High Throughput Sequence Analysis with MapReduce

Michael Schatz

June 18, 2009

JCVI Informatics Seminar

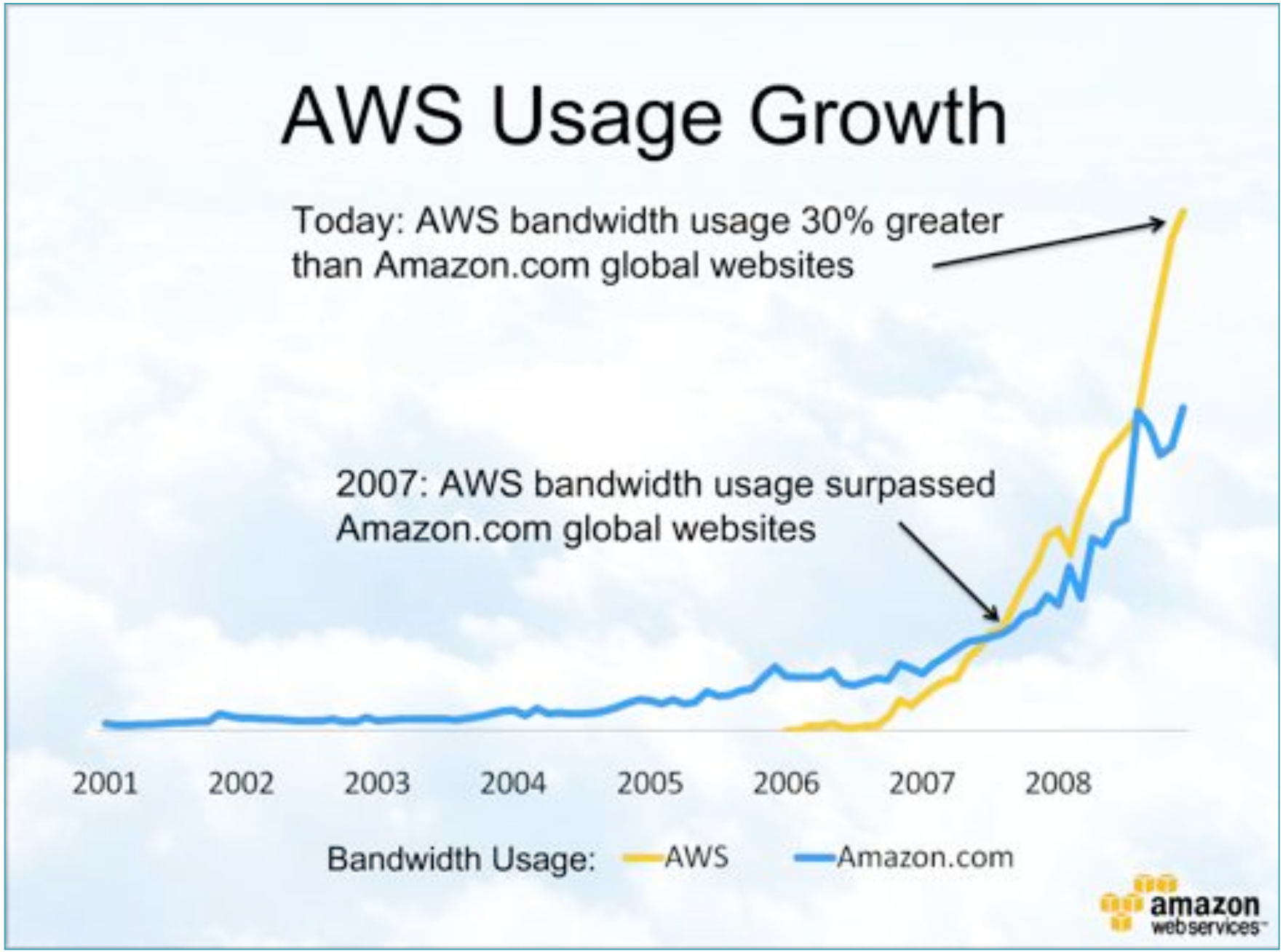
Outline

1. Parallel Computing Resources
    1. Amazon Cloud Services
    2. Hadoop & MapReduce
2. MapReduce Showcase
    1. Read Mapping: CloudBurst
    2. Mapping & SNP Discovery: CrossBow
    3. De novo Genome Assembly
3. Summary and Questions

A Brief History of the Amazon Cloud

- Urban Legend
  - Additional capacity added every fall for the holiday shopping season, underutilized rest of the year…
- Official Story
  - Amazon is a technology company
    - Different divisions of Amazon are loosely coupled
  - Amazon Web Services is the 3rd Business Division
    - Retail & Seller Businesses

amazon.com



Source: http://www.slideshare.net/tracylaxdal/aws-startup-event-seattle-2009-aws-overview

Source: http://www.slideshare.net/tracylaxdal/aws-startup-event-seattle-2009-aws-overview

Amazon Web Services

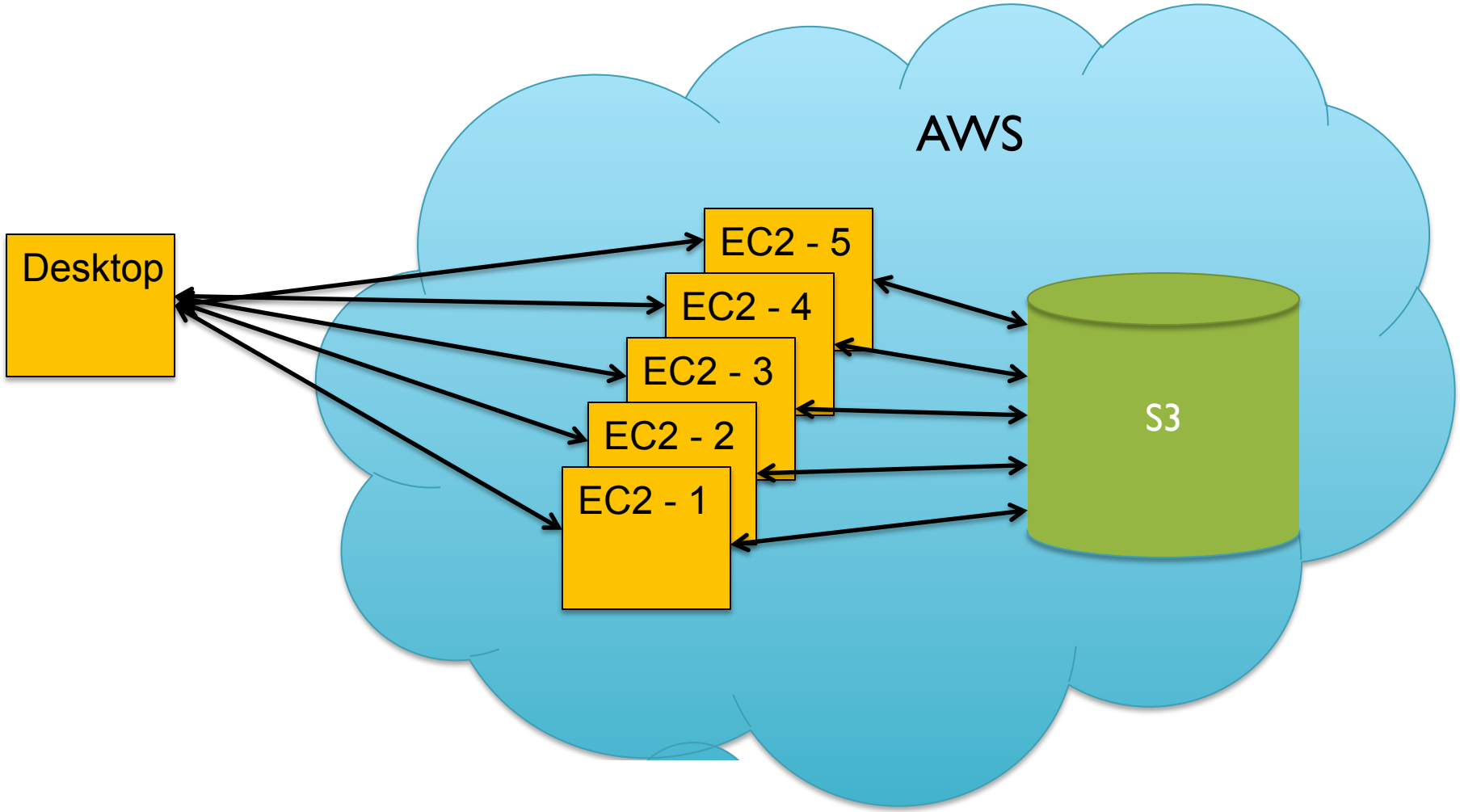
- Elastic Compute Cloud (EC2)
  - On demand computing power
    - Support for Windows, Linux, & OpenSolaris
- Simple Storage Service (S3)
  - Scalable data storage
    - 17¢ / GB transfer fee, 15¢ / GB monthly fee
- Elastic MapReduce (EMR)
  - Point-and-click Hadoop Workflows
    - Computation runs on EC2

