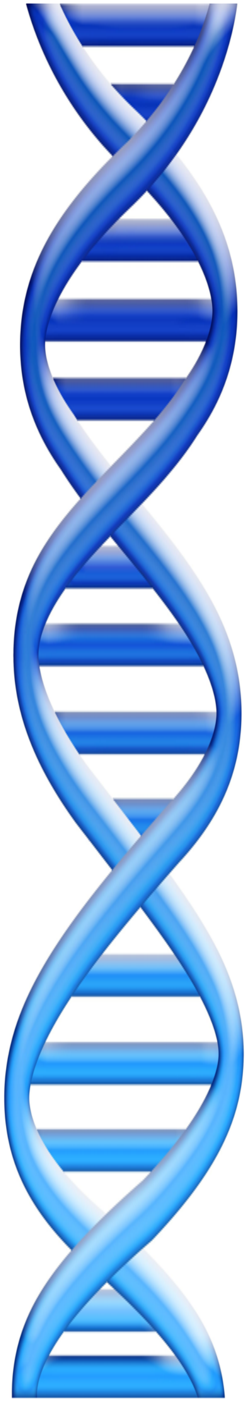
Whole Genome Assembly and Alignment

Michael Schatz

Nov 14, 2011

CSHL Sequencing Course

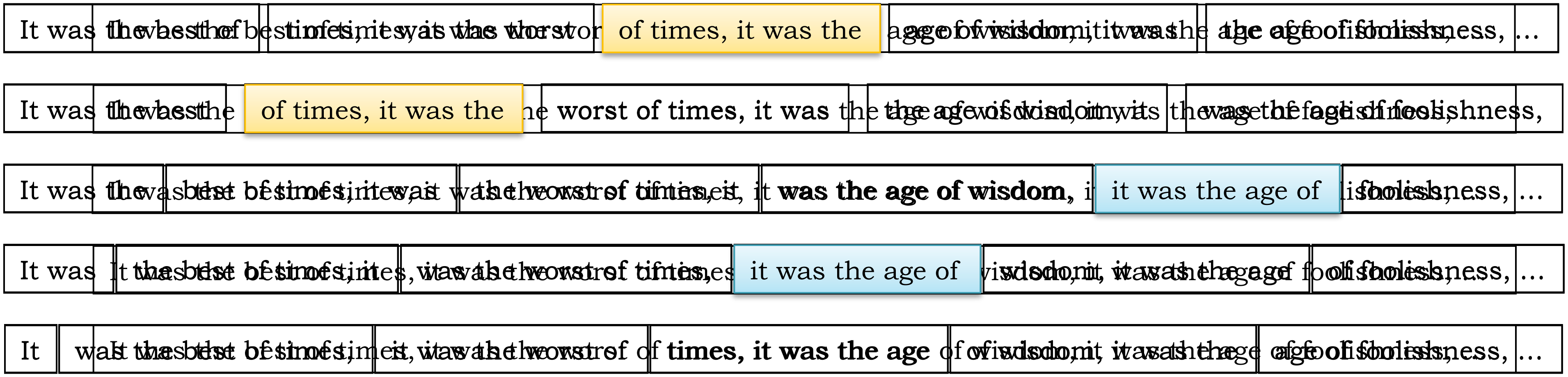


Outline

1. Assembly theory
   1. Assembly by analogy
   2. De Bruijn and Overlap graph
   3. Coverage, read length, repeats, and errors
2. Genome assemblers
   1. ALLPATHS-LG
   2. SOAPdenovo
   3. Celera Assembler
3. Whole Genome Alignment with MUMmer

Shredded Book Reconstruction

- Dickens accidentally shreds the first printing of 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

Greedy Reconstruction

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,

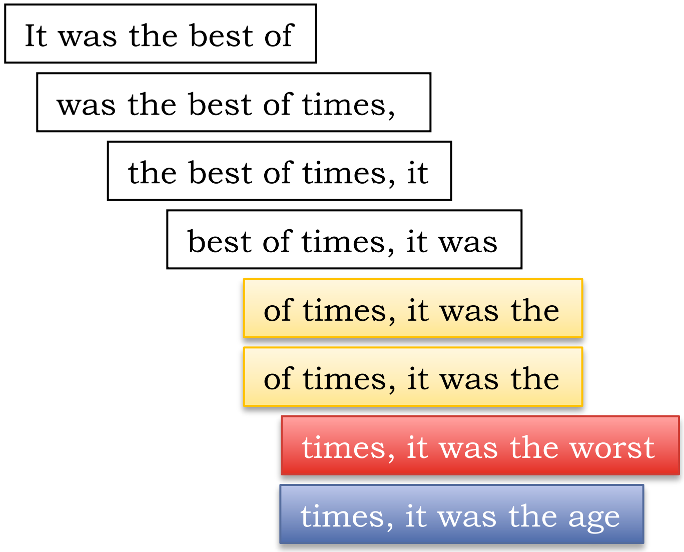
was the age of foolishness,

was the best of times,

was the worst of times,

wisdom, it was the age

worst of times, it was



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

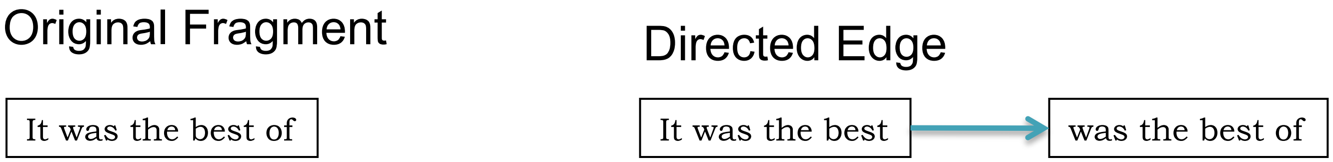
The repeated sequence make the correct reconstruction ambiguous

- It was the best of times, it was the [worst/age]

