INDEXING FINITE LANGUAGE REPRESENTATION OF POPULATION GENOTYPES

Jouni Sirén (with Niko Välimäki and Veli Mäkinen)
University of Helsinki

WABI 2011, Saarbrücken, Germany, September 5–7, 2011

INDEXING POPULATION GENOTYPES

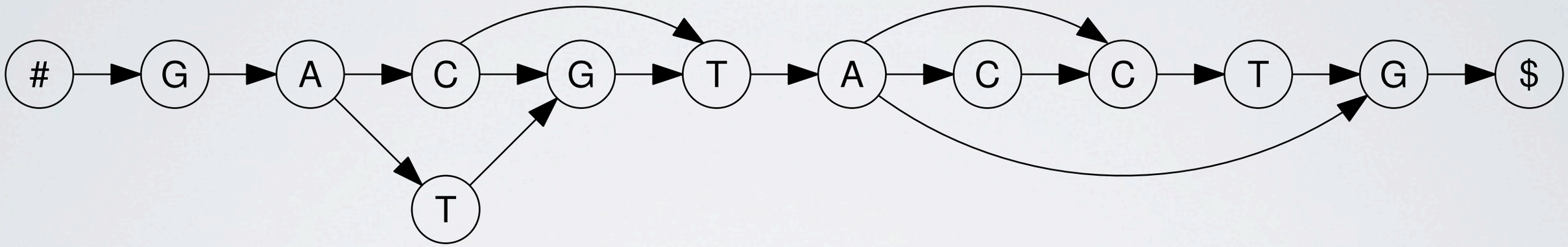
- Population genotype can be extrapolated from a sample of individual genomes.

- We have a space-efficient indexed representation for the genotype information.

- The representation is based on flexible, widely used sequence analysis techniques: BWT, suffix array, suffix tree.

- Application example: Better accuracy for read alignment / variation calling than by using a single reference sequence.

FINITE AUTOMATON



- Built from a multiple alignment of similar sequences.
- Alternatively from a reference sequence and a set of SNPs.

BURROWS-WHEELER TRANSFORM

- BWT was originally intended for data compression.
- Ferragina and Manzini used it for text indexing by simulating the suffix array.
- FM-index = Compressed Suffix Array = BWT
- Widely used in bioinformatics: Bowtie, BWA, SOAP2...
- Existing indexes support single sequences, collections of sequences, and trees.

BURROWS-WHEELER TRANSFORM

Suffixes	**BWT**
$	G
ACCTG$	T
ACGTACCTG$	G
CCTG$	A
CGTACCTG$	A
CTG$	C
G$	T
GACGTACCTG$	$
GTACCTG$	C
TACCTG$	G
TG$	C

We sort the suffixes in lexicographic order, and write down the previous character for each suffix.