EC2 Pricing and Models

	Small	High-CPU Medium	High-CPU Extra Large
Cores	1 @ 1.0 GHz	2 @ 2.5 GHz	8 @ 2.5 GHz
RAM	1.7 GB	1.7 GB	7 GB
Local Disk	150 GB	350 GB	1690 GB
Price	10¢ / hour	20¢ / hour	80¢ / hour

- Instances boot on demand from virtual disk image
  - Stock images available for common configurations & operating systems
  - Custom images "easy" to create
- Reserved Instances
  - Pay small upfront fee, lower hourly rate

AWS Use Model

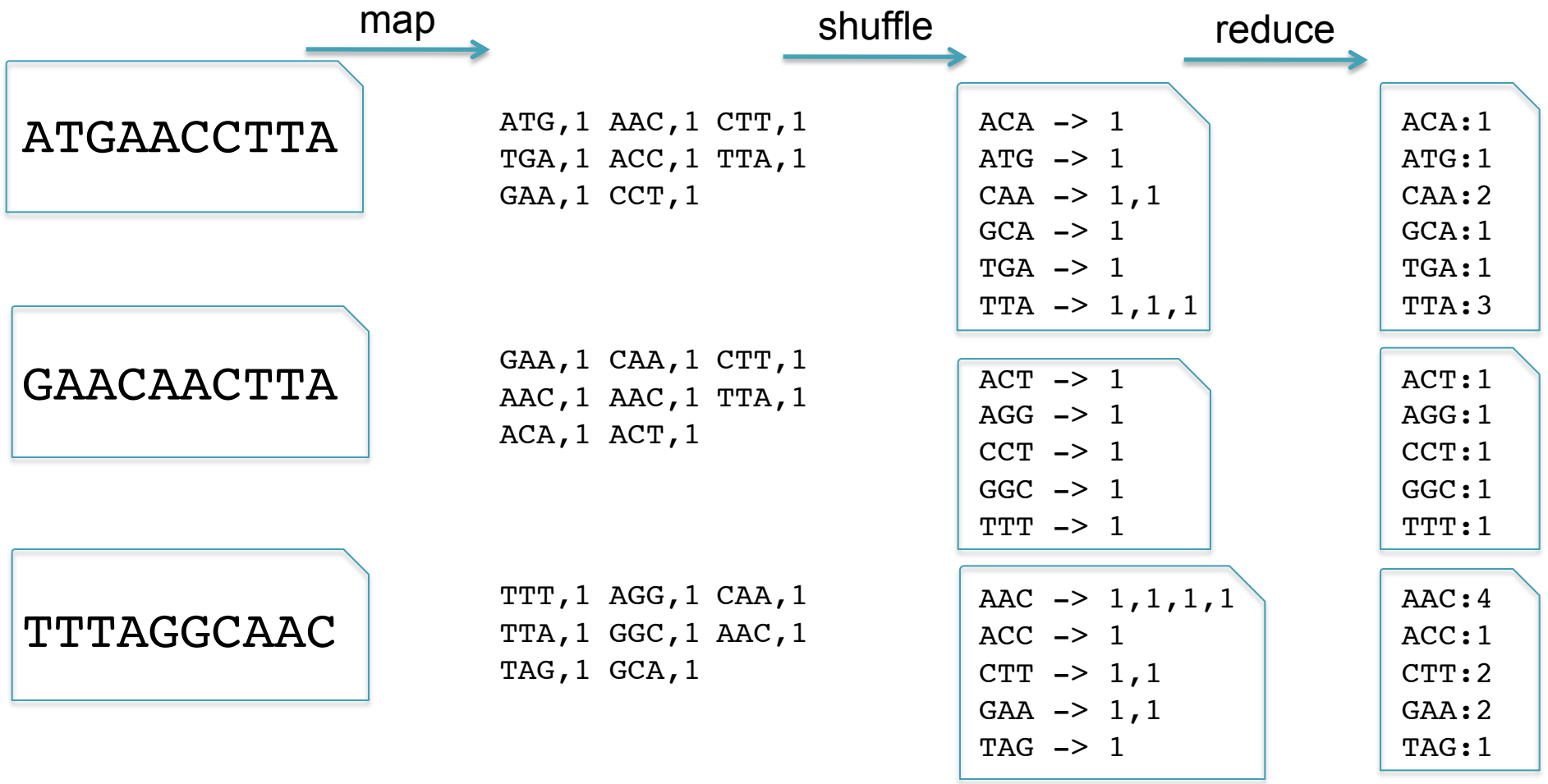


After machines spool up, ssh to instance them like any other server.
Use S3 for persistent data storage, with very fast interconnect to EC2

Hadoop MapReduce

- MapReduce is the parallel distributed framework invented by Google for large data computations.
  - Data and computations are spread over thousands of computers, processing petabytes of data each day (Dean and Ghemawat, 2004)
  - PankRank, Inverted Index Construction, Log Analysis, Distributed Grep,…
- Hadoop is the leading open source implementation of MapReduce
  - Sponsored by Yahoo, Google, Amazon, and other major vendors
  - Clusters with 10k nodes, petabytes of data
- Benefits
  - Scalable, Efficient, Reliable
  - Easy to Program
  - Open Source
- Challenges
  - Redesigning / Retooling applications
    - Not SunGrid, Not MPI

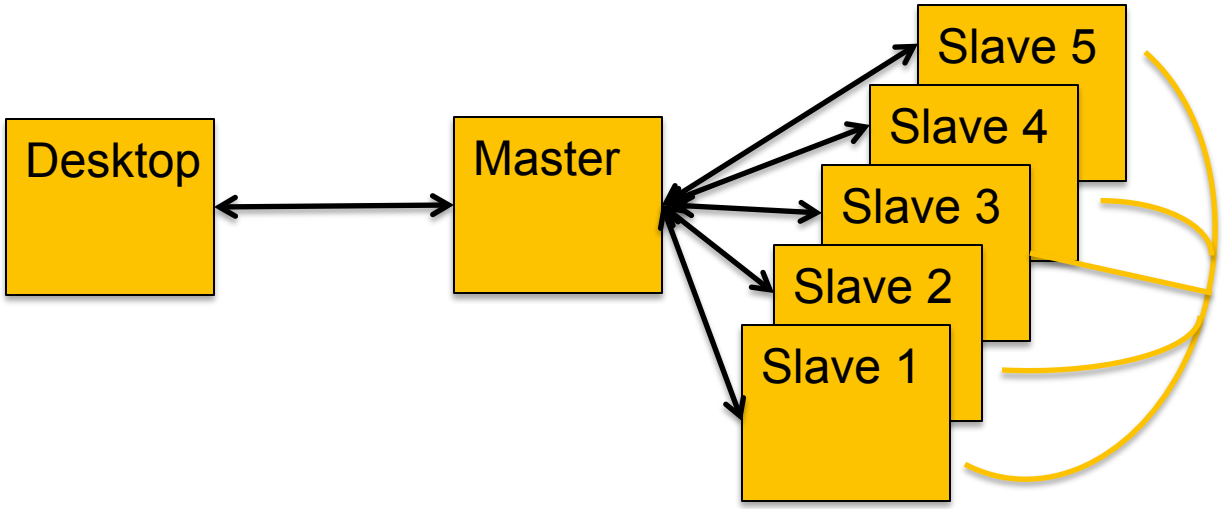
K-mer Counting with MapReduce



- Application developers focus on 2 (+1 internal) functions
  - Map: input -> key, value pairs
  - Shuffle: Group together pairs with same key
  - Reduce: key, value-lists -> output

Map, Shuffle & Reduce
All Run in Parallel

Hadoop Use Model



- Hadoop Distributed File System (HDFS)
  - Data files partitioned into large chunks (64MB), replicated on multiple nodes
  - NameNode stores metadata information (block locations, directory structure)

- Master node (JobTracker) schedules and monitors work on slaves
  - Computation moves to the data

Hadoop for Computational Biology?