Model the assembly 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



Original Fragment

It was the best of

Directed Edge

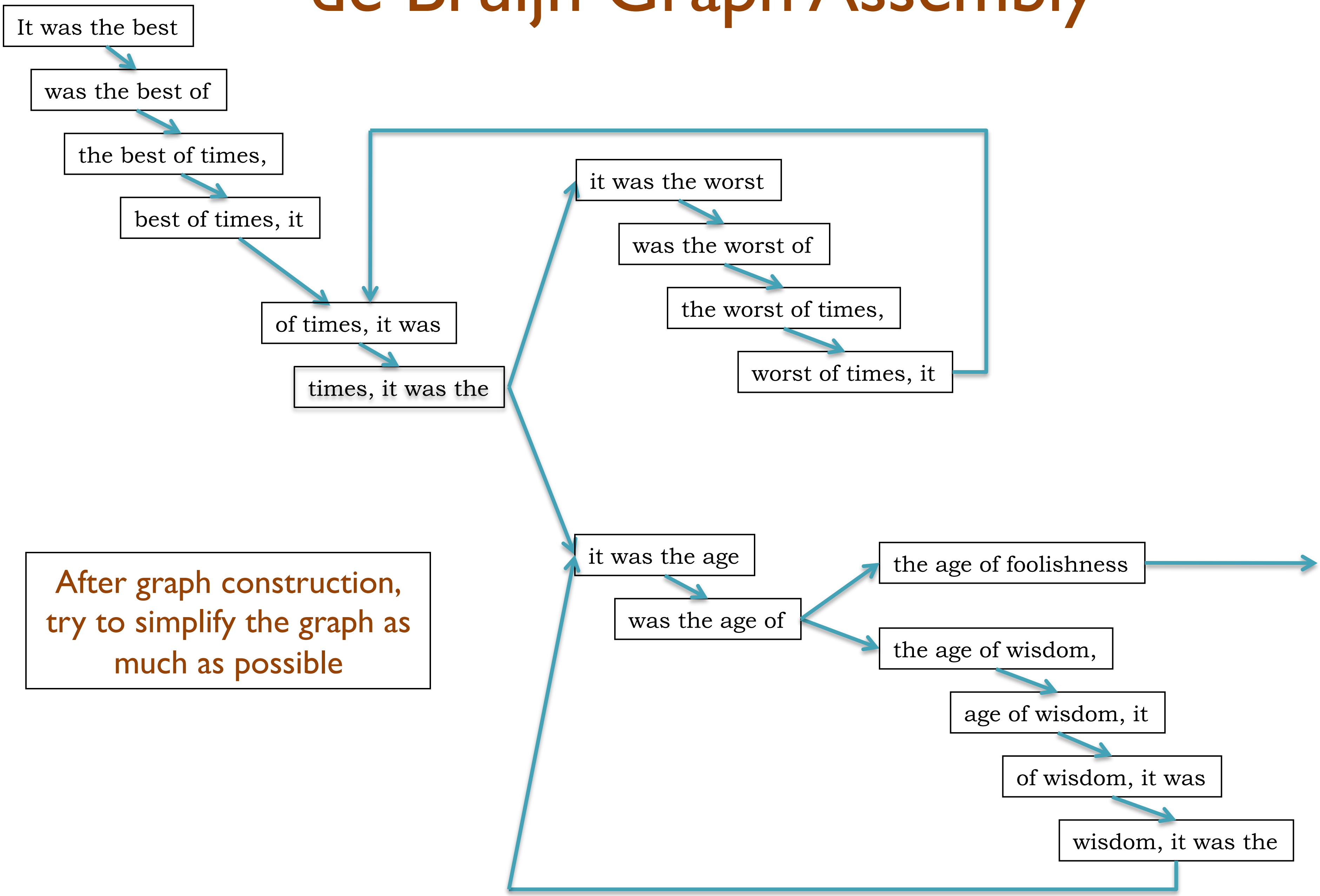
It was the best

was the best of

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

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

de Bruijn Graph Assembly



It was the best

was the best of

the best of times,

best of times, it

of times, it was

times, it was the

it was the worst

was the worst of

the worst of times,

worst of times, it

it was the age

was the age of

the age of foolishness

the age of wisdom,

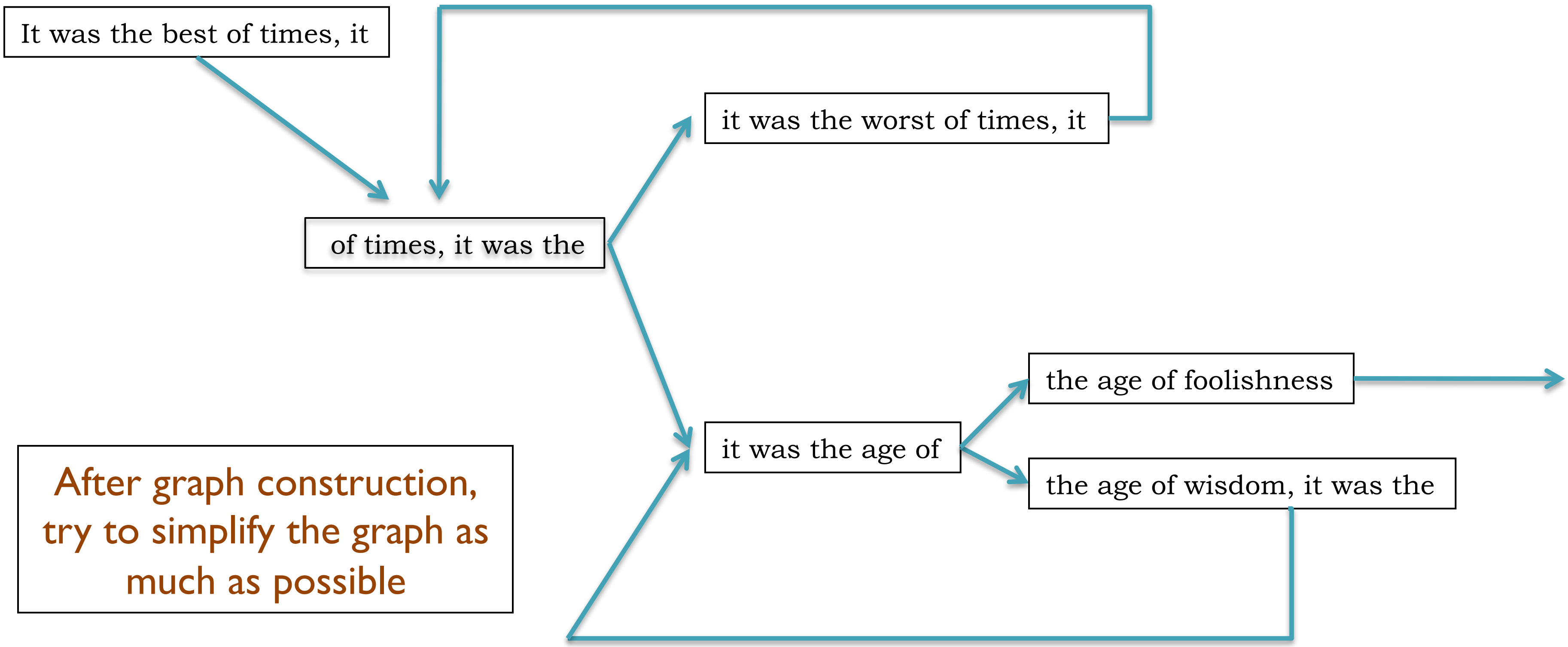
age of wisdom, it

of wisdom, it was

wisdom, it was the

After graph construction, try to simplify the graph as much as possible

de Bruijn Graph Assembly



It was the best of times, it

of times, it was the

it was the worst of times, it

it was the age of

the age of foolishness

the age of wisdom, it was the

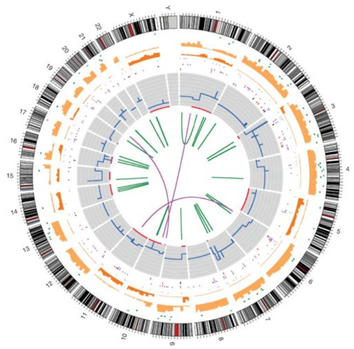
After graph construction, try to simplify the graph as much as possible

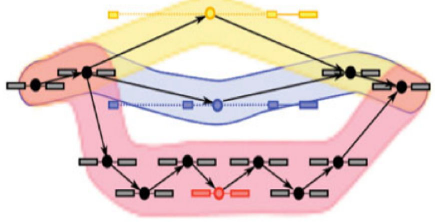
Assembly Applications

- Novel genomes

- Metagenomes

- Sequencing assays
  - – Structural variations
  - – Transcript assembly
  - – …

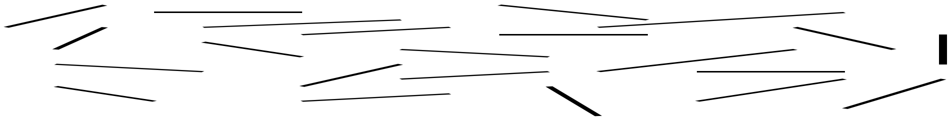




Like Dickens, we must computationally reconstruct a genome from short fragments

Assembling a Genome

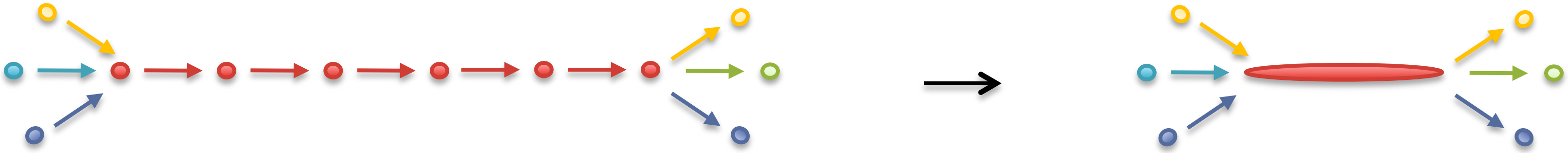
1. Shear & Sequence DNA



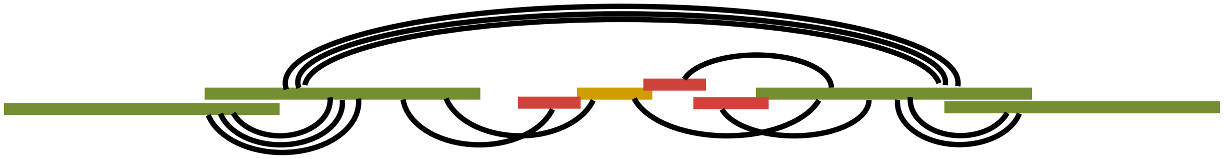
2. Construct assembly graph from overlapping reads

