Towards a *de novo* short read assembler for large genomes using cloud computing

Michael Schatz

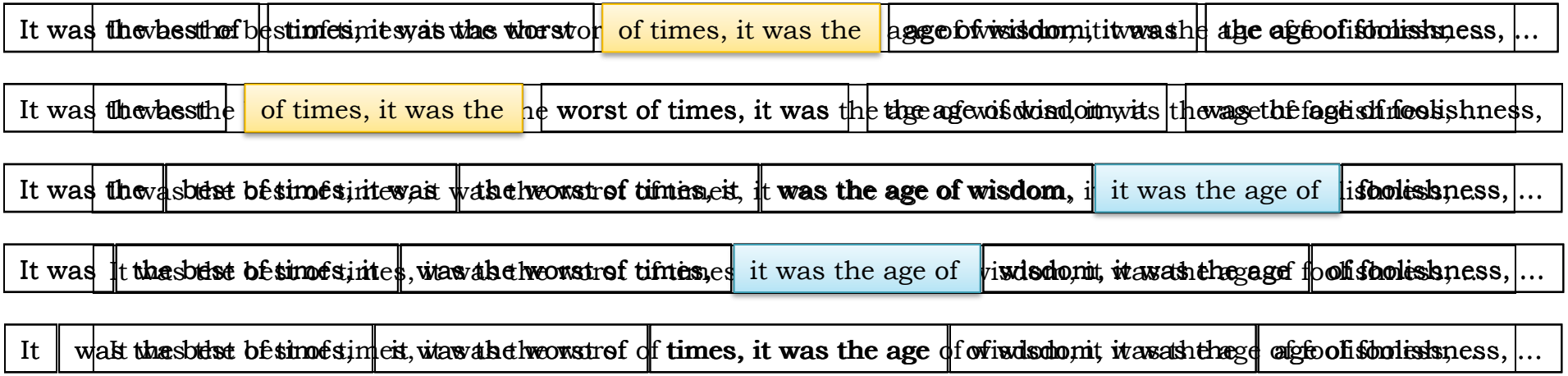
April 21, 2009
AMSC664 Advanced Scientific Computing

Outline

1. Genome assembly by analogy
2. DNA sequencing and assembly
3. MapReduce for genome assembly
4. Research plan and related work

Shredded Book Reconstruction

- Dickens accidently shreds the original A Tale of Two Cities
  - Text printed on 5 long spools



- How can he reconstruct the text?
  - 5 copies x 138, 656 words / 5 words per fragment = 138k fragments
  - The short fragments from every copy are mixed together
  - Some fragments are identical

It was the best of

age of wisdom, it was

best of times, it was

it was the age of

it was the age of

it was the worst of

of times, it was the

of times, it was the

of wisdom, it was the

the age of wisdom, it

the best of times, it

the worst of times, it

times, it was the age

times, it was the worst

was the age of wisdom,

was the age of foolishness,

was the best of times,

was the worst of times,

wisdom, it was the age

worst of times, it was

Overlaps

It was the best of
was the best of times,

4 word overlap

It was the best of
of times, it was the

1 word overlap

It was the best of
of wisdom, it was the

1 word overlap

- Generally prefer longer overlaps to shorter overlaps.
- In the presence of error, we might allow the overlapping fragments to differ by a small amount.
  - It was the beast of times, …

It was the best of
age of wisdom, it was
best of times, it was
it was the age of
it was the age of
it was the worst of
of times, it was the
of times, it was the
of wisdom, it was the
the age of wisdom, it
the best of times, it
the worst of times, it
times, it was the age
times, it was the worst
was the age of wisdom,
was the age of foolishness,
was the best of times,
was the worst of times,
wisdom, it was the age
worst of times, it was

Greedy Assembly

It was the best of
was the best of times,
the best of times, it
best of times, it was
of times, it was the
of times, it was the
times, it was the worst
times, it was the age

- The repeated sequence makes the correct reconstruction ambiguous
  - It was the best of times, it was the [worst/age]

Sequence Repeats

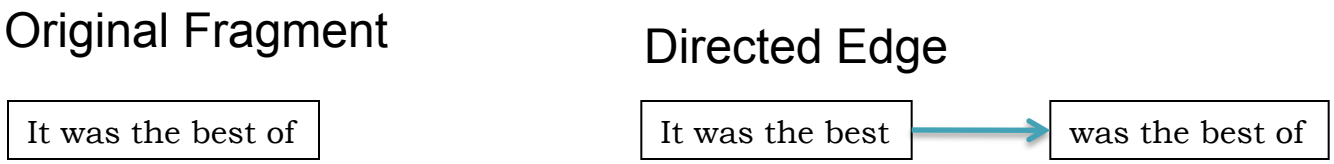
- Repeated sequences cause false overlaps between distant fragments
  - Transition between repeated and unique sequences forks the reconstruction

- Our sequence reconstruction algorithm must correctly resolve repeats or risk mis-assembly
  - It was the best of times, it was the age of wisdom, it was the age of foolishness, …

- Model the sequence structure and reconstruction problem as a graph problem.

de Bruijn Graph Construction

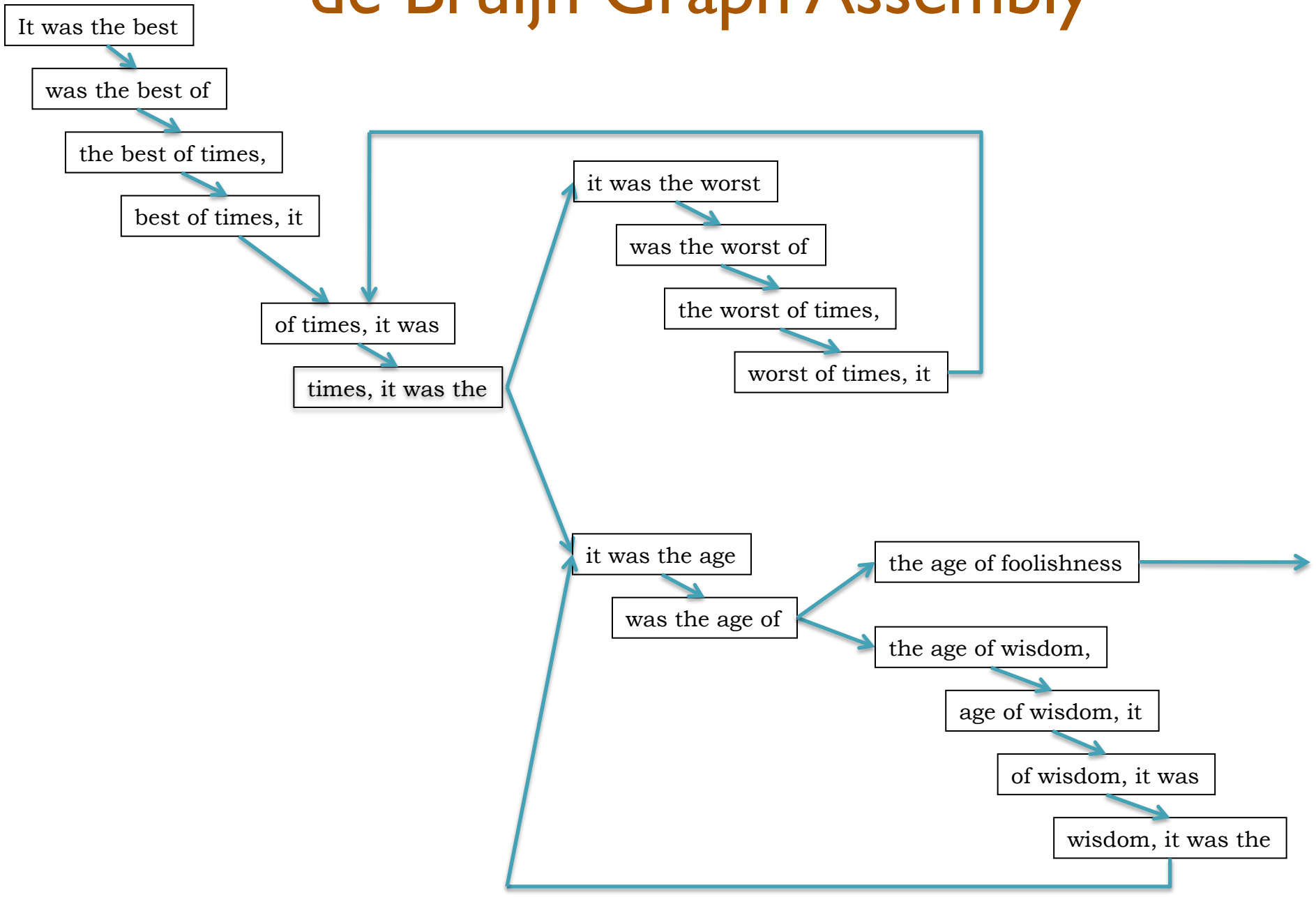
- $D_k = (V,E)$
  - V = All length-k subfragments (k < l)
  - E = Directed edges between consecutive subfragments
    - Nodes overlap by k-1 words