- Biological research requires an increasing amount of data

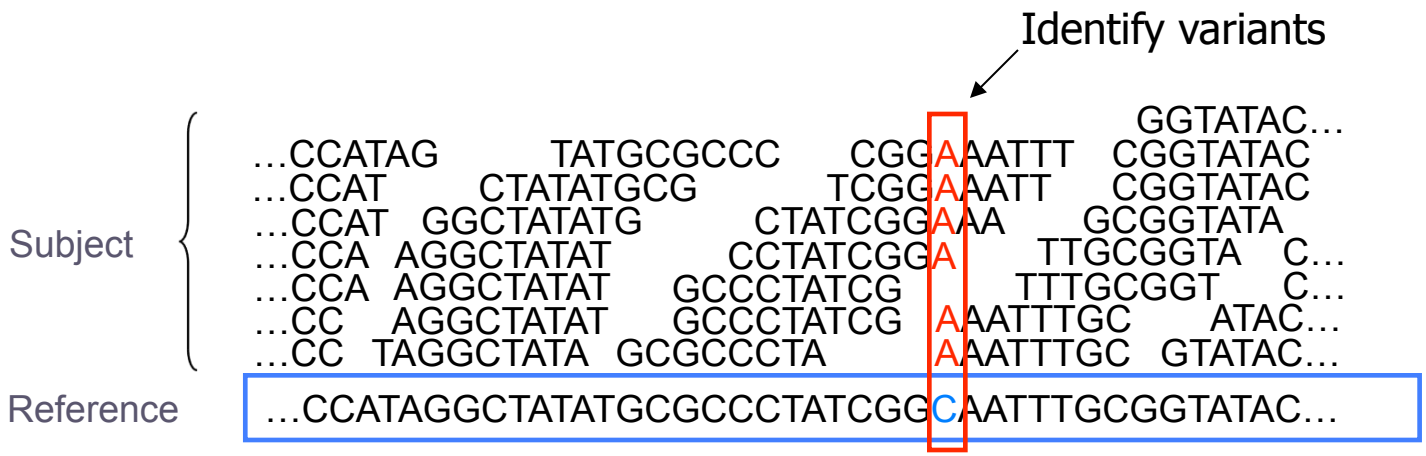
1000 Genomes

Global Ocean Survey

Human Microbiome

- Big data computing requires thinking in parallel
  - Hadoop may be an enabling technology

Short Read Mapping

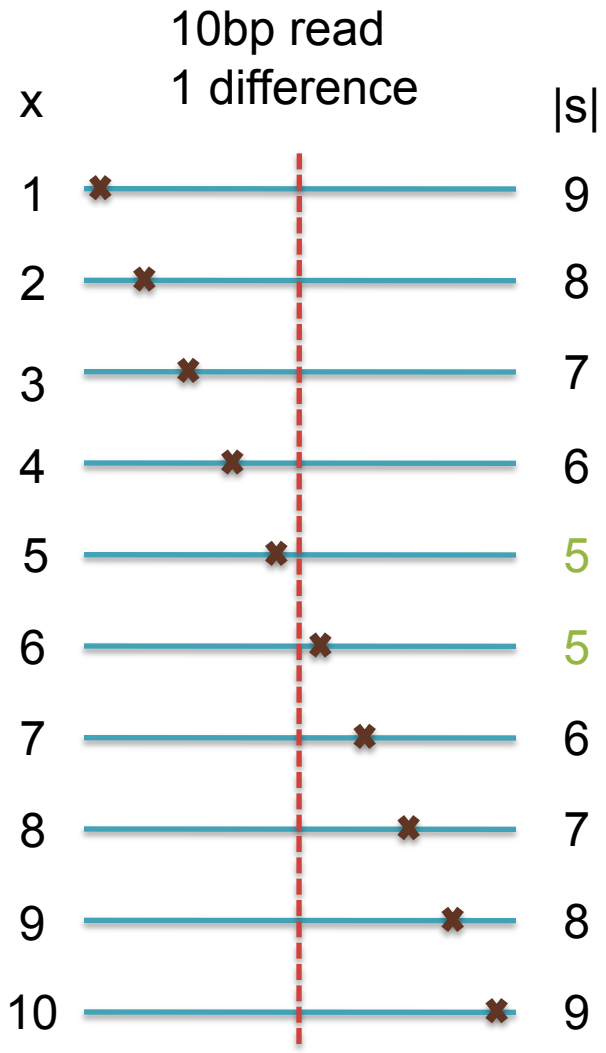


- Recent studies of entire human genomes analyzed billions of reads
  - Asian Individual Genome: 3.3 Billion 35bp, 104 GB (Wang et al., 2008)
  - African Individual Genome: 4.0 Billion 35bp, 144 GB (Bentley et al., 2008)

- Alignment computation required >10,000 CPU hours*
  - Alignments are "embarassingly parallel" by read
  - Variant detection is parallel by chromosome region

Seed and Extend

- Good alignments must have significant exact alignments
  - Use exact alignments to seed search for longer in-exact alignments
  - Minimal exact alignment length = $l/(k+1)$
- RMAP (Smith et al, 2008)
  - Catalog mers in reads as seeds
  - Stream across reference sequence
  - Count mismatches of reads anchored by seeds
- Expensive to scale to large experiments or highly sensitive searches
  - If only there was a way to parallelize execution…

