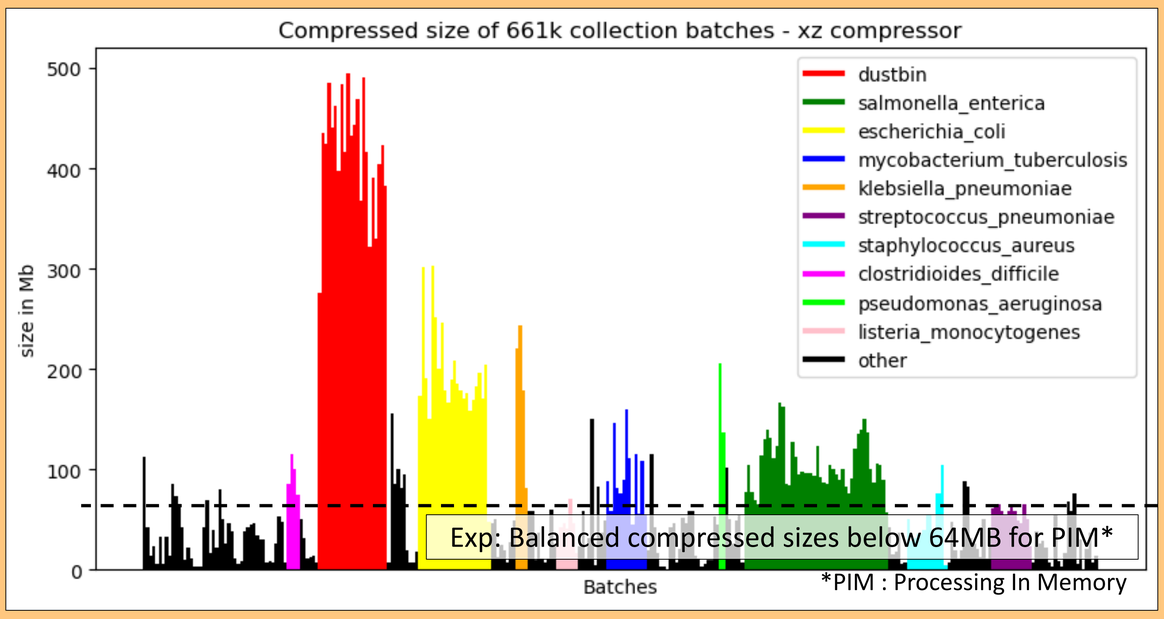
## COMPRESSION

### STEP 3: COMPRESSION




Lossless compression of 1-3 orders of magnitude

## CURRENT LIMITATION: Batching Results In Non-uniform Compressed Sizes




*PIM : Processing In Memory

**PROBLEM**

Unreliable data transmission over bad network

Hinder Parallelization

## ULTIMATE OBJECTIVE

**Given**

Clusters of genomes

Hardware platform

**Objective :**

$$minimize \sum ressource(batch)$$

**Per-batch Constraints :**

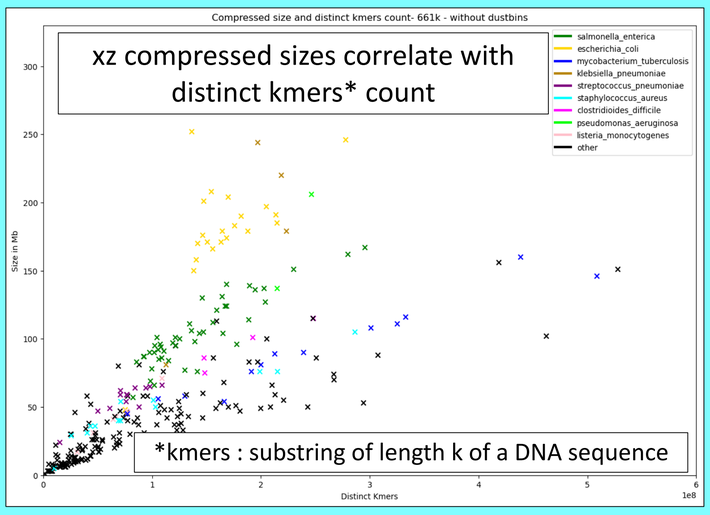
- Bounds on compressed size
- Bounds on decompressed size
- Bounds on number of genomes

➔ **For:** - Data Transmission - Parallelization

## CURRENT GOAL

Create balanced batches after compression that stay below a size constraint and minimizing the number of batches needed

## METHODS




Cardinality estimation using HyperLogLog sketching

Sketches : approximate data structures.
HyperLogLog sketches : bit patterns,
i.e. *hash(ATGCG)* ➔ 00010100, *hash(CGTAC)* ➔ 00000010.
Fast and efficient UNION operation for sketches.

