

Building an Index for a large number of genomes

Christina Boucher
Department of Computer and Information
Science and Engineering



From SPIRE 2008 to SPIRE 2022

On the structure of small motif recognition instances

Authors Christina Boucher, Daniel G Brown, Stephane Durocher

Publication date 2008/11/10

Conference International Symposium on String Processing and Information Retrieval

Pages 269-281

Publisher Springer, Berlin, Heidelberg

Description Given a set of sequences, S , and degeneracy parameter, d , the CONSENSUS SEQUENCE problem asks whether there exists a sequence that has Hamming distance at most d from each sequence in S . A *valid motif set* is a set of sequences for which such a consensus sequence exists, while a *decoy set* is a set of sequences that does not have a consensus sequence but whose pairwise Hamming distances are all at most $2d$. At present, no efficient solution is known to the CONSENSUS SEQUENCE problem when the number of sequences is greater than three. For instances of CONSENSUS SEQUENCE with binary sequences and cardinality four, we present a combinatorial characterization of decoy sets and a linear-time exact algorithm, resolving an open problem posed by Gramm *et al.* [7].

SPIRE 2009 to SPIRE 2010

Faster algorithms for sampling and counting biological sequences

Authors Christina Boucher

Publication date 2009/8/25

Conference International Symposium on String Processing and Information Retrieval

Pages 243-253

Publisher Springer, Berlin, Heidelberg

Why Large CLOSEST STRING Instances Are Easy to Solve in Practice

Authors Christina Boucher, Kathleen Wilkie

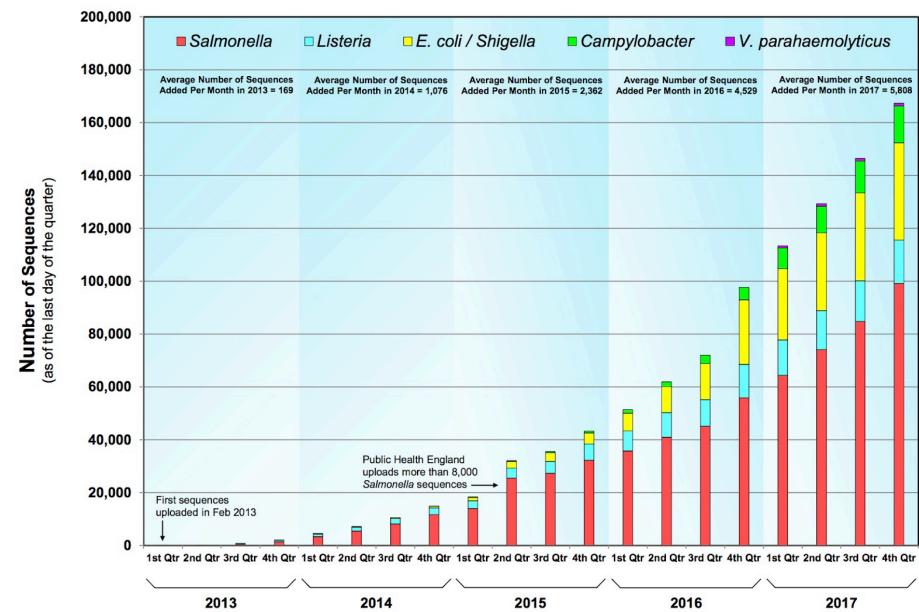
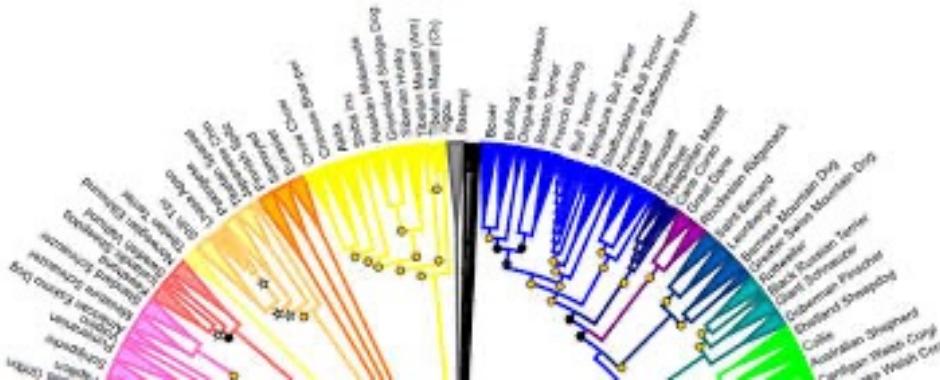
Publication date 2010/10/11

Conference International Symposium on String Processing and Information Retrieval

Pages 106-117

Publisher Springer, Berlin, Heidelberg

Biological Data vs. Moore's Law



Sequencing Cores == Accessibility



ABOUT US SERVICES & RESOURCES CORES TOOLS

— Illumina NovaSeq: S4

Illumina NovaSeq: S4

[UF | ICBR NextGen DNA Sequencing](#) // ICBR-NextGenSeq@ad.ufl.edu // 352.273.8050

Format	Lanes	UF Pricing	Non-Profit	Commercial	Reads*	Max Output**	UF Cost/Gb
2x150	Full FC	\$16687.50	\$19190.63	\$20859.38	10 Billion	3000	\$5.56
2x150	1	\$4218.47	\$4851.24	\$5273.09	2.5 Billion	750	\$5.62
2x100	Full FC	\$15052.57	\$17310.45	\$18815.71	10 Billion	2000	\$7.53
2x100	1	\$3809.73	\$4381.19	\$4762.17	2.5 Billion	500	\$7.62
1x35	Full FC	\$12364.62	\$14219.31	\$15455.77	10 Billion	350	\$35.33
1x35	1	\$3137.75	\$3608.41	\$3922.18	2.5 Billion	88	\$35.66

* - Number of SE reads per FC run or lane

** - Max output PE per FC run (Gb)

+ Illumina NovaSeq, Sp

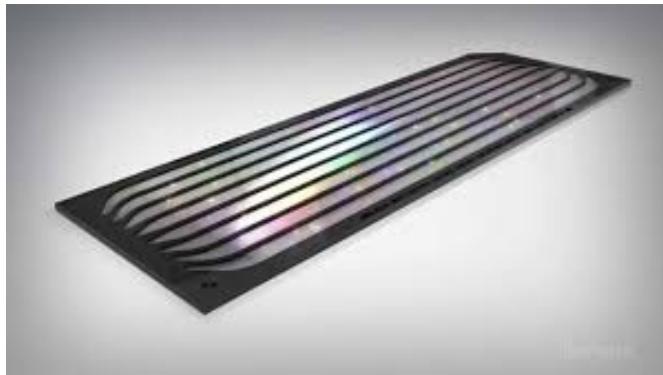
PacBio Library Construction Services

+ PacBio IsoSeq libraries

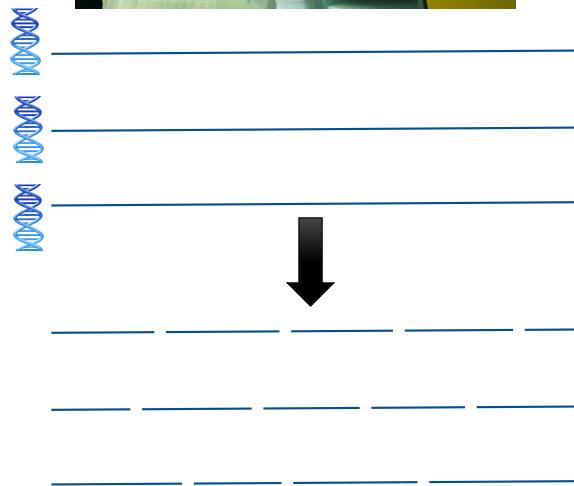
Sample Preparation



Sample Preparation



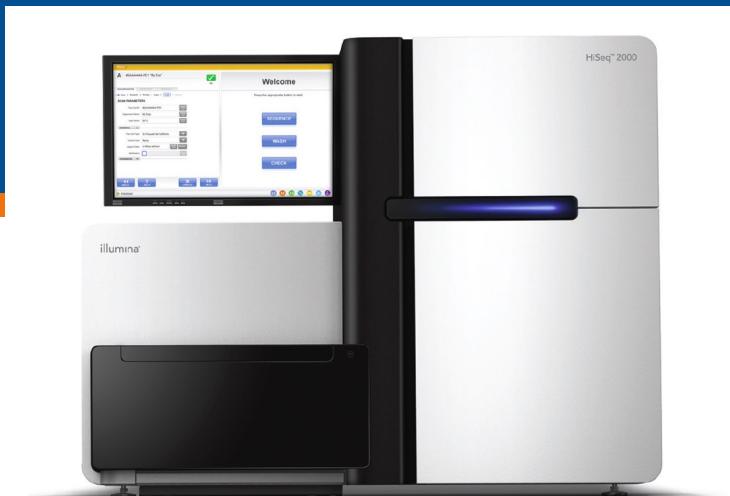
Fragments



Sample Preparation

Fragments

Sequencing



Next Generation Sequencing (NGS)

ACGTAGAATCGACCATG
GGGACGTAGAATAACGAC
ACGTAGAATAACGTAGAA

Reads

Sample Preparation

Fragments



Sequencing

Reads



Assembly / Alignment



ACGTAGAATACTAGAA
ACGTAGAATACTAGAA
ACGTAGAATCGACCATG ACGTAGAATACTAGAA
GGGACGTAGAATAACGAC ACGTAGAATACTAGAA
ACGTAGAATACTAGAA
ACGTAGAATACTAGAA

Sample Preparation

Fragments



Sequencing

Reads



Assembly / Alignment



Analysis

Sample Preparation

Fragments



Sequencing

Reads



Alignment



Analysis

Burrows Wheeler Transform

The **FM-index** which is BWT in combination with the SA. It allows for compression of the text but still enables fast substring queries.

Burrows-Wheeler Aligner

Introduction

BWA is a software package for mapping low-divergent sequences against a large reference genome, such as the human genome. It consists of three algorithms: BWA-backtrack, BWA-SW and BWA-MEM. The first algorithm is designed for Illumina sequence reads up to 100bp, while the rest two for longer sequences ranged from 70bp to 1Mbp. BWA-MEM and BWA-SW share similar features such as long-read support and split alignment, but BWA-MEM, which is the latest, is generally recommended for high-quality queries as it is faster and more accurate. BWA-MEM also has better performance than BWA-backtrack for 70-100bp Illumina reads.

