Augmented Thresholds for MONI

César Martínez-Guardiola[1] **Nathaniel K. Brown**[2]
Fernando Silva-Coira[1] Dominik Köppl[3] Travis Gagie[2]
Susana Ladra[1]

[1]CITIC
Universidade da Coruña
A Coruña, Spain

[2]FCS
Dalhousie University
Halifax, Canada

[3]M&D Data Science Center
TMDU
Tokyo, Japan

Data Compression Conference, March 2023

Introduction

- **Computational Pan-Genomics**:
  - Want to index many genomes in reasonable space
  - Pan-genome graphs one method
  - *Solution*: Versions of FM-Index based on run-length compressed BWT (RLBWT)

- **r-index**: efficient exact matching in runs-bounded space
  - Computes LF steps, samples SA at run boundaries

Maximal Exact Matches

- **Approximate Matching**: Exact matching is not ideal for most use cases
  - Instead, want to support approximate matching
- **Maximal Exact Matches (MEMs)**: The longest matching sub strings of a pattern P with a text T
  - $P[i..j]$ occurs in T
  - Neither $P[i-1..j]$ or $P[i..j+1]$ occurs in T
  - Supports approximate pattern matching in tools like BWA-MEM

Pan-genomics MEMs

- **MONI**: Rossi et al.'s *r*-index variant supporting MEM-finding
  - Computes MEMs using matching statistics of text
  - Two passes over the pattern, stores 'threshold' values at run boundaries
  - Can be made online without thresholds by using Longest Common Extension (LCE) queries
  - Online one-pass variant named **PHONI** (Boucher et al.)
- Online can be used in targeted approaches
  - Ahmed et al.'s **SPUMONI** for metagenomic classification using nanopore sequencing

MEMs

We show how to use both thresholds and LCE queries to compute MEMs online quickly by storing additional LCE information with thresholds.

Suffix Array

- **Suffix Array (SA)**: $\mathrm{SA}[i]$ is the starting position of the lexicographically ith suffix in a text $T[0..n-1]$
  - MONI samples at start/end of BWT-runs

- $\mathrm{BWT}[i]$ is character preceding $\mathrm{SA}[i]$
- $\mathrm{LF}(i)$ is position of $\mathrm{SA}[i-1]$ in SA

Longest Common Prefixes

- **LCP Array**: $\mathrm{LCP}[i]$ stores the LCP between suffixes at $\mathrm{SA}[i]$ and $\mathrm{SA}[i-1]$
- **Longest Common Extension**: $\mathrm{LCE}(i,j)$ returns the LCP between $T[i..n-1]$ and $T[j..n-1]$
  - Given SA samples: $\mathrm{LCE}(\mathrm{SA}[i], \mathrm{SA}[j])$
  - Range-minimum-query (RMQ) over the LCP array

Thresholds

	k	$\mathrm{SA}[k]$	$\mathrm{BWT}[k]$	$T[\mathrm{SA}[k]..n]$	$\mathrm{LCP}[k]$
	⋮	⋮	⋮	⋮	⋮
	1234	8765	A	GAGACATCA...	-
$e_1 =$	1235	1519	A	GATACATTA...	-
	1236	5450	C	GATAGATTA...	4
$j =$	1237	1004	G	GATATAGAA...	4
	1238	4242	G	GATCCAATA...	3
$t =$	1239	3110	G	GATTACATA...	3
	1240	1102	T	GATTACTTA...	6
	1241	1978	T	GATTAGATA...	5
$s_2 =$	1242	2505	A	GATTATCAT...	5
	1243	2022	A	GATTATGAA...	-
	⋮	⋮	⋮	⋮	⋮

Matching Statistics

- **Matching Statistics (MS)**: For pattern $P[0..m-1]$ and text $T[0..n-1]$, $\mathrm{MS}[0..m-1]$ defined as pair
  - $\mathrm{MS}[i].pos$ is the starting text position of the longest prefix of $P[i..m-1]$ that occurs in T
  - $\mathrm{MS}[i].len$ is the length of that prefix

MONI Process

- **Matching Statistics for** $P[0..m-1]$
  - $j = \mathrm{BWT}.select_{P[m-1]}(1)$
  - $\mathrm{MS}[m-1].pos = \mathrm{SA}[j]$
  - $j = LF(j)$, continue with $i = m-2, ..., 0$
  - **Case 1:** $\mathrm{BWT}[j] = P[i]$, $\mathrm{MS}[i].pos = \mathrm{MS}[i+1].pos - 1$
  - **Case 2:** $\mathrm{BWT}[j] \neq P[i]$
    - $\mathrm{BWT}[s_1..e_1] = \mathrm{BWT}[s_2..e_2] = P[i]$ where $e_1 < j < s_2$
    - Threshold t; if $j < t$ take $\mathrm{SA}[e_1]$, if $j \geq t$ take $\mathrm{SA}[s_2]$
    - Set j, $\mathrm{MS}[i].pos$ corresponding with choice
  - LF step and continue