Implemented in the Dashing[5] tool.
Average relative error [4] : $6.537 \times 10^{-4}$

**MAIN IDEA:**

Prediction of Genome Batch's Compression Size Using Distinct Kmers Estimation

HyperLogLog Based Load Balancing[6] and Bin Packing[7]

**HLL-binning:**

Given unlimited batches with capacity C, m genomes, put genomes into batches:

$$\boldsymbol{Minimize\ nb\_of\_batch\ B} = \sum_{1}^{n} \boldsymbol{b_j}\ , \qquad \boldsymbol{for\ (j = 1, \dots, n)}$$

**s.t.** $distinct\_kmers(b_j) < C,\ \text{for (j} = 1, \dots, \text{n)}$

Greedy implemetation : first-fit bin packing

https://github.com/tam-km-truong/HLL-Binning

**HLL-balancing:**

Given n batches, m genomes, put genomes into batches:

$$\boldsymbol{Minimize\ max\left(distinct\_kmers(b_j)\right), for\ (j = 1, \dots n)}$$

Greedy partitioning algorithm :

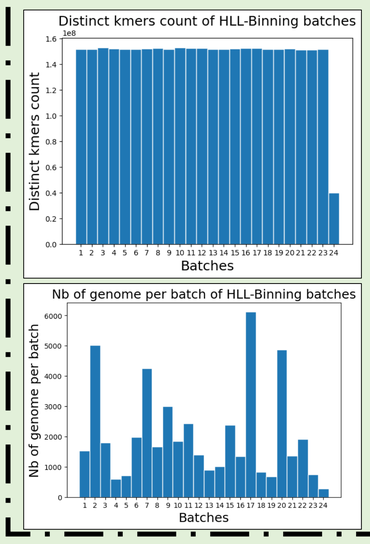
https://github.com/tam-km-truong/HLL-Balancing

## PRELIMINARY RESULTS

DATA : Genomes of Mycobacterium Tuberculosis (MTB) from the 661k Collection[2]

### STRATEGY 1 HLL-Binning

Batches Obtained From Strat. 1

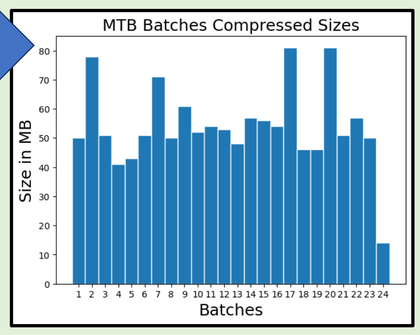



COMPRESS

Most batches have balance distinct kmers count

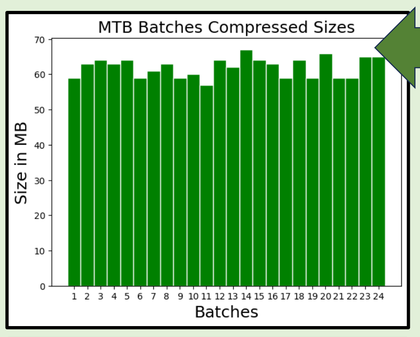
Batch capacity :
C = 152,000,000
(C obtained by linear regression)

Number of genome per batch varies




Most of the batches are balanced (between 40-50MB)

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

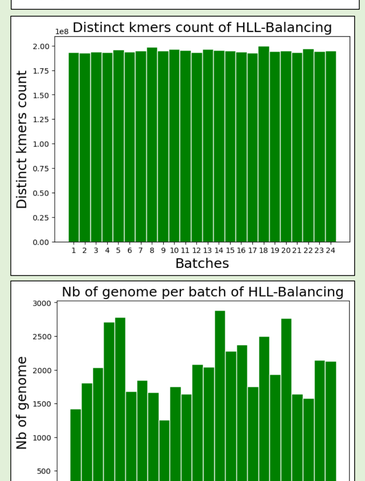



All Batches are well balanced (max size is 69MB)

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

### STRATEGY 2 : HLL-Balancing

Batches Obtained From Strat. 2




COMPRESS

Most batches have balance distinct kmers count

Nb of batch
B = 24

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

## CONCLUSION & PERSPECTIVES

**Batching by Predicting Compression Size Using HyperLogLog Distinct K-mer Estimation:**

Improves balancing of the final compressed sizes.

Allows for better control over compression sizes.

**Perspectives:**

Extend the results to the whole collection.

Enable control over the number of genomes in each batch.

## BIBLIOGRAPHY

*[1] Karel Břinda et al., 2024. Efficient and Robust Search of Microbial Genomes via Phylogenetic Compression. Nature Methods.*

*[2] Grace A. Blackwell et al., 2021. Exploring bacterial diversity via a curated and searchable snapshot of archived DNA sequences. PLOS Biology 19, 11*

*[3] Martin Hunt et a.l, 2024. AllTheBacteria - all bacterial genomes assembled, available and searchable. bioRxiv.*

*[4] Jessica K. Bonnie et al., 2024. DandD: Efficient measurement of sequence growth and similarity. iScience 27, 3*

*[5] Daniel N Baker, Ben Langmead, 2019. Dashing: Fast and Accurate Genomic Distances with HyperLogLog,, bioRxiv*

