DALHOUSIE UNIVERSITY

Cache-friendly Run-Length Compressed Burrows-Wheeler Transform

Yansong Li

CSCI 6057 Project Presentation

March 22nd, 2022

Course Instructor: Meng He

Introduction

- Compressing large highly competitive string (e.g. human genome) in bioinformatics is essential
  - Chromosome 19 has almost 59 million bases [1]
- Burrows-Wheeler Transform (BWT)
- Run-length compressed BWT (RLBWT)
- Inverse the BWT: Last-to-First mapping (L-F mapping)
  - High number of cache misses.

Introduction

- Nishimoto’s lookup table for L-F mapping
  - L-F Computation only needs constant time
  - Enormous number of cache misses
- Sirén’s method
  - Graph BWT
- Apply Sirén’s method on Nishimoto’s lookup table

Literature Review

- Nishimoto's lookup table for RLBWT [2].
  - Efficiency: do the L-F mapping of RLBWT in $O(1)$ time and $O(r)$ space.
  - Description: use lookup table to replace compressed spare bitvector for in order to compress a random permutation of a string.
  - Improvement requeried: $O(r)$ space is large, so it has to sit on the memory.

Literature Review

- Sirén’s Graph BWT implementation[3].
  - Partition BWT in to sub-BWTs according to the most significant character in the lexicographic order.
  - Encode each sub-BWT.
  - Assume a cache-friendly layout.

Objectives and Hypotheses

- Decrease the number of cache misses
- Rearrange the RLBWT table
  - Cut the table into blocks, build a graph using blocks, cluster the graph
- Assume a cache-friendly graph layout
  - Low out-degree of each blocks
  - Low expansion of the graph

Methodology – RLBWT table

- The RLBWT tables are provided by Nate Brown[4]

- Let $T[1 \ldots r]$ represent a RLBWT table of length r that was built from a BWT with of size n.
  - T_i is a quadruple that consisted of character c, length l, interval k_{LF}, and offset d_{LF}.

- row#4

G	10	7	0

Methodology – Graph building

- Cut T into blocks $B[1 \dots b]$ with size of 1024 rows

- Use the B as nodes V and the jumps of L-F mapping (index, k_{LF}) as edges E to build a graph $G = (V, E)$
  - Self-connected edges $e = (v, v), e \in E, v \in V$ are ignored

- $e \in E$ is weighted, and weight $w \in W$ of e is the total length of runs between two nodes $v \in V, u \in V$

Methodology – METIS Clustering

- G is clustered by METIS with different #clusters [5].
- Clustering Algorithm: Cluster v and u if $e = (v, u)$ has the largest w until all $v \in V$ is clustered.
- Calculate the total weight W_m of inter-clusters edges $e_t \in E_t$.

Methodology – Sequential Clustering and weight ratio

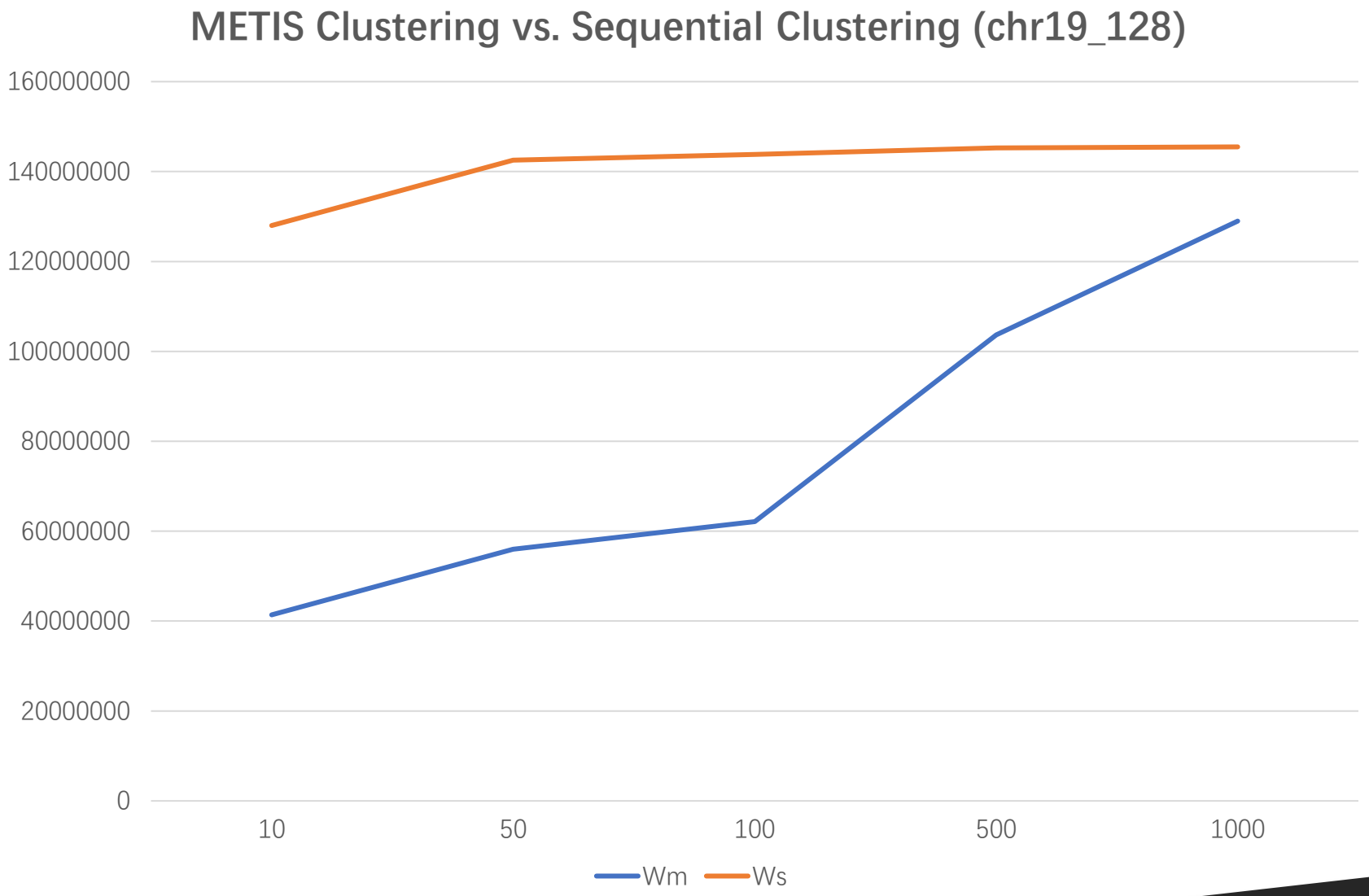
- G is clustered sequentially as a comparison.
- Clustering Algorithm: For each $v \in V$, cluster it by its index in T with a certain #clusters
- Calculate the total weight W_s of inter-clusters edges $e_t \in E_t$
- Calculate $\frac{W_s}{W_m}$