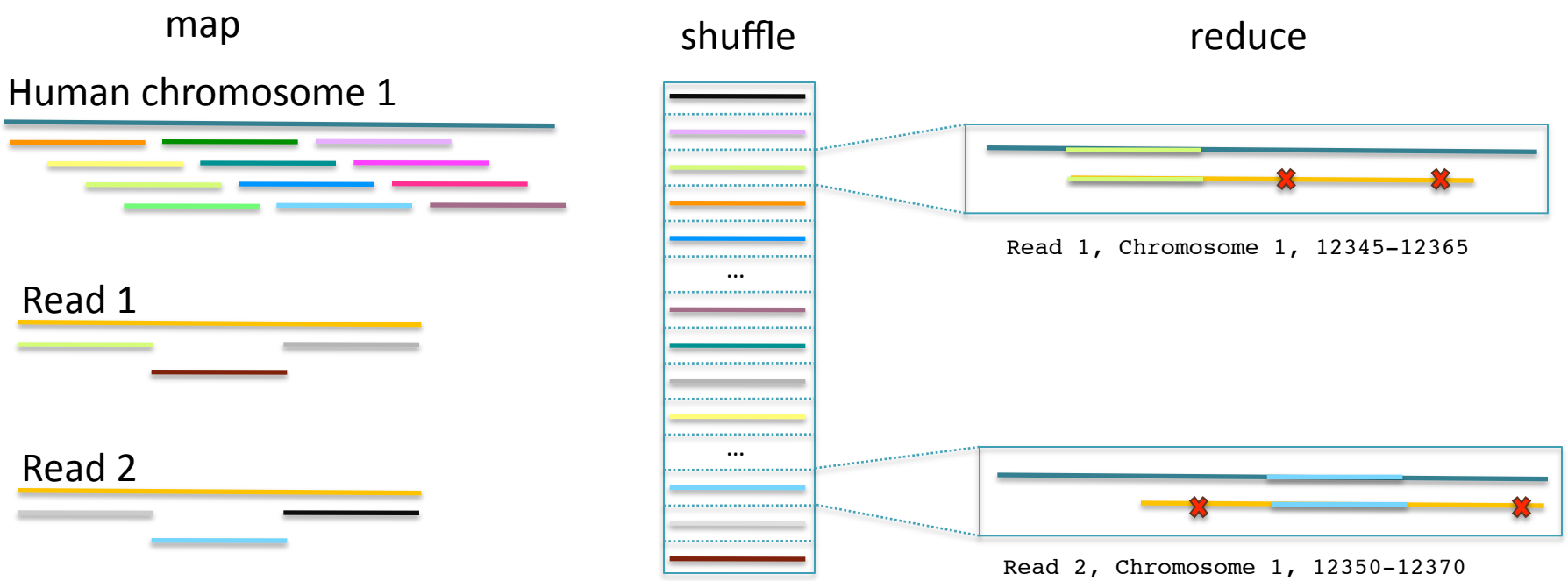


CloudBurst

1. Map: Catalog K-mers
   - Emit k-mers in the genome and reads

2. Shuffle: Collect Seeds
   - Conceptually build a hash table of k-mers and their occurrences

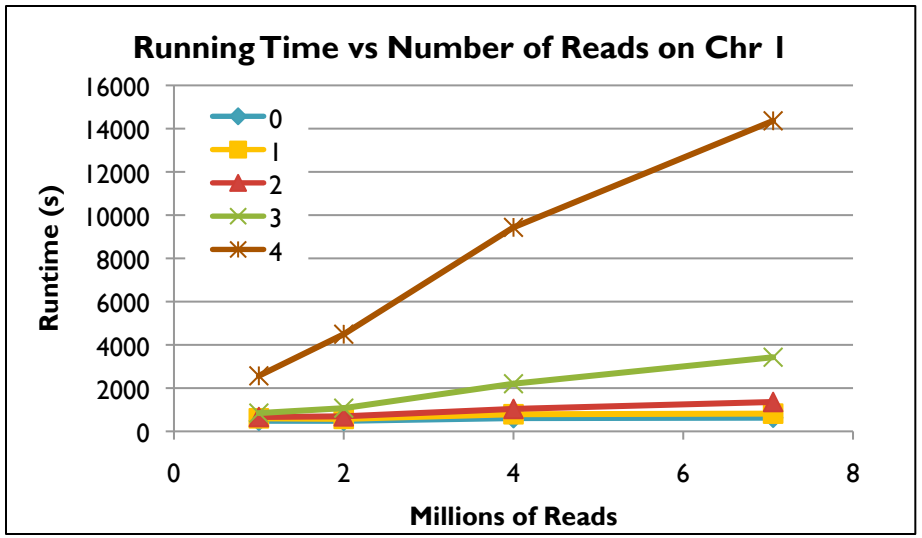
3. Reduce: End-to-end alignment
   - If read aligns end-to-end with ≤ k errors, record the alignment

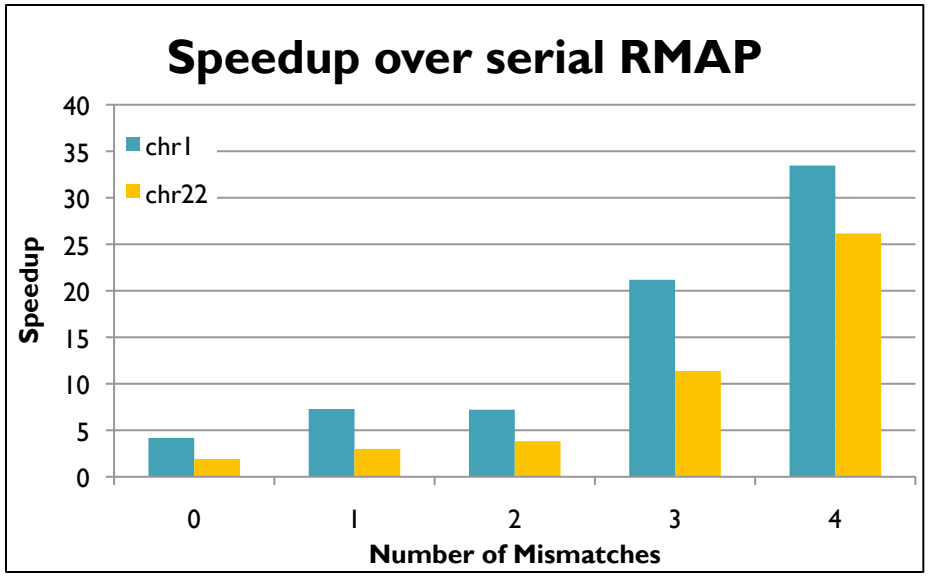


Schatz, MC (2009) CloudBurst: Highly Sensitive Read Mapping with MapReduce. *Bioinformatics*. 25:1363-1369

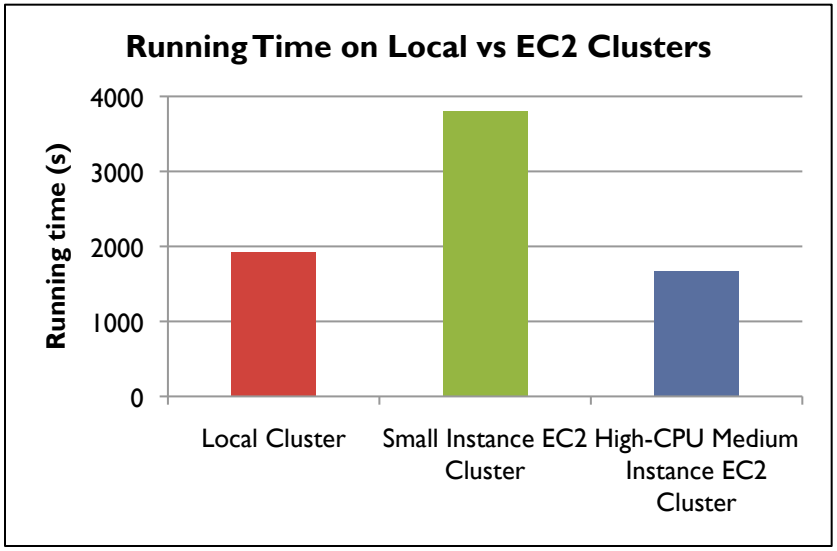
CloudBurst Results

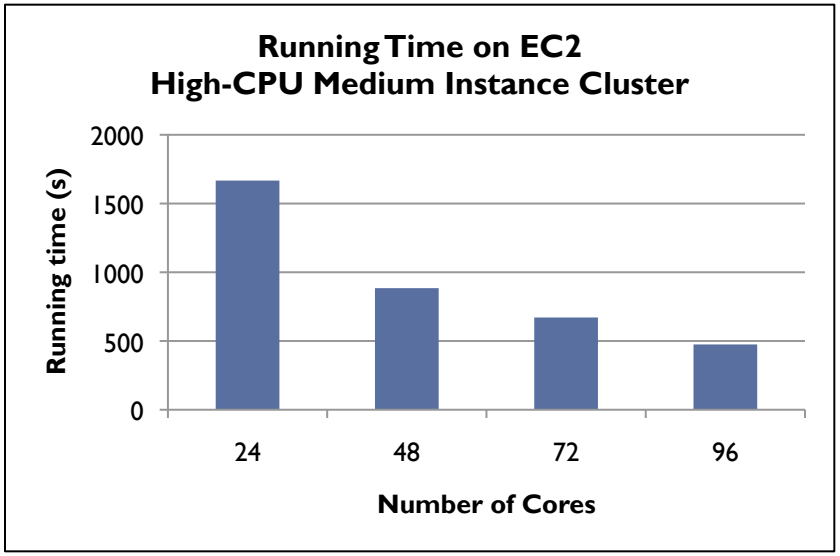
- Evaluate running time on local 24 core cluster
  - Running time increases linearly with the number of reads
- Compare to RMAP
  - Highly sensitive alignments have better than 24x linear speedup.
- Produces identical results in a fraction of the time