*[6] Mertens, Stephan, 2006, The Easiest Hard Problem: Number Partitioning, in Allon Percus; Gabriel Istrate; Cristopher Moore (eds.), Computational complexity and statistical physics, Oxford University Press US, p. 125.*

*[7] Coffman et al., 2012. Bin Packing Approximation Algorithms: Survey and Classification. 10.1007/978-1-4419-7997-1_35.*

# Motivation

**HLL-binning:**

$$\text{Minimize } B = \sum_{1}^{n} b_j \text{ , for } (j = 1, \dots, n)$$

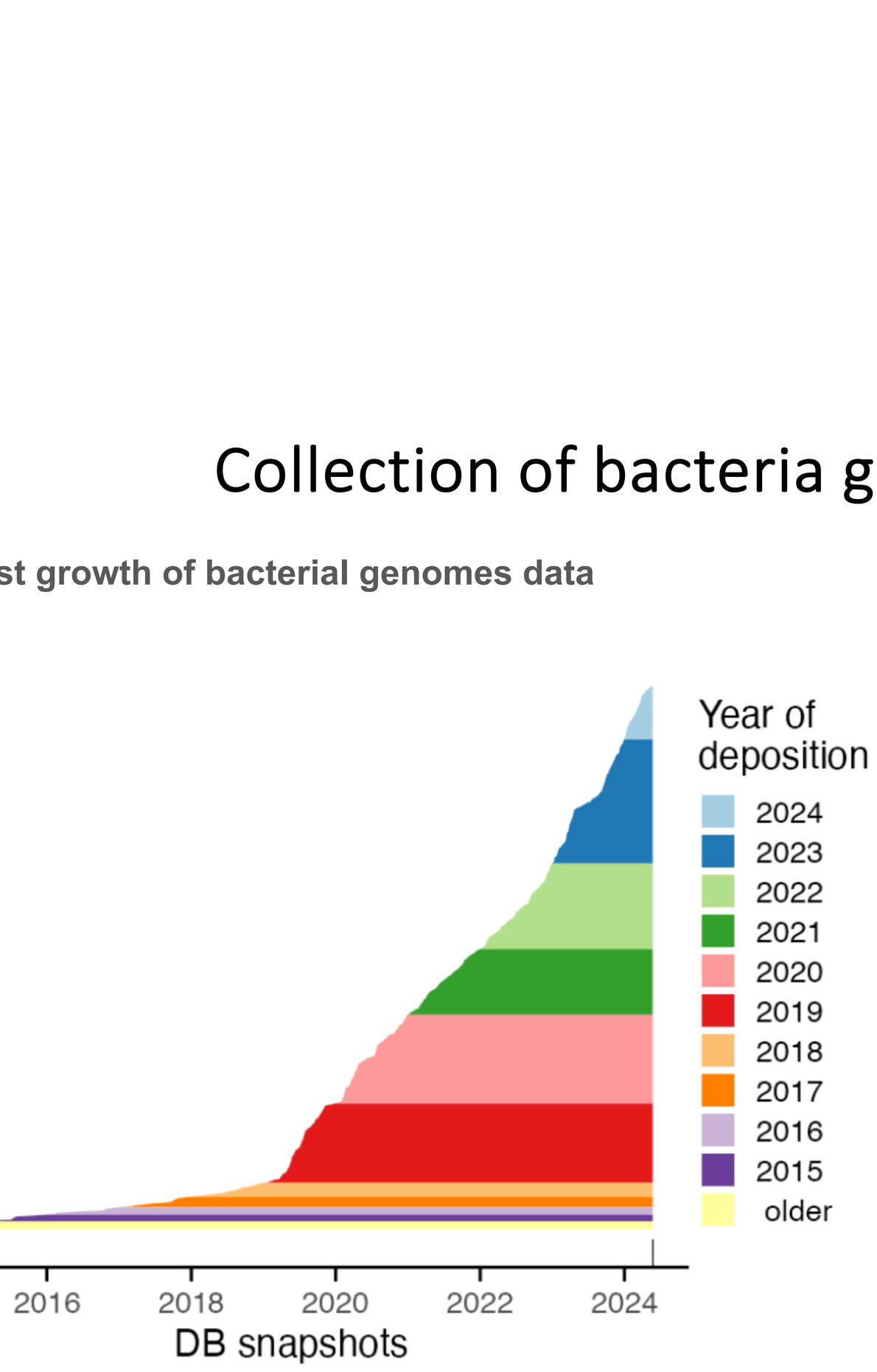
**s.t.**

$$|b_j| < C, \text{ for } (j = 1, \dots, n)$$

$$\sum_{i=1}^{m} x_{ij} = 1, \text{for } (j = 1, \dots, n)$$

$$b_j \in \{0,1\}, x_{ij} \in \{0,1\}$$

# Collection of bacteria genomes is growing rapidly




[Brinda et al.,bioRxiv, 2024]