BACKWARD SEARCHING

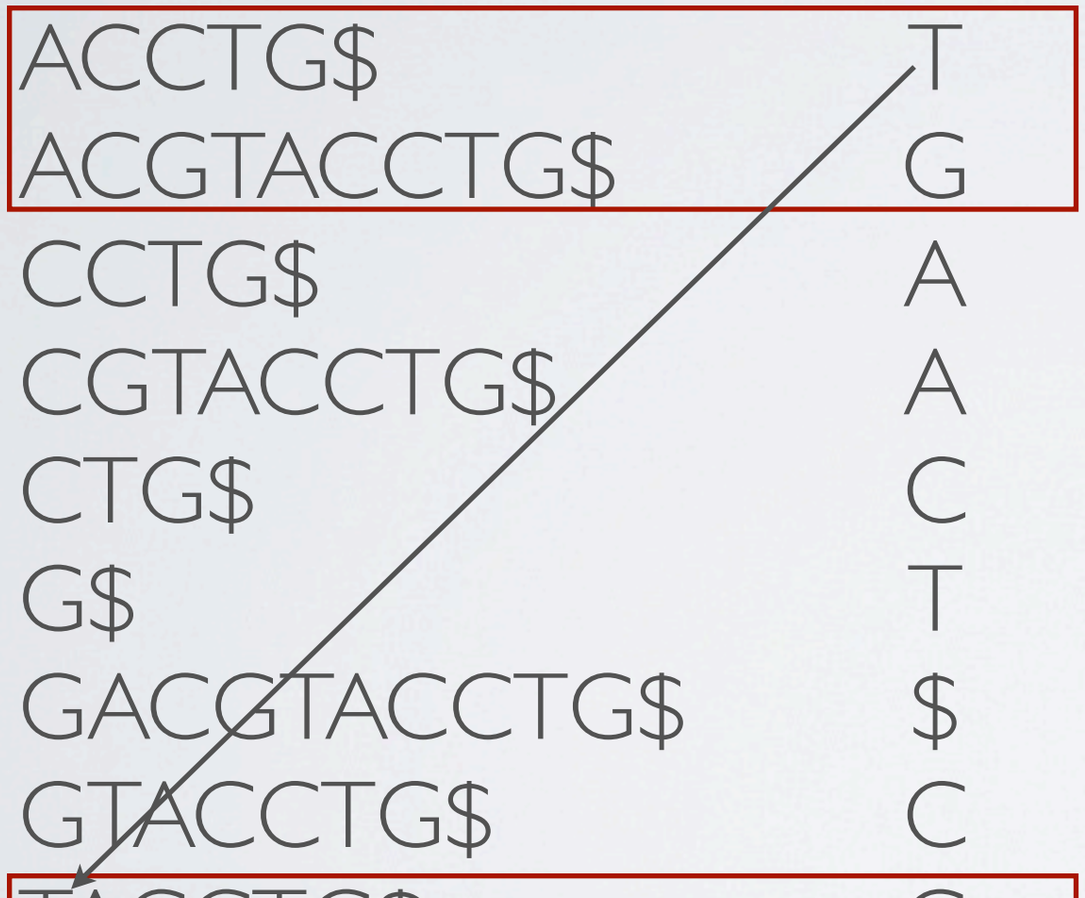
Suffixes	**BWT**
$	G
ACCTG$	T
ACGTACCTG$	G
CCTG$	A
CGTACCTG$	A
CTG$	C
G$	T
GACGTACCTG$	$
GTACCTG$	C
TACCTG$	G
TG$	C

