Compressed Full-Text Indexes for Highly Repetitive Collections

Lectio praecursoria
Jouni Sirén 29.6.2012

ALGORITHM

Grossi, Vitter: Compressed suffix arrays and suffix trees with

Navarro, Mäkinen: Compressed full-text indexes

Ferragina, Manzini: Indexing compressed text

Raman, Raman, Rao: Succinct indexable dictionaries with applications

Sadakane: New text indexing functionalities of the compressed suffix

Burrows, Wheeler: A block sorting lossless data compression algorithm

Sadakane: Compressed suffix trees with full functionality

Manber, Myers: Suffix arrays: A new method for on-line string searches

Ferragina, Manzini, Mäkinen, Navarro: Compressed representations of

Are there papers with *Sadakane* as the first author?

Grossi, Vitter: Compressed suffix arrays and suffix trees with

Navarro, Mäkinen: Compressed full-text indexes

Ferragina, Manzini: Indexing compressed text

Raman, Raman, Rao: Succinct indexable dictionaries with applications

Sadakane: New text indexing functionalities of the compressed suffix

Burrows, Wheeler: A block sorting lossless data compression algorithm

Sadakane: Compressed suffix trees with full functionality

Manber, Myers: Suffix arrays: A new method for on-line string searches

Ferragina, Manzini, Mäkinen, Navarro: Compressed representations of

How many papers have *Sadakane* as the first author?

Grossi, Vitter: Compressed suffix arrays and suffix trees with

Navarro, Mäkinen: Compressed full-text indexes

Ferragina, Manzini: Indexing compressed text

Raman, Raman, Rao: Succinct indexable dictionaries with applications

Sadakane: New text indexing functionalities of the compressed suffix

Burrows, Wheeler: A block sorting lossless data compression algorithm

Sadakane: Compressed suffix trees with full functionality

Manber, Myers: Suffix arrays: A new method for on-line string searches

Ferragina, Manzini, Mäkinen, Navarro: Compressed representations of

What are the papers with *Sadakane* as the first author?