**Increasing Availability of Larger Bacterial Genome**

661k (Blackwell et al., PLOS Biology, 2021) n = 6

AllTheBacteria (Hunt et al., bioRxiv, 2024) n = 2,4

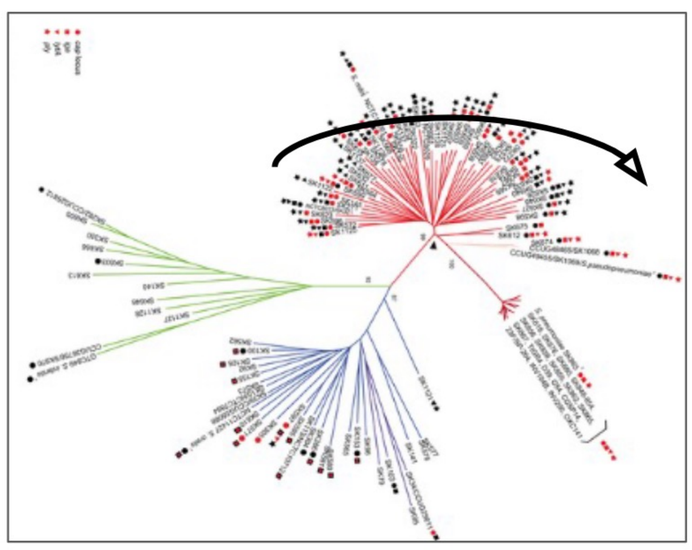
FUTURE : collection of ten of millions bacteria genomes,

# ic compression allows efficient storage and query on bacteria

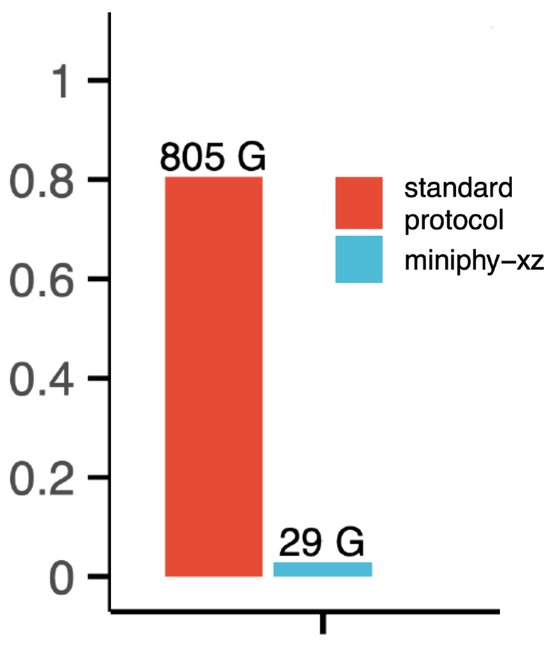
**elated genomes** are **extremely compressible**.

mpression: Utilizes evolutionary relationships to
on and search

Genomes that are closer to each other on the phylogenetic tree are more compressible



[Brinda et al.,bioRxiv, 2024]