…AGCCTAGACCTACAGGATGCGCGACACGT
GGATGCGCGACACGTCGCATATCCGGT…

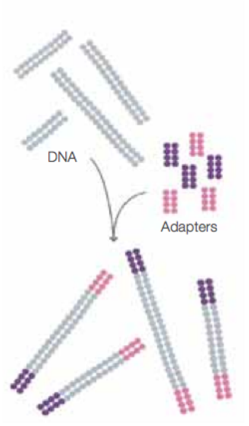
3. Simplify assembly graph



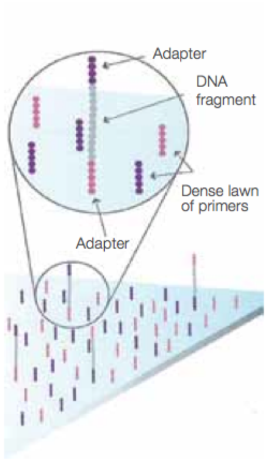
4. Detangle graph with long reads, mates, and other links



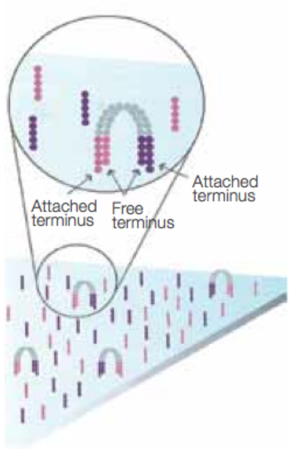
Illumina Sequencing by Synthesis



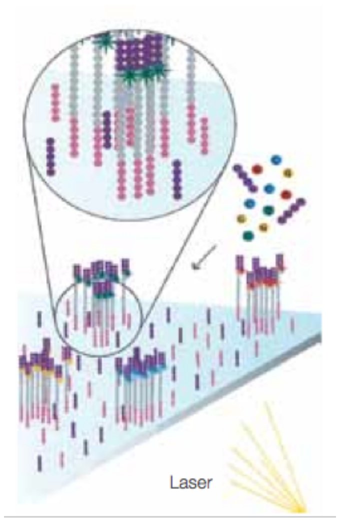
1. Prepare



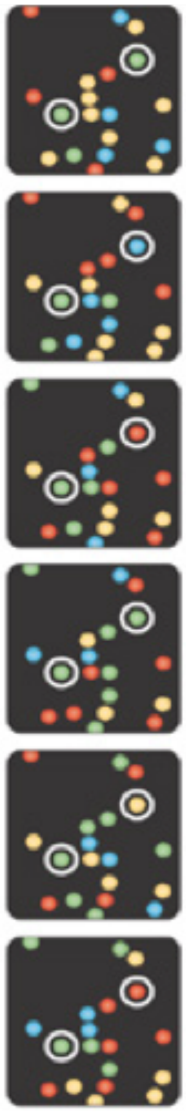
2. Attach



3. Amplify



4. Image



5. Basecall

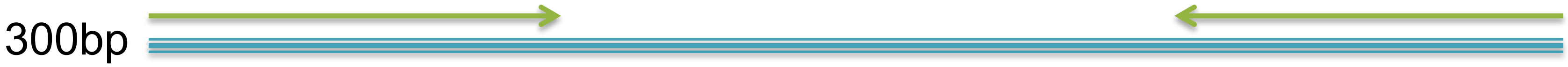
Metzker (2010) Nature Reviews Genetics 11:31-46

http://www.illumina.com/documents/products/techspotlights/techspotlight_sequencing.pdf

Paired-end and Mate-pairs

Paired-end sequencing

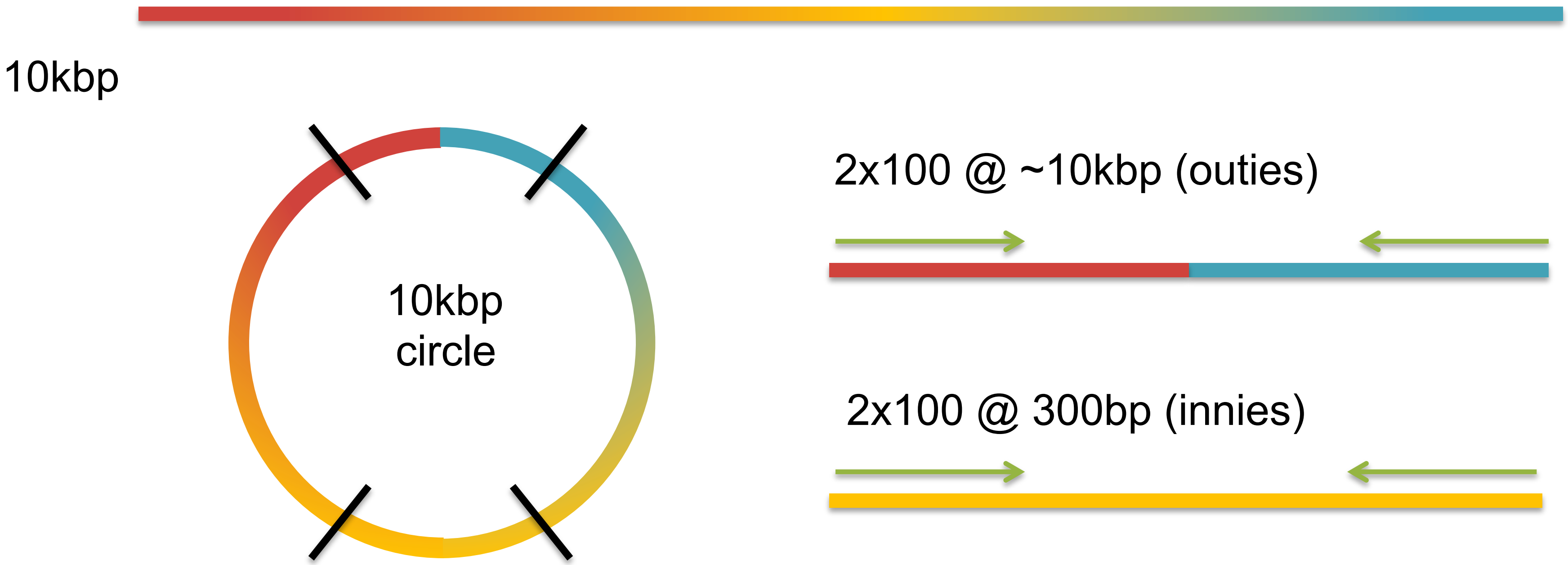
- Read one end of the molecule, flip, and read the other end
- Generate pair of reads separated by up to 500bp with inward orientation



300bp

Mate-pair sequencing

- Circularize long molecules (1-10kbp), shear into fragments, & sequence
- Mate failures create short paired-end reads



