Pangenomic k-mer distribution estimation

At low memory cost

Timothé Rouzé, Antoine Limasset & Rayan Chikhi

March 14, 2024

Context

- Project Logan, compression of unitigs

Introduction

Context

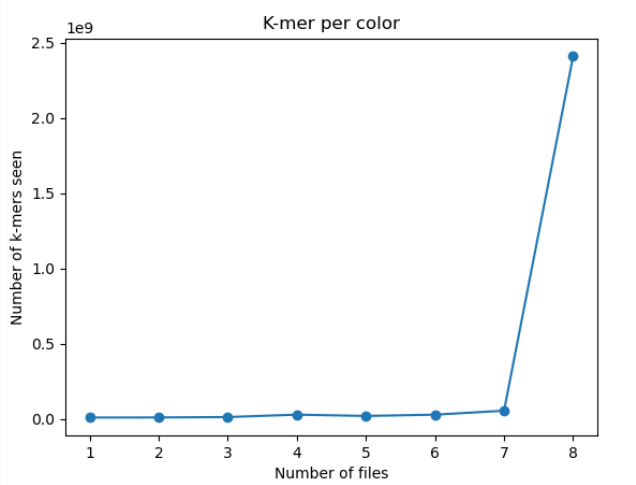
- Project Logan, compression of unitigs
- Compress a Human pangenome

Context

- Project Logan, compression of unitigs
- Compress a Human pangenome
- Computing k-mer distribution accross pangenome

Context

- Project Logan, compression of unitigs
- Compress a Human pangenome
- Computing k-mer distribution accross pangenome
- Need to count k-mers once per file accross every files



Existing tools

- Kmer counters, kmer index, ...
  - Needs tweaking to output the ✨histogram✨

Existing tools

- Kmer counters, kmer index, ...
  - Needs tweaking to output the ✨histogram✨
- Gene based tools
  - Not insightful for our research

Existing tools

- Kmer counters, kmer index, ...
  - Needs tweaking to output the ✨histogram✨
- Gene based tools
  - Not insightful for our research
- K-mer based tools
  - Pangrowth [1]
  - That's all....

Specificities

- Based on yak [2]

Specificities

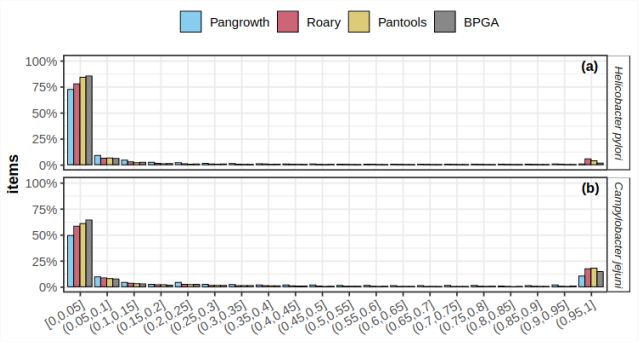
- Based on yak [2]
- Compared with gene based methods, gives a similar output

Specificities

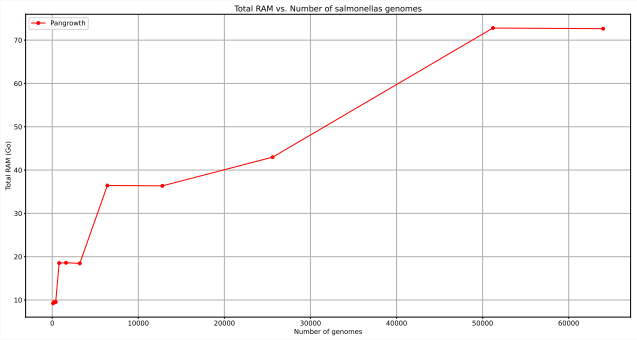
- Based on yak [2]
- Compared with gene based methods, gives a similar output
- uses the histogram to estimate pangenome ”openness”