**Phylign:** tool for sear
compressed 661 colle
laptop.

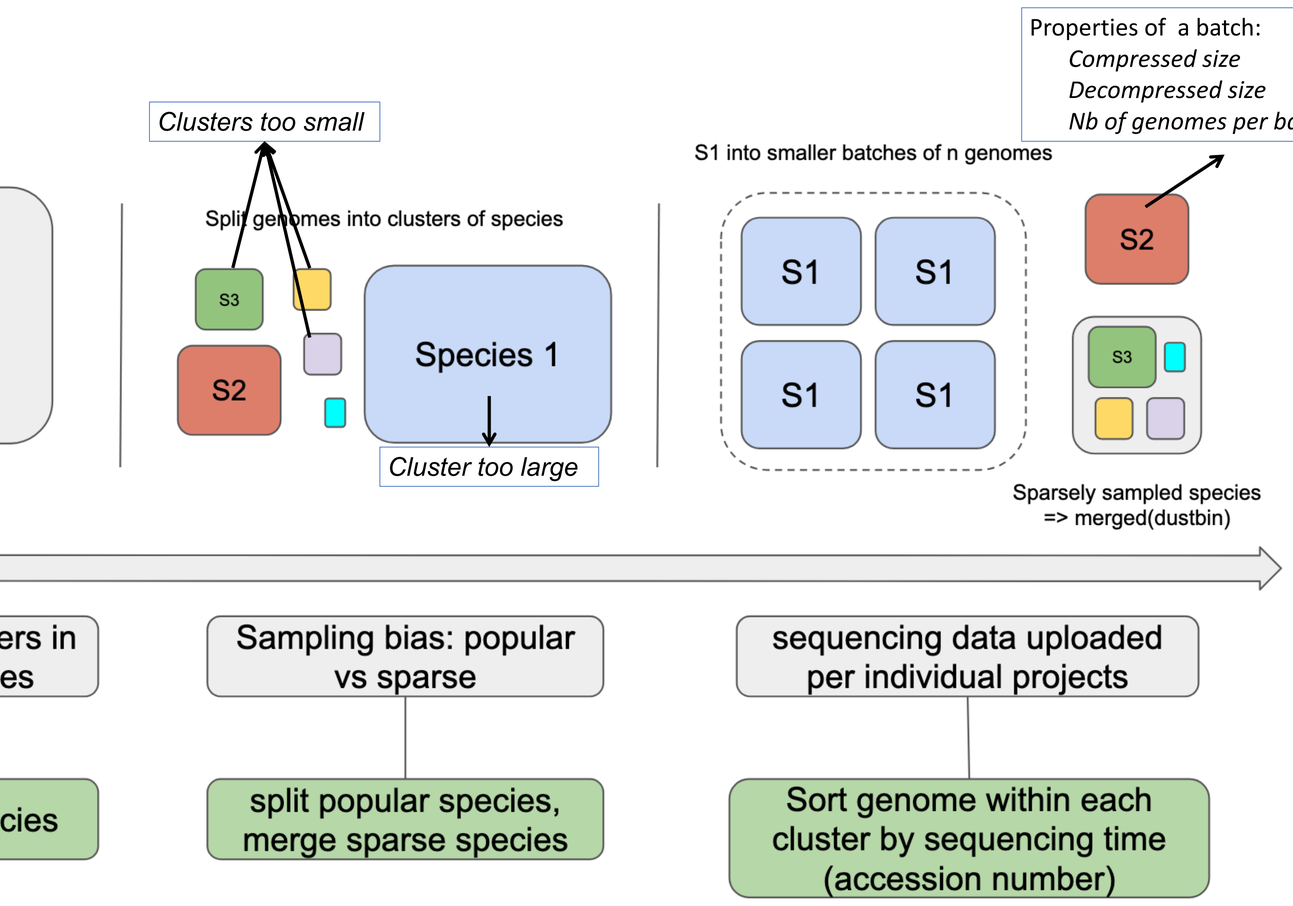
https://github.com/kar

661k

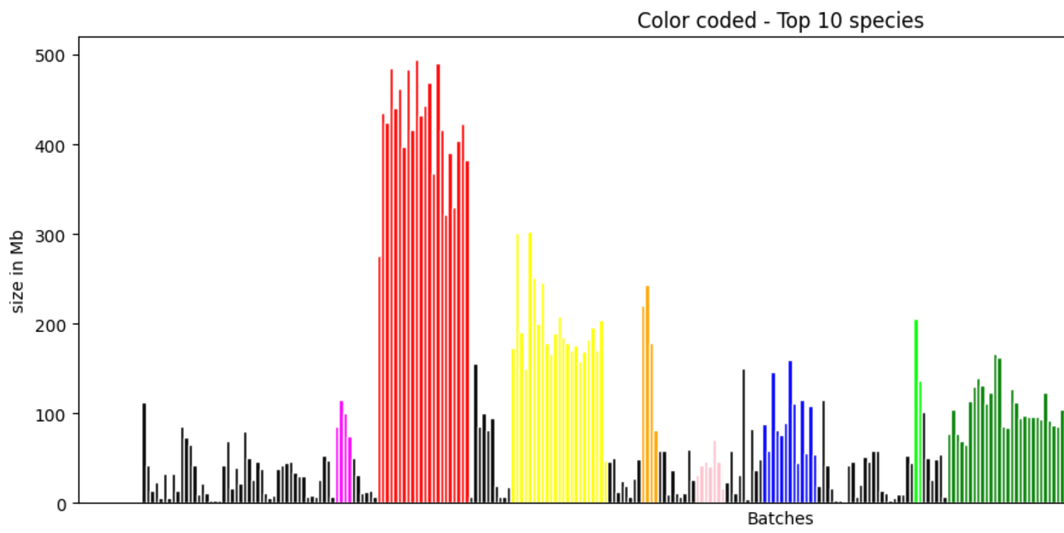
High compression ratio
Compressed using MiniPhy:
*https://github.com/karel-brinda/miniphy*

# State of the art

# Key idea in phylogenetic compression: clustering/batching




Compressed sizes of batches of the 661k




This batching enables compression and search (parallelization) or linear time (using a sing

This batching is not adapted for different hardw memory constraint
The compressed size of the batch is not

ced batches that fit within a constraint (i.e. memory) while minimizing the numb (maximize the number of of genomes in the batches)

MultiProcessor Scheduling
zation Problem

**Instance**:

tems, a size s(i) ∈ Z + for each i ∈ I.

ven b batches.

tion the items in set I into b we minimize the max size f the batches

- ➔ These problem are extensively studied
- ➔ NP-hard but there are efficient heuristic algorithms
- ➔ How to adapt these problem for DNA data