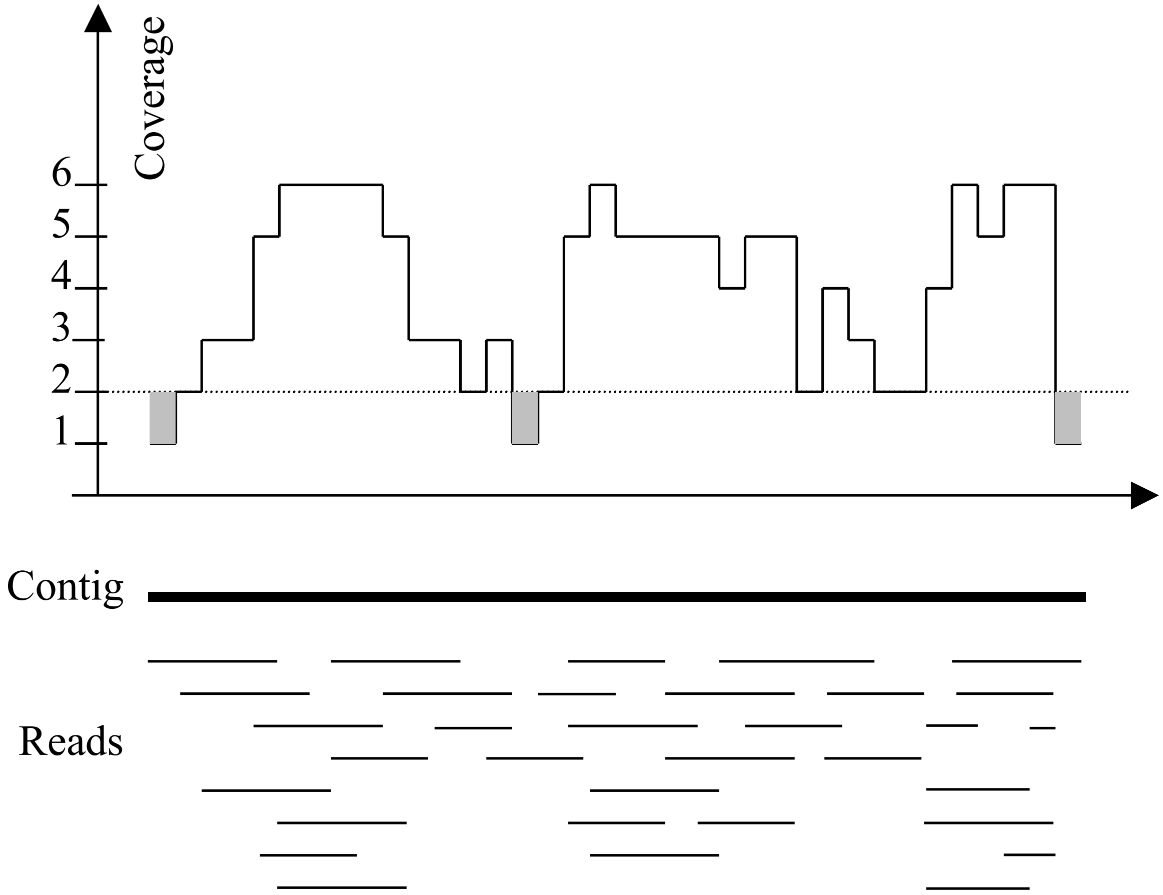
10kbp

10kbp
circle

2x100 @ ~10kbp (outies)

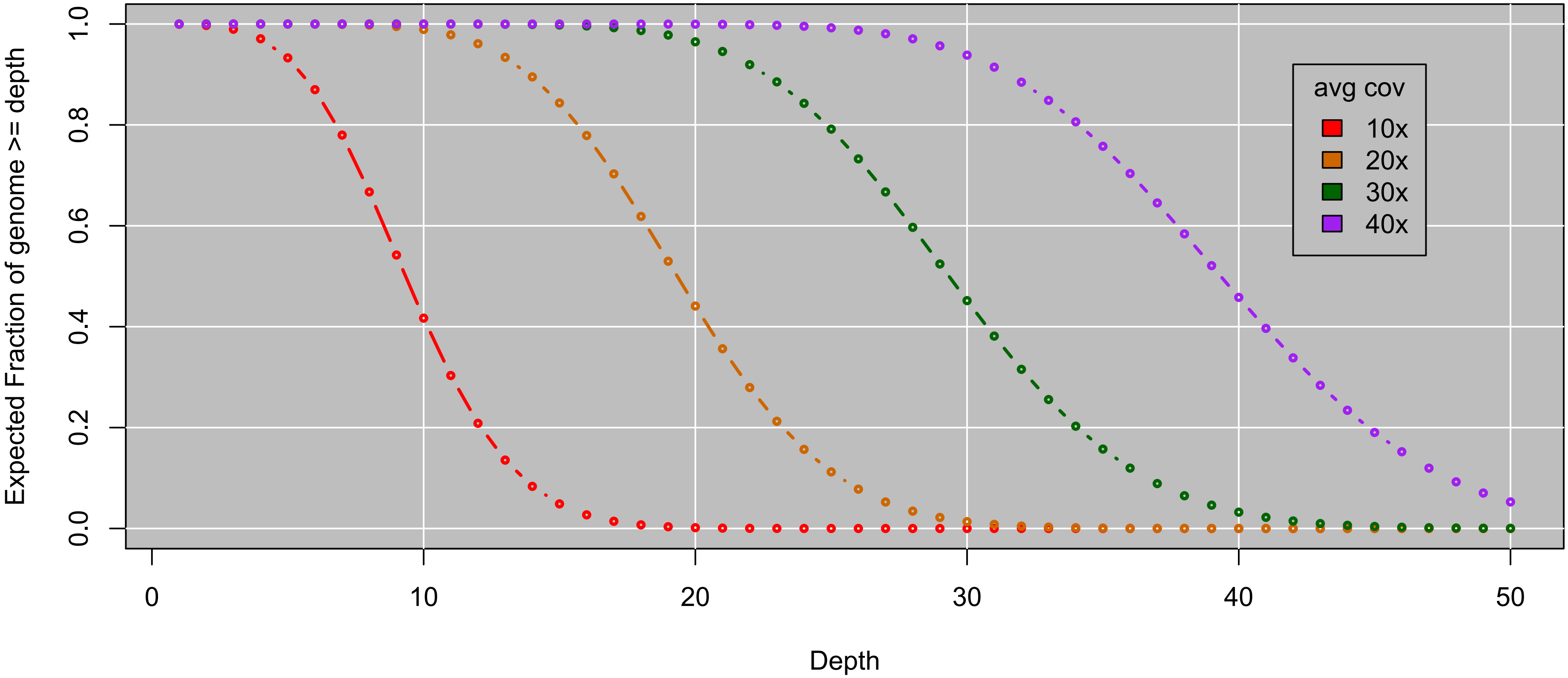
2x100 @ 300bp (innies)

Typical contig coverage



Imagine raindrops on a sidewalk

Genome Coverage Distribution

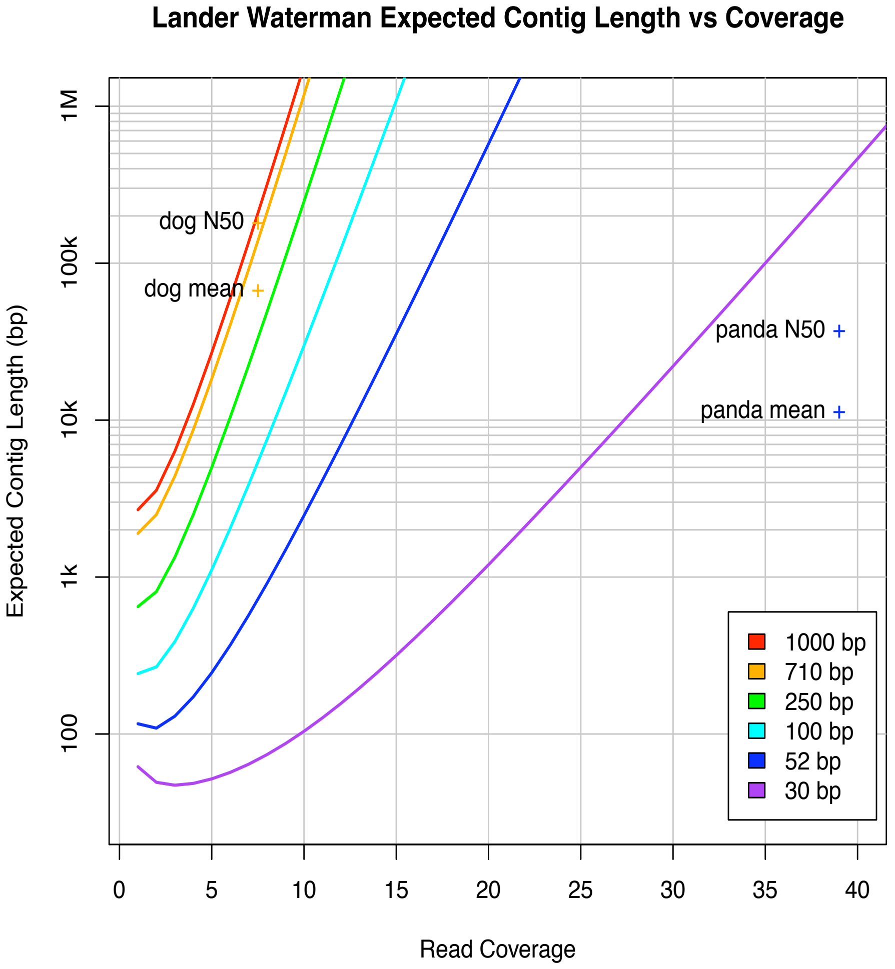


This is the mathematically model => reality may be much worse

Coverage and Read Length

Idealized Lander-Waterman model

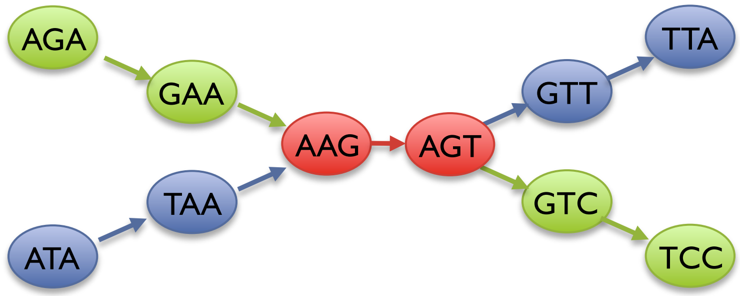
- Reads start at perfectly random positions
- Poisson distribution in coverage
  - Contigs end when there are no overlapping reads
- Contig length is a function of coverage and read length
  - Effective coverage reduced by *o/l*
  - Short reads require much higher coverage to reach same expected contig length



Assembly of Large Genomes using Second Generation Sequencing
Schatz MC, Delcher AL, Salzberg SL (2010) *Genome Research.* 20:1165-1173.