Grossi, Vitter: Compressed suffix arrays and suffix trees with

Navarro, Mäkinen: Compressed full-text indexes

Ferragina, Manzini: Indexing compressed text

Raman, Raman, Rao: Succinct indexable dictionaries with applications

Sadakane: New text indexing functionalities of the compressed suffix

Burrows, Wheeler: A block sorting lossless data compression algorithm

Sadakane: Compressed suffix trees with full functionality

Manber, Myers: Suffix arrays: A new method for on-line string searches

Ferragina, Manzini, Mäkinen, Navarro: Compressed representations of

Burrows, Wheeler: A block sorting lossless data compression algorithm

Ferragina, Manzini: Indexing compressed text

Ferragina, Manzini, Mäkinen, Navarro: Compressed representations of

Grossi, Vitter: Compressed suffix arrays and suffix trees with

Manber, Myers: Suffix arrays: A new method for on-line string searches

Navarro, Mäkinen: Compressed full-text indexes

Raman, Raman, Rao: Succinct indexable dictionaries with applications

Sadakane: Compressed suffix trees with full functionality

Sadakane: New text indexing functionalities of the compressed suffix

Burrows, Wheeler: A block sorting lossless data compression algorithm

Ferragina, Manzini: Indexing compressed text

Ferragina, Manzini, Mäkinen, Navarro: Compressed representations of

Grossi, Vitter: Compressed suffix arrays and suffix trees with

Manber, Myers: Suffix arrays: A new method for on-line string searches

Navarro, Mäkinen: Compressed full-text indexes

Raman, Raman, Rao: Succinct indexable dictionaries with applications

Sadakane: Compressed suffix trees with full functionality

Sadakane: New text indexing functionalities of the compressed suffix

Burrows, Wheeler: A block sorting lossless data compression algorithm

Ferragina, Manzini: Indexing compressed text

Ferragina, Manzini, Mäkinen, Navarro: Compressed representations of

Grossi, Vitter: Compressed suffix arrays and suffix trees with

Manber, Myers: Suffix arrays: A new method for on-line string searches