Bin Packing Optimizat

**Instance**:

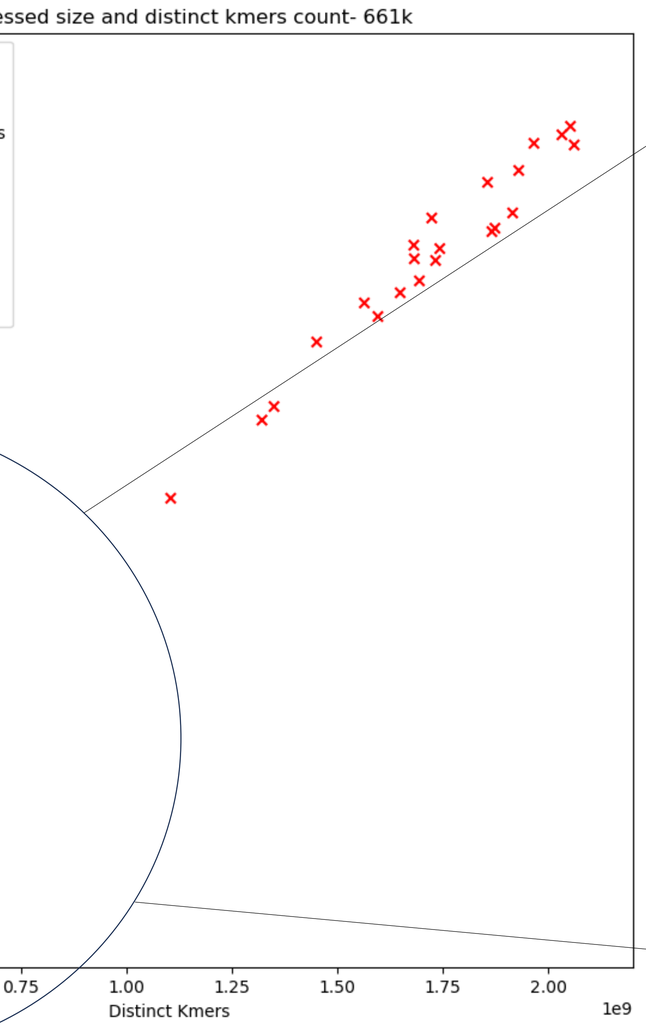
Finite set I of items, a siz each i ∈ I.

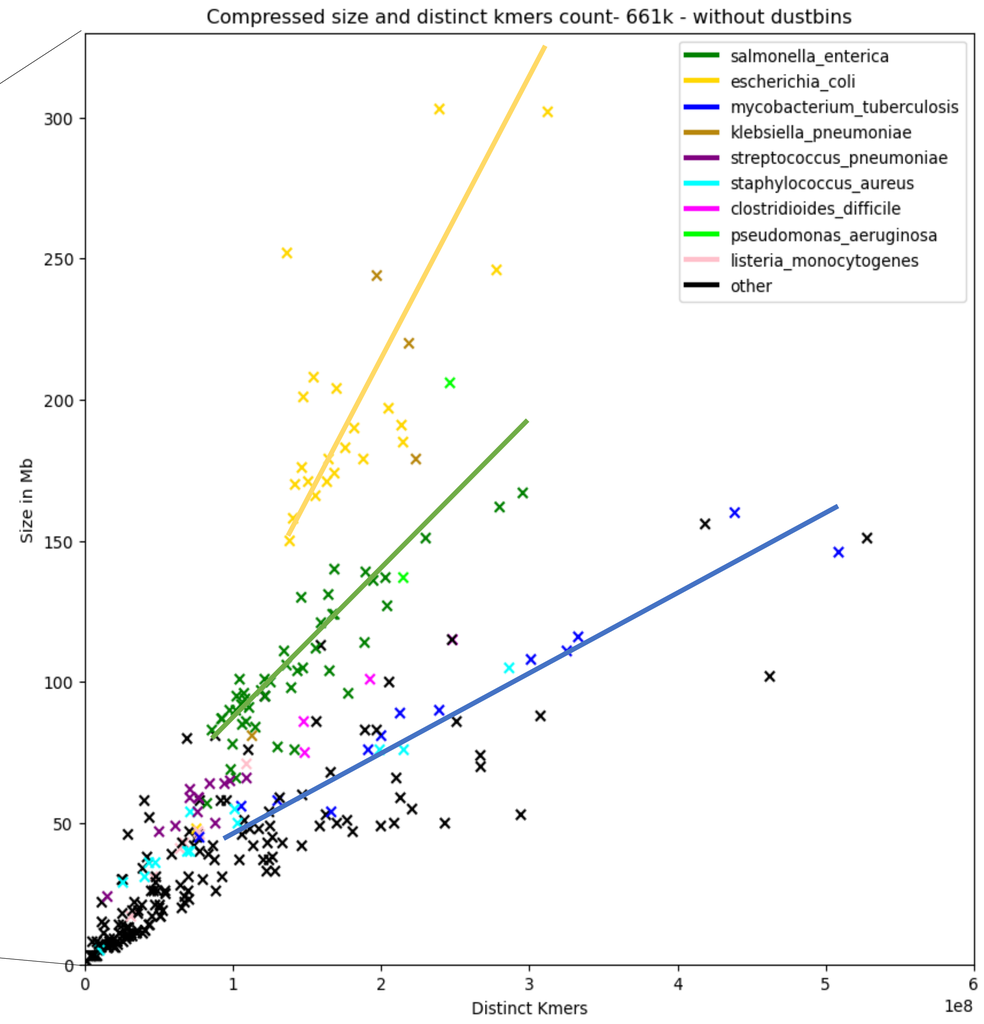
Given a capacity K for

**Objective**: Partition the ite batches and minimize th batches use

# Method

# mpressed size of batches can be estimated using biological prop




k-mers: substring of length k of DNA seque

Correlation between kmer and compressed batches

Different correlation for different species

➔ **Distinct kmers count can be used to se batches**

# ct kmer estimation using probabilistic counting – HyperLogLog

ating an approximate, compact
a sketch)

GATCGATCGA

G

...