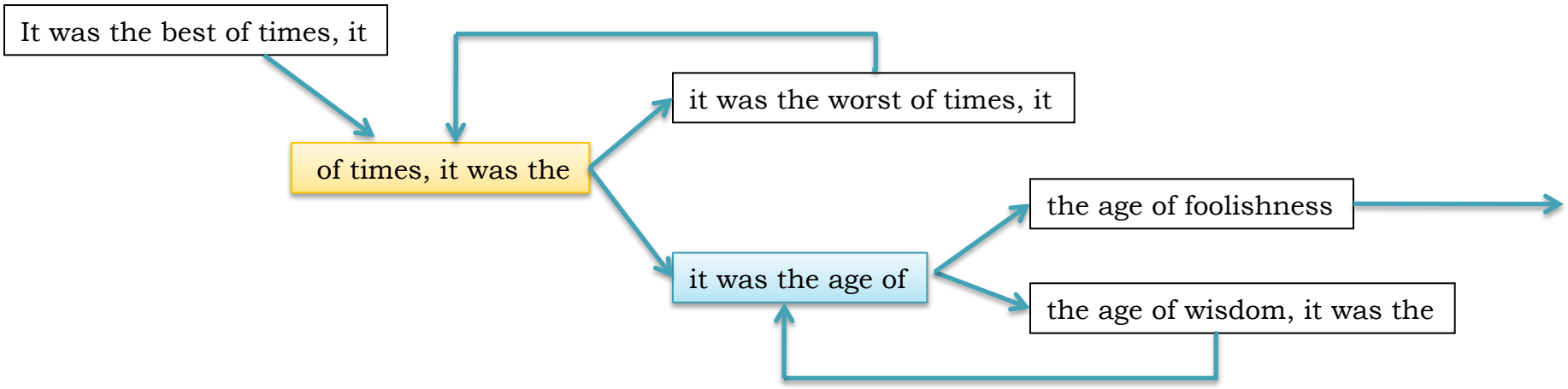
- Locally constructed graph reveals the global sequence structure
  - Overlaps implicitly computed

de Bruijn, 1946
Idury and Waterman, 1995
Pevzner, Tang, Waterman, 2001

de Bruijn Graph Assembly

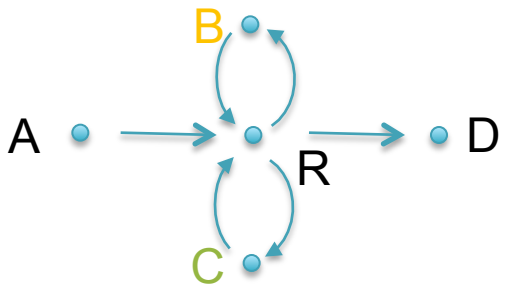


Compressed de Bruijn Graph



- Unambiguous non-branching paths replaced by single nodes
- An Eulerian/Chinese traversal of the graph spells a compatible reconstruction of the original text.
  - There may be many traversals of the graph
- Different sequences can have the same string graph
  - It was the best of times, it was the worst of times, it was the worst of times, it was the age of wisdom, it was the age of foolishness, …

Counting Eulerian Paths



ARBRCRD
or
ARCRBRD

- The number of compatible sequences increases exponentially with nested cycles.
  - Value computed by application of the BEST theorem (Hutchinson, 75)
  - Corrects for multiple edge orderings having the same node order

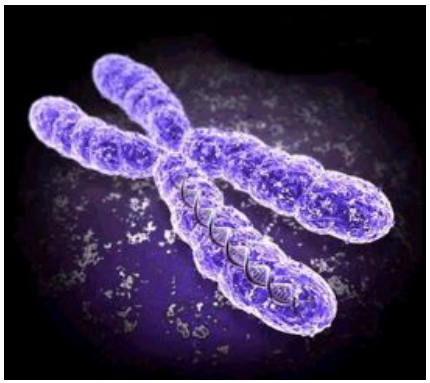
$$\mathcal{W}(G,t) = (\det L)\left\{\prod_{u\in V}(r_u-1)!\right\}\left\{\prod_{(u,v)\in E} a_{uv}!\right\}^{-1}$$

L = n x n matrix with r_u-a_{uu} along the diagonal and -a_{uv} in entry uv

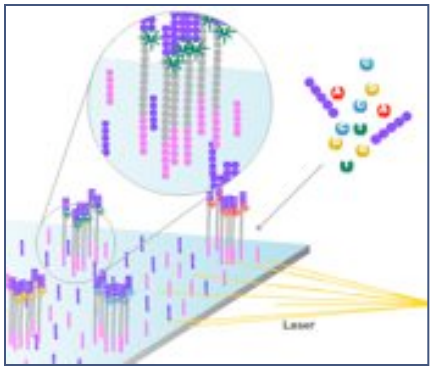
$r_u = d^+(u)+1$ if $u=t$, or $d^+(u)$ otherwise

a_{uv} = multiplicity of edge from u to v

DNA Sequencing



- Genome of an organism encodes the genetic information in long sequence of 4 DNA nucleotides: ACGT
  - Bacteria: ~5 million bp
  - Humans: ~3 billion bp



- Current DNA sequencing machines can generate 1-2 Gbp of sequence per day, in millions of short reads (25-300bp)
  - Shorter reads, but much higher throughput
  - Per-base error rate estimated at 1-2% (Simpson, et al, 2009)
  - Ends of the reads tend to have worse quality, other sequence dependent biases

ATCTGATAAGTCCCAGGACTTCAGT

GCAAGGCAAACCCGAGCCCAGTTT

TCCAGTTCTAGAGTTTCACATGATC

GGAGTTAGTAAAAGTCCACATTGAG

- Recent studies of entire human genomes have used 3.3 (Wang, et al., 2008) & 4.0 (Bentley, et al., 2008) billion 36bp reads
  - ~144 GB of compressed sequence data

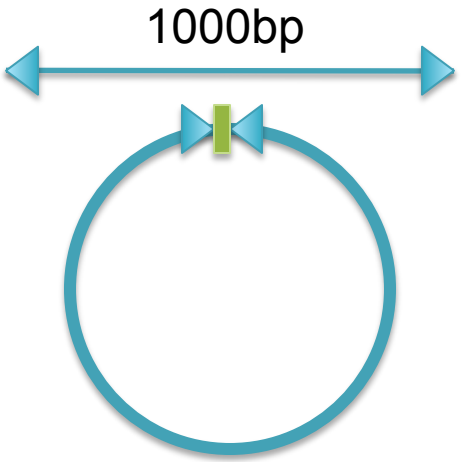
Sequencing Reads

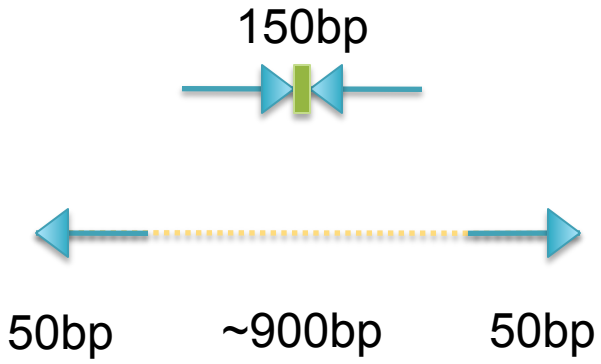
- DNA is double stranded
  - Nucleotides on one strand bond with complementary nucleotides of the other (A ⇔ T, C ⇔ G)
  - Orientation of each read is unknown

ATCTGATAAGTCCCAGGACTTCAGT

ACTGAAGTCCTGGGACTTATCAGAT

- Mate-pairs are pairs of reads separated by a known distance





Short Read Genome Assemblers

- Several new assemblers developed specifically for short read data
  - Old assemblers incompatible for technical and algorithmic reasons
  - Variations on compressed de Bruijn graphs
    - Velvet (Zerbino & Birney, 2008)
    - ALLPATHS (Butler et al, 2008)
    - EULER-USR (Chaisson et al, 2009)
    - ABySS (Simpson et al, 2009)

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

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

Large Genome Assembly

- Recent studies of the human genome primarily used comparative techniques to map individual reads to a reference genome
  - In contrast, *de novo* assembly has power to reveal new biology
  - Structural variations: Large Indels, Rearrangements, Copy Number Variations

- The new short read assemblers have been limited to small genomes, in part, because of the tremendous computational resources required
  - Velvet on 48x coverage of 2 Mbp *S. suis* requires > 2GB of RAM
  - ABySS on 42x coverage of human required ~4 days of parallel computation on ~200 CPUs