The screenshot shows the homepage of the Bowtie aligner. At the top, there is a dark header with the text "Bowtie" and "An ultrafast memory-efficient short read aligner". Below the header, there is a sidebar with the heading "BWA:" followed by links to "SF project page", "SF download page", "Mailing list", "BWA manual page", and "Repository". The main content area features the "mrFAST" logo and the text "Micro Read Fast Alignment Search Tool".

SOAP: short oligonucleotide alignment program FREE

Ruiqiang Li, Yingrui Li, Karsten Kristiansen, Jun Wang ✉ Author Notes

FM-index

Ferragina and Manzini [FOCS 2000]

Text index which allows fast substring queries in compressed space.

- Burrows-Wheeler transform (**BWT**).
- Suffix Array (**SA**) samples at constant distance intervals.

$$T = \text{ACAAACAC\$}$$

SA	\mathcal{M}	BWT
7	\$ACAAACAC	
2	AACAC\\$AC	
5	AC\\$ACAAAC	
0	ACAAACAC\$	
3	ACAC\\$ACA	
6	C\\$ACAAACA	
1	CAACAC\\$A	
4	CAC\\$ACAA	

FM-index

Ferragina and Manzini [FOCS 2000]

Text index which allows fast substring queries in compressed space.

- Burrows-Wheeler transform (**BWT**).
- Suffix Array (**SA**) samples at constant distance intervals.

Seed and extend: find exact short matches and then extend to find approximate alignments.

$$T = \text{ACAAACAC\$}$$

SA	\mathcal{M}	BWT
7		\$ACAAACAC
2		AACAC\\$AC
5		AC\\$ACAAAC
0		ACAAACAC\$
3		ACAC\\$ACA
6		C\\$ACAAACA
1		CAACAC\\$A
4		CAC\\$ACAA

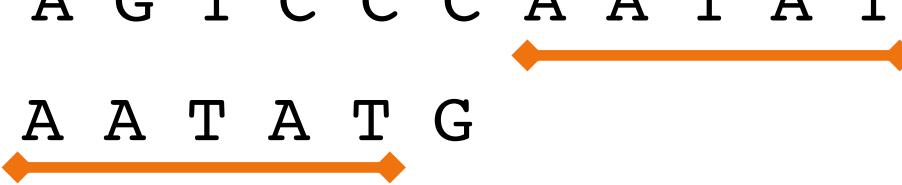


Maximal Exact Matches (MEMs)

Given a text $S[1..n]$ and a pattern $P[1..m]$. The substring $P[i..i + \ell - 1]$ of length ℓ is a **maximal exact match** (MEM) of P in S if:

$S: A\ G\ T\ T\ T\ A\ G\ T\ C\ C\ C\ A\ A\ T\ A\ T\ A\ T\ T\ T\ A$

$P: G\ G\ G\ C\ G\ A\ A\ T\ A\ T\ G$



1. $P[i .. i + \ell - 1]$ occurs in S ;
2. $P[i - 1..i + \ell - 1]$ and $P[i .. i + \ell]$ does not occur in S .

FM-index

Ferragina and Manzini [FOCS 2000]

Text index which allows fast substring queries in compressed space.

- Burrows-Wheeler transform (**BWT**).
- Suffix Array (**SA**) samples at constant distance intervals.

Indexing 500 GB of data would require 500 GB of memory.

$$T = \text{ACAAACAC\$}$$

SA	\mathcal{M}	BWT
7	\$ACAAACAC	
2	AACAC\\$AC	
5	AC\\$ACAAAC	
0	ACAAACAC\$	
3	ACAC\\$ACA	
6	C\\$ACAAACA	
1	CAACAC\\$A	
4	CAC\\$ACAA	

FM-index

Ferragina and Manzini [FOCS 2000]

Text index which allows fast substring queries in compressed space.

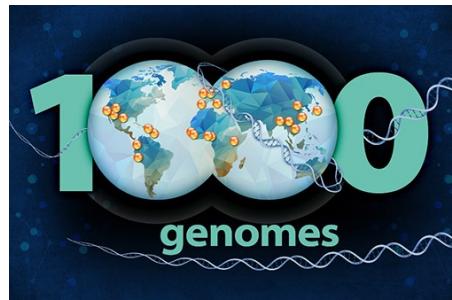
- Burrows-Wheeler transform (**BWT**).
- Suffix Array (**SA**) samples at constant distance intervals.

Indexing 5 TB of data would require 5 TB of memory.

$$T = \text{ACAAACAC\$}$$

SA	\mathcal{M}	BWT
7	\$ACAAACAC	
2	AACAC\\$AC	
5	AC\\$ACAAAC	
0	ACAAACAC\$	
3	ACAC\\$ACA	
6	C\\$ACAAACA	
1	CAACAC\\$A	
4	CAC\\$ACAA	

Our Goal



The 100,000 Genomes Project

Genomics England & Partners



Build a pangenomics index in sub-linear memory a manner that it can efficiently support read alignment.

RLBWT

Mäkinen and Navarro [TCS 2007]

Text index which allows fast substring queries in compressed space.

- Burrows-Wheeler transform (**BWT**).
- Suffix Array (**SA**) samples at constant distance intervals.

The RLBWT has size proportional to the number of runs in the BWT.

$$T = \text{ACAAACAC\$}$$



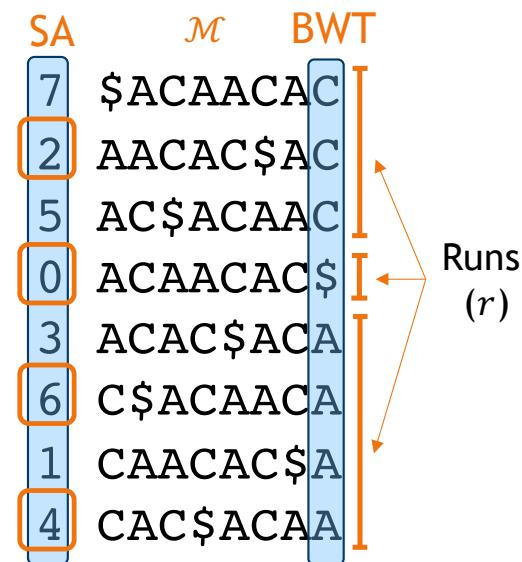
RLBWT

Text index which allows fast substring queries in compressed space.

- Burrows-Wheeler transform (**BWT**).
- Suffix Array (**SA**) samples at constant distance intervals.

The SA samples space scales linearly with the length of the text

$$T = \text{ACAAACAC\$}$$



r -index

Gagie, Navarro, Prezza [JACM 2020]

Gagie et al. defined a new SA sample that can be stored in $O(r)$ -space, where r is the number of runs of a character in the BWT.

- Burrows-Wheeler transform (**BWT**).
- Suffix Array (**SA**) samples at run boundaries.
- Additional (small) data structure to reconstruct **SA**.

$T = \text{ACAACAC\$}$

SA	\mathcal{M}	BWT
7	\$ACAACAC	
2	AACAC\$AC	
5	AC\$ACAAC	
0	ACAACAC\$	
3	ACAC\$ACA	
6	C\$ACAACA	
1	CAACAC\$A	
4	CAC\$ACAA	

Runs (r)

The diagram illustrates the construction of the BWT and SA for the string $T = \text{ACAACAC\$}$. The BWT is a 9x3 grid where each row is a suffix of T . The columns are labeled \mathcal{M} (middle column) and BWT (rightmost column). The SA is a list of indices [7, 2, 5, 0, 3, 6, 1, 4] indicating the position of each suffix in the BWT. Orange arrows point from the SA indices to their corresponding rows in the BWT grid. A bracket on the right side of the BWT grid is labeled 'Runs (r)'.

r -index

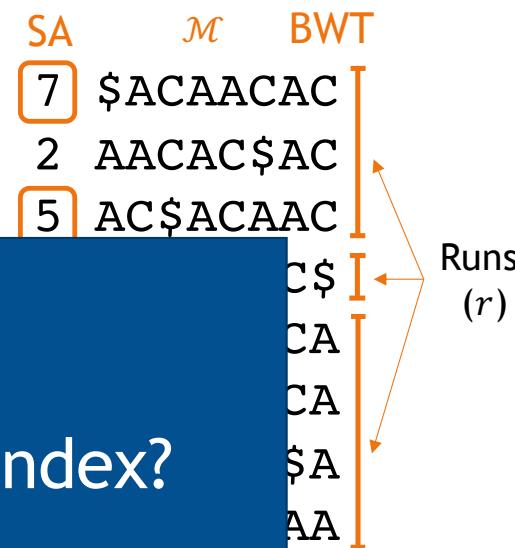
Gagie, Navarro, Prezza [JACM 2020]

Gagie et al. defined a new SA sample that can be stored in $O(r)$ -space, where r is the number of runs of a character in the BWT.

- Burrows-Wheeler transform (**BWT**).
- Suffix Array (**SA**) stores run boundaries.
- Additional reconstruction

Problem:
How to build the r -index?

$T = \text{ACAACAC\$}$



Prefix-free parsing

Boucher et al. [WABI 2018]

The **prefix-free parsing** P of S with **dictionary** D is computed by choosing a set E of strings of length w (***trigger strings***), and dividing S into overlapping phrases that start and end with a trigger string and does not contain any trigger string.

Prefix-free parsing

The **prefix-free parsing** P of S with **dictionary** D is computed by choosing a set E of strings of length w (*trigger strings*), and dividing S into overlapping phrases that start and end with a trigger string and does not contain any trigger string.

The r-index (SA + RLBWT) is constructed from P and D.