ATCG

TCGA

**Probabilistic counting/approximate counting:**

Sketches
Hash(item_1) = 0000... => Prob = 1 / $2^4$
Hash(item_2) = 0001...
Hash(item_3) = 0010...
Hash(item_4) = 0011...
Hash(item_5) = 0100...
Hash(item_6) = 0101...
...

Longest prefix of leading zeros in the hash values is **4**
Estimated cardinality ≈ **$2^4$**

**Set cardinality estimation wit**

Hyperloglog is a sketching algo advanced approximate counting

Baker, D.N., Langmead, B. imp tool dashing.

*https://github.com/dnbaker/das*

# Result

# Hyperloglog based bin packing and load balancing for genomes

**g:**

s into batches that fit within a user-defined
nstraint

**KETCHES, CAPACITY (max_distinct_kmers)**

s based on their accession number.
**H**:
**:**
n the first batch that can accommodate it (based on capacity).
e the batch and move on to the next genome.
**ES if Necessary**:
commodate the current genome, create a new batch and place the

m-truong/HLL-Binning

**HLL-balancing:**

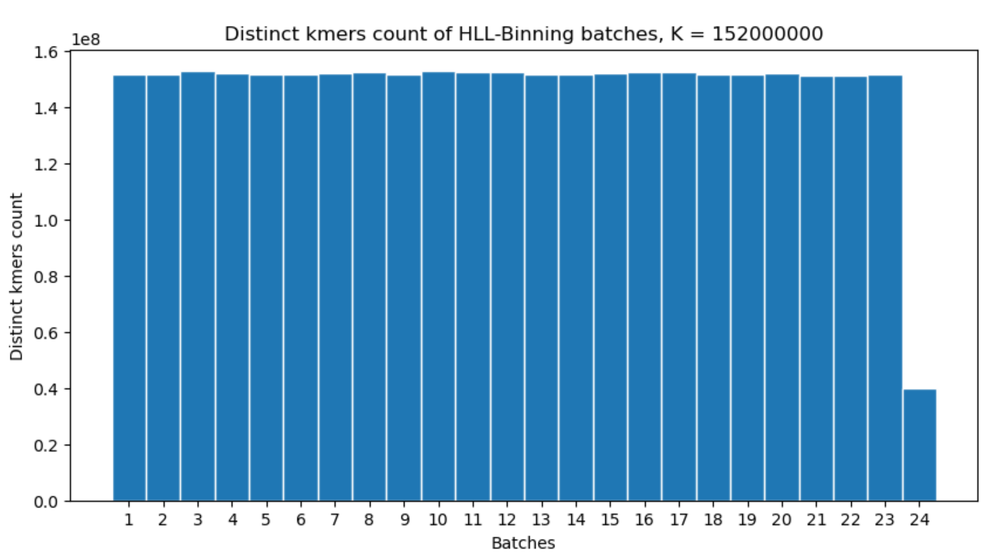
Put genomes into balanced, user-defined nu
batches

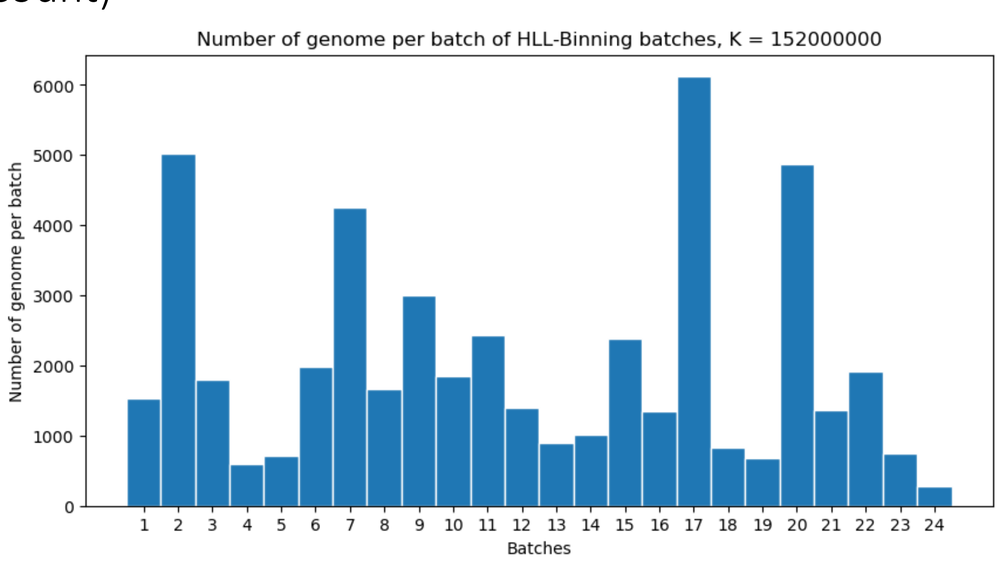
**INPUT: GENOMES SKETCHES, NUMBER OF BATCH b**
**Initialize**:
Sort the genomes based on their accession number.
**Initial Assignment**:
Assign the first b genomes directly to b batches.
**Greedy Assignment**:
For each remaining genomes, place it in the batches with the smallest kmers.
Update the batch's contents and size.