Result – Salmonella

- Salmonella dataset statistics:
  - n: 145,595,456; r = 12,823,516; $\frac{n}{r}$ = 11.3537.
  - $|B|$ = 12,523; $|V|$ = 62,556; W = 145,589,783.
- Clustering result

#Clusters	10	50	100	500	1000
W_m	41,393,086	56,002,575	62,137,611	103,647,402	128,998,476
W_s	127,980,098	142,549,115	143,815,338	145,258,185	145,482,565
$\frac{W_s}{W_m}$	3.09	2.55	2.31	1.4	1.13

Result – Salmonella

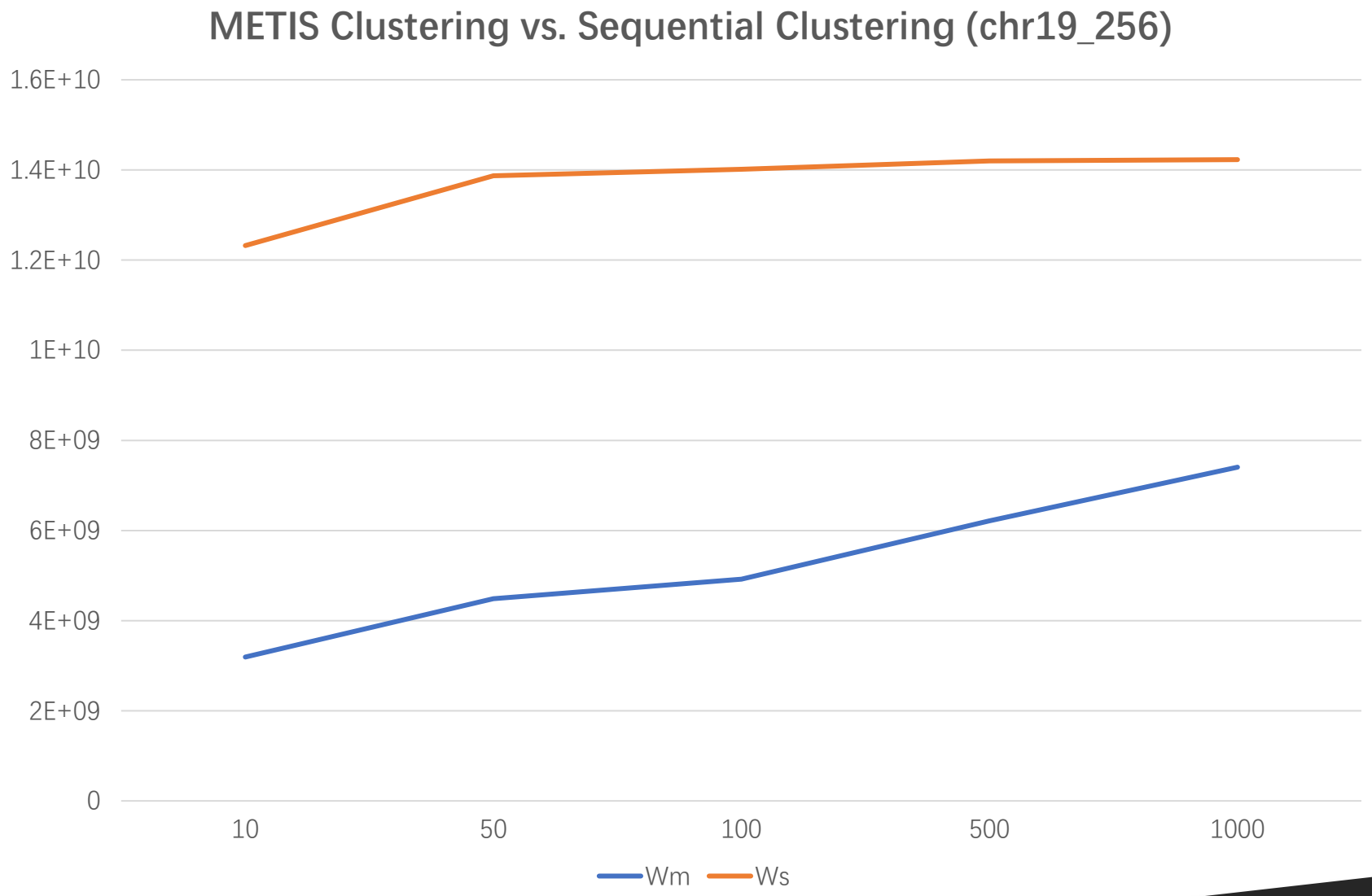


Result – Chromosome 19 (128 copies)

- Chr19_128 dataset statistics:
  - n: 7,568,015,632; $r = 34{,}053{,}959$; $\frac{n}{r} = 222.236.$
  - $|B| = 33{,}256$; $|V| = 166{,}104$; $W = 7{,}142{,}005{,}530.$
- Clustering result

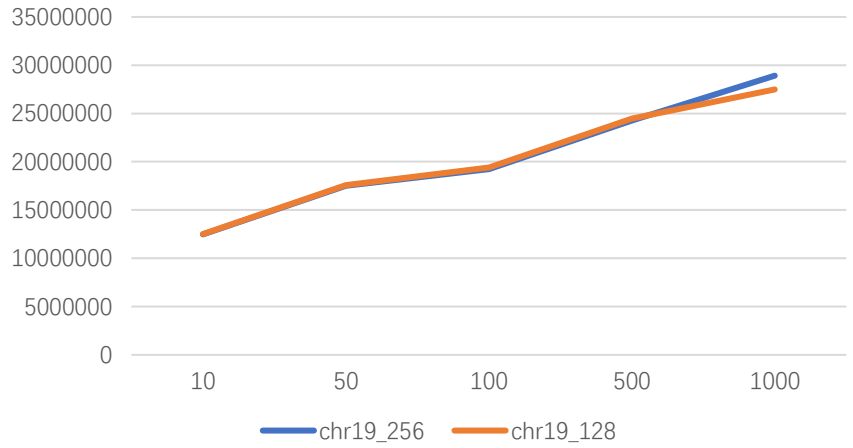
#Clusters	10	50	100	500	1000
W_m	1,601,510,950	2,249,127,139	2,483,372,420	3,132,482,264	3,518,451,513
W_s	6,149,247,602	6,934,085,819	7,010,231,320	7,093,603,030	7,110,940,769
$\frac{W_s}{W_m}$	3.84	3.09	2.82	2.26	2.02

Result – Chromosome 19 (128 copies)



Result – Chromosome 19 (256 copies)

- Chr19_256 dataset statistics:
  - $n, r, |B|, |V|$ increases by 4.5%.
  - $\frac{n}{r} = 424.934; W = 14{,}284{,}840{,}384$, which increases by 2 times.
- Each $w \in W$ increases by 2 times, didn't get better.



Discussions