Two Paradigms for Assembly

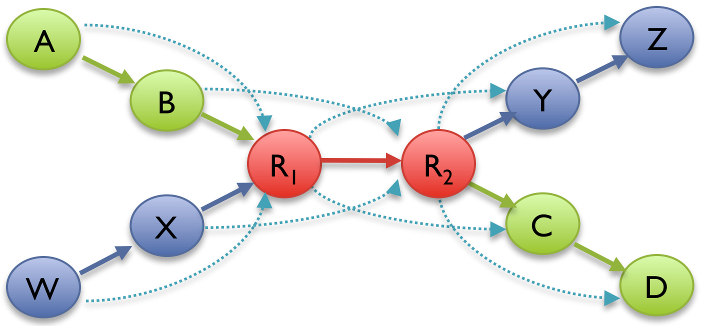
de Bruijn Graph



Short read assemblers

- Repeats depends on word length
- Read coherency, placements lost
- Robust to high coverage

Overlap Graph



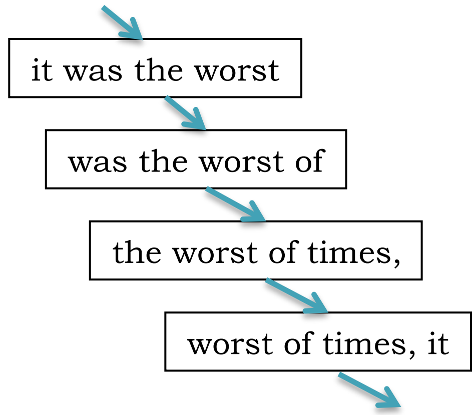
Long read assemblers

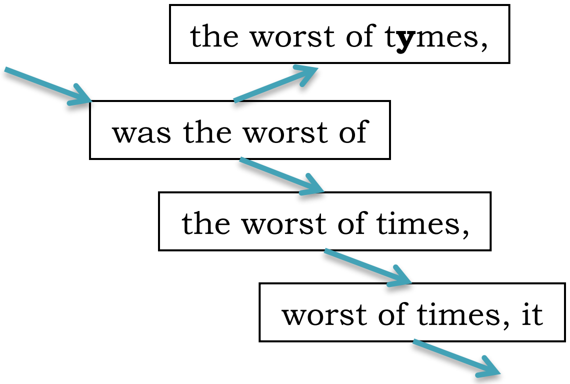
- Repeats depends on read length
- Read coherency, placements kept
- Tangled by high coverage

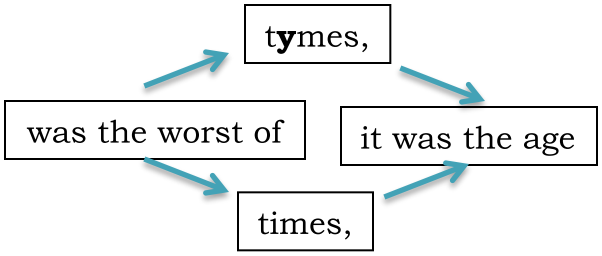
Assembly of Large Genomes using Second Generation Sequencing
Schatz MC, Delcher AL, Salzberg SL (2010) *Genome Research.* 20:1165-1173.

Simplifications and Corrections

Path Compression	Clip Tips	Pop Bubbles
it was the worst of was the worst of times, the worst of times, it	was the worst of times, was the worst of t**y**mes, the worst of times, it	was the worst of times, was the worst of t**y**mes, times, it was the age t**y**mes, it was the age