EC2 Evaluation

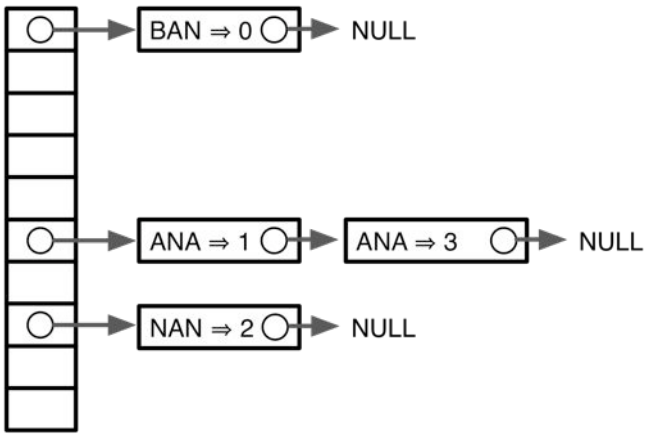




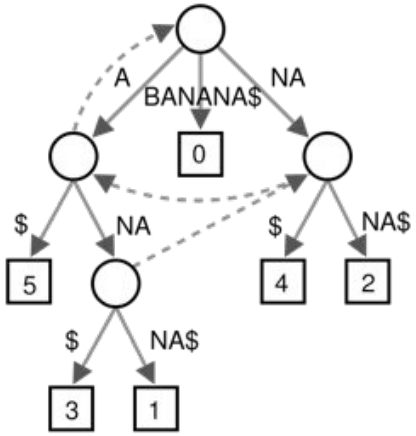
- CloudBurst running times for mapping 7M reads to human chromosome 22 with at most 4 mismatches on the local and EC 2 clusters.
- The 24-core Amazon High-CPU Medium Instance EC2 cluster is faster than the 24-core Small Instance EC2 cluster, and the 24-core local dedicated cluster.
- The 96-core cluster is 3.5x faster than the 24-core, and 100x faster than serial RMAP.

CloudBurst Reflections

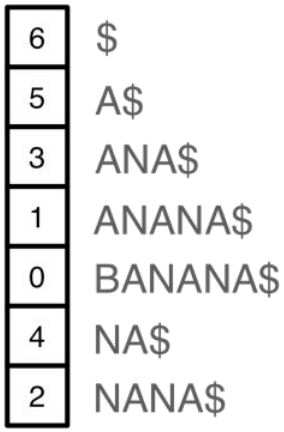
- CloudBurst efficiently reports every k-difference alignment of every read
  - But many applications only need the best alignment
  - Finding just the best alignment needs an in-memory index



Seed hash tables (>15 GB)
Many variants, incl. spaced seeds



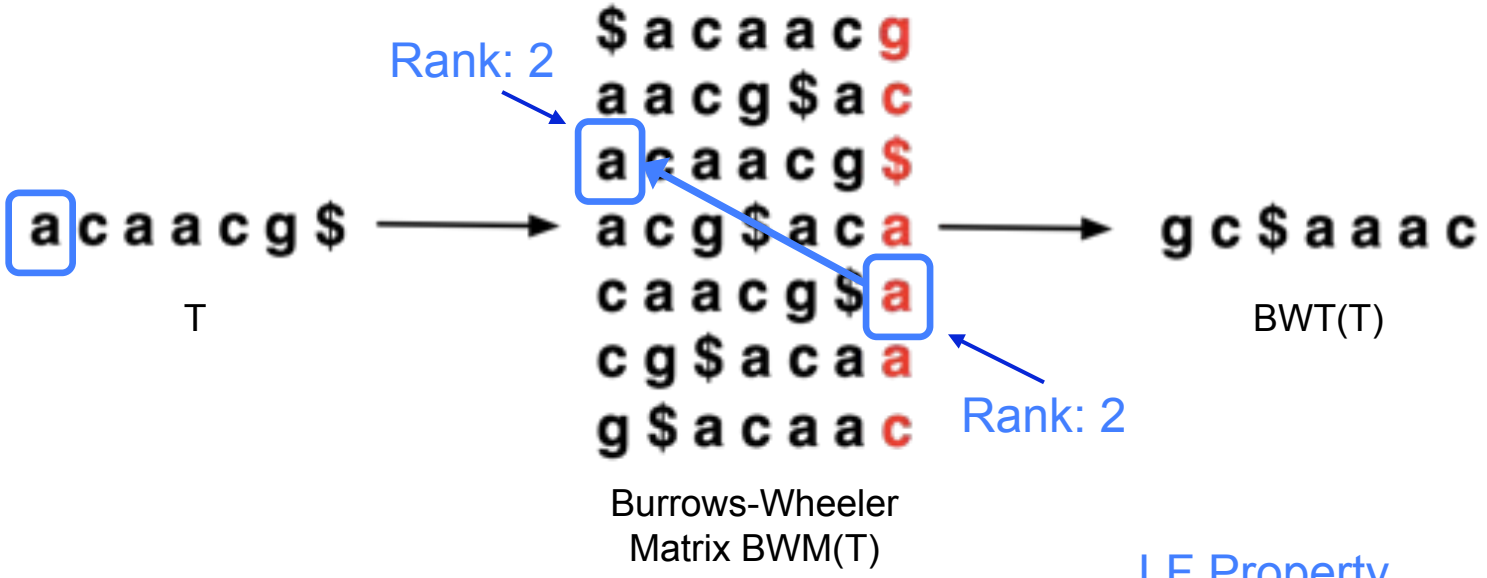
Suffix tree
(>51 GB)



Suffix array
(>15 GB)

Burrows-Wheeler Transform

- Reversible permutation of the characters in a text



LF Property implicitly encodes Suffix Array

- BWT(T) is the index for T

Burrows M, Wheeler DJ: A block sorting lossless data compression algorithm. *Digital Equipment Corporation, Palo Alto, CA* 1994, Technical Report 124; 1994

Bowtie: Ultrafast Short Read Aligner

- Quality-aware backtracking of BWT to rapidly find the best alignment(s) for each read

- BWT precomputed once, easy to distribute, and analyze in RAM
  - 3 GB for whole human genome

- Support for paired-end alignment, quality guarantees, etc…
  - Langmead B, Trapnell C, Pop M, Salzberg SL. (2009) Ultrafast and memory-efficient alignment of short DNA sequences to the human genome. *Genome Biology* 10:R25.

Bowtie algorithm

Reference

BWT(Reference)

Query:
A A T G A T A C G G C G A C C A C C G A G A T C T A

Bowtie algorithm

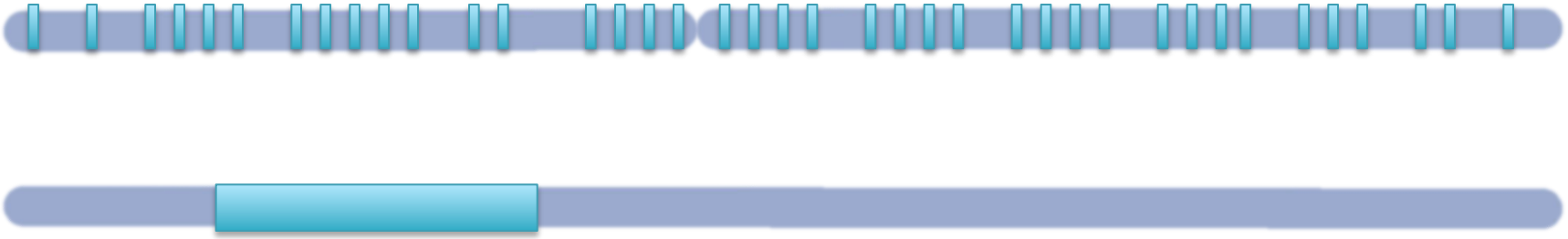
Reference

BWT(Reference)

Query:
A A T G A T A C G G C G A C C A C C G A G A T C T A

Bowtie algorithm

Reference

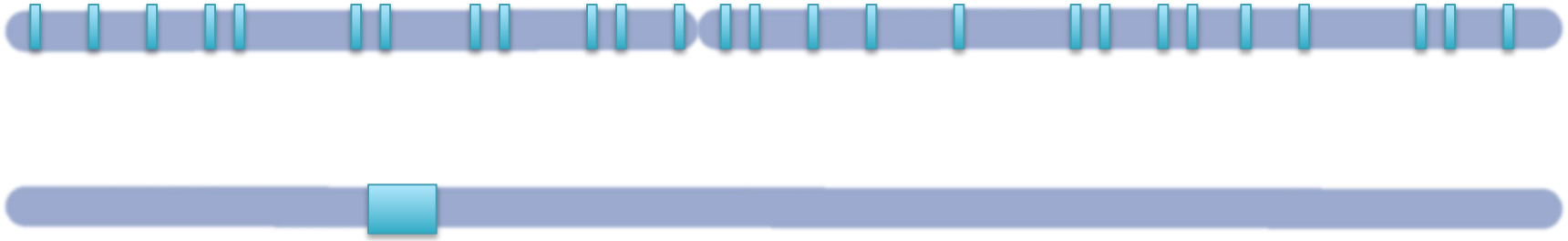


BWT(Reference)