- The METIS clustering gives meaningfully better results when #cluster < 100.
- Salmonella shows 2 times better, and chromosome 19 show 3 times better with 100 clusters.
- 1024 block size and 100 clusters is feasible based on the L1 and L3 cache size of Waverley and Timbelea.

Conclusion and future works

- The results is not significantly better, but still interesting.
- Prove the graph has a cache-friendly layout
  - The number of out edges per node is low
  - The graph is almost linear, a low expansion
- Make block size and cache size to parameters.
- Implement the cache-friendly feature

Thanks for listening

References

[1] NCBI. (2022). Genome Reference Consortium Human Build 38 patch release 14 (GRCh38.p14). https://www.ncbi.nlm.nih.gov/assembly/GCA_000001405.29

[2] Nishimoto, T., & Tabei, Y. (2021). Optimal-Time Queries on BWT-Runs Compressed Indexes. *ICALP*.

[3] Sirén, J., Garrison, E., Novak, A., Paten, E., & Durbin, R. (2020). Haplotype-aware graph indexes. *Bioinformatics*, Volume 36, Issue 2, pp. 400–407.

[4] Brown, N.K., Gagie, T., & Rossi, M. (2021). RLBWT Tricks. ArXiv, abs/2112.04271.

[5] Karypis, G., Kumar,V. (1999). A Fast and Highly Quality Multilevel Scheme for Partitioning Irregular Graphs. *SIAM Journal on Scientific Computing*, Vol. 20, No. 1, pp. 359—392.