Prefix-free parsing

$S: \text{GATTACAT\#GATACAT\#GATTAGATA}$

We consider S to be circular and we append w copies of $\#$

$S: \text{GATTACAT\#GATACAT\#GATTAGATA\#\#}$

$$E = \{\text{AC}, \text{AG}, \text{T\#}, \#\#\}$$

$S: \text{GATTACAT\#GATACAT\#GATTAGATA\#\#}$

$P = D[1] \quad D[2] \quad D[4] \quad D[2] \quad D[5] \quad D[3]$

$D = \{\#\#GATTAC, ACAT\#, AGATA\#\#, T\#GATAC, T\#GATTAG\}$

Prefix-free parsing

$S: \text{GATTACAT\#GATACAT\#GATTAGATA}$

We consider S to be circular and we append w copies of $\#$

$S: \text{GATTACAT\#GATACAT\#GATTAGATA\#\#}$

$$E = \{\text{AC}, \text{AG}, \text{T\#}, \#\#\}$$

$S: \text{GATTACAT\#GATACAT\#GATTAGATA\#\#}$

$P = D[1] \quad D[2] \quad D[4] \quad D[2] \quad D[5] \quad D[3]$

$D = \{\#\#GATTAC, ACAT\#, AGATA\#\#, T\#GATAC, T\#GATTAG\}$

Prefix-free parsing

$T: \{ \text{\#}\#\text{GATTAC}, \#\text{GATTAC}, \text{GATTAC}, \text{ATTAC}, \text{TTAC}, \text{TAC},$
 $\text{ACAT\#}, \text{CAT\#}, \text{AT\#},$
 $\text{AGATA\#\#\#}, \text{GATA\#\#\#}, \text{ATA\#\#\#}, \text{TA\#\#\#}, \text{A\#\#\#}$
 $\text{T\#\#GATAC}, \#\text{GATAC}, \text{GATAC}, \text{ATAC}, \text{TAC},$
 $\text{T\#\#GATTAG}, \#\text{GATTAG}, \text{GATTAG}, \text{ATTAG}, \text{TTAG}, \text{TAG} \}$

$P = D[1] \quad D[2] \quad D[4] \quad D[2] \quad D[5] \quad D[3]$

$D = \{ \text{\#}\#\text{GATTAC}, \text{ACAT\#}, \text{AGATA\#\#\#}, \text{T\#\#GATAC}, \text{T\#\#GATTAG} \}$

Prefix-free parsing

$T: \{\#\#GATTAC, \#GATTAC, GATTAC, ATT,$
 $ACAT\#, CAT\#, AT\#,$
 $AGATA##, GATA##, ATA##, TA##,$
 $T\#GATAC, \#GATAC, GATAC, ATAC,$
 $T\#GATTAG, \#GATTAG, GATTAG, AT$

If a sequence in T is a prefix of another sequence in T

#GATTAC
#GATTACTTA

$P = D[1] \quad D[2] \quad D[4] \quad D[2] \quad D[5] \quad D[3]$

$D = \{\#\#GATTAC, ACAT\#, AGATA##, T\#GATAC, T\#GATTAG\}$

Prefix-free parsing

$T: \{\#\#GATTAC, \#GATTAC, GATTAC, ATT,$
 $ACAT\#, CAT\#, AT\#,$
 $AGATA##, GATA##, ATA##, TA##,$
T: $E = \{\underline{AC}, \underline{AG}, T\#, \#\#\}$ TAC, ATAC,
T#GATTAG, #GATTAG, GATTAG, AT

If a sequence in T is a prefix of another sequence in T

#GATTAC
#GATTACTTA

$P = D[1] \quad D[2] \quad D[4] \quad D[2] \quad D[5] \quad D[3]$

$D = \{\#\#GATTAC, ACAT\#, AGATA##, T\#GATAC, T\#GATTAG\}$

Prefix-free parsing

$T: \{ \text{\#}\#\text{GATTAC}, \text{\#GATTAC}, \text{GATTAC}, \text{ATTA}$
 $\quad \text{ACAT\#}, \text{CAT\#}, \text{AT\#},$
 $\quad \text{AGATA##}, \text{GATA##}, \text{ATA##}, \text{TA##}, \text{A}$
 $\quad \text{T\#GATAC}, \text{\#GATAC}, \text{GATAC}, \text{ATAC}, \text{T}$
 $\quad \text{T\#GATTAG}, \text{\#GATTAG}, \text{GATTAG}, \text{ATTA}$

If a sequence in T is only a suffix of one sequence in D then we need only to consider D.

$P = D[1] \quad D[2] \quad D[4] \quad D[2] \quad D[5] \quad D[3]$

$D = \{ \text{\#}\#\text{GATTAC}, \text{ACAT\#}, \text{AGATA##}, \text{T\#GATAC}, \text{T\#GATTAG} \}$

Prefix-free parsing

$T: \{ \text{\#}\#\text{GATTAC}, \text{\#GATTAC}, \text{GATTAC}, \text{ATTA}$
 $\quad \text{ACAT\#}, \text{CAT\#}, \text{AT\#},$
 $\quad \text{AGATA##}, \text{GATA##}, \text{ATA##}, \text{TA##}, \text{A}$
 $\quad \text{T\#GATAC}, \text{\#GATAC}, \text{GATAC}, \text{ATAC}, \text{T}$
 $\quad \text{T\#GATTAG}, \text{\#GATTAG}, \text{GATTAG}, \text{ATTAG} \}$

If a sequence in T is only a suffix of one sequence in D then we need only to consider D.

$P = D[1] \quad D[2] \quad D[4] \quad D[2] \quad D[5] \quad D[3]$

$D = \{ \text{\#}\#\text{GATTAC}, \text{ACAT\#}, \text{AGATA##}, \text{T\#GATAC}, \text{T\#GATTAG} \}$

Prefix-free parsing

If a sequence in T is a suffix of more than one sequence in D then we consider D and P.

$$P = D[1] \quad D[2] \quad D[4] \quad D[2] \quad D[5] \quad D[3]$$

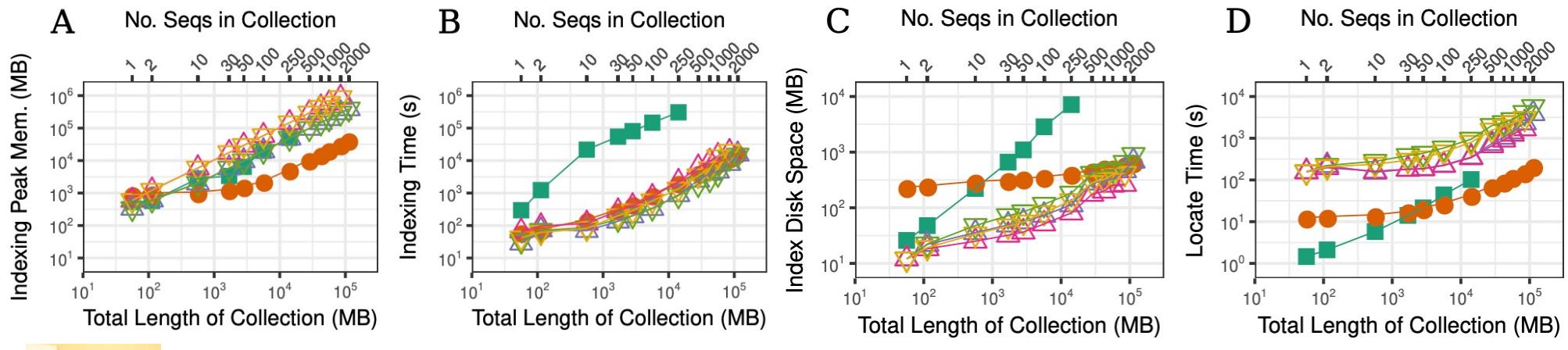
$$D = \{ \text{##GATTAC}, \text{ACAT\#}, \text{AGATA\#\#}, \text{T\#GATAC}, \text{T\#GATTAG} \}$$

C, GATTAC, ATTAC, TTAC, TAC,
#, #, #, #, #
, ATA##, TA##, A##
, GATAC, ATAC, TAC,
TAG, GATTAG, ATTAG, TTAG, TAG}

Toward a sub-linear solution

Mun et al. [RECOMB 2019]

We gave an algorithm for indexing repetitive text in sub-linear time [RECOMB 2019].

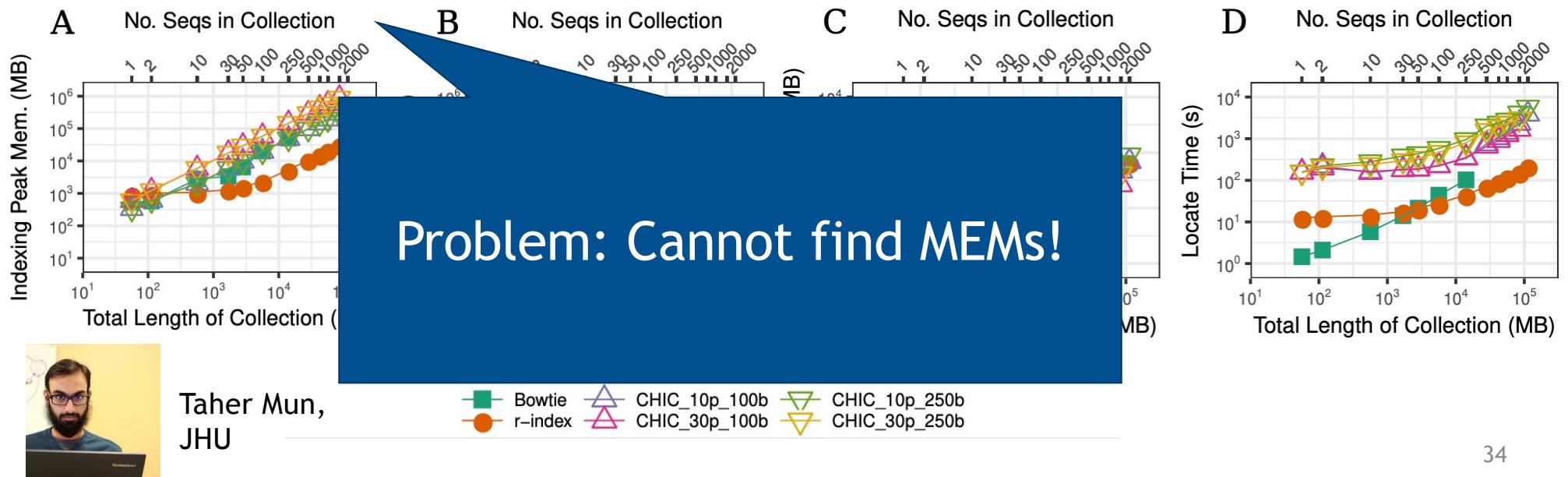


Taher Mun,
JHU

Toward a sub-linear solution

Mun et al. [RECOMB 2019]

We gave an algorithm for indexing repetitive text in sub-linear time [RECOMB 2019].



Matching statistics

The **matching statistics** of P with respect to S is the array $M[1..m]$ such that $M[i]$ is the length of the longest prefix of $P[i..m]$ that occurs in S .