- Second left-to-right pass to recover lengths
  - Using compressed straight line program (SLP) of T supporting random access

Threshold LCE Example

$\mathrm{MS}[i+1].pos = SA[j],\ P[i] = \mathrm{A}$

	k	$\mathrm{SA}[k]$	$\mathrm{BWT}[k]$	$T[\mathrm{SA}[k]..n]$	$\mathrm{LCP}[k]$
	⋮	⋮	⋮	⋮	⋮
	1234	8765	A	GAGACATCA...	-
$e_1 =$	1235	1519	A	GATACATTA...	-
	1236	5450	C	GATAGATTA...	4
$j =$	1237	1004	G	GATATAGAA...	4
	1238	4242	G	GATCCAATA...	3
$t =$	1239	3110	G	GATTACATA...	3
	1240	1102	T	GATTACTTA...	6
	1241	1978	T	GATTAGATA...	5
$s_2 =$	1242	2505	A	GATTATCAT...	5
	1243	2022	A	GATTATGAA...	-
	⋮	⋮	⋮	⋮	⋮

PHONI

- **PHONI**: Compute lengths with positions using LCEs

- **Start**: $\mathrm{MS}[m-1].len = 1$
- **Case 1 (match)**: $\mathrm{MS}[i].len = \mathrm{MS}[i+1].len + 1$
- **Case 2 (mismatch)**:
  - $maxLCE = max(\mathrm{LCE}(\mathrm{SA}[j], \mathrm{SA}[e_1]), \mathrm{LCE}(\mathrm{SA}[j], \mathrm{SA}[s_2]))$
  - $\mathrm{MS}[i+1].len = min(\mathrm{MS}[i+1].len, maxLCE) + 1$

Finding Lengths

$\mathrm{MS}[i+1].len = 4,\ P[i] = \mathrm{A}$

	k	$\mathrm{SA}[k]$	$\mathrm{BWT}[k]$	$T[\mathrm{SA}[k]..n]$	$\mathrm{LCP}[k]$
	⋮	⋮	⋮	⋮	⋮
	1234	8765	A	GAGACATCA...	-
$e_1 =$	1235	1519	A	GATACATTA...	-
	1236	5450	C	GATAGATTA...	4
$j =$	1237	1004	G	GATATAGAA...	4
	1238	4242	G	GATCCAATA...	3
	1239	3110	G	GATTACATA...	3
	1240	1102	T	GATTACTTA...	6
	1241	1978	T	GATTAGATA...	5
$s_2 =$	1242	2505	A	GATTATCAT...	5
	1243	2022	A	GATTATGAA...	-
	⋮	⋮	⋮	⋮	⋮

Finding Lengths

$\mathrm{MS}[i+1].len = 2,\ P[i] = \mathrm{A}$

	k	$\mathrm{SA}[k]$	$\mathrm{BWT}[k]$	$T[\mathrm{SA}[k]..n]$	$\mathrm{LCP}[k]$
	⋮	⋮	⋮	⋮	⋮
	1234	8765	A	GAGACATCA...	-
$e_1 =$	1235	1519	A	GATACATTA...	-
	1236	5450	C	GATAGATTA...	4
$j =$	1237	1004	G	GATATAGAA...	4
	1238	4242	G	GATCCAATA...	3
	1239	3110	G	GATTACATA...	3
	1240	1102	T	GATTACTTA...	6
	1241	1978	T	GATTAGATA...	5
$s_2 =$	1242	2505	A	GATTATCAT...	5
	1243	2022	A	GATTATGAA...	-
	⋮	⋮	⋮	⋮	⋮

Mismatches

- **Case 2a**: Max LCE computed $> \mathrm{MS}[i+1].\mathit{len}$
- **Case 2b**: Max LCE computed $\leq \mathrm{MS}[i+1].\mathit{len}$

- Can we avoid the LCEs in 2b?