Navarro, Mäkinen: Compressed full-text indexes

Raman, Raman, Rao: Succinct indexable dictionaries with applications

Sadakane: Compressed suffix trees with full functionality

Sadakane: New text indexing functionalities of the compressed suffix

Burrows, Wheeler: A block sorting lossless data compression algorithm

Ferragina, Manzini: Indexing compressed text

Ferragina, Manzini, Mäkinen, Navarro: Compressed representations of

Grossi, Vitter: Compressed suffix arrays and suffix trees with

Manber, Myers: Suffix arrays: A new method for on-line string searches

Navarro, Mäkinen: Compressed full-text indexes

Raman, Raman, Rao: Succinct indexable dictionaries with applications

Sadakane: Compressed suffix trees with full functionality

Sadakane: New text indexing functionalities of the compressed suffix

Burrows, Wheeler: A block sorting lossless data compression algorithm

Ferragina, Manzini: Indexing compressed text

Ferragina, Manzini, Mäkinen, Navarro: Compressed representations of

Grossi, Vitter: Compressed suffix arrays and suffix trees with

Manber, Myers: Suffix arrays: A new method for on-line string searches

Navarro, Mäkinen: Compressed full-text indexes

Raman, Raman, Rao: Succinct indexable dictionaries with applications

Sadakane: Compressed suffix trees with full functionality

Sadakane: New text indexing functionalities of the compressed suffix

Burrows, Wheeler: A block sorting lossless data compression algorithm

Ferragina, Manzini: Indexing compressed text

Ferragina, Manzini, Mäkinen, Navarro: Compressed representations of

Grossi, Vitter: Compressed suffix arrays and suffix trees with

Manber, Myers: Suffix arrays: A new method for on-line string searches

Navarro, Mäkinen: Compressed full-text indexes

Raman, Raman, Rao: Succinct indexable dictionaries with applications

Sadakane: Compressed suffix trees with full functionality

Sadakane: New text indexing functionalities of the compressed suffix

DATA STRUCTURE

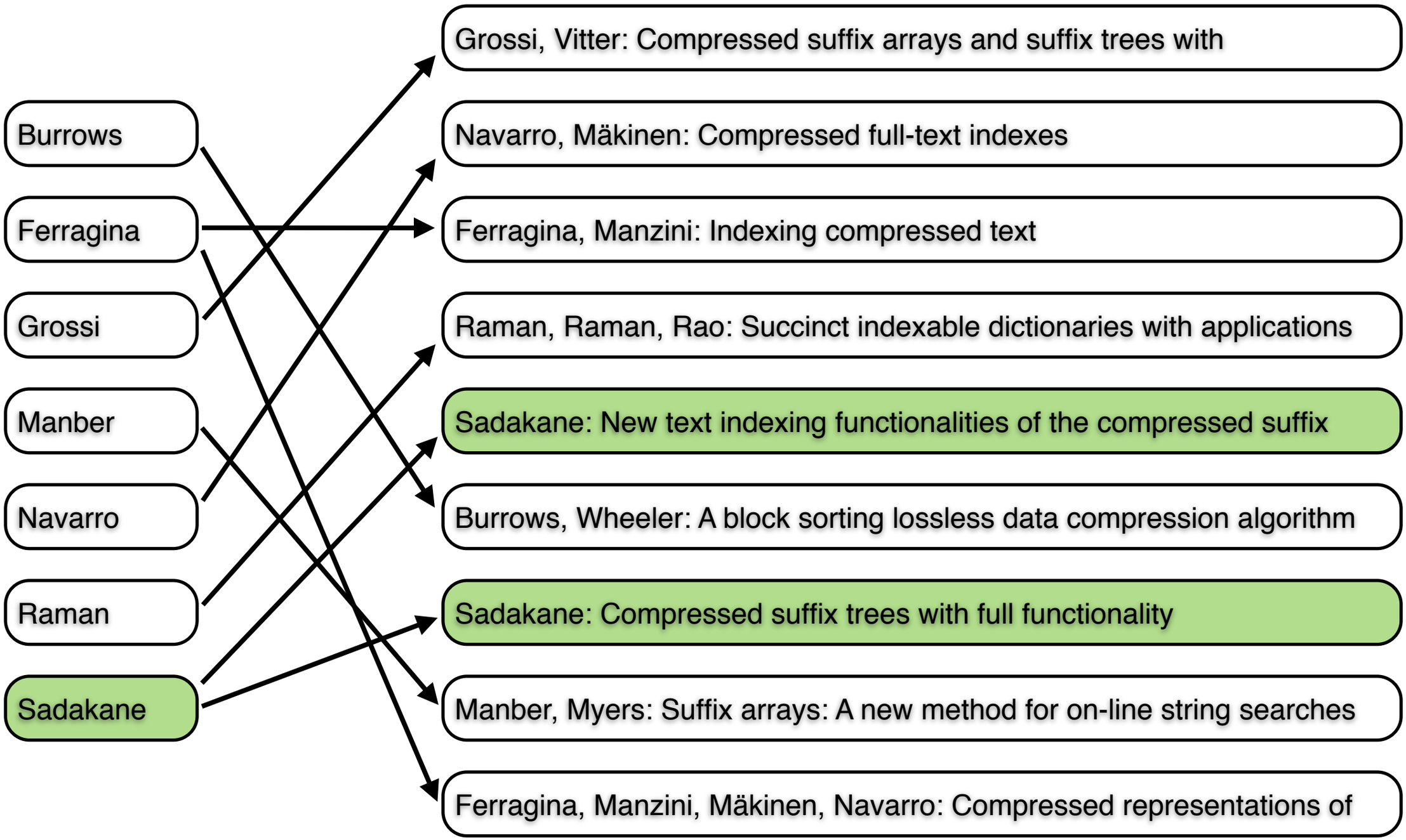
- What if we have to preserve the original order of the records?