- If only there was a framework for large data sets…

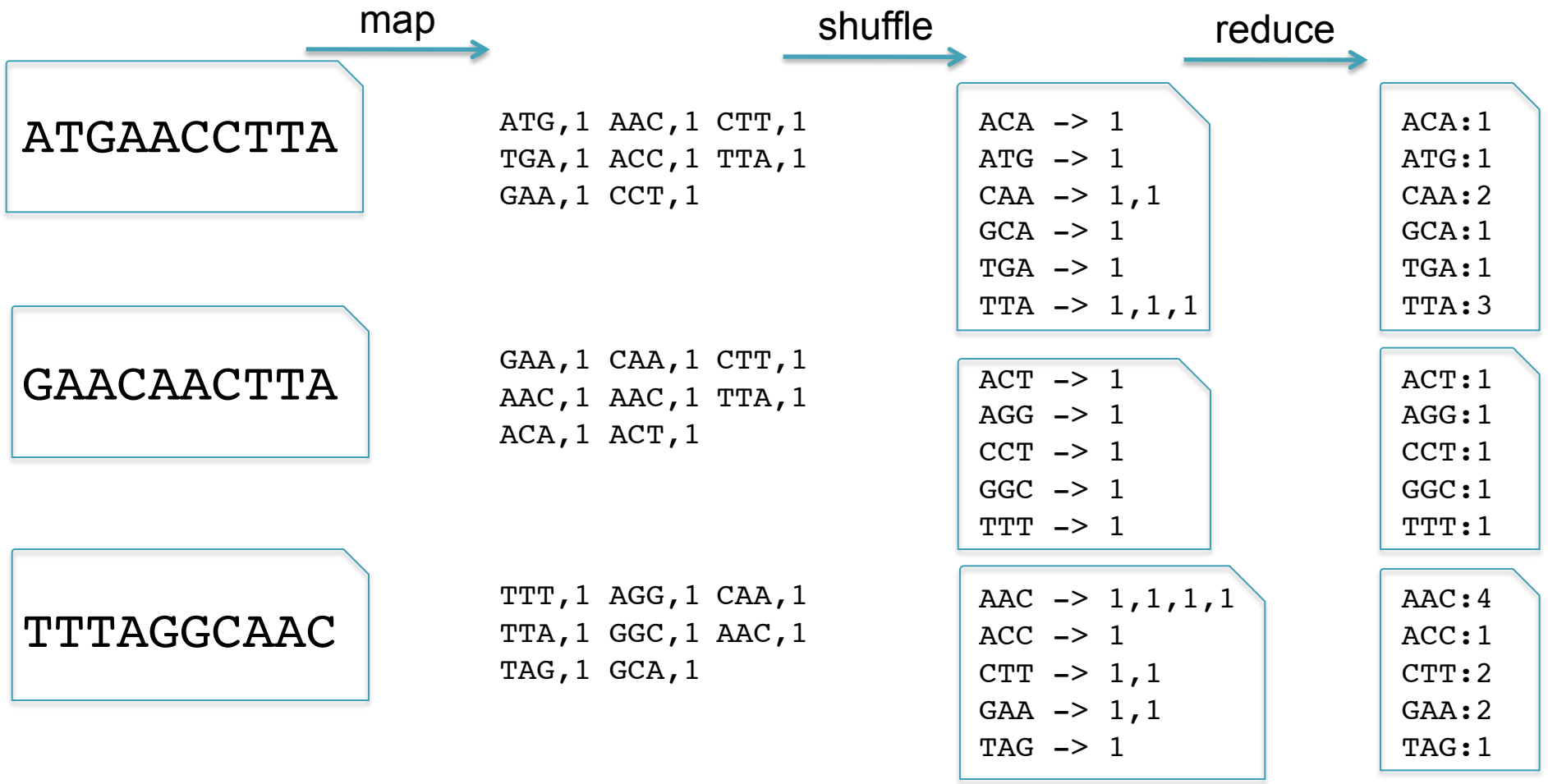
Hadoop MapReduce

- MapReduce is the parallel distributed framework invented by Google for their large data computations.
  - Data and computations are spread over thousands of computers, processing petabytes of data each day (Dean and Ghemawat, 2004)

- 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
  - 100k jobs per month

- Benefits
  - Scalable, Efficient, Reliable
  - Cloud Ready: Ready to run on remote resources
    - As little as 10¢ per hour per machine

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

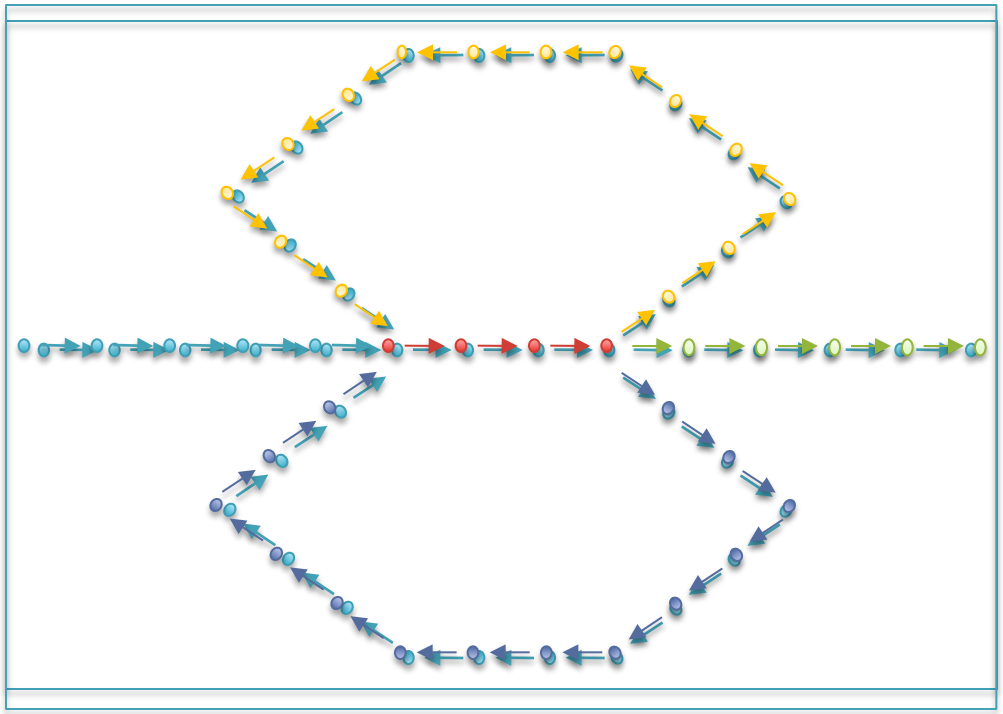
Genome Assembly with MapReduce

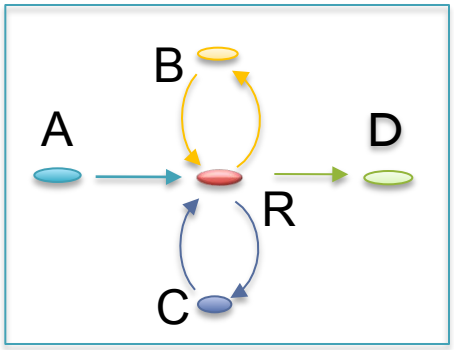
- Advantages
  - Proven system for processing huge datasets
    - PageRank: Significance in web graph of >1 trillion pages
    - CloudBurst: Highly Sensitive Alignment of Short Reads
  - Simple programming model
    - Reliability, redundancy, scalability built-in

- Challenges
  - How to efficiently implement assembly graph algorithms when adjacent nodes are stored on different machines?
    - Restricted programming model (not Shared Memory, not MPI)

de Bruijn Graph Construction

- Map: Scan reads and emit (k_i,k_{i+1}) for consecutive k-mers
  - Also consider reverse complement k-mers, build bi-directed graph
- Reduce: Save adjacency representation of graph (n, (nodeinfo, ni))





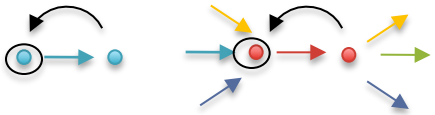
Mark Compressible Nodes

- Input: Graph stored as (n, (nodeinfo, ni))
- Map:
  - For all nodes, emit (n, (nodeinfo, ni))
  - If node n has unique predecessor p, emit (p, (unique-pred, n))