$S: GATTACAT\$ GATACAT\$ GATTAGATA\$$

$P: TATACAGAT$

$M: 2 5 4 3 2 4 3 2 1$

Matching statistics using the LCP Array

$S: GATTACAT\$GATACAT\$GATTAGATA\#$
 $BWT(S): ATTTCCTGGGAAA\$#\$AAATATAA$

$P: TATACAGAT$

$M:$

SA	LCP	BWT	F	\mathcal{M}	L
26	0	A	#GATTACAT\\$GATACAT\\$GATTAGATA		
8	0	T	\\$GATACAT\\$GATTAGATA\#GATTACAT		
16	4	T	\\$GATTAGATA\#GATTACAT\\$GATACAT		
25	0	T	A\#GATTACAT\\$GATACAT\\$GATTAGATA		
4	1	T	ACAT\\$GATACAT\\$GATTAGATA\#GATT		
12	8	T	ACAT\\$GATAGATAG\#ATTACAT\\$GAT		
21	1	T	AGATA\#GATTACAT\\$GATACAT\\$GATT		
6	1	C	AT\\$GATACAT\\$GATTAGATA\#GATTAC		
14	6	C	AT\\$GATAGATA\#GATTACAT\\$GATAC		
23	2	G	ATA\#GATTACAT\\$GATACAT\\$GATTAG		
10	3	G	ATACAT\\$GATTAGATA\#GATTACAT\\$G		
1	2	G	ATTACAT\\$GATACAT\\$GATTAGATA\#G		
18	4	G	ATTAGATA\#GATTACAT\\$GATACAT\\$G		
5	0	A	CAT\\$GATACAT\\$GATTAGATA\#GATT		
13	7	A	CAT\\$GATAGATA\#GATTACAT\\$GATA		
22	0	A	GATA\#GATTACAT\\$GATACAT\\$GATT		
9	4	\\$	GATACAT\\$GATTAGATA\#GATTACAT\\$		
0	3	#	GATTACAT\\$GATACAT\\$GATTAGATA\#		
17	5	\\$	GATTAGATA\#GATTACAT\\$GATACAT\\$		
7	0	A	T\\$GATACAT\\$GATTAGATA\#GATTAC		
15	5	A	T\\$GATAGATA\#GATTACAT\\$GATAC		
24	1	A	TA\#GATTACAT\\$GATACAT\\$GATTAGA		
3	2	T	TACAT\\$GATACAT\\$GATTAGATA\#GAT		
11	9	A	TACAT\\$GATAGATA\#GATTACAT\\$GA		
20	2	T	TAGATA\#GATTACAT\\$GATACAT\\$GAT		
2	1	A	TTACAT\\$GATACAT\\$GATTAGATA\#CA		
19	3	A	TTAGATA\#GATTACAT\\$GATACAT\\$GA		

Matching statistics using the LCP Array

$S: GATTACAT\$GATACAT\$GATTAGATA\#$
 $BWT(S): ATTTCCTGGGAAA\$#\$AAATATAA$

$P: T A T A C A G A T$

$M:$

SA	LCP	BWT	F	\mathcal{M}	L
26	0	A	#GATTACAT\\$GATACAT\\$GATTAGATA		
8	0	T	\$GATACAT\\$GATTAGATA#GATTACAT		
16	4	T	\$GATTAGATA#GATTACAT\\$GATACAT		
25	0	T	A#GATTACAT\\$GATACAT\\$GATTAGAT		
4	1	T	ACAT\\$GATACAT\\$GATTAGATA#GATT		
12	8	T	ACAT\\$GATAGATAG#ATTACAT\\$GAT		
21	1	T	AGATA#GATTACAT\\$GATACAT\\$GATT		
6	1	C	AT\\$GATACAT\\$GATTAGATA#GATTAC		
14	6	C	AT\\$GATAGATA#GATTACAT\\$GATAC		
23	2	G	ATA#GATTACAT\\$GATACAT\\$GATTAG		
10	3	G	ATACAT\\$GATTAGATA#GATTACAT\\$G		
1	2	G	ATTACAT\\$GATACAT\\$GATTAGATA#G		
18	4	G	ATTAGATA#GATTACAT\\$GATACAT\\$G		
5	0	A	CAT\\$GATACAT\\$GATTAGATA#GATT		
13	7	A	CAT\\$GATAGATA#GATTACAT\\$GATA		
22	0	A	GATA#GATTACAT\\$GATACAT\\$GATT		
9	4	S	GATACAT\\$GATTAGATA#GATTACAT\\$		
0	3	#	GATTACAT\\$GATACAT\\$GATTAGATA#		
17	5	S	GATTAGATA#GATTACAT\\$GATACAT\\$		
7	0	A	T\\$GATACAT\\$GATTAGATA#GATTAC		
15	5	A	T\\$GATAGATA#GATTACAT\\$GATACA		
24	1	A	TA#GATTACAT\\$GATACAT\\$GATTAGA		
3	2	T	TACAT\\$GATACAT\\$GATTAGATA#GAT		
11	9	A	TACAT\\$GATAGATA#GATTACAT\\$GA		
20	2	T	TAGATA#GATTACAT\\$GATACAT\\$GAT		
2	1	A	TTACAT\\$GATACAT\\$GATTAGATA#CA		
19	3	A	TTAGATA#GATTACAT\\$GATACAT\\$GA		

Matching statistics using the LCP Array

$S: GATTACAT\$GATACAT\$GATTAGATA\#$
 $BWT(S): ATTTCCTGGGAAA\$#\$AAATATAA$

$P: T A T A C A G A T$

$M:$ 1

SA	LCP	BWT	F	\mathcal{M}	L
26	0	A	#GATTACAT\\$GATACAT\\$GATTAGATA		
8	0	T	\\$GATACAT\\$GATTAGATA#GATTACAT		
16	4	T	\\$GATTAGATA#GATTACAT\\$GATACAT		
25	0	T	A#GATTACAT\\$GATACAT\\$GATTAGATA		
4	1	T	ACAT\\$GATACAT\\$GATTAGATA#GATT		
12	8	T	ACAT\\$GATAGATAG#ATTACAT\\$GAT		
21	1	T	AGATA#GATTACAT\\$GATACAT\\$GATT		
6	1	C	AT\\$GATACAT\\$GATTAGATA#GATTAC		
14	6	C	AT\\$GATAGATA#GATTACAT\\$GATAC		
23	2	G	ATA#GATTACAT\\$GATACAT\\$GATTAG		
10	3	G	ATACAT\\$GATTAGATA#GATTACAT\\$G		
1	2	G	ATTACAT\\$GATACAT\\$GATTAGATA#G		
18	4	G	ATTAGATA#GATTACAT\\$GATACAT\\$G		
5	0	A	CAT\\$GATACAT\\$GATTAGATA#GATT		
13	7	A	CAT\\$GATAGATA#GATTACAT\\$GATA		
22	0	A	GATA#GATTACAT\\$GATACAT\\$GATT		
9	4	\\$	GATACAT\\$GATTAGATA#GATTACAT\\$		
0	3	#	GATTACAT\\$GATACAT\\$GATTAGATA#		
17	5	\\$	GATTAGATA#GATTACAT\\$GATACAT\\$		
7	0	A	T\\$GATACAT\\$GATTAGATA#GATTAC		
15	5	A	T\\$GATAGATA#GATTACAT\\$GATAC		
24	1	A	TA#GATTACAT\\$GATACAT\\$GATTAGA		
3	2	T	TAACAT\\$GATACAT\\$GATTAGATA#GAT		
11	9	A	TAACAT\\$GATAGATA#GATTACAT\\$GA		
20	2	T	TAAGATA#GATTACAT\\$GATACAT\\$GAT		
2	1	A	TTACAT\\$GATACAT\\$GATTAGATA#CA		
19	3	A	TTAGATA#GATTACAT\\$GATACAT\\$GA		

Matching statistics using the LCP Array

$S: GATTACAT\$GATACAT\$GATTAGATA\#$
 $BWT(S): ATTTCCTGGGAAA\$#\$AAATATAA$

$P: T A T A C A G A T$

$M:$ 1

SA	LCP	BWT	F	\mathcal{M}	L
26	0	A	#GATTACAT\\$GATACAT\\$GATTAGATA		
8	0	T	\\$GATACAT\\$GATTAGATA#GATTACAT		
16	4	T	\\$GATTAGATA#GATTACAT\\$GATACAT		
25	0	T	A#GATTACAT\\$GATACAT\\$GATTAGATA		
4	1	T	ACAT\\$GATACAT\\$GATTAGATA#GATT		
12	8	T	ACAT\\$GATAGATAG#ATTACAT\\$GAT		
21	1	T	AGATA#GATTACAT\\$GATACAT\\$GATT		
6	1	C	AT\\$GATACAT\\$GATTAGATA#GATTAC		
14	6	C	AT\\$GATTAGATA#GATTACAT\\$GATAC		
23	2	G	ATA#GATTACAT\\$GATACAT\\$GATTAG		
10	3	G	ATACAT\\$GATTAGATA#GATTACAT\\$G		
1	2	G	ATTACAT\\$GATACAT\\$GATTAGATA#G		
18	4	G	ATTAGATA#GATTACAT\\$GATACAT\\$G		
5	0	A	CAT\\$GATACAT\\$GATTAGATA#GATT		
13	7	A	CAT\\$GATTAGATA#GATTACAT\\$GATA		
22	0	A	GATA#GATTACAT\\$GATACAT\\$GATT		
9	4	\\$	GATACAT\\$GATTAGATA#GATTACAT\\$		
0	3	#	GATTACAT\\$GATACAT\\$GATTAGATA#		
17	5	\\$	GATTAGATA#GATTACAT\\$GATACAT\\$		
7	0	A	T\\$GATACAT\\$GATTAGATA#GATTAC		
15	5	A	T\\$GATTAGATA#GATTACAT\\$GATAC		
24	1	A	TA#GATTACAT\\$GATACAT\\$GATTAGA		
3	2	T	TACAT\\$GATACAT\\$GATTAGATA#GAT		
11	9	A	TACAT\\$GATTAGATA#GATTACAT\\$GA		
20	2	T	TAGATA#GATTACAT\\$GATACAT\\$GAT		
2	1	A	TTACAT\\$GATACAT\\$GATTAGATA#CA		
19	3	A	TTAGATA#GATTACAT\\$GATACAT\\$GA		