Specificities

- Based on yak [2]
- Compared with gene based methods, gives a similar output
- uses the histogram to estimate pangenome ”openness”
- Does not scale easily



A word on scalability



Memory is critical

- A 96GB RAM machine is >4x price 16GB on AWS

K-mer Layers Estimation using Bloom filters

Structure

- Directly inspired by Pangrowth

K-mer Layers Estimation using Bloom filters

Structure

- Directly inspired by Pangrowth
- Bloom filters

K-mer Layers Estimation using Bloom filters

Structure

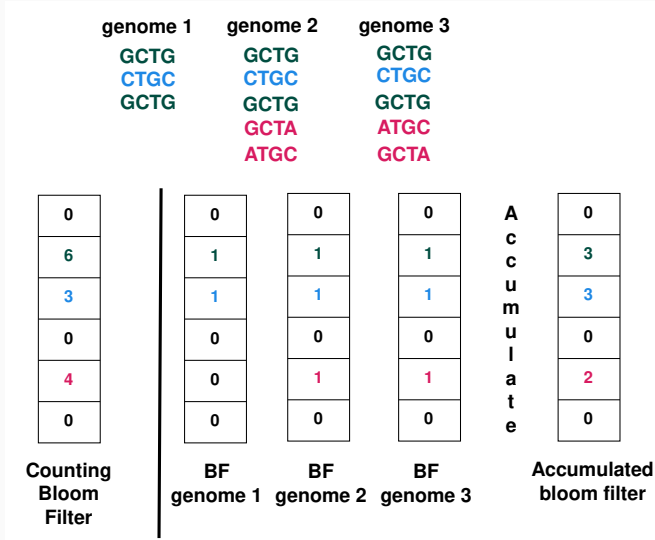
- Directly inspired by Pangrowth
- Bloom filters
  - Arbitrary low memory cost
  - Some false positive values
  - One per file
  - Temporary

K-mer Layers Estimation using Bloom filters

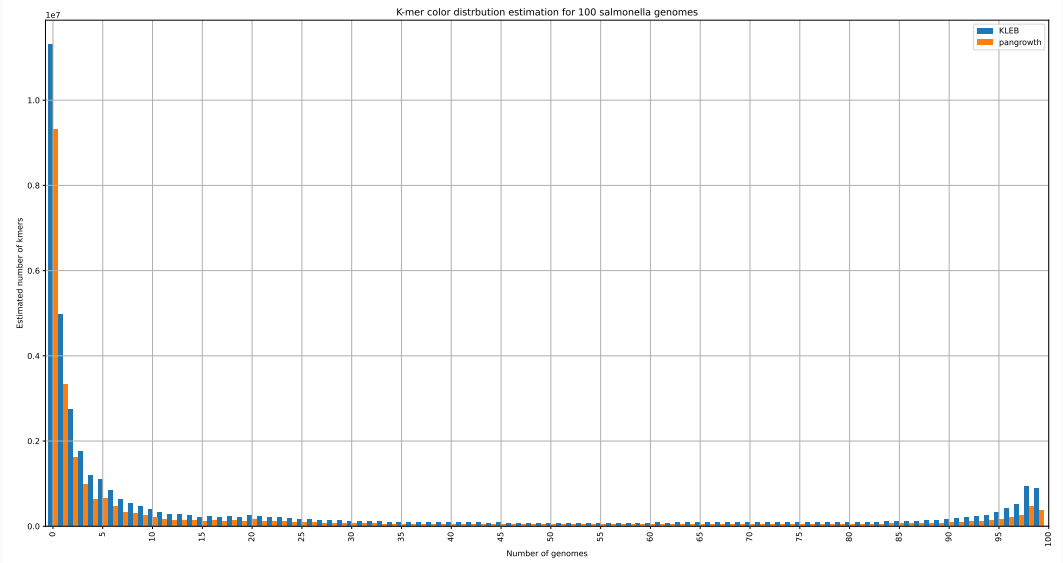
Structure

- Directly inspired by Pangrowth
- Bloom filters
  - Arbitrary low memory cost
  - Some false positive values
  - One per file
  - Temporary
- Accumulated Bloom Filter
  - Novel data structure
  - Between Counting bloom filters [3] and Agregating bloom filters [4]