Query:
A A T G A T A C G G C G A C C A C C G A G A T C T A

Bowtie algorithm

Reference



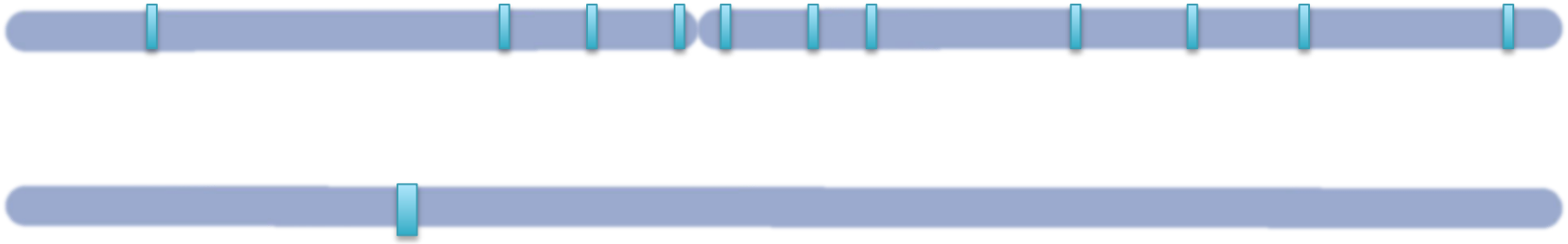
BWT(Reference)

Query:

A A T G A T A C G G C G A C C A C C G A G A T C T A

Bowtie algorithm

Reference



BWT(Reference)

Query:

A A T G A T A C G G C G A C C A C C G A G A T C T A

Bowtie algorithm

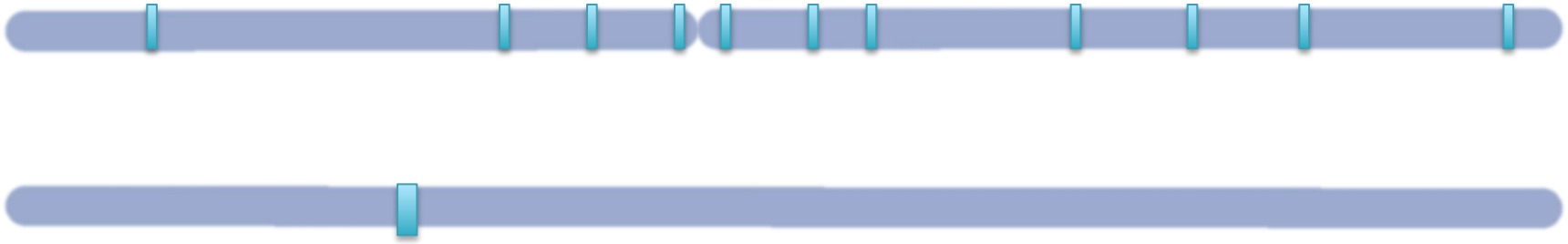
Reference

BWT(Reference)

Query:
AATG ATACGGCGACCACCGAGATCTA

Bowtie algorithm

Reference



BWT(Reference)

Query:

A A T G T T A C G G C G A C C A C C G A G A T C T A

Bowtie algorithm

Reference

BWT(Reference)

Query:

A A T G T T A C G G C G A C C A C C G A G A T C T A

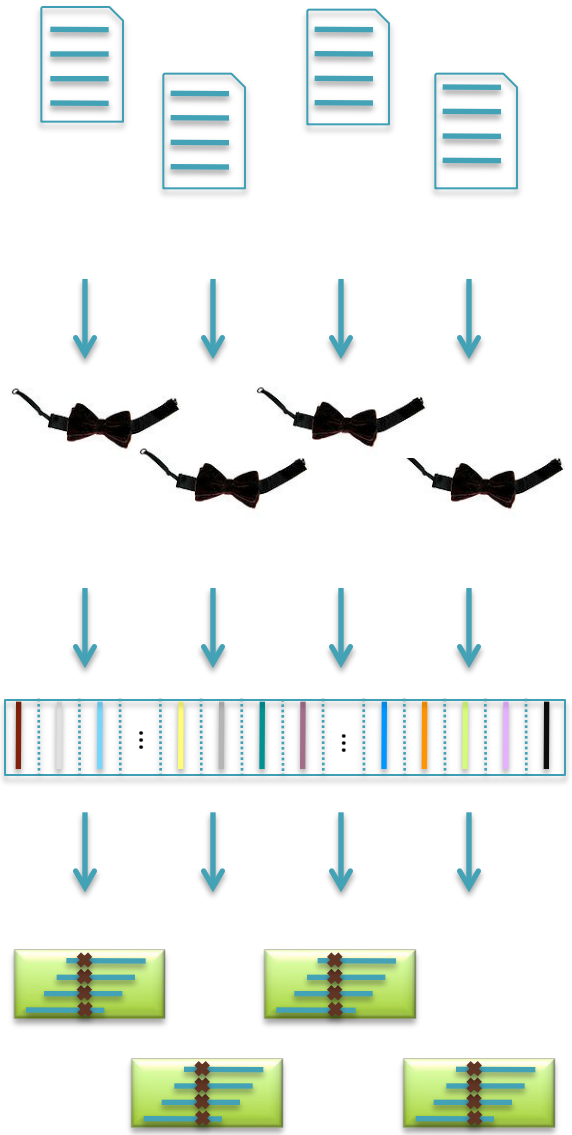
Comparison to MAQ & SOAP

	CPU time	Wall clock time	Reads mapped per hour (millions)	Peak virtual memory footprint (MB)	Bowtie speedup	Reads aligned (%)
Bowtie –v 2 (server)	15m:07s	15m:41s	33.8	1,149	-	67.4
SOAP (server)	91h:57m:35s	91h:47m:46s	0.10	13,619	351x	67.3
Bowtie (server)	17m:58s	18m:26s	28.8	1,353	-	71.9
Bowtie --best (server)	46m:54s	47m:23s	11.2	2,383	2.6x	72.0
Maq (server)	32h:56m:53s	32h:58m:39s	0.27	804	107x	74.7

Performance and sensitivity of Bowtie v0.9.6, SOAP v1.10, Maq v0.6.6 when aligning 8.84M reads from the 1000 Genome project [NCBI Short Read Archive:SRR001115] trimmed to 35 base pairs. The "soap.contig" version of the SOAP binary was used. SOAP could not be run on the PC because SOAP's memory footprint exceeds the PC's physical memory. For the SOAP comparison, Bowtie was invoked with "-v 2" to mimic SOAP's default matching policy (which allows up to 2 mismatches in the alignment and disregards quality values). For the Maq comparison Bowtie is run with its default policy, which mimics Maq's default policy of allowing up to 2 mismatches in the first 28 bases and enforcing an overall limit of 70 on the sum of the quality values at all mismatched positions. To make Bowtie's memory footprint more comparable to Maq's, Bowtie is invoked with the "-z" option in all experiments to ensure only the forward or mirror index is resident in memory at one time.