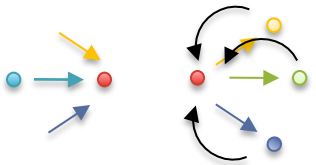
MapReduce Message Passing

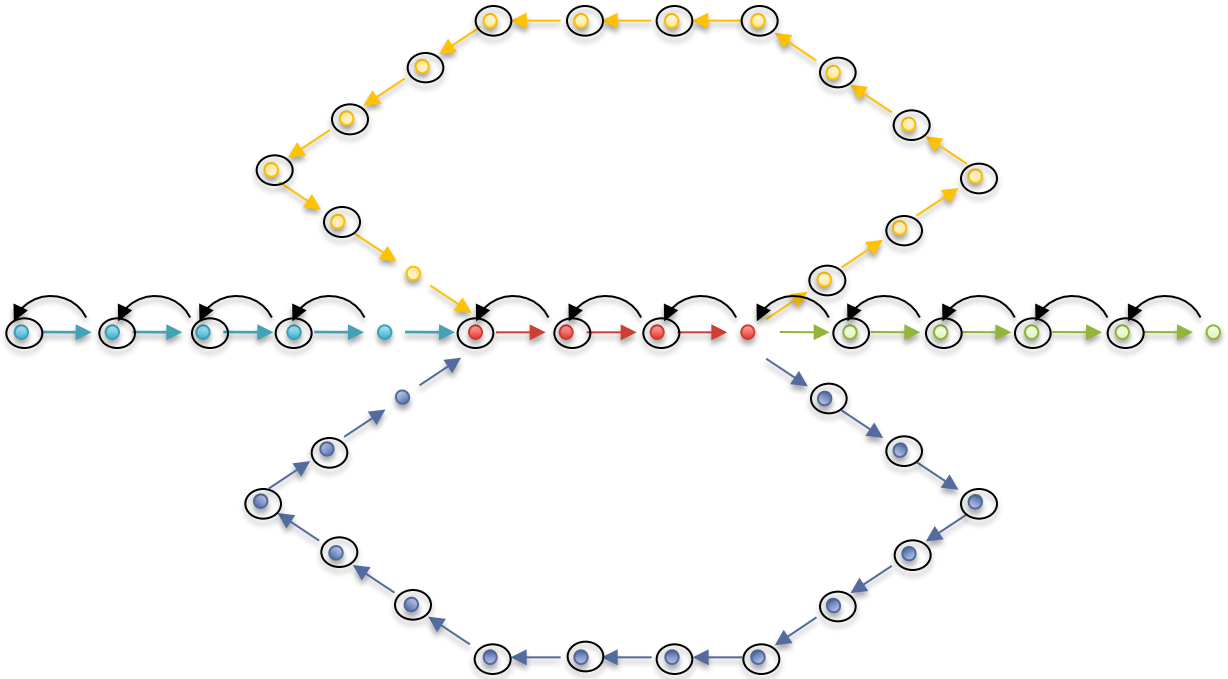
- Reduce:
  - If node n has unique successor s, and received (unique-pred, s),
    - Mark node as compressible
  - Save (n, nodeinfo)

Compressible



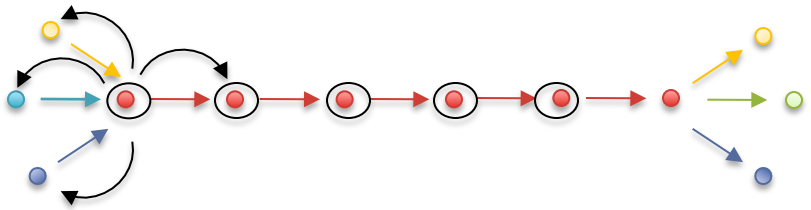
Not Compressible





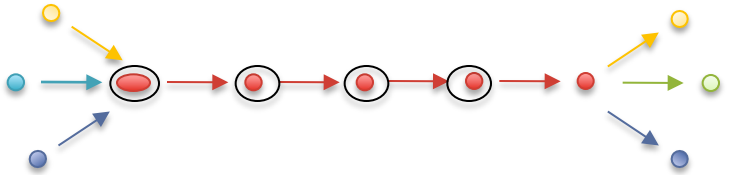
Linear Path Compression

- Iteratively identify and collapse the beginning of each chain



- Map:
  - Emit messages to the neighbors of the head of each chain

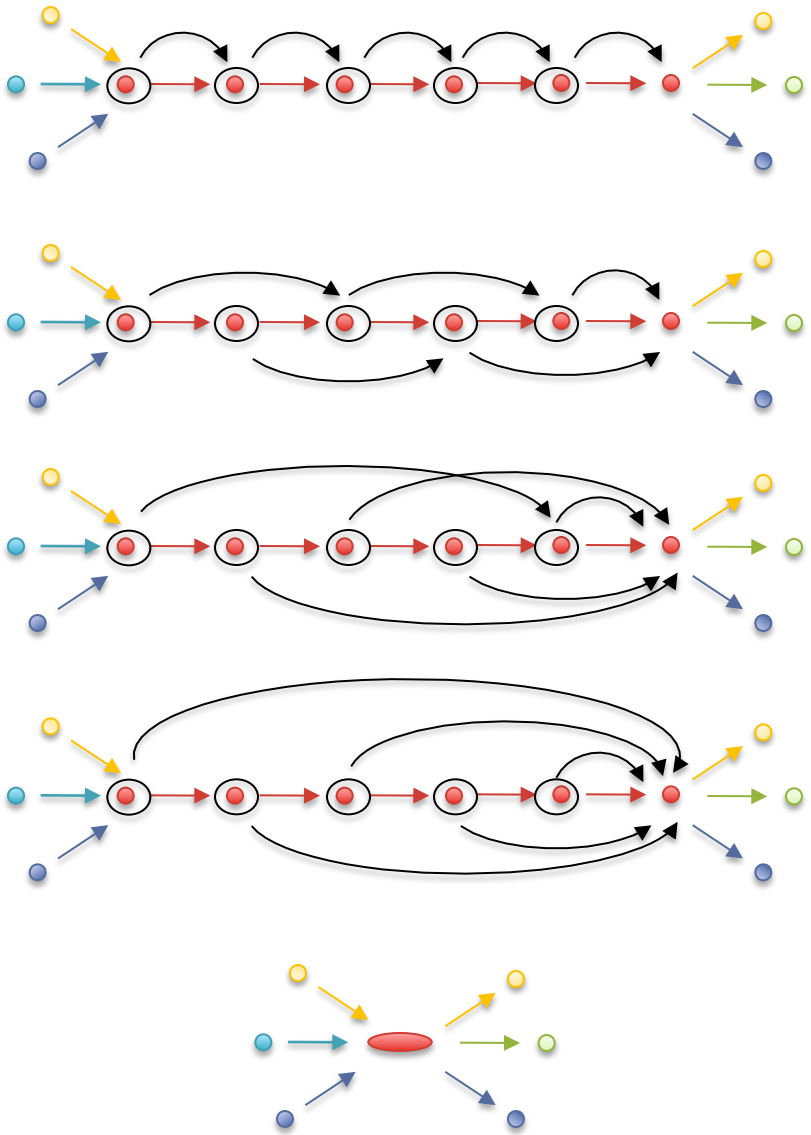
- Reduce:
  - Update links, node label



- Requires S MapReduce cycles, where S is the length of the longest simple path
  - *B. anthracis*: L=5.2Mbp S=268,925 bp
  - *H. sapiens* chr 22: L=49.6Mbp S=33,832 bp
  - *H. sapiens* chr 1: L=247.2Mbp S=37,172 bp

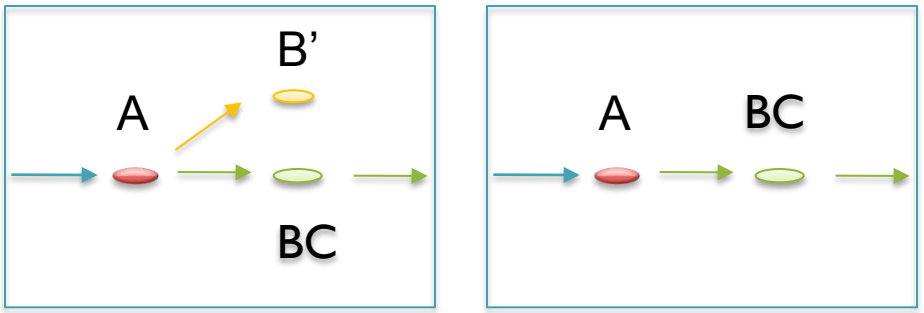
Parallel Path Compression

- 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

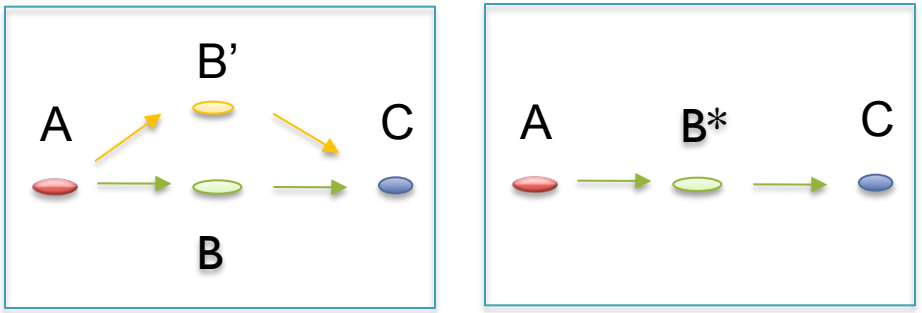


Error Correction

- Sequence error distorts idealized graph structure
  - Errors at end of read
    - Trim off 'dead-end' tips