Suffixes matching pattern AC

Suffixes starting with T

BACKWARD SEARCHING

Suffixes	BWT
$	G
ACCTG$	T
ACGTACCTG$	G
CCTG$	A
CGTACCTG$	A
CTG$	C
G$	T
GACGTACCTG$	$
GTACCTG$	C
TACCTG$	G
TG$	C



Suffixes matching pattern AC

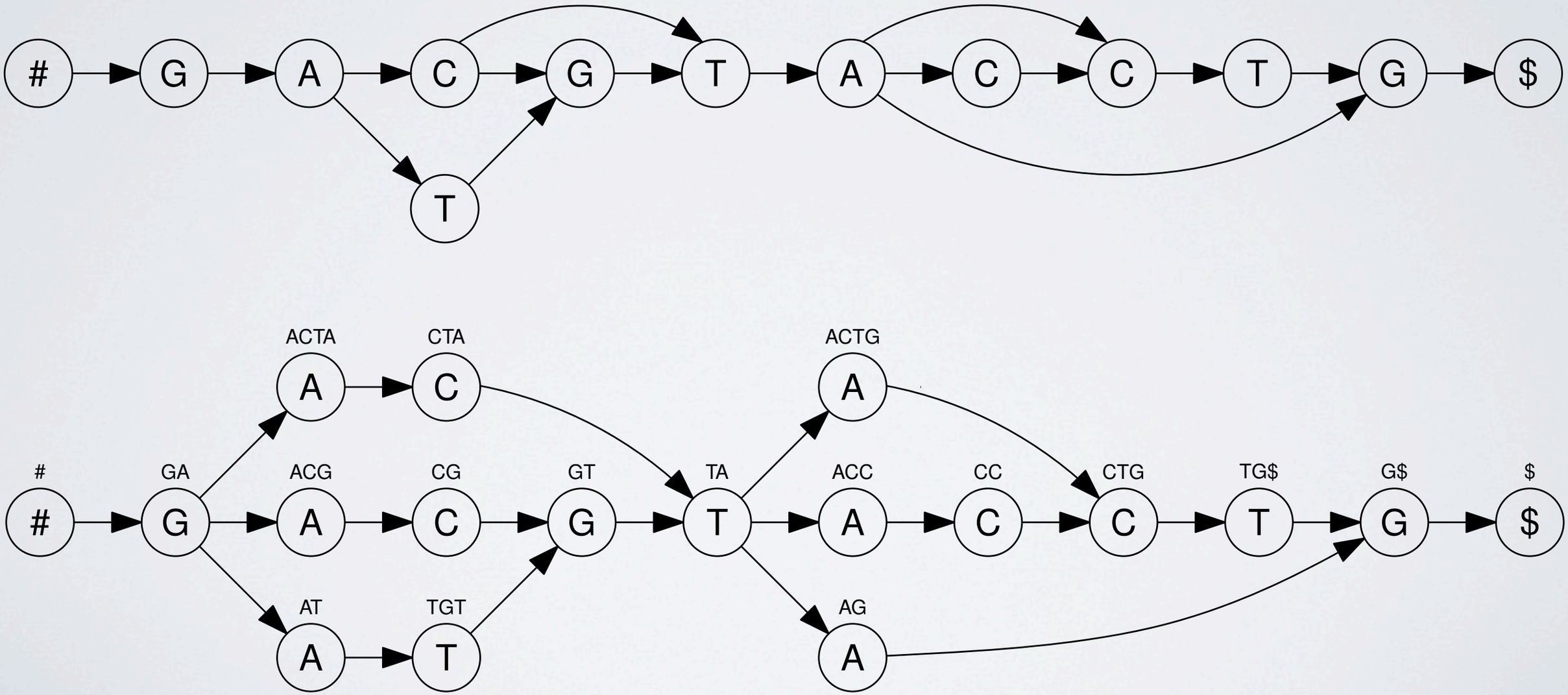
Suffixes matching pattern TAC

Nodes with label c must be in the same order as nodes having a predecessor with label c.