- We may want even faster queries.

- Perhaps there are too many records to fit into memory.

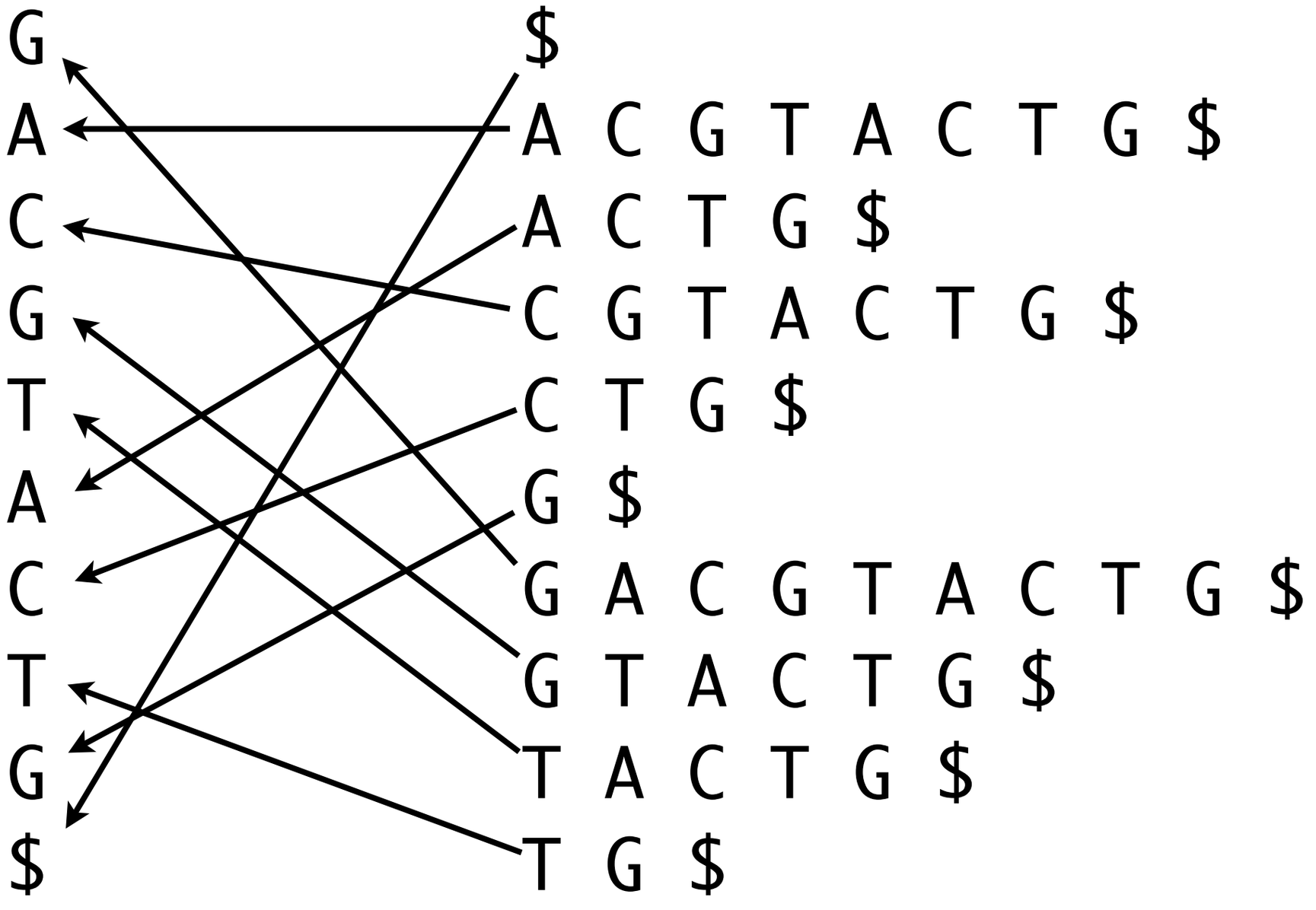
- Then we probably need another data structure.