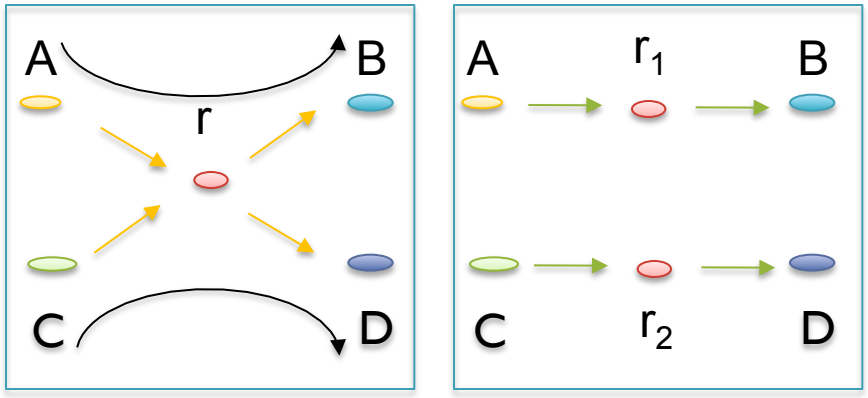
  - Errors in middle of read
    - Pop Bubbles
    - Record Hetrozygote sites



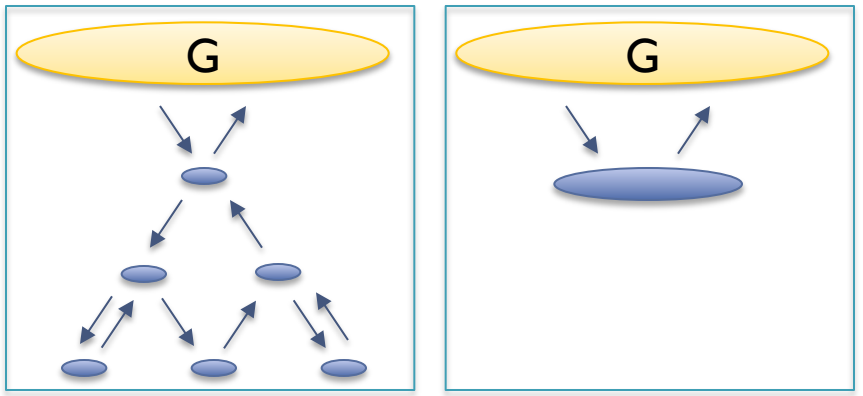
  - Multiple errors throughout
    - Small disconnected node

Graph Simplifications

- X-cut
  - Individual reads span the repeat
  - Especially for hybrid assemblies with short and long read
  - (Pevzner, Tang, Waterman, 2001)



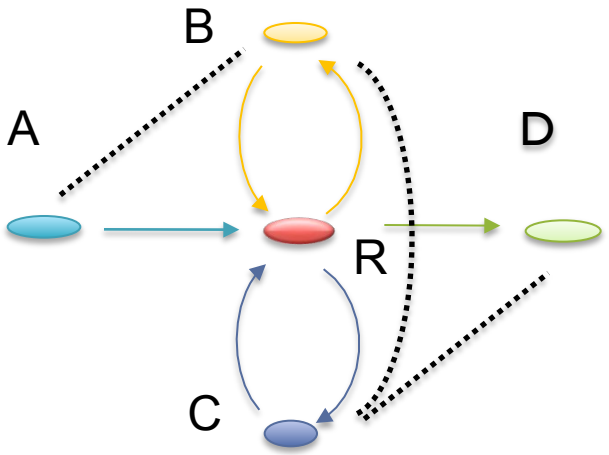
- Cycle tree compression
  - There is a unique path through nodes with a tree structure (Pevzner, 1989)
  - Recursive resolve those regions
  - (Kingsford, Schatz, Pop, 2009*)



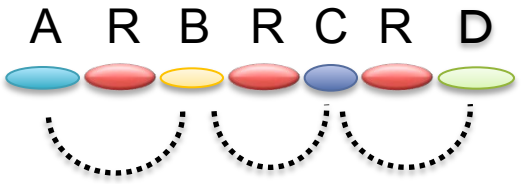
- Other simplifications possible

Scaffolding

- Use mate-pairs to resolve correct order through string graph
  - Place sequence to satisfy the mate constraints
  - Mates through repeat nodes are tangled



- Bambus (Pop, Kosack, Salzberg, 2004)
  - Edge Bundling
  - Contig Orientation & Ordering
    - Iteratively join strongest connection



- Parallel Bambus
  - Identify repeat nodes (A-stat)
  - Identify (strongly) connected components
  - Concurrently execute serial Bambus on different components

Research Plan

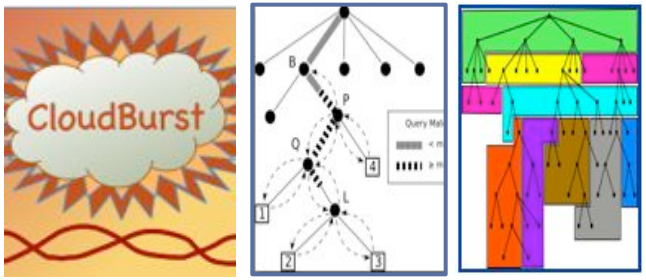
- Design & Implement Parallel Assembler
  - Trimming & Chimeric reads
  - ✓ de Bruijn graph construction
  - ✓ Path Compression
  - ✓ Error correction
  - Graph Simplifications
  - Connected components
  - Scaffolding: Bambus, Celera Assembler, Velvet
- Evaluate performance
  - Computational & Biological performance
  - Bacteria / Velvet
  - Human / ABySS
- Investigate MapReduce compatible algorithms
  - Randomized List Ranking (Anderson & Miller, 1990)
  - Priority Queues

(Chaisson et al., 2009)

Related Work : Methods

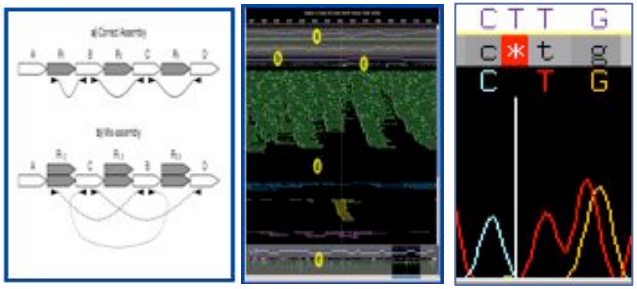
Parallel Short Read Mapping

- CloudBurst (**Schatz**, 2009)
- MUMmerGPU 2 (Trapnell, **Schatz**, 2009*)
- MUMmerGPU (**Schatz**, Trapnell, Delcher, Varshney, 2008)



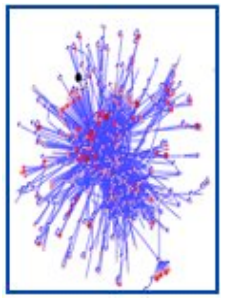
Assembly Correctness

- Assembly Forensics (Phillippy, **Schatz**, Pop, 2008)
- Hawkeye (**Schatz**, Phillippy, Shneiderman, Salzberg, 2007)
- AutoEditor (Gajer, **Schatz**, Salzberg, 2004)



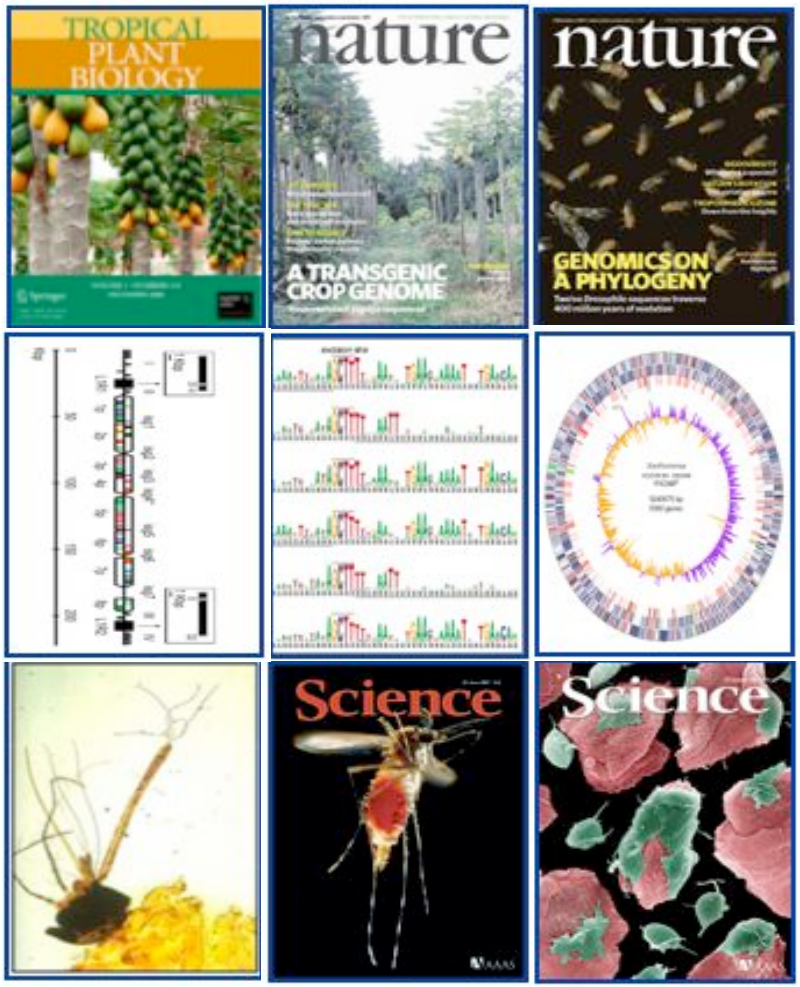
Machine Learning on PPI Networks

- Graph Summarization (Navlakha, **Schatz**, Kingsford, 2008)



* Submitted for publication

Related Work : Results



Genome Assembly & Analysis

- *N. ceranae* (Cornman et al, 2009*)
- *B. taurus* (Zimin et al, 2009*)
- *G. indiensis* (Desjardins et al, 2009)
- *C. papaya* (Ming et al, 2008),
- *C. papaya* (Suzuki et al, 2008)
- *X. oryzae* (Salzberg et al, 2008)
- *T. vaginas* (Carlton et al, 2007)
- Drosophila species (Drosophila 12 genomes consortium, 2007)
- *A. aegypti* (Nene et al, 2007)
- *B. malayi* (Ghedin et al, 2007)
- *G. indiensis* (Desjardins et al, 2007)
- Campylobacter species (Fouts et al, 2005)

* Submitted for publication

Grand Challenge of Biology

"NextGen sequencing has completely outrun the ability of good bioinformatics people to keep up with the data and use it well… We need a MASSIVE effort in the development of tools for "normal" biologists to make better use of massive sequence databases."