Crossbow: Rapid Whole Genome SNP Analysis

- Align billions of reads and find SNPs
  - Reuse software components: Hadoop Streaming
- Map: Bowtie
  - Emit (chromome region, alignment)
- Shuffle: Hadoop
  - Group and sort alignments by region
- Reduce: SoapSNP (Li et al, 2009)
  - Scan alignments for divergent columns
  - Accounts for sequencing error, known SNPs



Results coming soon

Short Read Genome Assembly

- Several new assemblers developed for short read data
  - Variations on compressed de Bruijn graphs

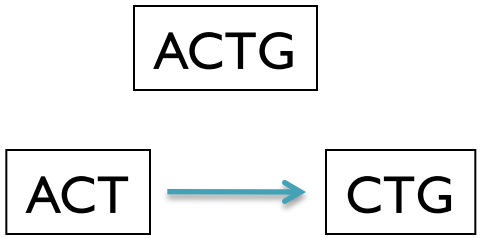
    Velvet (Zerbino & Birney, 2008)
    EULER-USR (Chaisson et al, 2009)
    ALLPATHS (Butler et al, 2008)
    ABySS (Simpson et al, 2009)

- Short Read Assembler Outline
  1. Construct compressed de Bruijn Graph
  2. Remove sequencing error from graph
  3. Use mate-pairs to resolve ambiguities

- Successful for small to medium genomes
  - 2Mbp bacteria – 39Mbp fungal assembly

de Bruijn Graph Construction

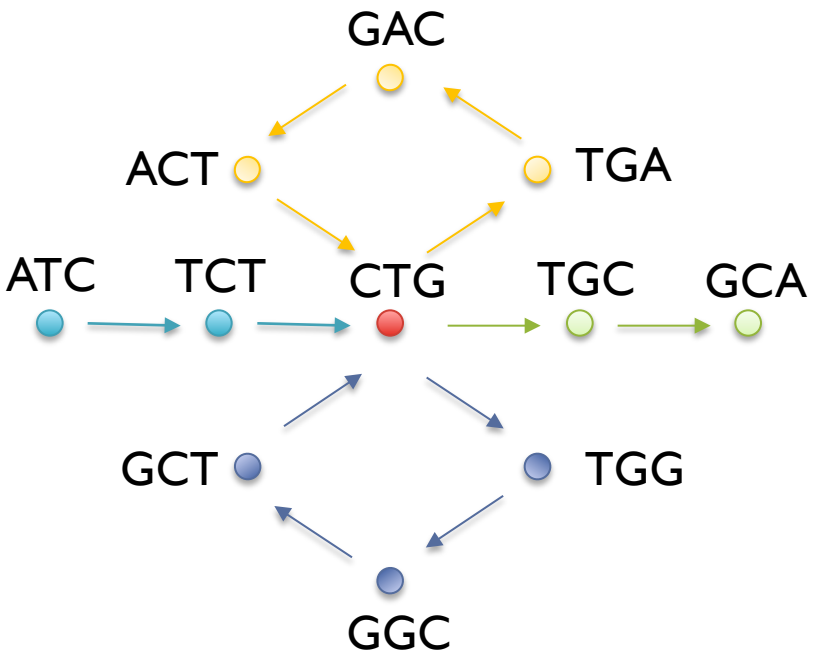
- $D_k = (V,E)$
  - V = All length-k mers (k < l)
  - E = Directed edges between consecutive mers



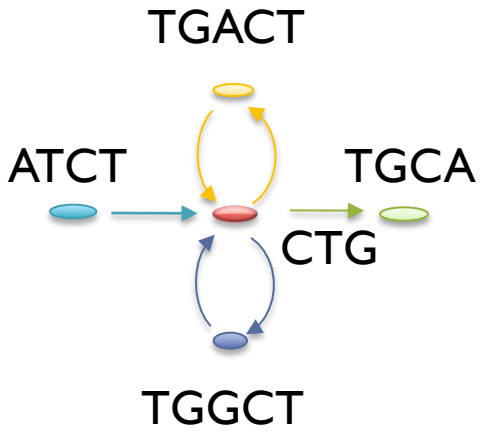
Reads

ACTG
ATCT
CTGA
CTGG
CTGC
GACT
GCTG
GGCT
TCTG
TGAC
TGCA
TGGC

De Bruijn Graph



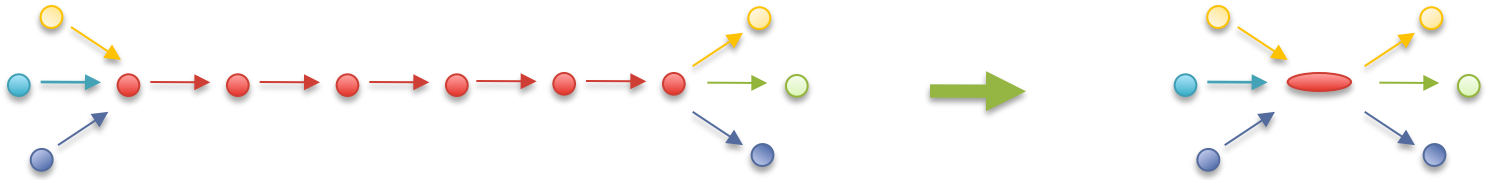
Compressed Graph



Genome Assembly with MapReduce

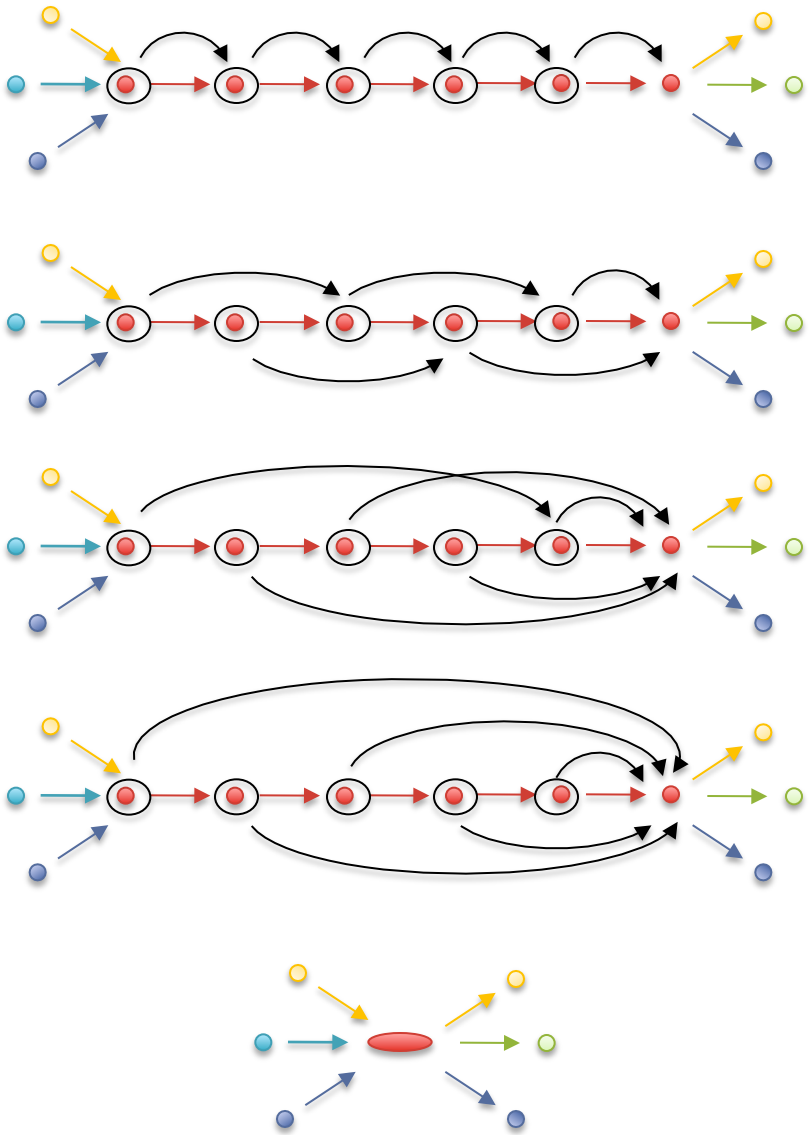
- The new short read assemblers require tremendous computation
  - Velvet on 48x coverage of 2 Mbp *S. suis* requires > 2GB of RAM
  - ABySS on 42x coverage of human required ~4 days on 168 cores

- Challenge: How to efficiently implement assembly algorithms in (restricted) MapReduce environment?
  - Different nodes of the graph are stored on different machines.



