Burrows-Wheeler Transform for Graphs

Jouni Sirén, University of Chile

Jouni Sirén, Niko Välimäki, Veli Mäkinen: **Indexing Graphs for Path Queries with Applications in Genome Research**. Manuscript in review, 2013. Early version in WABI 2011.

Burrows-Wheeler transform for a class of graphs that includes DAGs and de Bruijn graphs. In principle a black-box replacement for BWT for sequences, but the practice is always more complicated.

Burrows-Wheeler transform

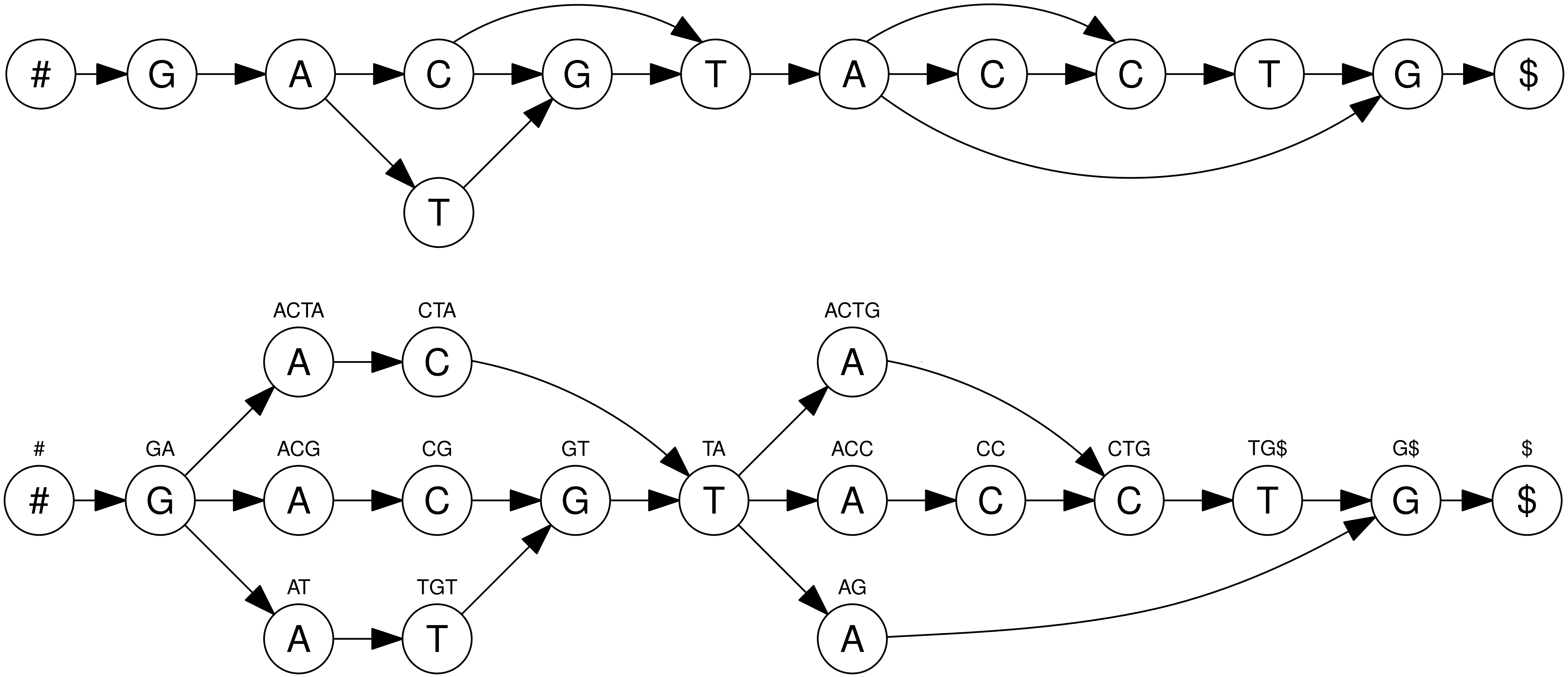
- Sort the suffixes in lexicographic order and take the previous character for each of the suffixes.

- Easy to compress, can be used to simulate the suffix tree and the suffix array.

- Key property: Suffixes starting with c are in same order as suffixes preceded by c.

BWT for DAGs

1. Build an automaton representing the reference sequence and variation.
2. Determinize the automaton.
3. Use prefix-doubling to build an equivalent automaton that can be indexed.



	$	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

We consider paths of length 1, 2, 4, 8, 16, ..., until no two paths starting from different nodes have the same label.

Each doubling step starts with a relational join:

$$(u, v, k) \oplus (v, w, k') \mapsto (u, w, (k, k'))$$

The records are then sorted by key values, and the key pairs are replaced by integer keys.

Exponential in the worst case, linear in the expected case under reasonable assumptions.

Index construction

1. Build an automaton representing the reference sequence and variation.
2. Determinize the automaton.
3. Use prefix-doubling to build an equivalent automaton that can be indexed.
4. Run out of memory.

Human chromosomes 3, 6, 8, 11, 16, 17, and 18 are hard. In doubling step 8 (path length 128 → 256), the number of paths increases e.g. from 100 million to 100 billion.