Matching statistics using the LCP Array

$S: GATTACAT\$GATACAT\$GATTAGATA\#$
 $BWT(S): ATTTCCTGGGAA\$#\$AAATATAA$

$P: T A T A C A G \textcolor{orange}{A} T$

$M:$ 2 1

SA	LCP	BWT	F	\mathcal{M}	L
26	0	A	#GATTACAT\\$GATACAT\\$GATTAGATA		
8	0	T	\\$GATACAT\\$GATTAGATA\#GATTACAT		
16	4	T	\\$GATTAGATA\#GATTACAT\\$GATACAT		
25	0	T	A\#GATTACAT\\$GATACAT\\$GATTAGATA		
4	1	T	ACAT\\$GATACAT\\$GATTAGATA\#GATT		
12	8	T	ACAT\\$GATAGATAG\#ATTACAT\\$GAT		
21	1	T	AGATA\#GATTACAT\\$GATACAT\\$GATT		
6	1	C	AT\\$GATACAT\\$GATTAGATA\#GATTAC		
14	6	C	AT\\$GATTAGATA\#GATTACAT\\$GATAC		
23	2	G	ATA\#GATTACAT\\$GATACAT\\$GATTAG		
10	3	G	ATACAT\\$GATTAGATA\#GATTACAT\\$G		
1	2	G	ATTACAT\\$GATACAT\\$GATTAGATA\#G		
18	4	G	ATTAGATA\#GATTACAT\\$GATACAT\\$G		
5	0	A	CAT\\$GATACAT\\$GATTAGATA\#GATT		
13	7	A	CAT\\$GATTAGATA\#GATTACAT\\$GATA		
22	0	A	GATA\#GATTACAT\\$GATACAT\\$GATT		
9	4	\\$	GATACAT\\$GATTAGATA\#GATTACAT\\$		
0	3	\#	GATTACAT\\$GATACAT\\$GATTAGATA\#		
17	5	\\$	GATTAGATA\#GATTACAT\\$GATACAT\\$		
7	0	A	T\\$GATACAT\\$GATTAGATA\#GATTAC		
15	5	A	T\\$GATTAGATA\#GATTACAT\\$GATAC		
24	1	A	TA\#GATTACAT\\$GATACAT\\$GATTAGA		
3	2	T	TACAT\\$GATACAT\\$GATTAGATA\#GAT		
11	9	A	TACAT\\$GATTAGATA\#GATTACAT\\$GA		
20	2	T	TAGATA\#GATTACAT\\$GATACAT\\$GAT		
2	1	A	TTACAT\\$GATACAT\\$GATTAGATA\#CA		
19	3	A	TTAGATA\#GATTACAT\\$GATACAT\\$GA		

Matching statistics using the LCP Array

$S: GATTACAT\$GATACAT\$GATTAGATA\#$
 $BWT(S): ATTTCCTGGGAA\$#\$AAATATAA$

$P: T A T A C A \textcolor{orange}{G} A T$

$M: \quad \quad \quad 4 \ 3 \ 2 \ 1$

SA	LCP	BWT	F	\mathcal{M}	L
26	0	A	#GATTACAT\\$GATACAT\\$GATTAGATA		
8	0	T	\\$GATACAT\\$GATTAGATA\#GATTACAT		
16	4	T	\\$GATTAGATA\#GATTACAT\\$GATACAT		
25	0	T	A\#GATTACAT\\$GATACAT\\$GATTAGAT		
4	1	T	ACAT\\$GATACAT\\$GATTAGATA\#GATT		
12	8	T	ACAT\\$GATAGATAG\#ATTACAT\\$GAT		
21	1	\textcolor{red}{T}	\textcolor{red}{AGATA}\#GATTACAT\\$GATACAT\\$GATT		
6	1	C	AT\\$GATACAT\\$GATTAGATA\#GATTAC		
14	6	C	AT\\$GATAGATA\#GATTACAT\\$GATAC		
23	2	G	ATA\#GATTACAT\\$GATACAT\\$GATTAG		
10	3	G	ATACAT\\$GATTAGATA\#GATTACAT\\$G		
1	2	G	ATTACAT\\$GATACAT\\$GATTAGATA\#G		
18	4	G	ATTAGATA\#GATTACAT\\$GATACAT\\$G		
5	0	A	CAT\\$GATACAT\\$GATTAGATA\#GATT		
13	7	A	CAT\\$GATAGATA\#GATTACAT\\$GATA		
22	0	A	GATA\#GATTACAT\\$GATACAT\\$GATT		
9	4	\\$	GATACAT\\$GATTAGATA\#GATTACAT\\$		
0	3	\#	GATTACAT\\$GATACAT\\$GATTAGATA\#		
17	5	\\$	GATTAGATA\#GATTACAT\\$GATACAT\\$		
7	0	A	T\\$GATACAT\\$GATTAGATA\#GATTAC		
15	5	A	T\\$GATAGATA\#GATTACAT\\$GATAC		
24	1	A	TA\#GATTACAT\\$GATACAT\\$GATTAGA		
3	2	T	TACAT\\$GATACAT\\$GATTAGATA\#GAT		
11	9	A	TACAT\\$GATAGATA\#GATTACAT\\$GA		
20	2	T	TAAGATA\#GATTACAT\\$GATACAT\\$GAT		
2	1	A	TTACAT\\$GATACAT\\$GATTAGATA\#CA		
19	3	A	TTAGATA\#GATTACAT\\$GATACAT\\$GA		

Matching statistics using the LCP Array

$S: G A T T A C A T \$ G A T A C A T \$ G A T T A G A T A \#$

$BWT(S): A T T T T T C C G G G G A A A \$ \# \$ A A A T A T A A$

$P: T A T A C A G A T$

$M: 4 3 2 1$

Find the longest prefix of $A G A T$ that is preceded by C in S .

It is the longest prefix of the suffix of S corresponding to either the preceding C or the following C in the BWT of S .

We use the LCP array to find the length of the longest prefix.

SA	LCP	BWT	F	\mathcal{M}	L
26	0	A	#GATTACAT\$GATACAT\$GATTAGATA		
8	0	T	\$GATACAT\$GATTAGATA#GATTACAT		
16	4	T	\$GATTAGATA#GATTACAT\$GATACAT		
25	0	T	A#GATTACAT\$GATACAT\$GATTAGATA		
4	1	T	ACAT\$GATACAT\$GATTAGATA#GATT		
12	8	T	ACAT\$GATAGATAG#ATTACAT\$GAT		
21	1	I	AGATA#GATTACAT\$GATACAT\$GATT		
6	1	I	AT\$GATACAT\$GATTAGATA#GATTAC		
14	6	C	AT\$GATAGATA#GATTACAT\$GATAC		
23	2	G	ATA#GATTACAT\$GATACAT\$GATTAG		
10	3	G	ATACAT\$GATTAGATA#GATTACAT\$G		
1	2	G	ATTACAT\$GATACAT\$GATTAGATA#G		
18	4	G	ATTAGATA#GATTACAT\$GATACAT\$G		
5	0	A	CAT\$GATACAT\$GATTAGATA#GATT		
13	7	A	CAT\$GATAGATA#GATTACAT\$GATA		
22	0	A	GATA#GATTACAT\$GATACAT\$GATT		
9	4	\$	GATACAT\$GATTAGATA#GATTACAT\$		
0	3	#	GATTACAT\$GATACAT\$GATTAGATA#		
17	5	\$	GATTAGATA#GATTACAT\$GATACAT\$		
7	0	A	T\$GATACAT\$GATTAGATA#GATTAC		
15	5	A	T\$GATAGATA#GATTACAT\$GATACA		
24	1	A	TA#GATTACAT\$GATACAT\$GATTAGA		
3	2	T	TACAT\$GATACAT\$GATTAGATA#GAT		
11	9	A	TACAT\$GATAGATA#GATTACAT\$GA		
20	2	T	TAGATA#GATTACAT\$GATACAT\$GAT		
2	1	A	TTACAT\$GATACAT\$GATTAGATA#CA		
19	3	A	TTAGATA#GATTACAT\$GATACAT\$GA		

Matching statistics using the LCP Array

$S: G A T T A C A T \$ G A T A C A T \$ G A T T A G A T A \#$

$BWT(S): A T T T T T C C G G G G A A A \$ \# \$ A A A T A T A A$

$P: T A T A C A \textcolor{orange}{G} A T$

$M: \quad \quad \textcolor{orange}{4} \ 3 \ 2 \ 1$

Find the longest prefix of $\textcolor{orange}{A} \ G \ A \ T$ that is preceded by C in S .

It is the longest prefix of the suffix of S corresponding to either the preceding C or the following C in the BWT of S .

We use the LCP array to find the length of the longest prefix.

SA	LCP	BWT	F	\mathcal{M}	L
26	0	A	#GATTACAT\$GATACAT\$GATTAGATA		
8	0	T	\$GATACAT\$GATTAGATA#GATTACAT		
16	4	T	\$GATTAGATA#GATTACAT\$GATACAT		
25	0	T	$\textcolor{orange}{A}$ #GATTACAT\$GATACAT\$GATTAGATA		
4	1	T	ACAT\$GATACAT\$GATTAGATA#GATT		
12	8	T	ACAT\$GATAGATAG#ATTACAT\$GAT		
21	1	T	AGATA#GATTACAT\$GATACAT\$GATT		
6	1	C	AT\$GATACAT\$GATTAGATA#GATTAC		
14	6	C	AT\$GATAGATA#GATTACAT\$GATAC		
23	2	G	ATA#GATTACAT\$GATACAT\$GATTAG		
10	3	G	ATACAT\$GATTAGATA#GATTACAT\$G		
1	2	G	ATTACAT\$GATACAT\$GATTAGATA#G		
18	4	G	ATTAGATA#GATTACAT\$GATACAT\$G		
5	0	A	CAT\$GATACAT\$GATTAGATA#GATT		
13	7	A	CAT\$GATAGATA#GATTACAT\$GATA		
22	0	A	GATA#GATTACAT\$GATACAT\$GATT		
9	4	\$	GATACAT\$GATTAGATA#GATTACAT\$		
0	3	#	GATTACAT\$GATACAT\$GATTAGATA#		
17	5	\$	GATTAGATA#GATTACAT\$GATACAT\$		
7	0	A	T\$GATACAT\$GATTAGATA#GATTAC		
15	5	A	T\$GATAGATA#GATTACAT\$GATACA		
24	1	A	TA#GATTACAT\$GATACAT\$GATTAGA		
3	2	T	TACAT\$GATACAT\$GATTAGATA#GAT		
11	9	A	TACAT\$GATAGATA#GATTACAT\$GA		
20	2	T	TAGATA#GATTACAT\$GATACAT\$GAT		
2	1	A	TTACAT\$GATACAT\$GATTAGATA#CA		
19	3	A	TTAGATA#GATTACAT\$GATACAT\$GA		

Matching statistics using the LCP Array

$S: G A T T A C A T \$ G A T A C A T \$ G A T T A G A T A \#$

$BWT(S): A T T T T T C C G G G G A A A \$ \# \$ A A A T A T A A$

$P: T A T A C A \textcolor{orange}{G} A T$

$M: \quad \quad \textcolor{orange}{4} \ 3 \ 2 \ 1$

Find the longest prefix of $\textcolor{orange}{A} \ G \ A \ T$ that is preceded by C in S .