https://github.com/tam-km-truong/HLL-Balancing

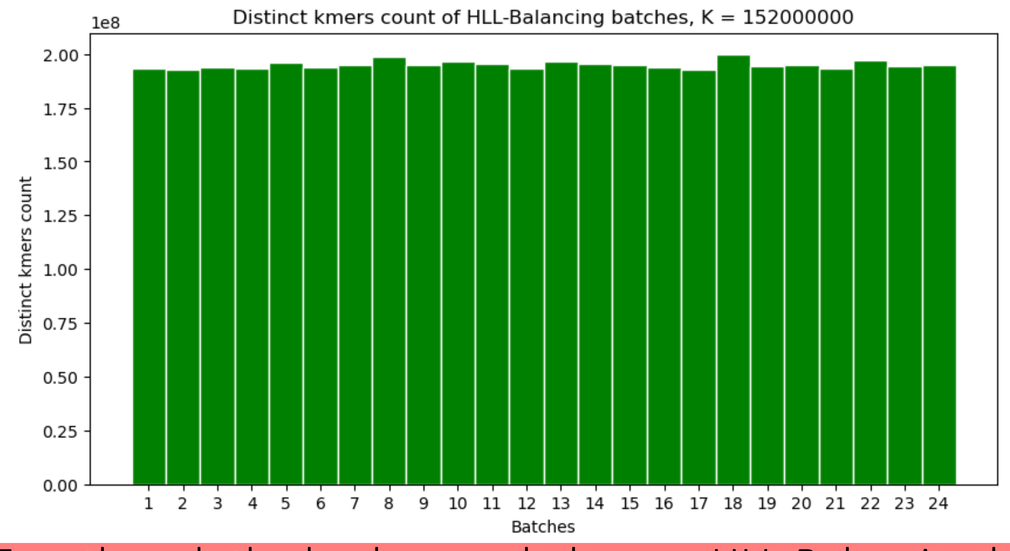
# EXPERIMENT: batching of Mycobacterium Turberculosis

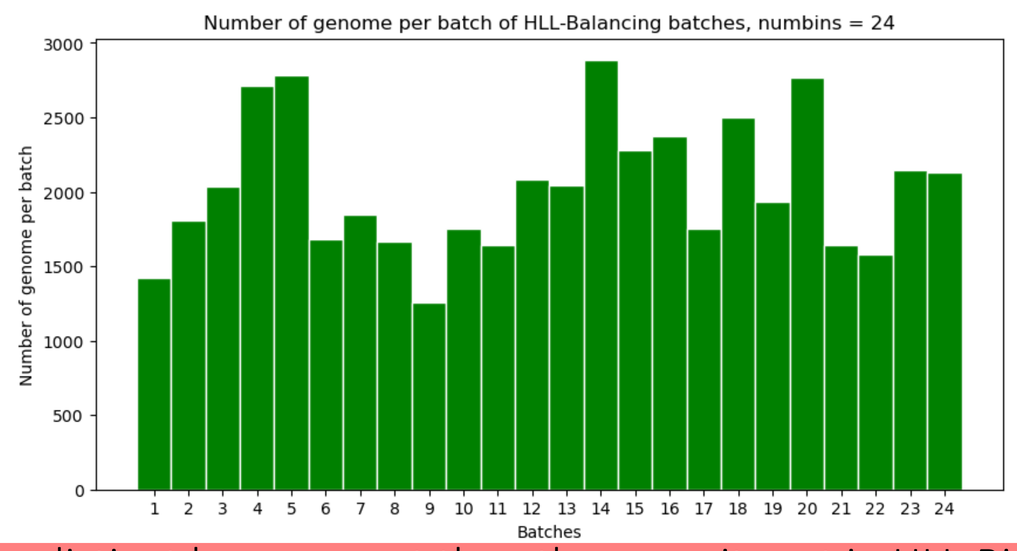
mers for batches: 153000000 kmers so that the compressed size stays under 64MB (caculated by using the correlation of compressed siz
kmers count)




Using the number off batches found by HLL-Binning, run HLL-Balancing with numbins b = 24




Eventhough the baches are balances, HLL-Balancing batches have higher distinct kmers count than the capacity set in HLL-Binning

# Discussion