Jonathan Eisen – JGI Users Meeting – 3/28/09

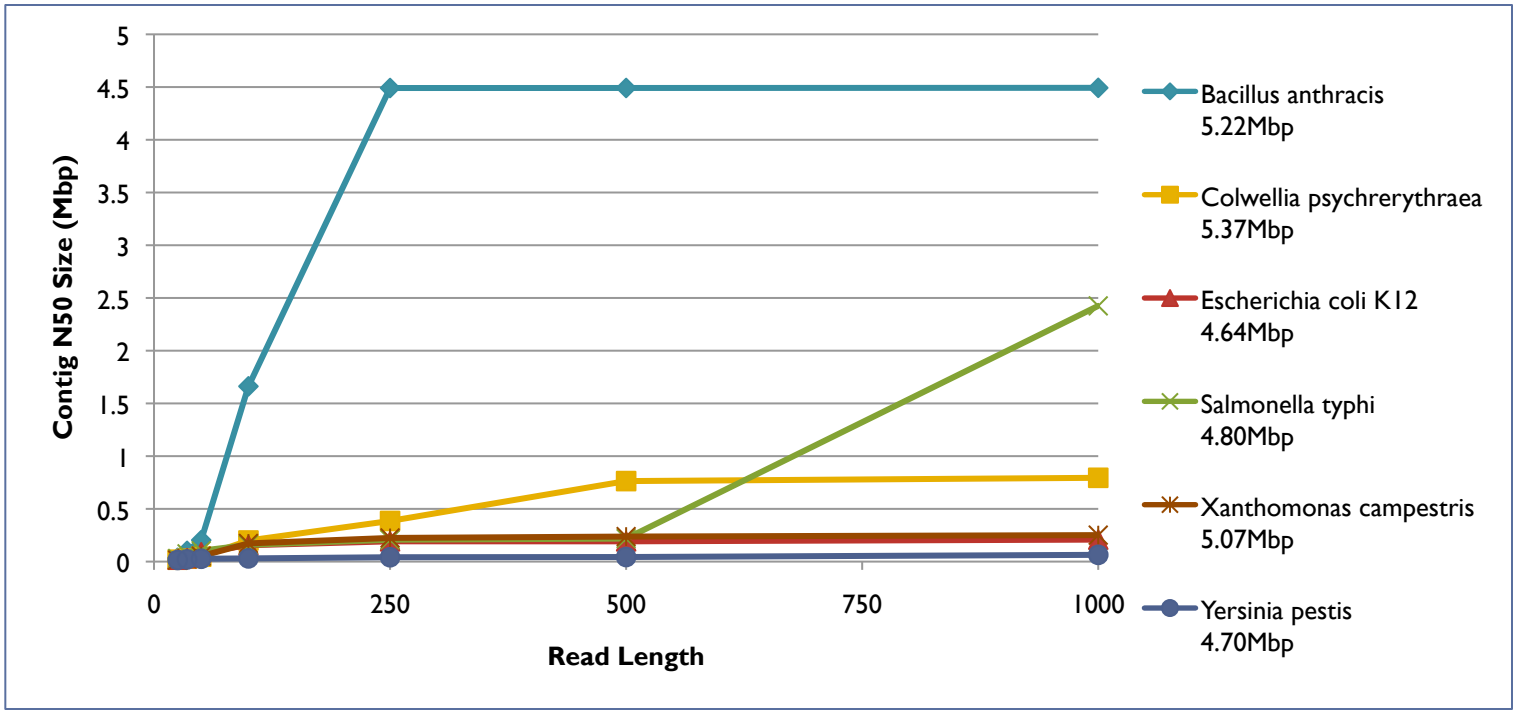
- Computational Biology
  - Make the problem of assembly of large genomes from short reads feasible and accessible to individual researchers
- High Performance Computing
  - Research Parallel Algorithms for MapReduce and Multicore systems

Acknowledgements

- Committee:
  - Steven Salzberg, Mihai Pop, Amitabh Varshney
- UMD Faculty
  - Art Delcher, Carl Kingsford, Ben Shneiderman, James Yorke, Jimmy Lin, Aleksey Zimin, Michael Roberts
- CBCB Students
  - Adam Phillippy, Cole Trapnell, Saket Navlakha, Ben Langmead, James White, Megan Smedinghoff

Thank You!

Short Reads and Mate-pairs



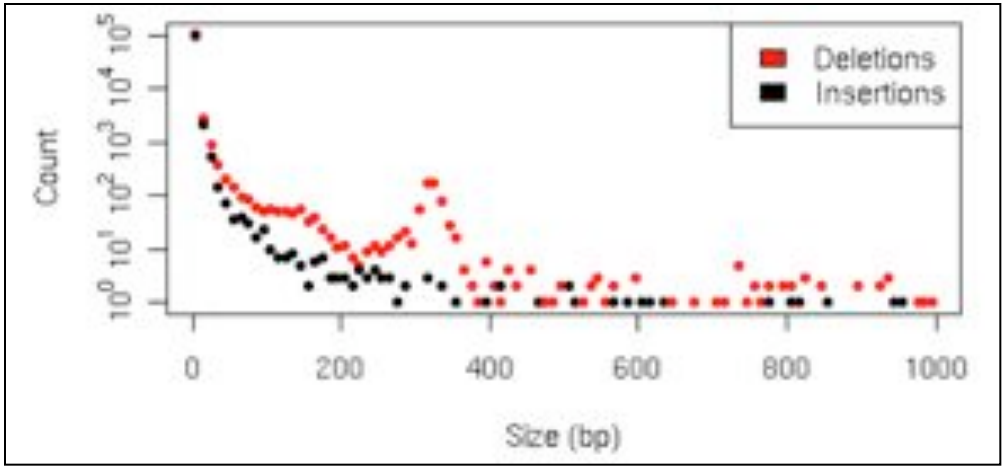
- Explore the relationship between read length and contig N50 size
  - Perfect reads, lengths: 25, 35, 50, 100, 250, 500, 1000
  - Long reads are limiting case for short mated reads, perfectly compute the insert sequence
  - (Kingsford, Schatz, Pop, 2009*)

ABySS Results

- Assemble 42x 36bp reads
- Mate pairs double the size of the contigs
  - Insert size 210bp
- Identify 100k insertions and deletions
  - Pronounced deletion peak corresponds to *Alu* family of retrotransposons

Table 3. Assembly statistics for data from the NA18507 Yoruba individual

Contig Statistics	$k = 27$, Without Paired-End Information		$k = 27$, With Paired-End Information	
	Contigs ≥100bp	Contigs ≥1,000bp	Contigs ≥100bp	Contigs ≥1,000bp
# Contigs	4,348,132	549,522	2,762,173	680,203
Median size (bp)	253	1,463	435	1,696
Mean size (bp)	484	1,703	791	2,093
Max. size (bp)	15,911	15,911	18,800	18,800
N50 size	870	1,731	1,499	2,282
# Contigs > N50	674,953	188,171	408,890	202,166
Sum (Gbp)	2.10	0.94	2.18	1.42



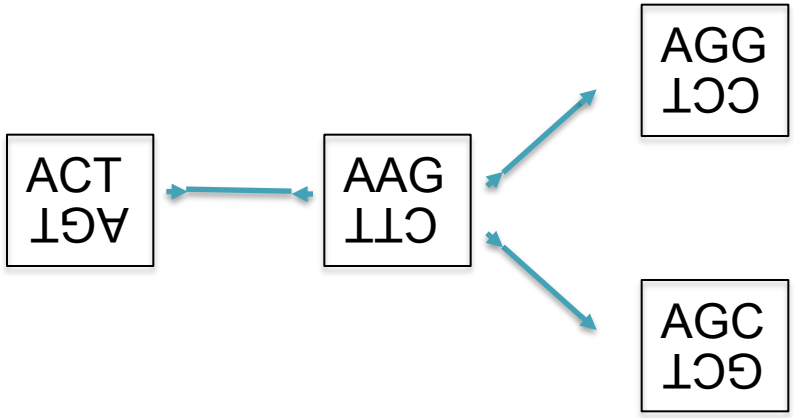
Bidirectional de Bruijn Graph

- Designate a representative mer for each mer/rc(mer) pair
  - Use the lexigraphically smaller mer