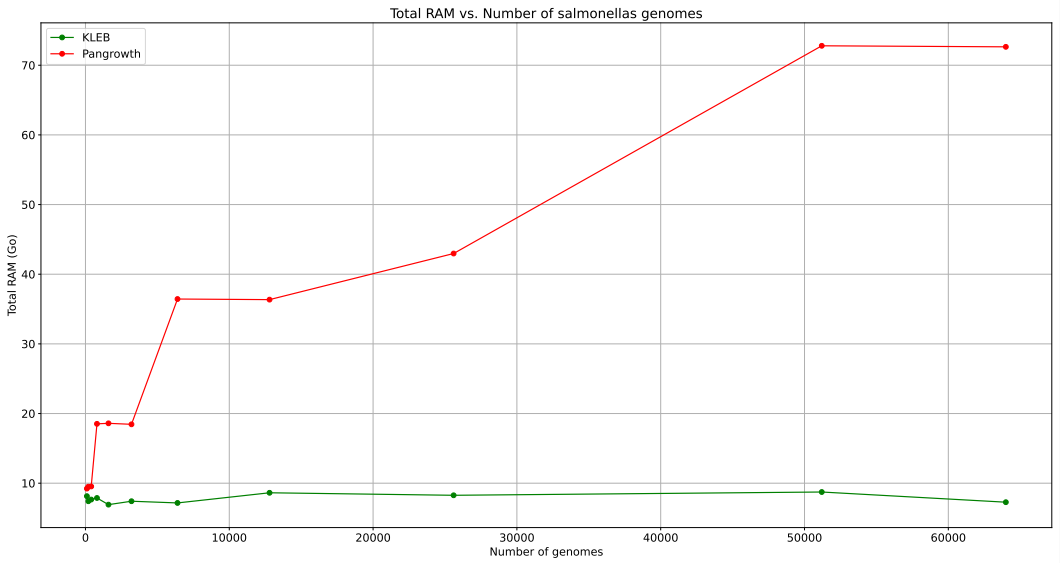
Accumulated bloom filter



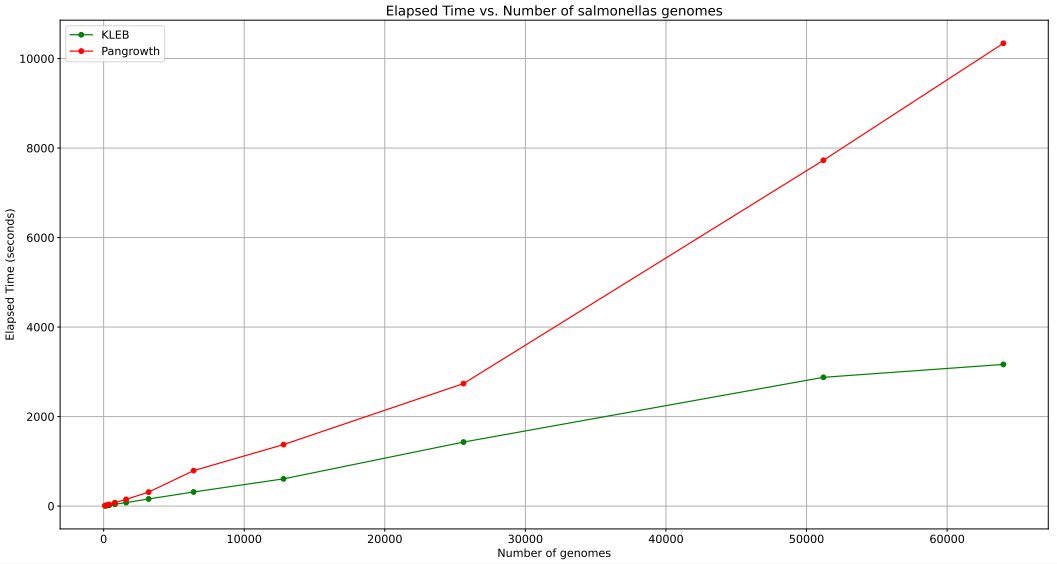
Histogram comparison



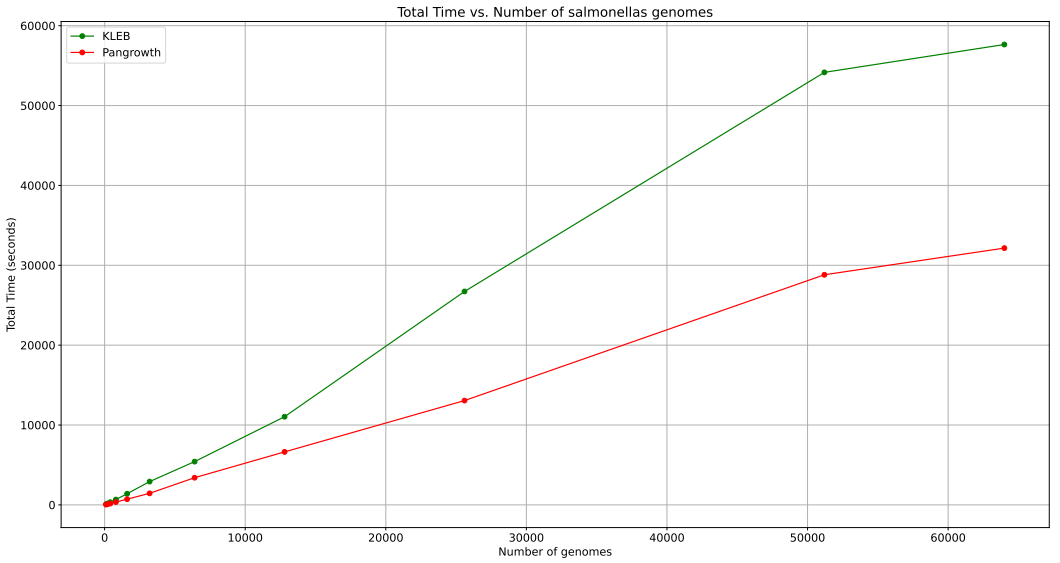
Memory consumption



Elapsed time on increasing number of genomes



CPU TIME (WIP)


Total Time vs. Number of salmonellas genomes
KLEB
Pangrowth
Total Time (seconds)
Number of genomes
0
10000
20000
30000
40000
50000
60000

Take home messages

- K-mer distribution
  - Gain better understanding on species
  - Guidance for genomic data compression
- Perspectives
  - Better qualitative analysis
  - Improve computational time
  - LSH methods

- KLEB
  - Arbitrarily lighter in memory
  - handling of big collections of genomes
  - still a work in progress
  - independant of k value

Appendix

References I

Luca Parmigiani, Roland Wittler, and Jens Stoye.
Revisiting pangenome openness with k-mers.
bioRxiv, pages 2022–11, 2022.

Haoyu Cheng, Gregory T Concepcion, Xiaowen Feng, Haowen Zhang, and Heng Li.
Haplotype-resolved de novo assembly using phased assembly graphs with hifiasm.
Nature methods, 18(2):170–175, 2021.

Li Fan, Pei Cao, Jussara Almeida, and Andrei Z Broder.
Summary cache: a scalable wide-area web cache sharing protocol.
IEEE/ACM transactions on networking, 8(3):281–293, 2000.

Camille Marchet and Antoine Limasset.
Scalable sequence database search using partitioned aggregated bloom comb trees.
Bioinformatics, 39(Supplement_1):i252–i259, 2023.