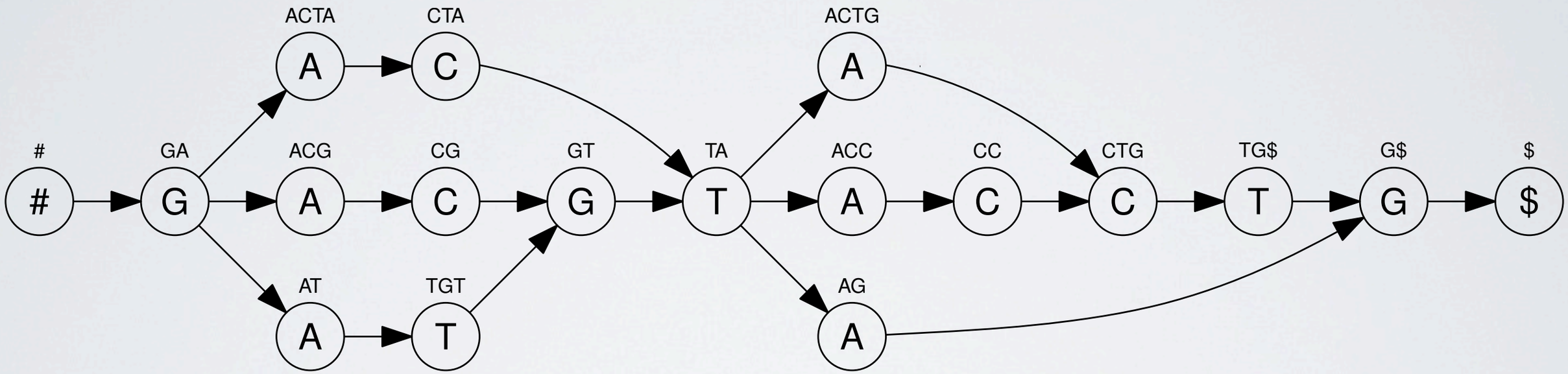
REQUIREMENTS FOR AUTOMATON

- Problem: Multiple suffixes can be recognized from a node.
- The lexicographic rank of a node should not depend on the particular suffix used as a sort key.
- Each node should correspond to a lexicographic range of suffixes, and the ranges should not overlap.
- We can build an automaton with this property for all finite languages.

PREFIX-SORTING



INDEX CONSTRUCTION



	$	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	$
Nodes	1	1	1	1	1	1	1	1	1	1	10	10	100	10	100	1	1	1
Edges	1	1	1	1	1	1	1	1	1	1	10	10	111	10	111	1	1	1