- Bidirected edges record if connection is between forward or reverse mer

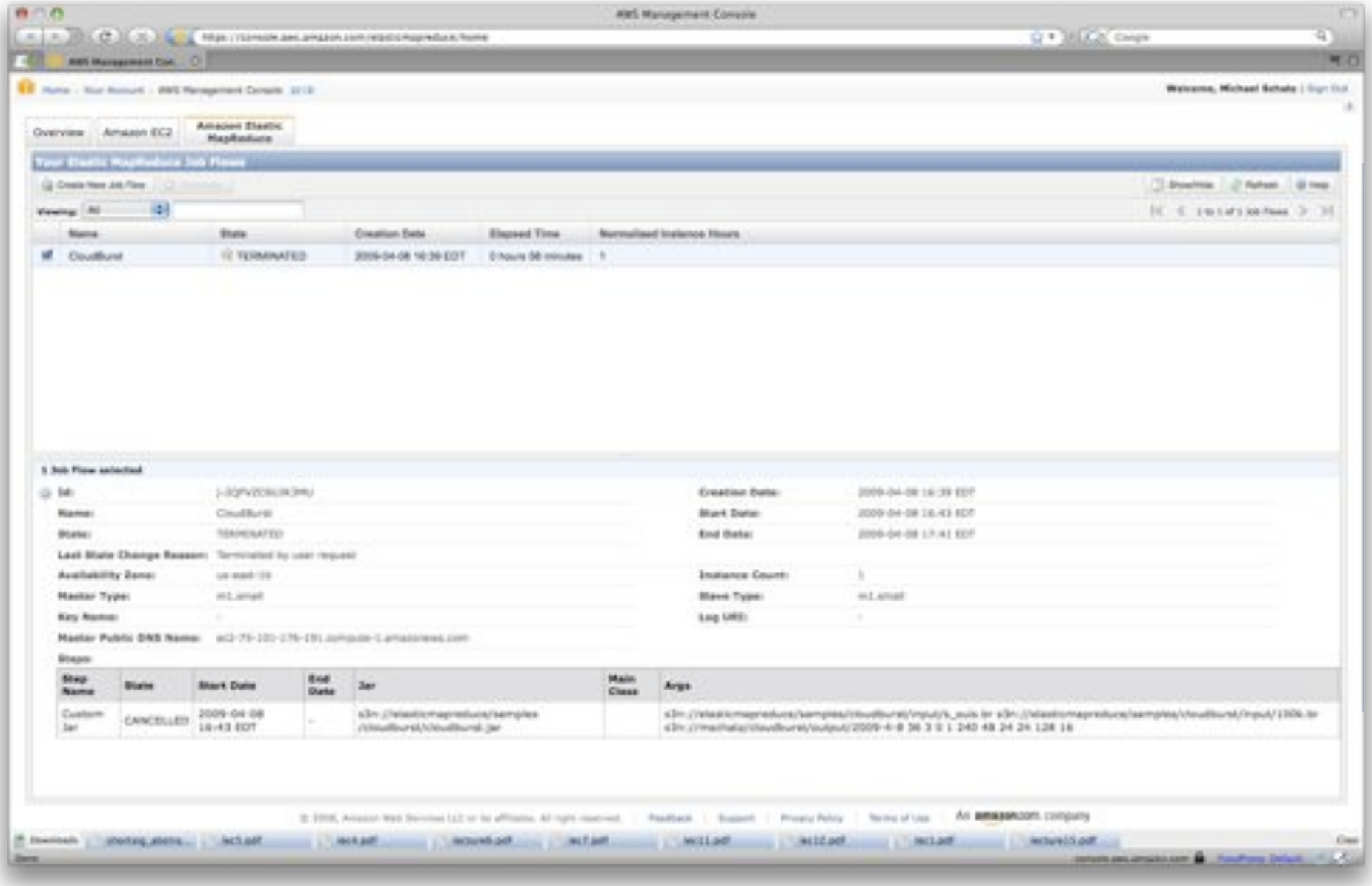
- In practice, keep separate adjacency lists for the forward and reverse mers

AAGG [CCTT]: AAG^{+} -> AGG^{+}
ACTT [AAGA]: ACT^{+} -> AAG^{-}
GCTT [AAGC]: AGC^{-} -> AAG^{-}
AAG^{+} -> AGC^{+}



(Medvedev et al, 2007)

Amazon Elastic MapReduce



EC2 Pricing

Pricing

Amazon Elastic MapReduce currently is available in the US region only. Pay only for what you use – there is no minimum fee. Amazon Elastic MapReduce pricing is in addition to normal Amazon EC2 and Amazon S3 pricing.

Standard Amazon EC2 Instances	Amazon EC2 Price per hour (On-Demand Instances)	Amazon Elastic MapReduce Price per hour
Small (Default)	$0.10 per hour	$0.015 per hour
Large	$0.40 per hour	$0.06 per hour
Extra Large	$0.80 per hour	$0.12 per hour
High CPU Instances	**Amazon EC2 Price per hour (On-Demand Instances)**	**Amazon Elastic MapReduce Price per hour**
Medium	$0.20 per hour	$0.03 per hour
Extra Large	$0.80 per hour	$0.12 per hour

Amazon EC2 and Amazon S3 charges are billed separately. Pricing for Amazon Elastic MapReduce is per instance-hour consumed for each instance type, from the time job flow began processing until it is terminated. Each partial instance-hour consumed will be billed as a full hour. For additional details on Amazon EC2 Instance Types, Amazon EC2 Reserved Instances Pricing, or Amazon S3 Pricing, follow the links below:

Amazon EC2 Instance Types

Amazon EC2 Reserved Instances Pricing

Amazon S3 Pricing

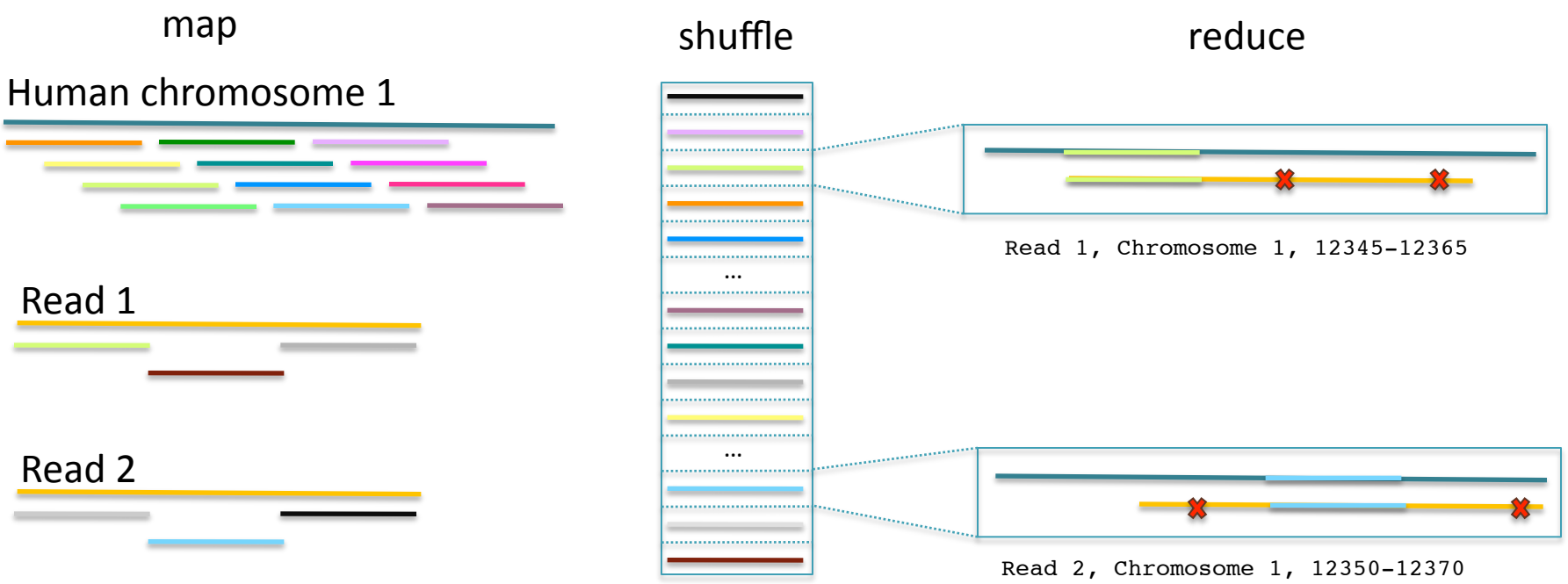
CloudBurst: Highly Sensitive Read Mapping with MapReduce

(Schatz, 2009)