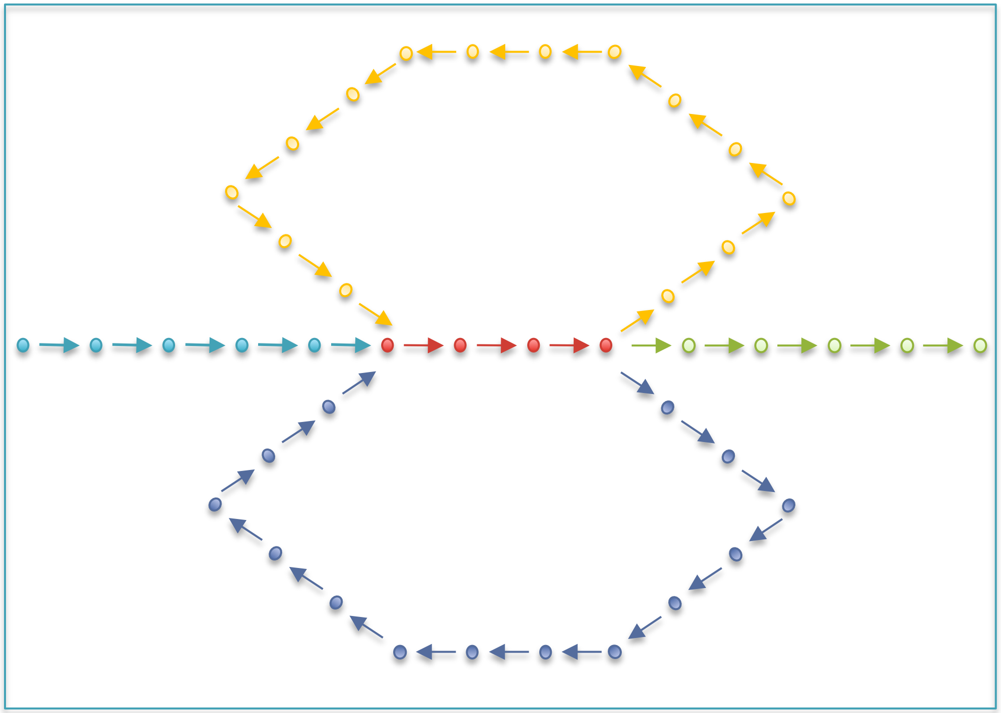


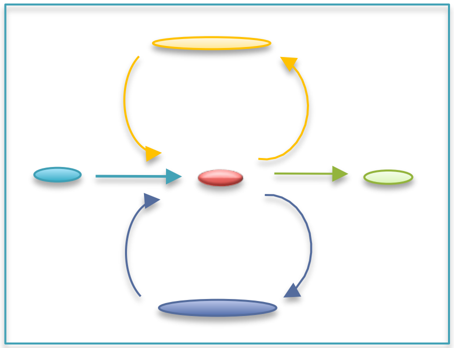




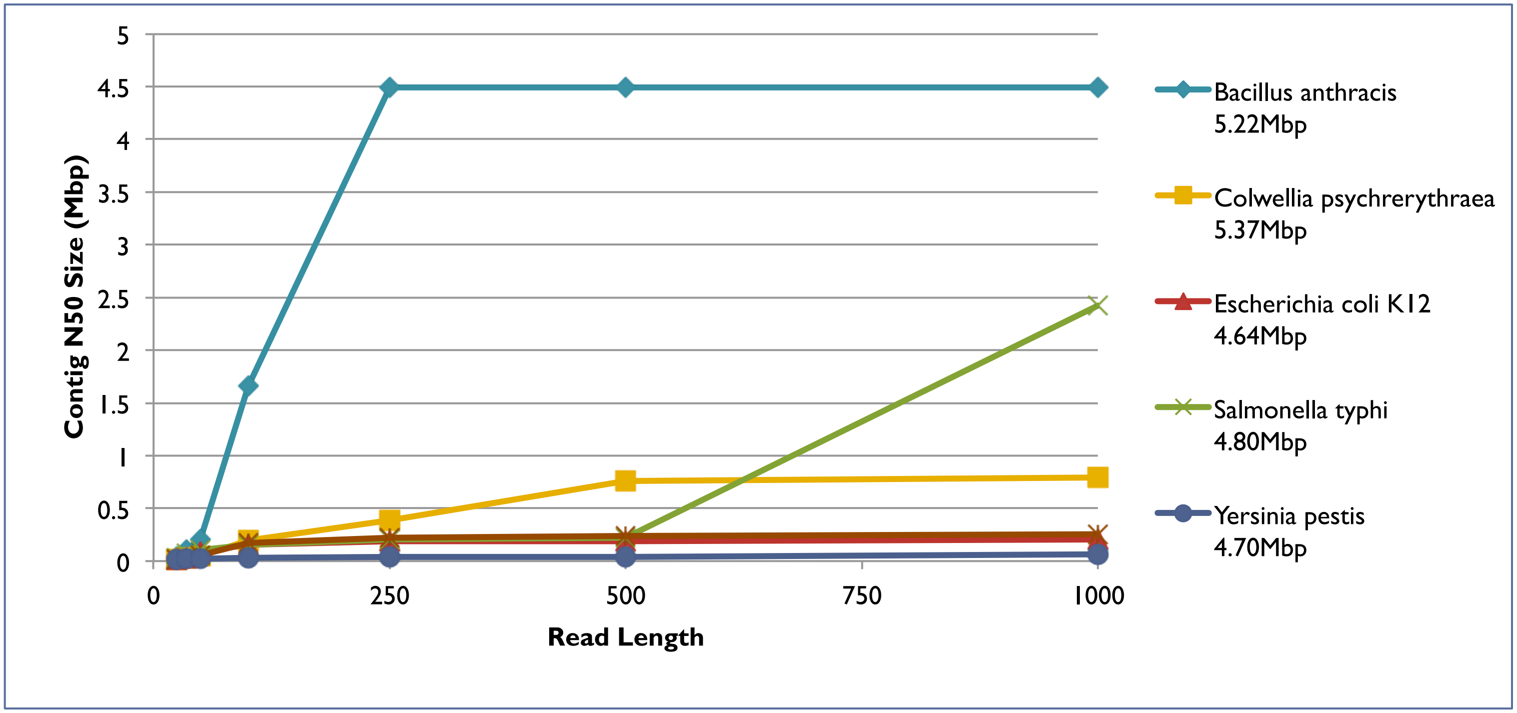
Initial Contigs

- After simplification and correction, compress graph down to its non-branching initial contigs
  - Aka "unitigs", "unipaths"





Repeats and Read Length



- Explore the relationship between read length and contig N50 size
  - Idealized assembly of read lengths: 25, 35, 50, 100, 250, 500, 1000
  - Contig/Read length relationship depends on specific repeat composition

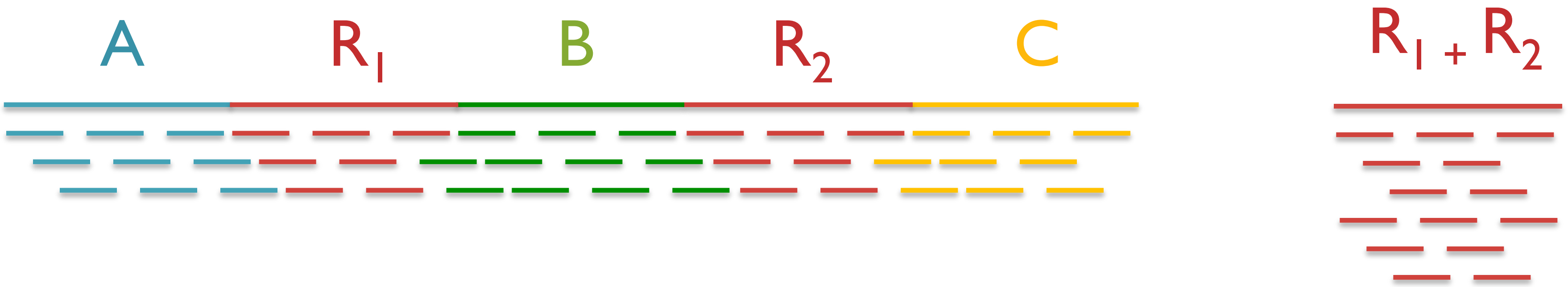
Assembly Complexity of Prokaryotic Genomes using Short Reads.
Kingsford C, Schatz MC, Pop M (2010) *BMC Bioinformatics*. 11:21.

Repetitive regions

- Over 50% of the human genome is repetitive

Repeat Type	Definition / Example	Prevalence
Low-complexity DNA / Microsatellites	$(b_1b_2\ldots b_k)^N$ where $1 \leq k \leq 6$ CACACACACACACACACA	2%
SINEs (Short Interspersed Nuclear Elements)	*Alu* sequence (~280 bp) Mariner elements (~80 bp)	13%
LINEs (Long Interspersed Nuclear Elements)	~500 – 5,000 bp	21%
LTR (long terminal repeat) retrotransposons	Ty1-copia, Ty3-gypsy, Pao-BEL (~100 – 5,000 bp)	8%
Other DNA transposons		3%
Gene families & segmental duplications		4%

Repeats and Coverage Statistics



- If *n* reads are a uniform random sample of the genome of length *G*, we expect *k*=*n Δ* /*G* reads to start in a region of length *Δ* .
  - If we see many more reads than k (if the arrival rate is > A) , it is likely to be a collapsed repeat
  - Requires an accurate genome size estimate

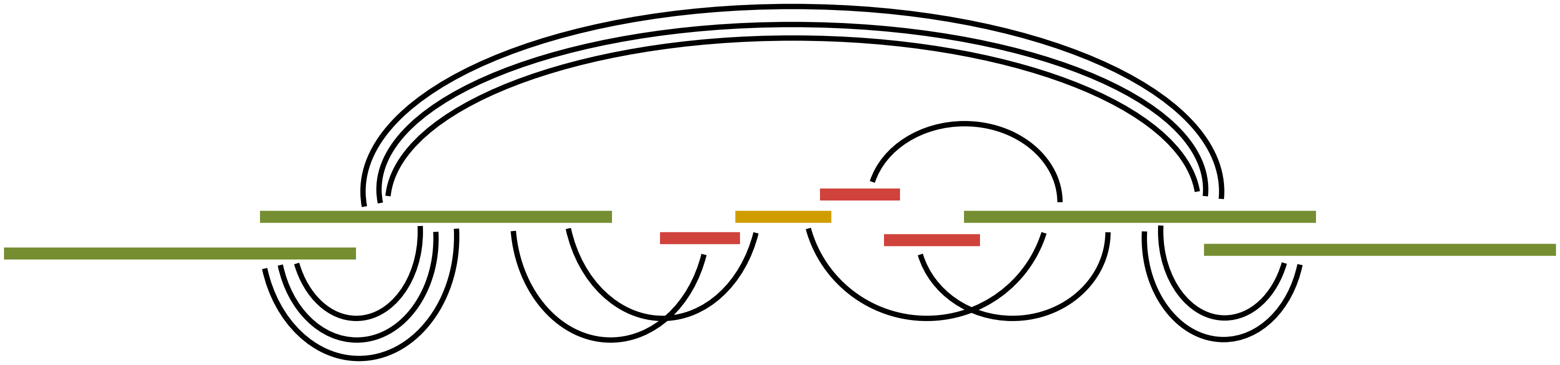
$$\Pr(X-copy) = \binom{n}{k}\left(\frac{X\Delta}{G}\right)^{k}\left(\frac{G-X\Delta}{G}\right)^{n-k}$$

$$A(\Delta,k) = \ln\left(\frac{\Pr(1-copy)}{\Pr(2-copy)}\right) = \ln\left(\frac{\frac{(\Delta n/G)^{k}}{k!}e^{\frac{-\Delta n}{G}}}{\frac{(2\Delta n/G)^{k}}{k!}e^{\frac{-2\Delta n}{G}}}\right) = \frac{n\Delta}{G} - k\ln 2$$

Scaffolding

- Initial contigs (*aka* unipaths, unitigs) terminate at
  - *Coverage gaps*: especially extreme GC regions
  - *Conflicts*: sequencing errors, repeat boundaries