Parallel Path Compression

- List Ranking Problem
  - General problem of parallel linked list operations
- Pointer Jumping (Wyllie, 1979)
  - Add tail pointer to each node
  - In each round, advance (double) the tail pointer
  - Collect and concatenate all the nodes in a simple path
  - O(log S) parallel cycles
    - *B. anthracis* 268,925 -> 19+1 cycles
    - Human: 37,172 ->16+1 cycles



Summary

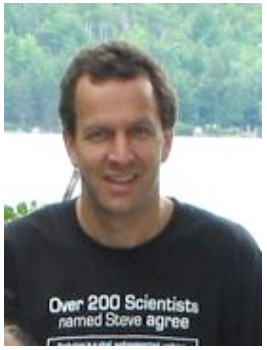
1. Hadoop is well suited to big data biological computation

2. Hadoop Streaming for easy scaling of existing software

3. Cloud computing is an attractive platform to augment resources

4. Look for many cloud computing & MapReduce solutions this year

Acknowledgements



Steven Salzberg


Mihai Pop


Jimmy Lin


Ben Langmead

hadoop

CBCB
Center for Bioinformatics & Computational Biology

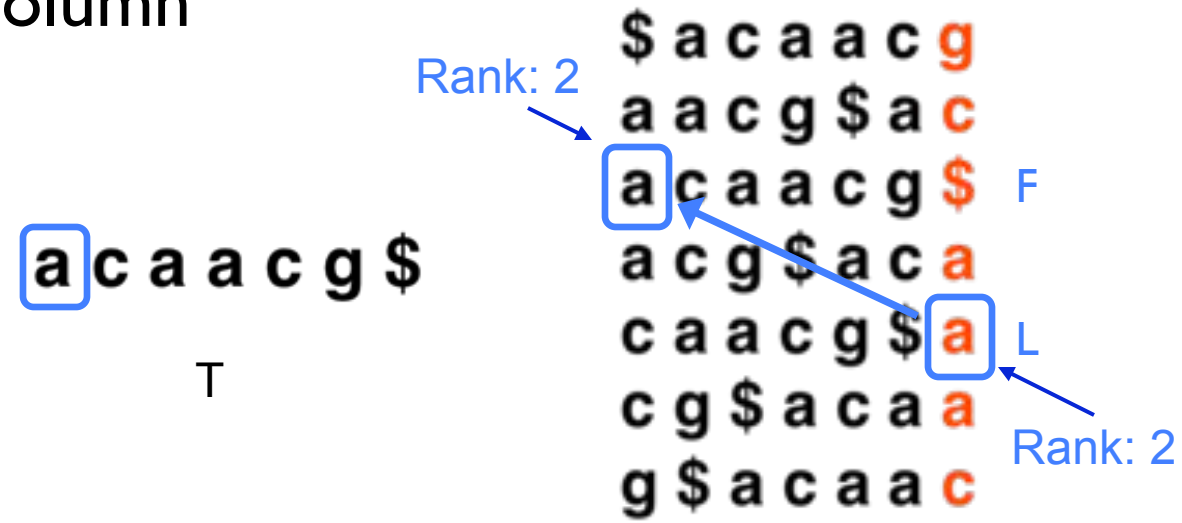
amazon web services™

Thank You!

http://www.cbcb.umd.edu/~mschatz

Burrows-Wheeler Transform

- Property that makes BWT(T) reversible is LF Mapping
  - i^{th} occurrence of a character in **L**ast column is same *text* occurrence as the i^{th} occurrence in **F**irst column



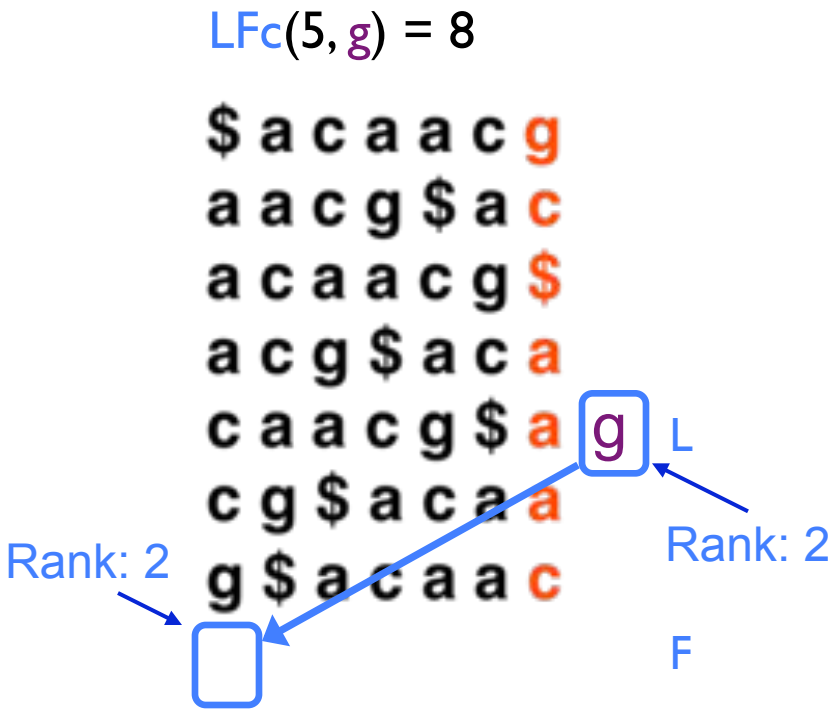
  - **LF**(*r*): function taking **L** row *r* to corresponding **F**

Burrows-Wheeler Transform

- Recreating T from BWT(T)
  - Start in the first row and apply **LF** repeatedly, accumulating predecessors along the way

Exact Matching

- **LFc**(r, c) does the same thing as **LF**(r) but it ignores r's actual final character and "pretends" it's c:



Exact Matching

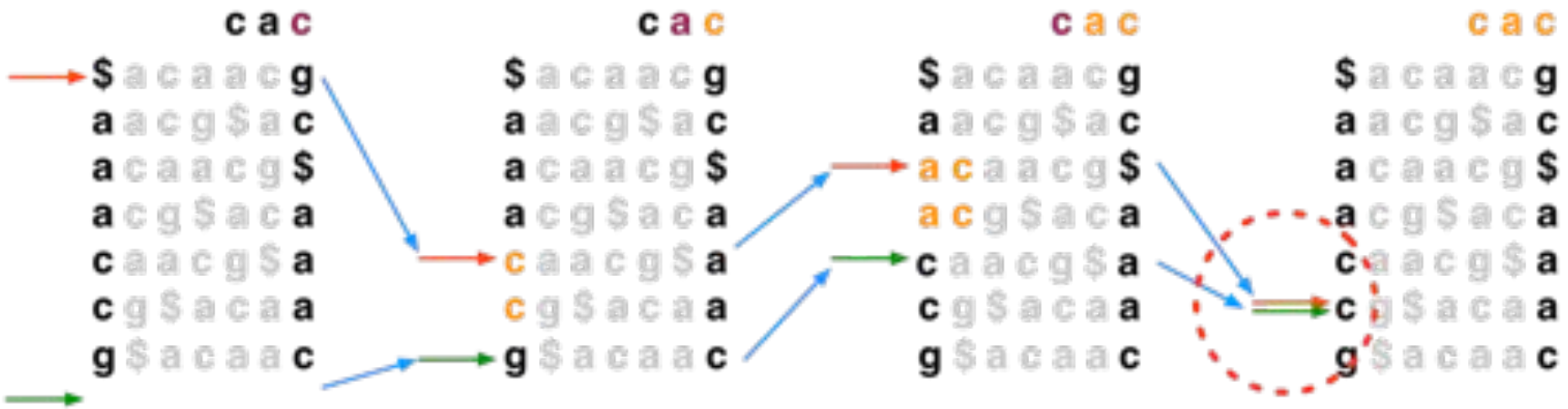
- Start with a range, (**top**, **bot**) encompassing all rows and repeatedly apply **LFc**:

  top = **LFc**(**top**, **qc**); **bot** = **LFc**(**bot**, **qc**)

  qc = the next character to the left in the query

Ferragina P, Manzini G: Opportunistic data structures with applications. *FOCS. IEEE Computer Society; 2000.*

Exact Matching



- If range becomes empty (**top** = **bot**) the query suffix (and therefore the query as a whole) does not occur