It is the longest prefix of the suffix of S corresponding to either the preceding C or the following C in the BWT of S .

We use the LCP array to find the length of the longest prefix.

SA	LCP	BWT	F	\mathcal{M}	L
26	0	A	#GATTACAT\$GATACAT\$GATTAGATA		
8	0	T	\$GATACAT\$GATTAGATA#GATTACAT		
16	4	T	\$GATTAGATA#GATTACAT\$GATACAT		
25	0	T	$\textcolor{orange}{A}$ #GATTACAT\$GATACAT\$GATTAGATA		
4	1	T	ACAT\$GATACAT\$GATTAGATA#GATT		
12	8	T	ACAT\$GATAGATAG#ATTACAT\$GAT		
21	1	T	AGATA#GATTACAT\$GATACAT\$GATT		
6	1	C	AT\$GATACAT\$GATTAGATA#GATTAC		
14	6	C	AT\$GATAGATA#GATTACAT\$GATAC		
23	2	G	ATA#GATTACAT\$GATACAT\$GATTAG		
10	3	G	ATACAT\$GATTAGATA#GATTACAT\$G		
1	2	G	ATTACAT\$GATACAT\$GATTAGATA#G		
18	4	G	ATTAGATA#GATTACAT\$GATACAT\$G		
5	0	A	CAT\$GATACAT\$GATTAGATA#GATT		
13	7	A	CAT\$GATAGATA#GATTACAT\$GATA		
22	0	A	GATA#GATTACAT\$GATACAT\$GATT		
9	4	\$	GATACAT\$GATTAGATA#GATTACAT\$		
0	3	#	GATTACAT\$GATACAT\$GATTAGATA#		
17	5	\$	GATTAGATA#GATTACAT\$GATACAT\$		
7	0	A	T\$GATACAT\$GATTAGATA#GATTAC		
15	5	A	T\$GATAGATA#GATTACAT\$GATACA		
24	1	A	TA#GATTACAT\$GATACAT\$GATTAGA		
3	2	T	TACAT\$GATACAT\$GATTAGATA#GAT		
11	9	A	TACAT\$GATAGATA#GATTACAT\$GA		
20	2	T	TAGATA#GATTACAT\$GATACAT\$GAT		
2	1	A	TTACAT\$GATACAT\$GATTAGATA#CA		
19	3	A	TTAGATA#GATTACAT\$GATACAT\$GA		

Matching statistics using the LCP Array

$S: GATTACAT\$GATACAT\$GATTAGATA\#$
 $BWT(S): ATTTCCTGGGAAA\$#\$AAATATAA$

$P: T A T A C A G A T$

$M: \begin{matrix} 2 & 4 & 3 & 2 & 1 \end{matrix}$

SA	LCP	BWT	F	\mathcal{M}	L
26	0	A	#GATTACAT\\$GATACAT\\$GATTAGATA		
8	0	T	\\$GATACAT\\$GATTAGATA#GATTACAT		
16	4	T	\\$GATTAGATA#GATTACAT\\$GATACAT		
25	0	T	A#GATTACAT\\$GATACAT\\$GATTAGAT		
4	1	T	ACAT\\$GATACAT\\$GATTAGATA#GATT		
12	8	T	ACAT\\$GATAGATAG#ATTACAT\\$GAT		
21	1	T	AGATA#GATTACAT\\$GATACAT\\$GATT		
6	1	C	AT\\$GATACAT\\$GATTAGATA#GATTAC		
14	6	C	AT\\$GATAGATA#GATTACAT\\$GATAC		
23	2	G	ATA#GATTACAT\\$GATACAT\\$GATTAG		
10	3	G	ATACAT\\$GATTAGATA#GATTACAT\\$G		
1	2	G	ATTACAT\\$GATACAT\\$GATTAGATA#G		
18	4	G	ATTAGATA#GATTACAT\\$GATACAT\\$G		
5	0	I A	CAT\\$GATACAT\\$GATTAGATA#GATT		
13	7	A	CAT\\$GATAGATA#GATTACAT\\$GATA		
22	0	A	GATA#GATTACAT\\$GATACAT\\$GATT		
9	4	\\$	GATACAT\\$GATTAGATA#GATTACAT\\$		
0	3	#	GATTACAT\\$GATACAT\\$GATTAGATA#		
17	5	\\$	GATTAGATA#GATTACAT\\$GATACAT\\$		
7	0	A	T\\$GATACAT\\$GATTAGATA#GATTAC		
15	5	A	T\\$GATAGATA#GATTACAT\\$GATAC		
24	1	A	TA#GATTACAT\\$GATACAT\\$GATTAGA		
3	2	T	TACAT\\$GATACAT\\$GATTAGATA#GAT		
11	9	A	TACAT\\$GATAGATA#GATTACAT\\$GA		
20	2	T	TAAGATA#GATTACAT\\$GATACAT\\$GAT		
2	1	A	TTACAT\\$GATACAT\\$GATTAGATA#CA		
19	3	A	TTAGATA#GATTACAT\\$GATACAT\\$GA		

Matching statistics using the LCP Array

$S: GATTACAT\$GATACAT\$GATTAGATA\#$
 $BWT(S): ATTTCCTGGGAAA\$#\$AAATATAA$

$P: T A T \textcolor{orange}{A C A} G A T$

$M: \quad \textcolor{orange}{2} \ 4 \ 3 \ 2 \ 1$

SA	LCP	BWT	F	\mathcal{M}	L
26	0	A	#GATTACAT\\$GATACAT\\$GATTAGATA		
8	0	T	\\$GATACAT\\$GATTAGATA\#GATTACAT		
16	4	T	\\$GATTAGATA\#GATTACAT\\$GATACAT		
25	0	T	A\#GATTACAT\\$GATACAT\\$GATTAGAT		
4	1	T	ACAT\\$GATACAT\\$GATTAGATA\#GATT		
12	8	T	ACAT\\$GATTAGATA\#ATTACAT\\$GAT		
21	1	T	AGATA\#GATTACAT\\$GATACAT\\$GATT		
6	1	C	AT\\$GATACAT\\$GATTAGATA\#GATTAC		
14	6	C	AT\\$GATTAGATA\#GATTACAT\\$GATAC		
23	2	G	ATA\#GATTACAT\\$GATACAT\\$GATTAG		
10	3	G	ATACAT\\$GATTAGATA\#GATTACAT\\$G		
1	2	G	ATTACAT\\$GATACAT\\$GATTAGATA\#G		
18	4	G	ATTAGATA\#GATTACAT\\$GATACAT\\$G		
5	0	\textcolor{orange}{I} A	CAT\\$GATACAT\\$GATTAGATA\#GATT		
13	7	A	CAT\\$GATTAGATA\#GATTACAT\\$GATA		
22	0	A	GATA\#GATTACAT\\$GATACAT\\$GATT		
9	4	\\$	GATACAT\\$GATTAGATA\#GATTACAT\\$		
0	3	#	GATTACAT\\$GATACAT\\$GATTAGATA\#		
17	5	\\$	GATTAGATA\#GATTACAT\\$GATACAT\\$		
7	0	A	T\\$GATACAT\\$GATTAGATA\#GATTAC		
15	5	A	T\\$GATTAGATA\#GATTACAT\\$GATAC		
24	1	A	TA\#GATTACAT\\$GATACAT\\$GATTAGA		
3	2	T	TACAT\\$GATACAT\\$GATTAGATA\#GAT		
11	9	A	TACAT\\$GATTAGATA\#GATTACAT\\$GA		
20	2	T	TAAGATA\#GATTACAT\\$GATACAT\\$GAT		
2	1	A	TTACAT\\$GATACAT\\$GATTAGATA\#CA		
19	3	A	TTAGATA\#GATTACAT\\$GATACAT\\$GA		

Matching statistics using the LCP Array

$S: GATTACAT\$GATACAT\$GATTAGATA\#$
 $BWT(S): ATTTCCTGGGAA\$#\$AAATATAA$

$P: T A T \textcolor{orange}{A C A} G A T$

$M: \textcolor{orange}{3} \ 2 \ 4 \ 3 \ 2 \ 1$

SA	LCP	BWT	F	\mathcal{M}	L
26	0	A	#GATTACAT\\$GATACAT\\$GATTAGATA		
8	0	T	\\$GATACAT\\$GATTAGATA\#GATTACAT		
16	4	T	\\$GATTAGATA\#GATTACAT\\$GATACAT		
25	0	T	A\#GATTACAT\\$GATACAT\\$GATTAGAT		
4	1	T	\textcolor{red}{ACAT\\$GATACAT\\$GATTAGATA\#GATT}		
12	8	T	\textcolor{red}{ACAT\\$GATTAGATAG\#ATTACAT\\$GAT}		
21	1	T	AGATA\#GATTACAT\\$GATACAT\\$GATT		
6	1	C	AT\\$GATACAT\\$GATTAGATA\#GATTAC		
14	6	C	AT\\$GATTAGATA\#GATTACAT\\$GATAC		
23	2	G	ATA\#GATTACAT\\$GATACAT\\$GATTAG		
10	3	G	ATACAT\\$GATTAGATA\#GATTACAT\\$G		
1	2	G	ATTACAT\\$GATACAT\\$GATTAGATA\#G		
18	4	G	ATTAGATA\#GATTACAT\\$GATACAT\\$G		
5	0	A	CAT\\$GATACAT\\$GATTAGATA\#GATT		
13	7	A	CAT\\$GATTAGATA\#GATTACAT\\$GATA		
22	0	A	GATA\#GATTACAT\\$GATACAT\\$GATT		
9	4	\\$	GATACAT\\$GATTAGATA\#GATTACAT\\$		
0	3	\#	GATTACAT\\$GATACAT\\$GATTAGATA\#		
17	5	\\$	GATTAGATA\#GATTACAT\\$GATACAT\\$		
7	0	A	T\\$GATACAT\\$GATTAGATA\#GATTAC		
15	5	A	\textcolor{brown}{T\\$GATTAGATA\#GATTACAT\\$GATAC}		
24	1	A	TA\#GATTACAT\\$GATACAT\\$GATTAGA		
3	2	T	TACAT\\$GATACAT\\$GATTAGATA\#GAT		
11	9	A	TACAT\\$GATTAGATA\#GATTACAT\\$GA		
20	2	T	TAAGATA\#GATTACAT\\$GATACAT\\$GAT		
2	1	A	TTACAT\\$GATACAT\\$GATTAGATA\#CA		
19	3	A	TTAGATA\#GATTACAT\\$GATACAT\\$GA		