INDEX


Grossi, Vitter: Compressed suffix arrays and suffix trees with
Navarro, Mäkinen: Compressed full-text indexes
Ferragina, Manzini: Indexing compressed text
Raman, Raman, Rao: Succinct indexable dictionaries with applications
Sadakane: New text indexing functionalities of the compressed suffix
Burrows, Wheeler: A block sorting lossless data compression algorithm
Sadakane: Compressed suffix trees with full functionality
Manber, Myers: Suffix arrays: A new method for on-line string searches
Ferragina, Manzini, Mäkinen, Navarro: Compressed representations of
Burrows
Ferragina
Grossi
Manber
Navarro
Raman
Sadakane

FULL-TEXT INDEX

Suffix Array



Suffix Array

G	10
A	2
C	6
G	3
T	7
A	9
C	1
T	4
G	5
$	8

- While a character takes 1 byte, each pointer requires 4 or 8 bytes.

- Suffix array usually requires 5 or 9 times more space than the text.

- We need something smaller to handle large texts.

COMPRESSED INDEX

- Ferragina, Manzini 2000, 2005: FM-index
- Grossi, Vitter 2000, 2005: Compressed Suffix Array
- Use Burrows-Wheeler transform to simulate the suffix array.
- Compresses to 40% to 80% of text size.
- Yet some data should compress better.

HIGHLY REPETITIVE DATA

Individual Genomes

```
G A C G T A - C T G C A G A T G - T A A T G C
G A C G T A - C T G C A G A T G C T A A T C C
G A C G T A - - - G C A G A T G C T A A T G C
G A C G T A - C T G C A G - T G C T A A T G C
G A C G T A - - - G C A G A T G C T A A T C C
G A C G T A - C T G C T G A T G C T A A T G C
G A C G T A C C T G C A G A T G C T A A T G C
G A C G T A C C T G C A G - T G C T A A T G C
G A C G T A - C T G C T G A T G C T A A T G C
G A C G T A - C T G C A G A T G C T A A T C C
```