LCEs and Thresholds

- **LCE Queries**: With threshold, need only one LCE
  - $\mathrm{LCE}(\mathrm{SA}[i], \mathrm{SA}[e_1])$ where $e_1 < i < t$
  - or $\mathrm{LCE}(\mathrm{SA}[i], \mathrm{SA}[s_2])$ where $t \leq i < s_2$

- **LCEs as RMQ**:
  - Consider $i \geq t$ (other case symmetrical)
  - $\mathrm{LCE}(\mathrm{SA}[i], \mathrm{SA}[s_2]) = min(\mathrm{LCP}[i+1..s_2])$
  - $\mathrm{LCE}(\mathrm{SA}[t], \mathrm{SA}[s_2]) = min(\mathrm{LCP}[t+1..s_2])$
  - Since $t \leq i < s_2$,
    $\mathrm{LCE}(\mathrm{SA}[t], \mathrm{SA}[s_2]) \leq \mathrm{LCE}(\mathrm{SA}[i], \mathrm{SA}[s_2])$

Finding Lengths

$\mathrm{MS}[i+1].len = 2,\ P[i] = \mathrm{A}$

	k	$\mathrm{SA}[k]$	$\mathrm{BWT}[k]$	$T[\mathrm{SA}[k]..n]$	$\mathrm{LCP}[k]$
	⋮	⋮	⋮	⋮	⋮
	1234	8765	A	GAGACATCA...	-
$e_1 =$	1235	1519	A	GATACATTA...	-
	1236	5450	C	GATAGATTA...	4
$j =$	1237	1004	G	GATATAGAA...	4
	1238	4242	G	GATCCAATA...	3
$t =$	1239	3110	G	GATTACATA...	3
	1240	1102	T	GATTACTTA...	6
	1241	1978	T	GATTAGATA...	5
$s_2 =$	1242	2505	A	GATTATCAT...	5
	1243	2022	A	GATTATGAA...	-
	⋮	⋮	⋮	⋮	⋮

Augmenting Thresholds

- Store $\mathrm{LCE}(\mathrm{SA}[t], \mathrm{SA}[e_1])$ and $\mathrm{LCE}(\mathrm{SA}[t], \mathrm{SA}[s_2])$
- e.g. $\mathrm{LCE}(\mathrm{SA}[t], \mathrm{SA}[s_2]) \leq \mathrm{MS}[i+1].\mathit{len}$
  - Set $\mathrm{MS}[i].\mathit{len} = \mathrm{MS}[i+1].\mathit{len} + 1$
  - Skip explicit LCE computation

- **Augmented Thresholds**:
  - Store threshold positions (one for each run)
  - With each threshold store the corresponding LCEs for before/after

Reducing LCEs

	MONI	*PHONI*	**AUG**
Case 1	None	None	None
Case 2a	1 SLP	1 or 2 LCEs	1 LCE
Case 2b	1 SLP	1 or 2 LCEs	None
thresholds?	yes	no	yes

Finding Threshold LCEs

- MONI computes thresholds efficiently using Boucher et al's prefix free parsing (PFP)
  - Finds the minimum in the LCP array between run boundaries
- Threshold LCEs are computed similarly
  - Minimum between first boundary and threshold
  - Minimum after threshold and next boundary

Compressing Values

- For each distinct character, threshold positions form increasing subsequences
- When $t = e_1 + 1$ or $t = s_2$, some threshold LCEs unused
  - Occurs up to roughly $1/3$ of time on test data
  - Can 'null' these values
- Although LCPs can be large, expect median to be small
  - 99% fit in a byte on test data

Threshold LCE Variants

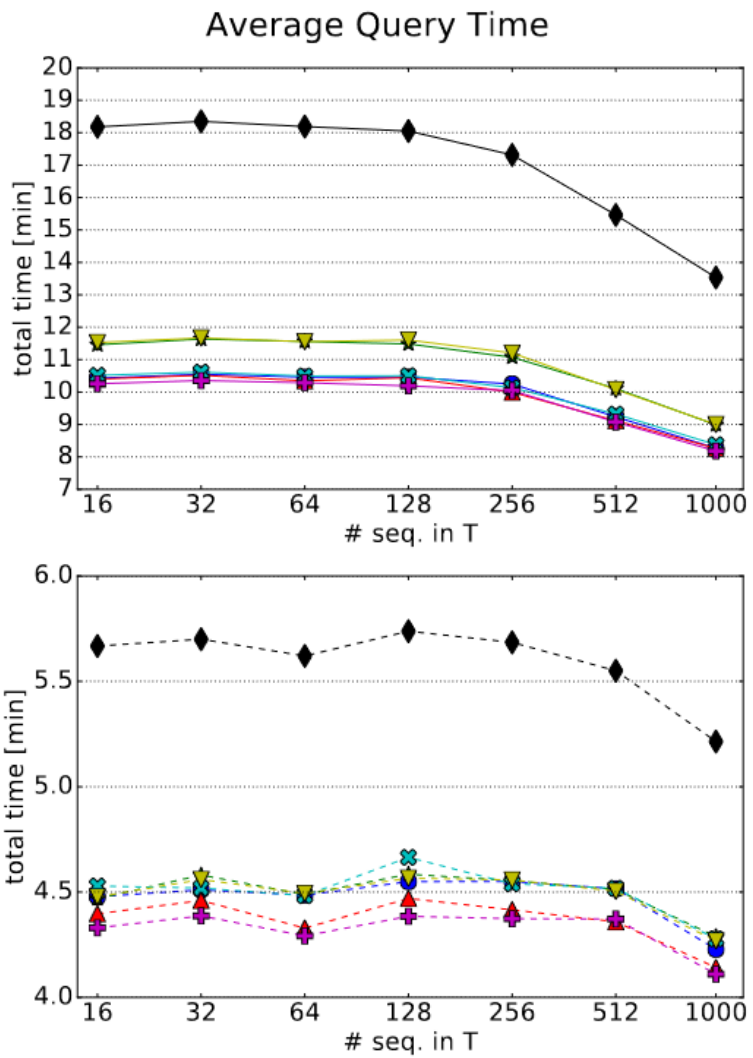
- `PHONI`: Standard version of one-pass MONI
- `Aug-Full`: One-pass MONI modified with augmented thresholds
- `Aug-1`: One byte per threshold LCE, perform LCE on overflow
- `Aug-BV-Full`: Bitvector marks threshold LCEs which are non-zero and stored
- `Aug-BV-1`: As above, but one byte per threshold LCE (default to LCE query)
- `Aug-DAC`: Stores threshold LCEs using a directly addressable code (DAC)
- `Aug-BV-DAC`: Same as `Aug-BV-Full`, but substituting in a DAC