Matching statistics using the LCP Array

$S: GATTACAT\$GATACAT\$GATTAGATA\#$
 $BWT(S): ATTTCCTGGGAAA\$#\$AAATATAA$

$P: \textcolor{orange}{T} A T A C A G A T$

$M: \textcolor{orange}{2} 5 4 3 2 4 3 2 1$

SA	LCP	BWT	F	\mathcal{M}	L
26	0	A	#GATTACAT\\$GATACAT\\$GATTAGATA		
8	0	T	\\$GATACAT\\$GATTAGATA\#GATTACAT		
16	4	T	\\$GATTAGATA\#GATTACAT\\$GATACAT		
25	0	T	A\#GATTACAT\\$GATACAT\\$GATTAGAT		
4	1	T	ACAT\\$GATACAT\\$GATTAGATA\#GATT		
12	8	T	ACAT\\$GATTAGATA\#ATTACAT\\$GAT		
21	1	\textcolor{orange}{T}	AGATA\#GATTACAT\\$GATACAT\\$GATT		
6	1	C	AT\\$GATACAT\\$GATTAGATA\#GATTAC		
14	6	C	AT\\$GATTAGATA\#GATTACAT\\$GATAC		
23	2	G	ATA\#GATTACAT\\$GATACAT\\$GATTAG		
10	3	G	ATACAT\\$GATTAGATA\#GATTACAT\\$G		
1	2	G	ATTACAT\\$GATACAT\\$GATTAGATA\#G		
18	4	G	ATTAGATA\#GATTACAT\\$GATACAT\\$G		
5	0	A	CAT\\$GATACAT\\$GATTAGATA\#GATT		
13	7	A	CAT\\$GATTAGATA\#GATTACAT\\$GATA		
22	0	A	GATA\#GATTACAT\\$GATACAT\\$GATT		
9	4	\\$	GATACAT\\$GATTAGATA\#GATTACAT\\$		
0	3	\#	GATTACAT\\$GATACAT\\$GATTAGATA\#		
17	5	\\$	GATTAGATA\#GATTACAT\\$GATACAT\\$		
7	0	A	T\\$GATACAT\\$GATTAGATA\#GATTAC		
15	5	A	T\\$GATTAGATA\#GATTACAT\\$GATAC		
24	1	A	\textcolor{orange}{TA}\#GATTACAT\\$GATACAT\\$GATTAGA		
3	2	T	\textcolor{orange}{TACAT\\$GATACAT\\$GATTAGATA\#GAT}		
11	9	A	\textcolor{orange}{TACAT\\$GATTAGATA\#GATTACAT\\$GA}		
20	2	T	\textcolor{orange}{TAGATA\#GATTACAT\\$GATACAT\\$GAT}		
2	1	A	TTACAT\\$GATACAT\\$GATTAGATA\#CA		
19	3	A	TTAGATA\#GATTACAT\\$GATACAT\\$GA		

Matching statistics using the LCP Array

$S: GATTACAT\$GATACAT\$GATTAGATA\#$
 $BWT(S): ATTTCCTGGGAAA\$#\$AAATATAA$

$P: \textcolor{orange}{TA}TACAGAT$

$M: \textcolor{orange}{2} 5 4 3 2 4 3 2 1$

Problem: LCP requires linear time to build!

SA	LCP	BWT	F	\mathcal{M}	L
26	0	A	#GATTACAT\\$GATACAT\\$GATTAGATA		
8	0	T	\\$GATACAT\\$GATTAGATA\#GATTACAT		
16	4	T	\\$GATTAGATA\#GATTACAT\\$GATACAT		
25	0	T	A\#GATTACAT\\$GATACAT\\$GATTAGATA		
4	1	T	ACAT\\$GATACAT\\$GATTAGATA\#GATT		
12	8	T	ACAT\\$GATAGATA\#ATTACAT\\$GAT		
21	1	T	AGATA\#GATTACAT\\$GATACAT\\$GATT		
6	1	C	AT\\$GATACAT\\$GATTAGATA\#GATTAC		
14	6	C	AT\\$GATAGATA\#GATTACAT\\$GATAC		
G		G	ATA\#GATTACAT\\$GATACAT\\$GATTAG		
G		G	ATACAT\\$GATTAGATA\#GATTACAT\\$G		
G		G	ATTACAT\\$GATACAT\\$GATTAGATA\#G		
G		G	ATTAGATA\#GATTACAT\\$GATACAT\\$G		
A		A	CAT\\$GATACAT\\$GATTAGATA\#GATT		
A		A	CAT\\$GATAGATA\#GATTACAT\\$GATA		
A		A	GATA\#GATTACAT\\$GATACAT\\$GATT		
\\$		\\$	GATACAT\\$GATTAGATA\#GATTACAT\\$		
#		#	GATTACAT\\$GATACAT\\$GATTAGATA\#		
\\$		\\$	GATAGATA\#GATTACAT\\$GATACAT\\$		
A		A	T\\$GATACAT\\$GATTAGATA\#GATTACA		
A		A	T\\$GATAGATA\#GATTACAT\\$GATAC		
A		A	TA\#GATTACAT\\$GATACAT\\$GATTAGA		
T		T	TA\#CAT\\$GATACAT\\$GATTAGATA\#GAT		
A		A	TA\#CAT\\$GATAGATA\#GATTACAT\\$GA		
T		T	TA\#GATA\#GATTACAT\\$GATACAT\\$GAT		
A		A	TTACAT\\$GATACAT\\$GATTAGATA\#CA		
20	2				
2	1				
19	3				

Thresholds

$P: \text{T A T T A T A C A C G}$

SA	BWT	LCP	F	\mathcal{M}
:	:	:	:	
67	G	7		ACACAG#GACACAG#...
27	T	11		ACACAG#GACAGAT#...
52	T	12		ACACAG#GACAGT#G...
91	T	11		ACACAG#GACAT#GA...
83	C	4		ACACGG#TACACAG#...
77	C	3		ACAG#CACACGG#TA...
69	C	5		ACAG#GACACAG#CA...
29	C	9		ACAG#GACAGAT#AG...
54	C	10		ACAG#GACAGT#GAC...
93	C	9		ACAG#GACAT#GACA...
35	G	4		ACAGAT#AGACAGAT...
44	G	7		ACAGAT#TACACAG#...
111	G	6		ACAGATA!GATTACA...
60	G	4		ACAGT#GACACAG#G...
105	G	3		ACAT#GACAGATA!G...
99	G	9		ACAT#GACAT#GACA...
4	T	7		ACAT#GATACAT#GA...
12	T	8		ACAT#GATTAGAT#T...
85	C	2		ACGG#TACACAG#GA...
:	:	:	:	

Thresholds

Find the longest prefix of **A C A C G**
that is preceded by **T** in S .

It follows either the previous **T** or
the next **T** in the BWT of S .

For each pair of consecutive runs of the
same character, we can store a **threshold**
 i such that:

- If the match is *above* i , we “jump up”.
- Otherwise, we “jump down”.

SA	BWT	LCP	F	\mathcal{M}
:	:	:	:	
67	G	7		ACACAG#GACACAG#...
27	T	11		ACACAG#GACAGAT#...
52	T	12		ACACAG#GACAGT#G...
91	T	11		ACACAG#GACAT#GA...
83	C	4		ACACGG#TACACAG#...
77	C	3		ACAG#CACACGG#TA...
69	C	5		ACAG#GACACAG#CA...
29	C	9		ACAG#GACAGAT#AG...
54	C	10		ACAG#GACAGT#GAC...
93	C	9		ACAG#GACAT#GACA...
35	G	4		ACAGAT#AGACAGAT...
44	G	7		ACAGAT#TACACAG#...
111	G	6		ACAGATA!GATTACA...
60	G	4		ACAGT#GACACAG#G...
105	G	3		ACAT#GACAGATA!G...
99	G	9		ACAT#GACAT#GACA...
4	T	7		ACAT#GATACAT#GA...
12	T	8		ACAT#GATAGAT#T...
85	C	2		ACGG#TACACAG#GA...
:	:	:	:	

Thresholds

Rossi et al. [RECOMB 2021]

Between two consecutive runs $BWT[i'..i]$ and $BWT[j..j']$ of the same character c , we store the threshold position k such that:

1. For all suffixes $i \leq x < k$, $lcp(SA[x], SA[i]) \geq lcp(SA[x], SA[j])$
2. For all suffixes $k \leq x \leq j$, $lcp(SA[x], SA[i]) \leq lcp(SA[x], SA[j])$

A threshold position k between two consecutive runs $BWT[i'..i]$ and $BWT[j..j']$ of the same character c , is the position of the minimum value of $LCP[i + 1..j]$.