Version History

```
dhcp-eduroam-hy-138-42:thesis jltsiren$ svn diff -r 662 thesis.tex
Index: thesis.tex
===================================================================
--- thesis.tex (revision 662)
+++ thesis.tex (working copy)
@@ -23,7 +23,7 @@
 \isbnpdf{978-952-10-8052-4}
 \issn{1238-8645}
 \printhouse{Unigrafia}
-\pubpages{108 + 72} % FIXME
+\pubpages{97 + 63}
 \supervisorlist{Veli Mäkinen, University of Helsinki, Finland}
 \preexaminera{Kunihiko Sadakane, National Institute of Informatics, Japan}
 \preexaminerb{Jorma Tarhio, Aalto University, Finland}
```

Finnish language Wikipedia with full version history
42 gigabytes

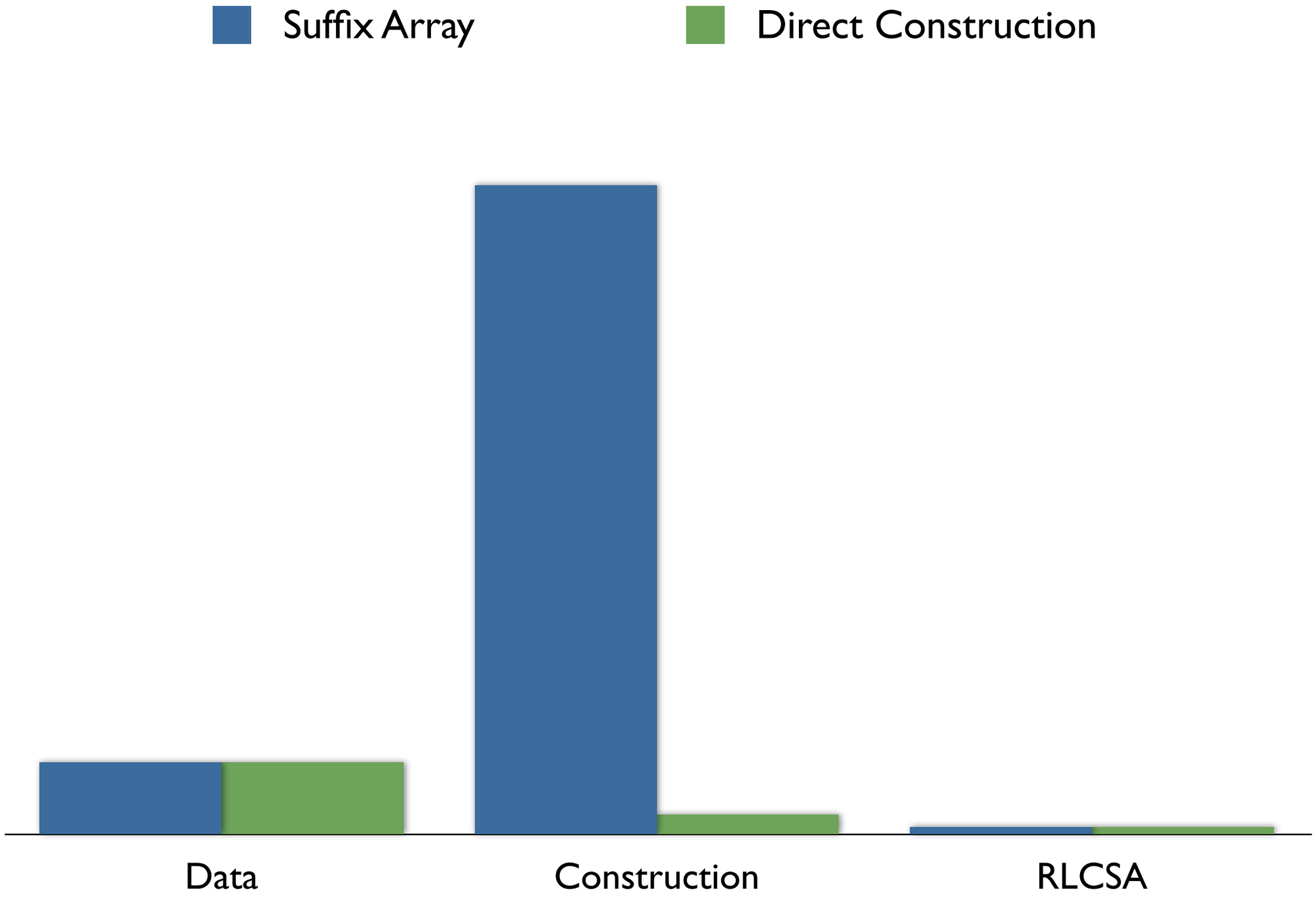
Suffix array construction
378 gigabytes

Run-length compressed suffix array
4.4 gigabytes

Do we have 378 gigabytes of memory?