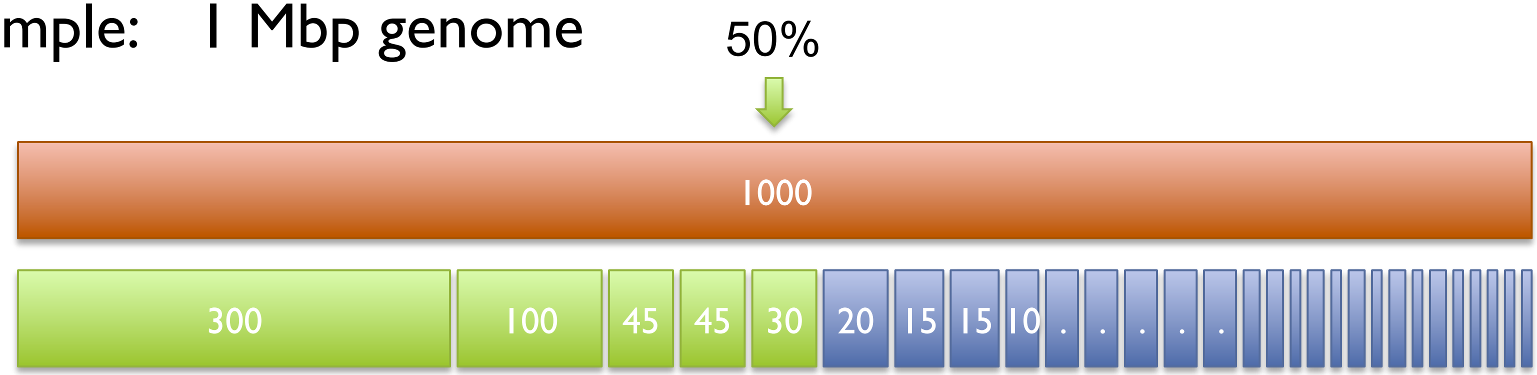
- Iteratively resolve longest, 'most unique' contigs
  - Both overlap graph and de Bruijn assemblers initially collapse repeats into single copies
  - Uniqueness measured by a statistical test on coverage



N50 size

Def: 50% of the genome is in contigs larger than N50

Example: 1 Mbp genome



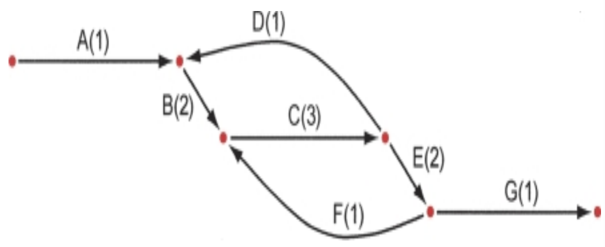
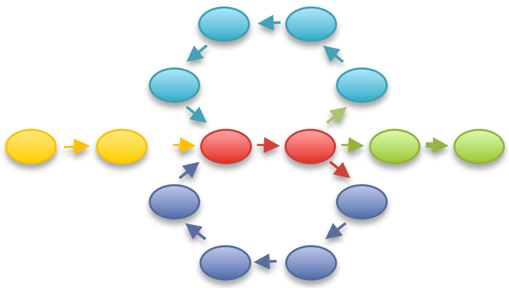
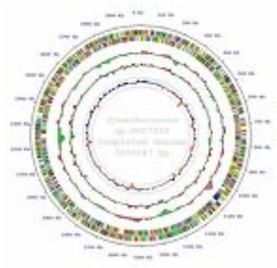
N50 size = 30 kbp

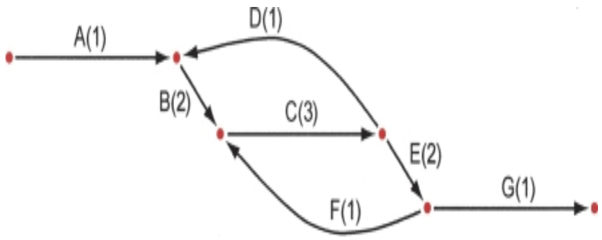
(300k+100k+45k+45k+30k = 520k >= 500kbp)

Note:

N50 values are only meaningful to compare when base genome size is the same in all cases

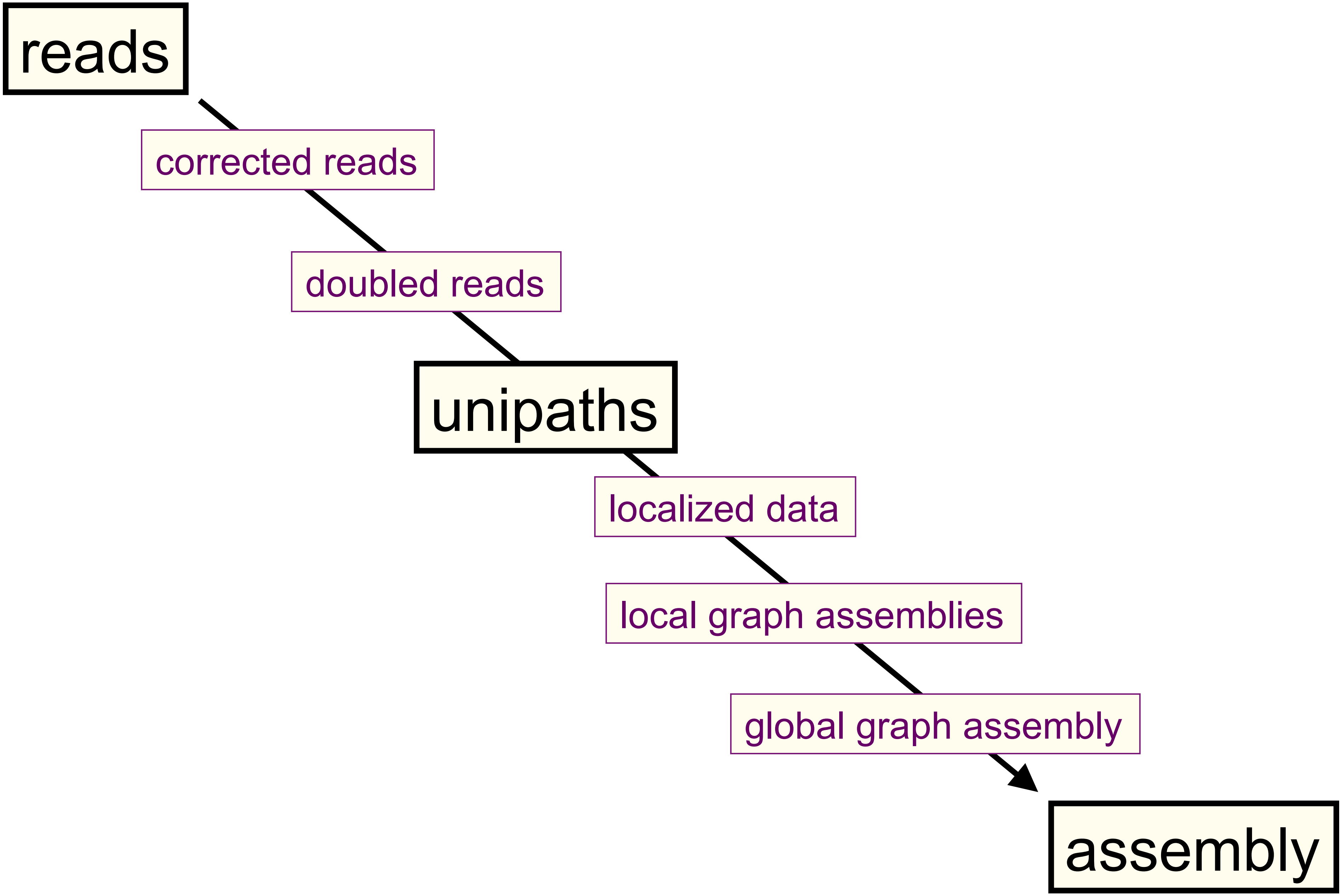
Assembly Algorithms

ALLPATHS-LG	SOAPdenovo	Celera Assembler
		
Broad's assembler (Gnerre et al. 2011)	BGI's assembler (Li et al. 2010)	JCVI's assembler (Miller et al. 2008)
De bruijn graph Short + PacBio (patching)	De bruijn graph Short reads	Overlap graph Medium + Long reads
Easy to run if you have compatible libraries	Most flexible, but requires a lot of tuning	Supports Illumina/454/PacBio Hybrid assemblies
http://www.broadinstitute.org/software/allpaths-lg/blog/	http://soap.genomics.org.cn/soapdenovo.html	http://wgs-assembler.sf.net



Genome assembly with ALLPATHS-LG
Iain MacCallum

How ALLPATHS-LG works



reads

corrected reads

doubled reads

unipaths

localized data

local graph assemblies

global graph assembly

assembly

ALLPATHS-LG sequencing model

Libraries (insert types)	Fragment size (bp)	Read length (bases)	Sequence coverage (x)	Required
Fragment	180*	≥ 100	45	yes
Short jump	3,000	≥ 100 preferable	45	yes
Long jump	6,000	≥ 100 preferable	5	no**
Fosmid jump	40,000	≥ 26	1	no**

*See next slide.