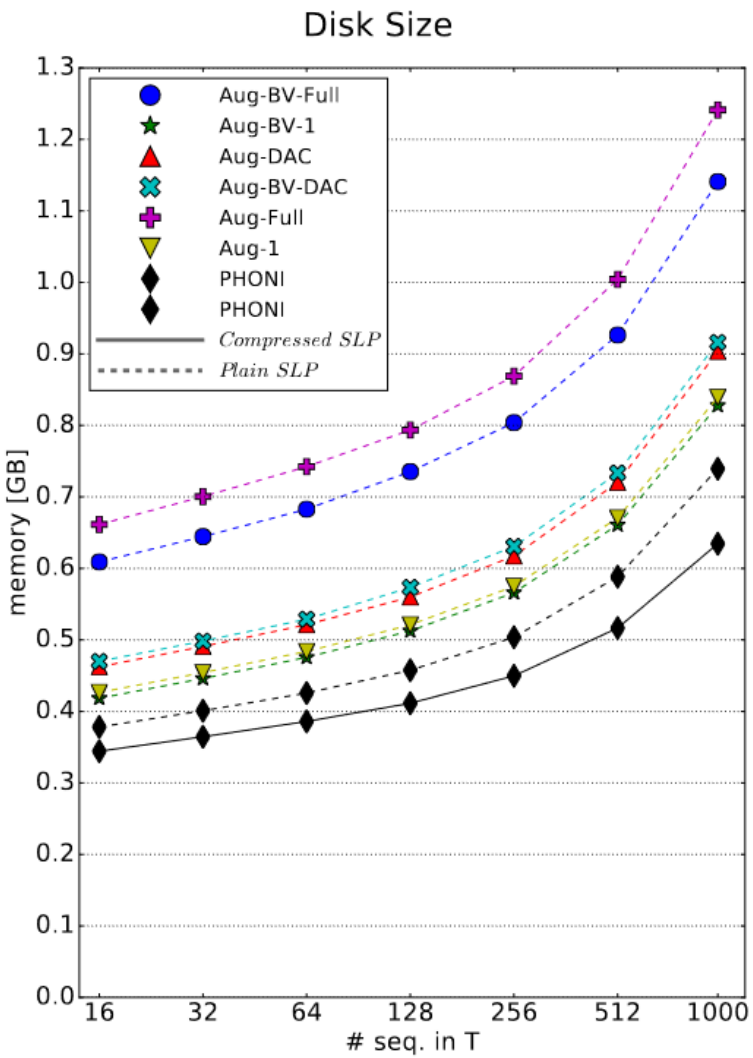
Datasets

- Patterns: 10 distinct chromosome-19 genomes
- References: chromosome-19 genomes of 16, 32, 64, 128, 256, 512 and 1000 copies
- Try compressed and plain SLPs

#	$n/10^6$	n/r	SLP_{comp} [MB]	SLP_{plain} [MB]
16	946.01	29.20	36.10	70.54
32	1892.01	57.64	37.80	74.75
64	3784.01	113.50	39.48	79.84
128	7568.01	222.24	42.11	88.89
256	15136.04	424.93	47.43	102.52
512	30272.08	771.54	58.00	131.09
1,000	59125.12	1287.38	80.63	186.98

Results





Thanks

- Email: `nathaniel.brown@dal.ca`
- Full-paper: `https://arxiv.org/abs/2211.07794`