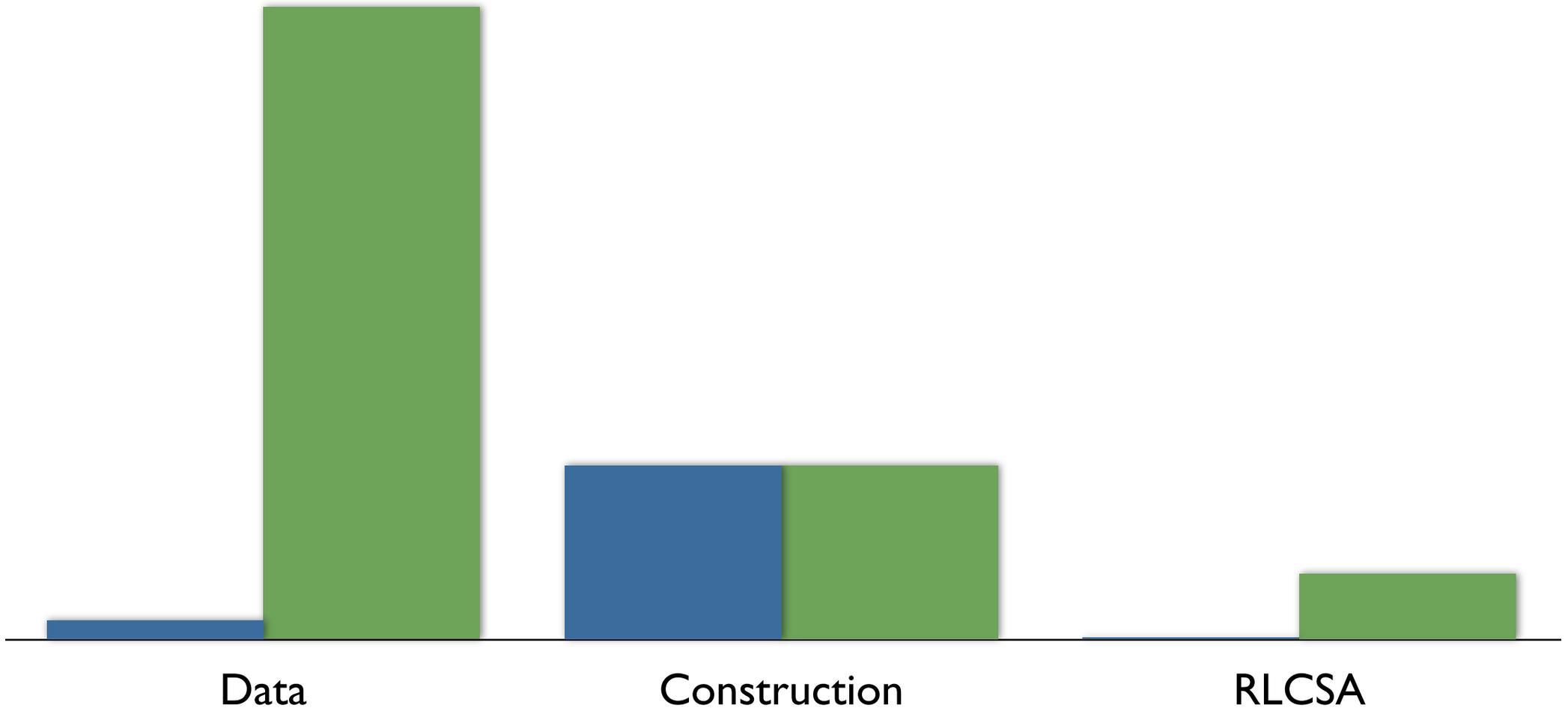
INDEX

CONSTRUCTION

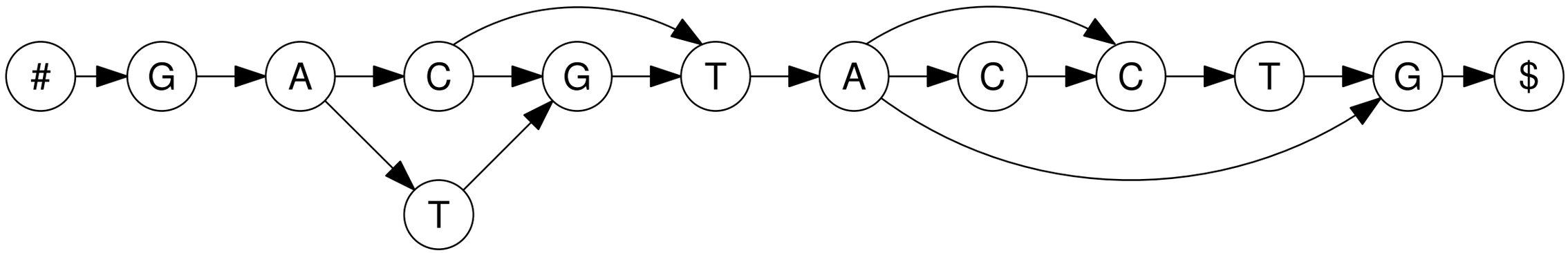

Suffix Array
Direct Construction
Data
Construction
RLCSA

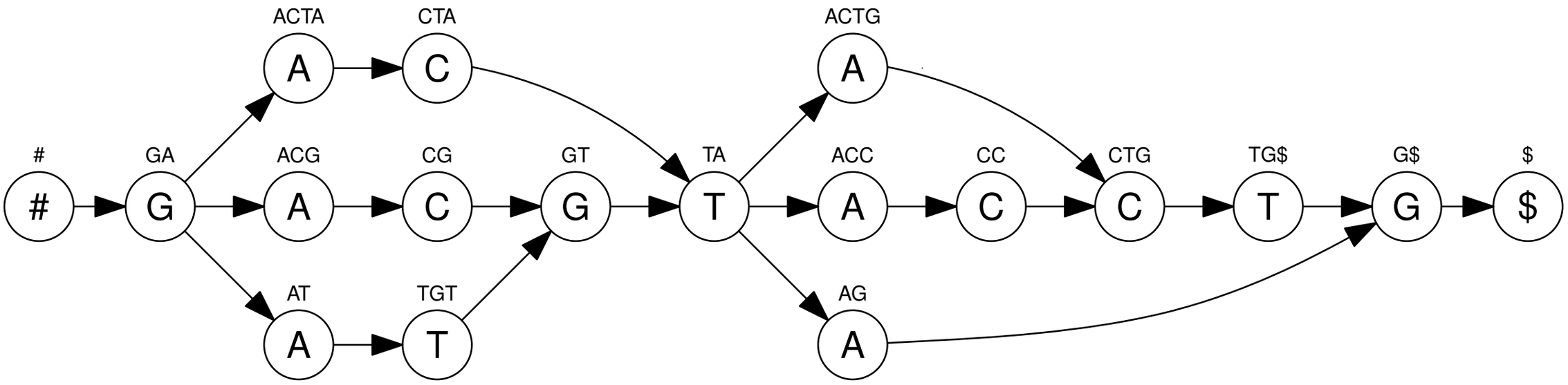
Suffix Array

Direct Construction



INDEXING AUTOMATA





	$	ACC	ACG	ACTA	ACTG	AG	AT	CC	CG	CTA	CTG	G$	GA	GT	TA	TG$	TGT	#
BWT	G	T	G	G	T	T	G	A	A	A	AC	AT	#	CT	CG	C	A	$
Edges	1	1	1	1	1	1	1	1	1	1	1	1	100	1	100	1	1	1

Compressed Full-Text Indexes for Highly Repetitive Collections