1. Map: Catalog K-mers
   - Emit every k-mer in the genome and non-overlapping k-mers in the reads
   - Non-overlapping k-mers sufficient to guarantee an alignment will be found

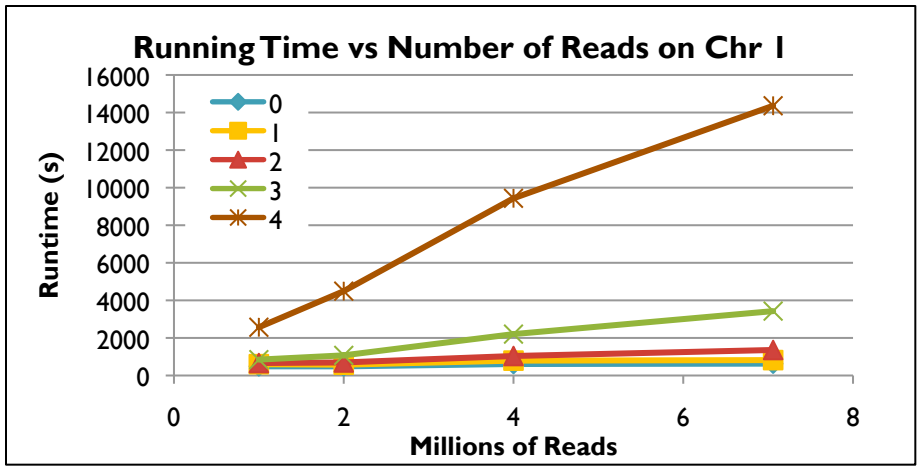
2. Shuffle: Coalesce Seeds
   - Hadoop internal shuffle groups together k-mers shared by the reads and the reference
   - Conceptually build a hash table of k-mers and their occurrences

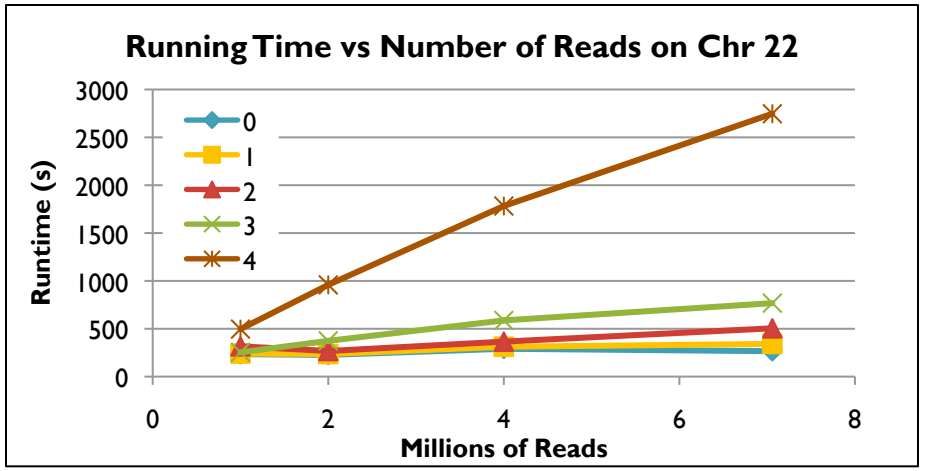
3. Reduce: End-to-end alignment
   - Locally extend alignment beyond seeds by counting mismatches, or with Landau-Vishkin k-difference algorithm to allow for indels.
   - If read aligns end-to-end, record the alignment



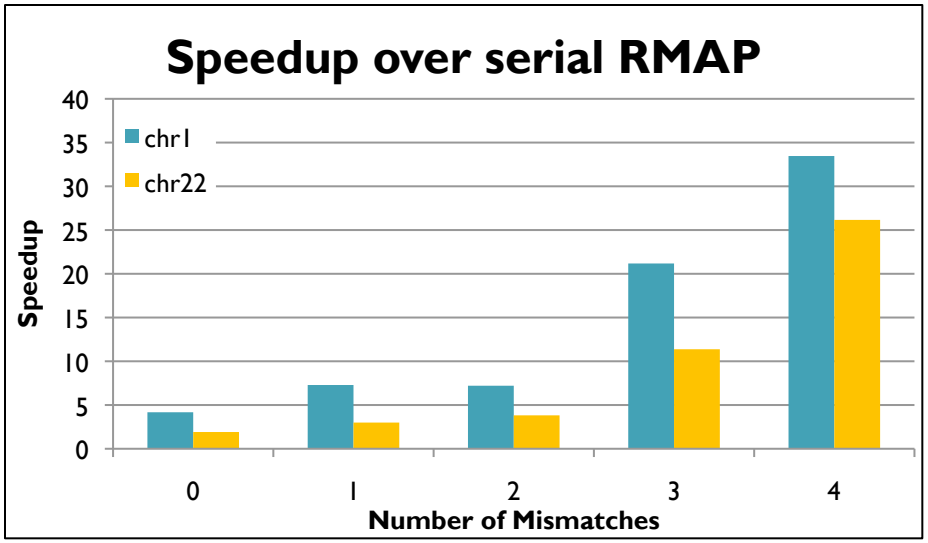
CloudBurst Results on Local CBCB Cluster

- Evaluation of CloudBurst running time while scaling the number of reads and the number of allowed mismatches while mapping to human chromosomes 1 (top) and 22 (bottom) on the local cluster with 24 cores.
- Colored lines indicate timings allowing 0 (fastest) through 4 (slowest) mismatches between a read and the reference.
- As the number of reads increases, the running time increases linearly.
- As the number of allowed mismatches increases, the running time increases super-linearly from the exponential increase in seed instances.





Comparison to RMAP

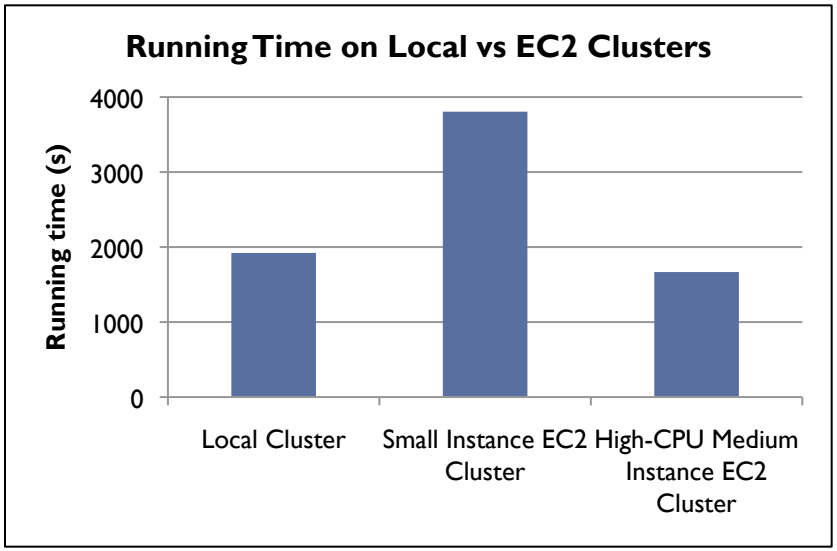


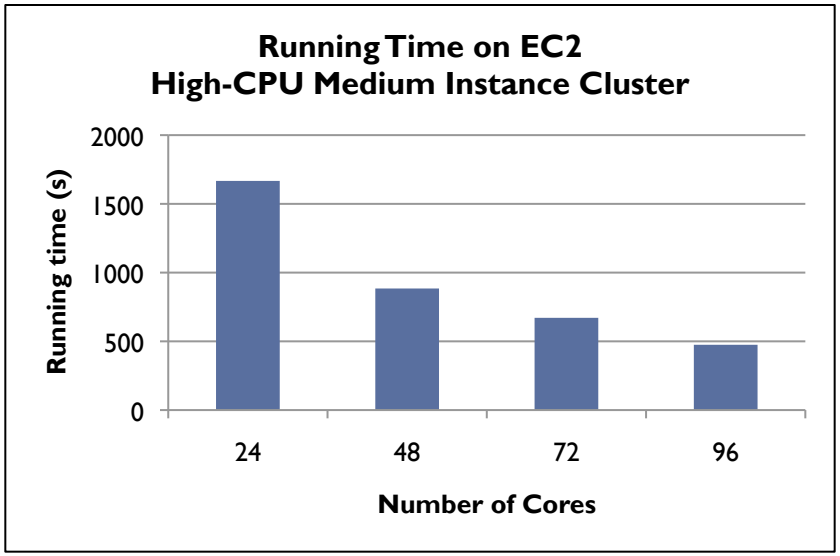
- CloudBurst running time compared to *RMAP* for mapping 7M reads, showing the speedup of CloudBurst running on 24 cores compared to *RMAP* running on 1 core.

- As the number of allowed mismatches increases, the relative overhead decreases allowing CloudBurst to meet and exceed 24x linear speedup.

- Produces identical results in a fraction of the time, especially for highly sensitive alignments.

Amazon 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.
- As the number of cores increase, the running time decreases with near linear speedup. The 96-core cluster is 3.5x faster than the 24-core, and 100x faster than a serial run of RMAP.