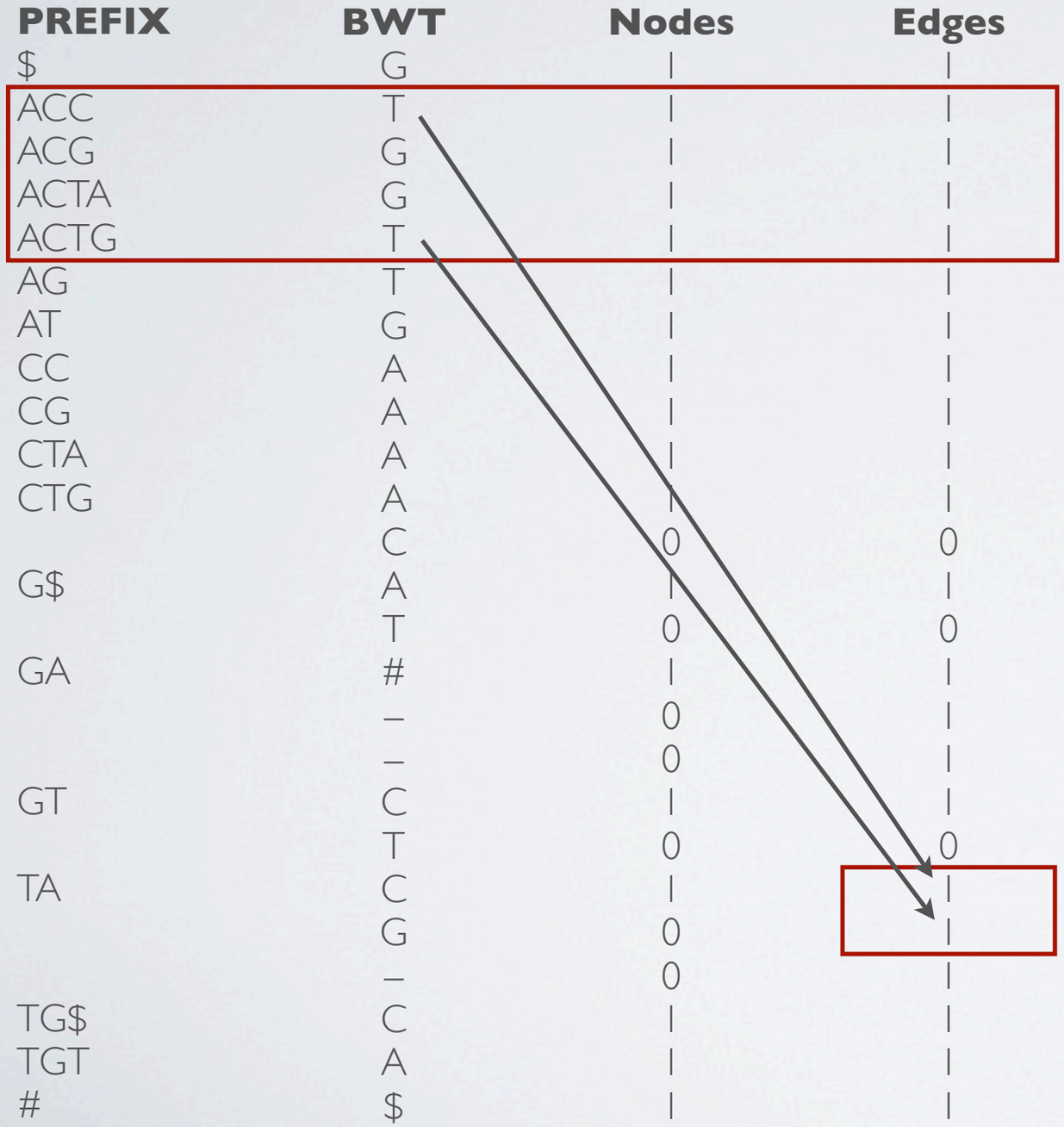
BACKWARD SEARCHING

PREFIX	BWT	Nodes	Edges
$	G	I	I
ACC	T	I	I
ACG	G	I	I
ACTA	G	I	I
ACTG	T	I	I
AG	T	I	I
AT	G	I	I
CC	A	I	I
CG	A	I	I
CTA	A	I	I
CTG	A	I	I
	C	0	0
G$	A	I	I
	T	0	0
GA	#	I	I
	–	0	I
	–	0	I
GT	C	I	I
	T	0	0
TA	C	I	I
	G	0	I
	–	0	I
TG$	C	I	I
TGT	A	I	I
#	$	I	I