SA	BWT	LCP	F	\mathcal{M}
:	:	:	:	
67	G	7		ACACAG#GACACAG#...
27	T	11		ACACAG#GACAGAT#...
52	T	12		ACACAG#GACAGT#G...
91	T	11		ACACAG#GACAT#GA...
83	C	4		ACACG#TACACAG#...
77	C	3		ACAG#CACACGG#TA...
69	C	5		ACAG#GACACAG#CA...
29	C	9		ACAG#GACAGAT#AG...
54	C	10		ACAG#GACAGT#GAC...
93	C	9		ACAG#GACAT#GACA...
35	G	4		ACAGAT#AGACAGAT...
44	G	7		ACAGAT#TACACAG#...
111	G	6		ACAGATA!GATTACA...
60	G	4		ACAGT#GACACAG#G...
105	G	3		ACAT#GACAGATA!G...
99	G	9		ACAT#GACAT#GACA...
4	T	7		ACAT#GATACAT#GA...
12	T	8		ACAT#GATAGAT#T...
85	C	2		ACGG#TACACAG#GA...
:	:	:	:	

Thresholds

Rossi et al. [RECOMB 2021]

Between two consecutive runs $BWT[i'..i]$ and $BWT[j..j']$ of the same character c , we store the threshold position k such that:

1. For all suffixes $i \leq x < k$, $lcp(SA[x], SA[i]) \geq lcp(SA[x], SA[j])$
2. For all suffixes $k \leq x \leq j$, $lcp(SA[x], SA[i]) \leq lcp(SA[x], SA[j])$

A threshold position k between two consecutive runs $BWT[i..i']$ and $BWT[j..j']$ of the same character c is the minimum position x such that

Build Thresholds at the same time as BWT and SA Samples.

SA	BWT	LCP	F	\mathcal{M}
:	:	:	:	
67	G	7		ACACAG#GACACAG#...
27	T	11		ACACAG#GACAGAT#...
52	T	12		ACACAG#GACAGT#G...
91	T	11		ACACAG#GACAT#GA...
83	C	4		ACACG#TACACAG#...
77	C	3		ACAG#CACACGG#TA...
69	C	5		ACAG#GACACAG#CA...
29	C	9		ACAG#GACAGAT#AG...
54	C	10		ACAG#GACAGT#GAC...
93	C	9		ACAG#GACAT#GACA...
35	G	4		ACAGAT#AGACAGAT...
44	G	7		ACAGAT#TACACAG#...
111	C	6		ACAGATA!GATTACA...
				CAGT#GACACAG#G...
				CAT#GACAGATA!G...
				CAT#GACAT#GACA...
				CAT#GATACAT#GA...
				CAT#GATTAGAT#T...
				CGG#TACACAG#GA...
:	:	:		

Moni: Pangenomics Indexing

The Finnish term “moni” matches the English term “a lot”

- Build an auxiliary data structure (called thresholds) that takes $O(r)$ space that lets us compute MEMs.
- We modify prefix free parsing so it is constructed at the same time as the r-index.

Experimental results – MEMs

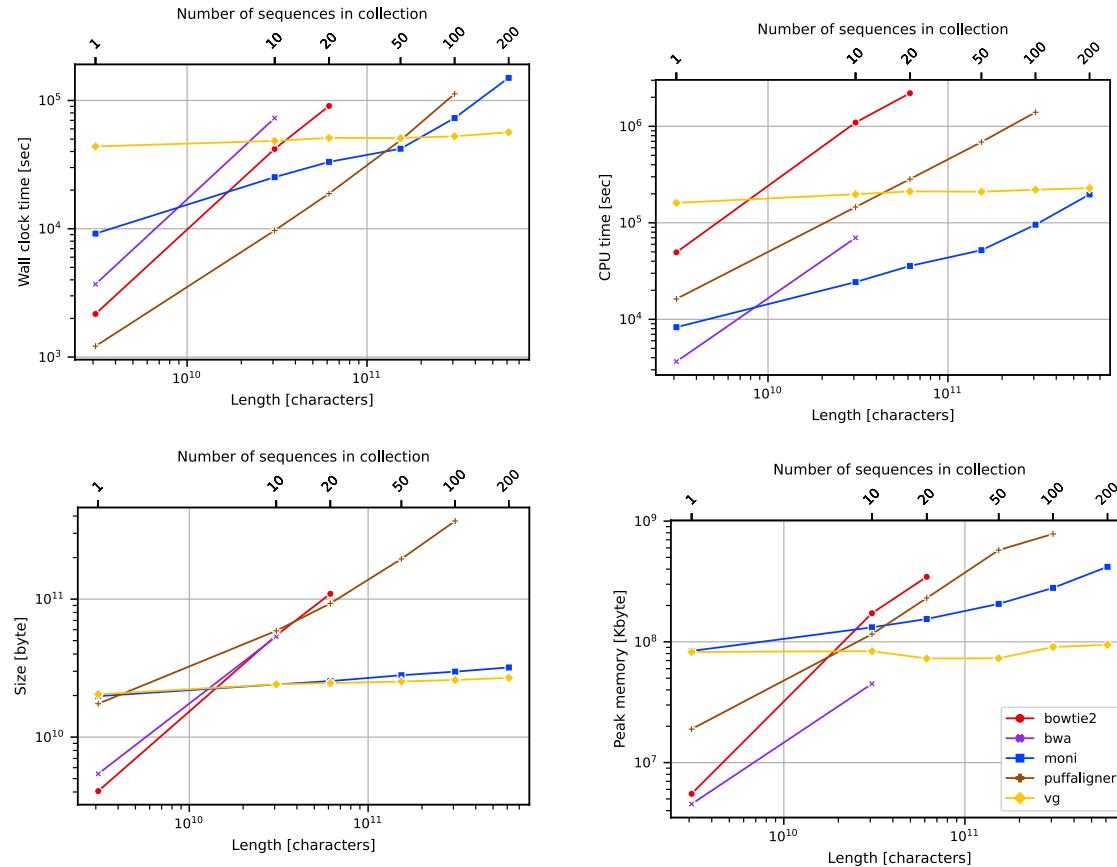
Thresholds for a collection of Human genomes.

- ref: GRCh37 human reference genome
- ref.10: GRCh37 human reference genome + 9 variants
- ref.20: GRCh37 human reference genome + 19 variants
- ref.50: GRCh37 human reference genome + 49 variants
- ref.100: GRCh37 human reference genome + 99 variants
- ref.200: GRCh37 human reference genome + 199 variants

611 400 000 reads from “The Simons genome diversity project”

Minimum length	ref	ref.10	ref.20	ref.50	ref.100	ref.200
25	411,665,608	412,292,276(+0.15%)	412,383,562(+0.17%)	412,580,107(+0.22%)	412,678,277(+0.25%)	412,818,387(+0.28%)
50	309,876,128	311,825,333(+0.63%)	311,986,530(+0.68%)	312,172,012(+0.74%)	312,298,469(+0.78%)	312,460,499(+0.83%)
75	253,953,551	264,406,220(+4.12%)	264,941,235(+4.33%)	265,311,230(+4.47%)	265,510,475(+4.55%)	265,770,839(+4.65%)

Experimental results – MEMs



Recap

- Building the BWT and SA samples of Gagie et al. using PFP
[Kuhnle et al., RECOMB 2018]

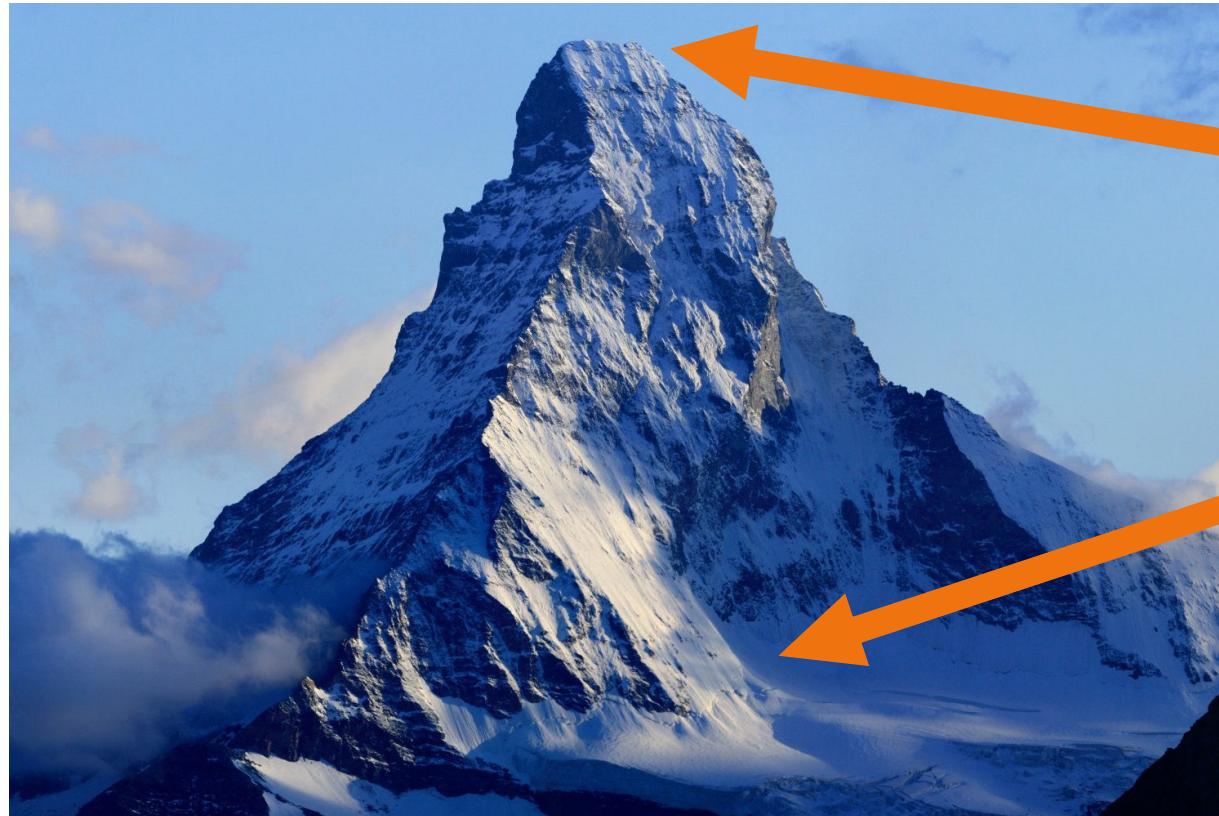


- Computing MEMs from matching statistics.
- Computing matching Statistics from Thresholds

[Bannai et al. 2020]

- Defining Thresholds as min LCP
- Computing Thresholds via PFP
[Rossi et al., in submission]

Conclusions and Future Work



Pangenomics
Alignment

We are here!

Thank you

Massimiliano Rossi



Marco Oliva



Travis Gagie



Ben Langmead



Funded by:

NSF IIS (Grant No. 1618814)

NIH BDMA (grants 308030, 314170, and 323233).

UF | UNIVERSITY *of* FLORIDA