This is probably caused by variation in repetitive regions.

Various heuristics can be used to handle these chromosomes.

Index construction

Index	Time	Space	Size
GCSA	14 h	215 GB	2.8 GB
BWA	1.5 h	4.2 GB	4.2 GB
RLCSA (fast construction)	0.2 h	47 GB	2.5 GB

Human reference genome and the Finnish subset of frequent variation from dbSNP. Construction parallelized on 24 CPU cores.

Pattern matching

Index	0 errors	1 error	2 errors	3 errors
GCSA	86.47 % 80.20 %	91.94 % 84.21 %	94.04 % 85.33 %	95.54 % 86.02 %
RLCSA	82.70 % 76.67 %	91.40 % 83.67 %	93.87 % 85.12 %	95.44 % 85.86 %

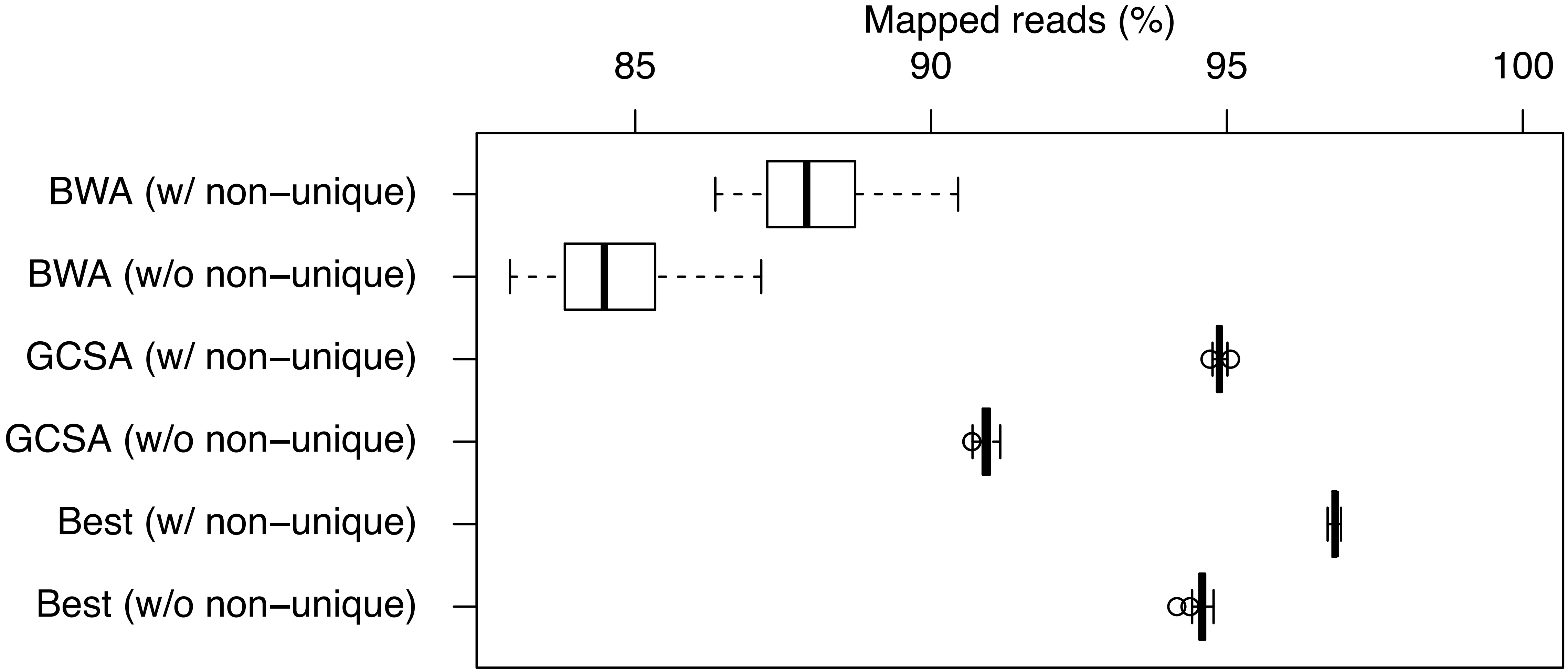
Total number of matches and unique matches with 10 million reads of length 56.

Read mapping

Index	TP	FP	TN	FN
GCSA	9,956,085	31,573	9,999,776	12,556
BWA	9,951,808	41,000	9,984,877	22,315

Variathon 2013 frequent variations: 10 million simulated read pairs and 10 million decoy pairs of length 70.

Highly polymorphic regions



Simulated reads from highly polymorphic regions in Finnish genotypes (1000 Genomes Project).

100x slower than BWA

2x	Fundamental differences
5x	Implementation choices
2x	Reverse complements
5x	Backtracking heuristics

Alexander Bowe, Taku Onodera, Kunihiko Sadakane, Tetsuo Shibuya: **Succinct de Bruijn Graphs**. WABI 2012.

Different terminology and different design choices, but the core combinatorial structure is essentially the same generalization of BWT for graphs.

Conclusions

- We can build BWT for DAGs and de Bruijn graphs.
- This is not always a black-box replacement for BWT for sequences.
- Construction is expensive, but can be improved.