Nodes matching pattern AC

Edges starting with T

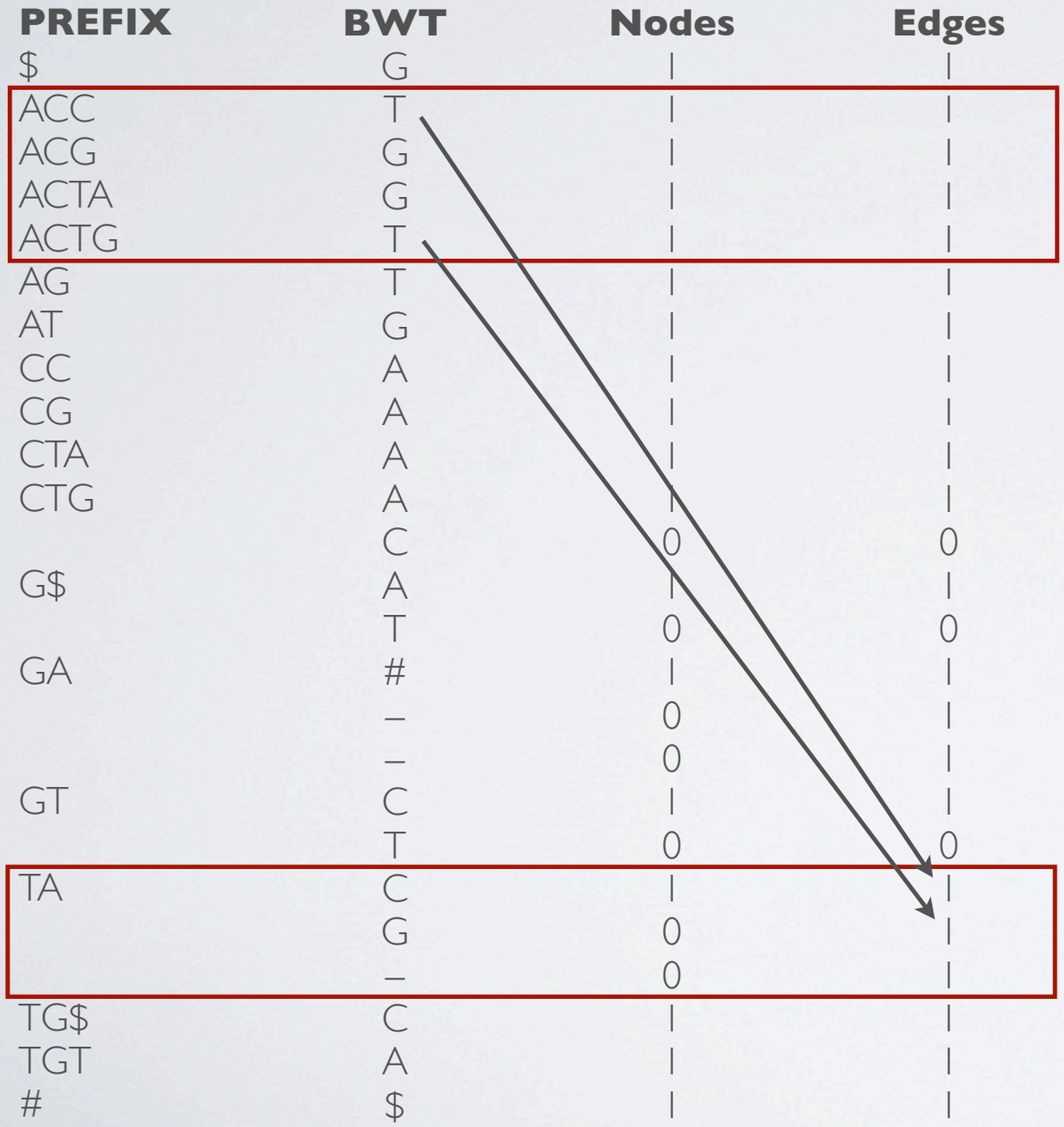
BACKWARD SEARCHING



Nodes matching pattern AC

Edges matching pattern TAC

BACKWARD SEARCHING



Nodes matching pattern AC

Nodes matching pattern TAC

EXPERIMENTS

- Multiple alignment of 4 assemblies of human chromosome 18 (~76 Mbases each).
- 10 million Illumina / Solexa reads of length 56 from the entire genome.
- A system with 2 quad-core 2.53 GHz Xeon E5540 processors and 32 GB of memory.
- Three indexes: GCSA, RLCSA, BWA.

CONSTRUCTION (WABI PAPER)

Index	Time	Space
GCSA-2	342 min	77 GB
GCSA-4	295 min	29 GB
GCSA-8	286 min	22 GB
RLCSA	11 min	2.3 GB
BWA	4 min	1.4 GB

CONSTRUCTION (FULL PAPER)

Index	Time	Space
GCSA-2	10 min	7.0 GB
GCSA-4	10 min	6.7 GB
GCSA-8	9 min	4.8 GB
RLCSA	5 min	2.3 GB
BWA	4 min	1.4 GB

EXACT MATCHING

Index	Time	Matches
GCSA-2	20 min	388.9k
GCSA-4	18 min	388.2k
GCSA-8	17 min	387.7k
RLCSA	7 min	384.4k
BWA	5 min	384.4k

APPROXIMATE MATCHING

Errors	1	2	3
GCSA time	101 min	283 min	1,730 min
GCSA matches	620k	876k	1,146k
RLCSA time	39 min	111 min	721 min
RLCSA matches	609k	856k	1,118k

CONCLUSIONS

- We extended BWT-based indexes to finite automata.
- Performance penalty is currently 2.5x to 3x (2x possible).
- Prefix-sorting increases automaton size exponentially (but less than 2x in practice).
- Time and space requirements for index construction are large but not unreasonable.
- Future work: Investigate potential applications (e.g. read mapping / variation calling).

THANK YOU!