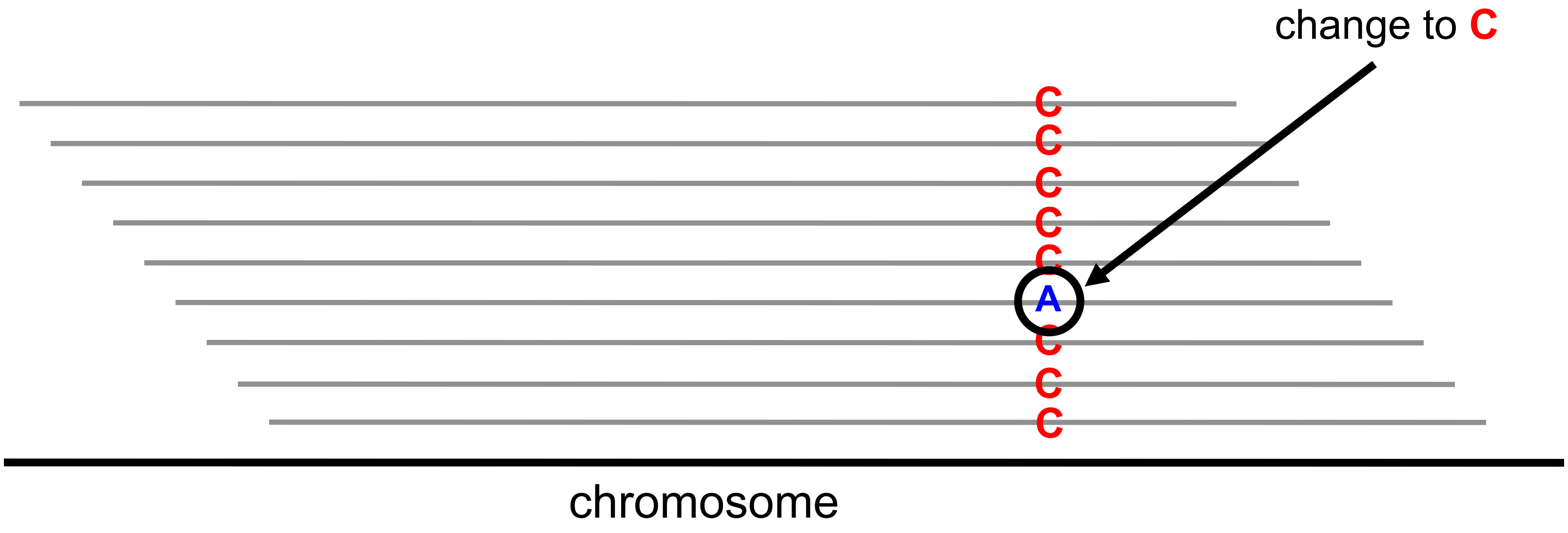
**For best results. Normally not used for small genomes. However essential to assemble long repeats or duplications.

Cutting coverage in half still works, with some reduction in quality of results.

All: protocols are either available, or in progress.

Error correction

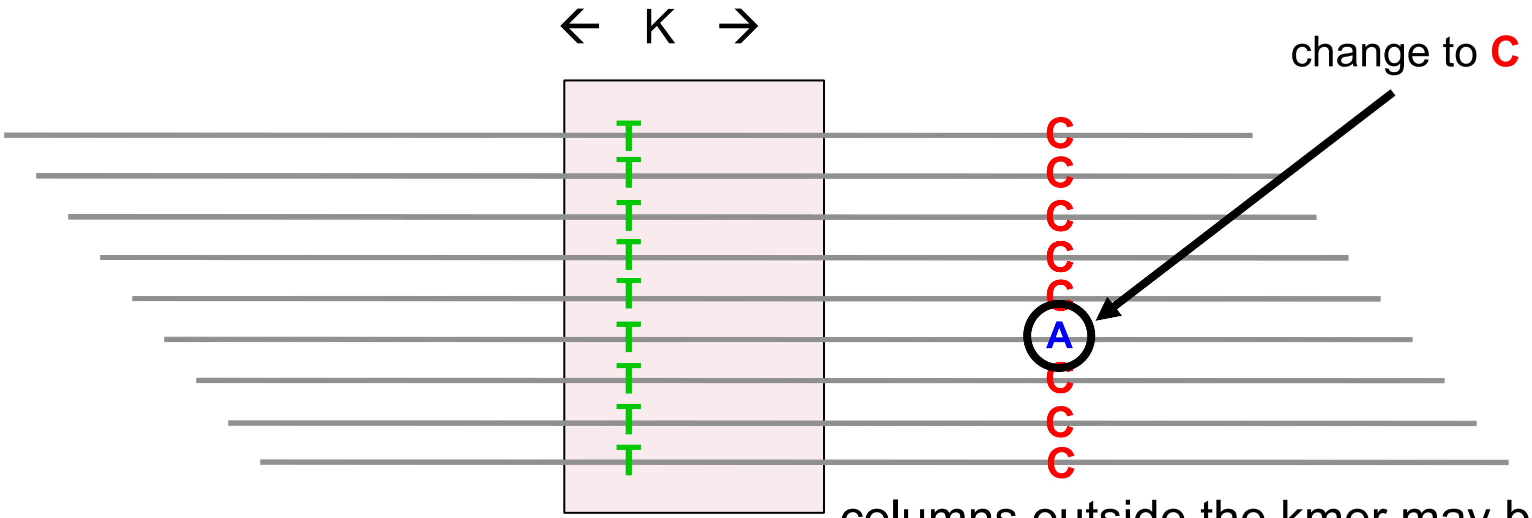
Given a crystal ball, we could stack reads on the chromosomes they came from (with homologous chromosomes separate), then let each column 'vote':



But we don't have a crystal ball....

Error correction

ALLPATHS-LG. For every K-mer, examine the stack of all reads containing the K-mer. Individual reads may be edited if they differ from the overwhelming consensus of the stack. If a given base on a read receives conflicting votes (arising from membership of the read in multiple stacks), it is not changed. (K=24)



change to **C**

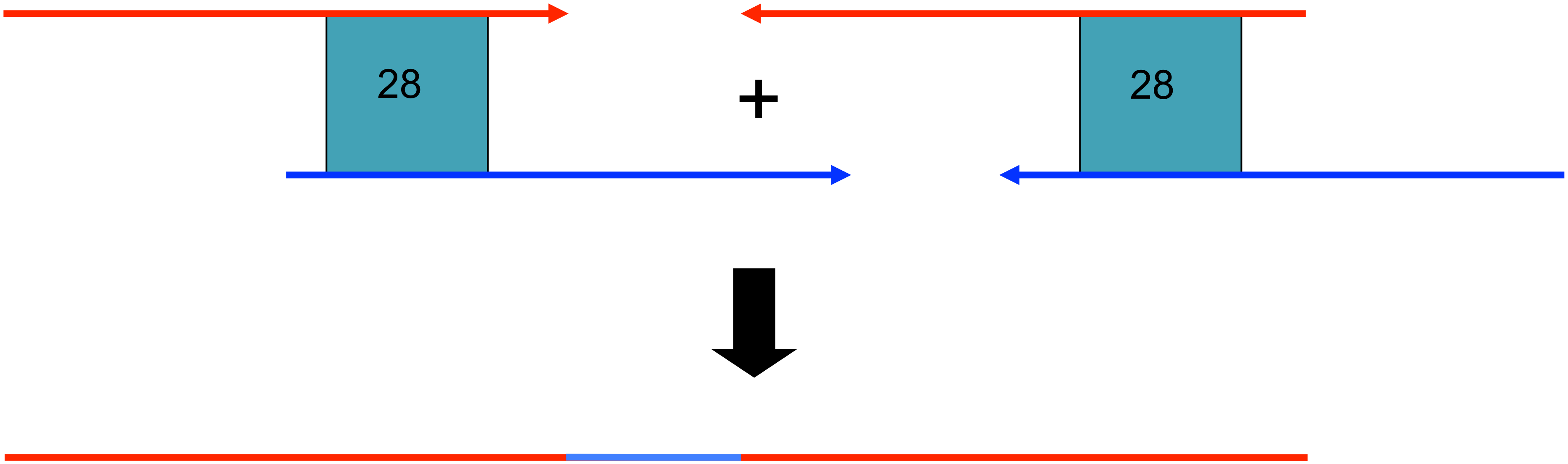
columns outside the kmer may be mixed

columns inside the kmer are homogeneous

Two calls at Q20 or better are enough to protect a base

Read doubling

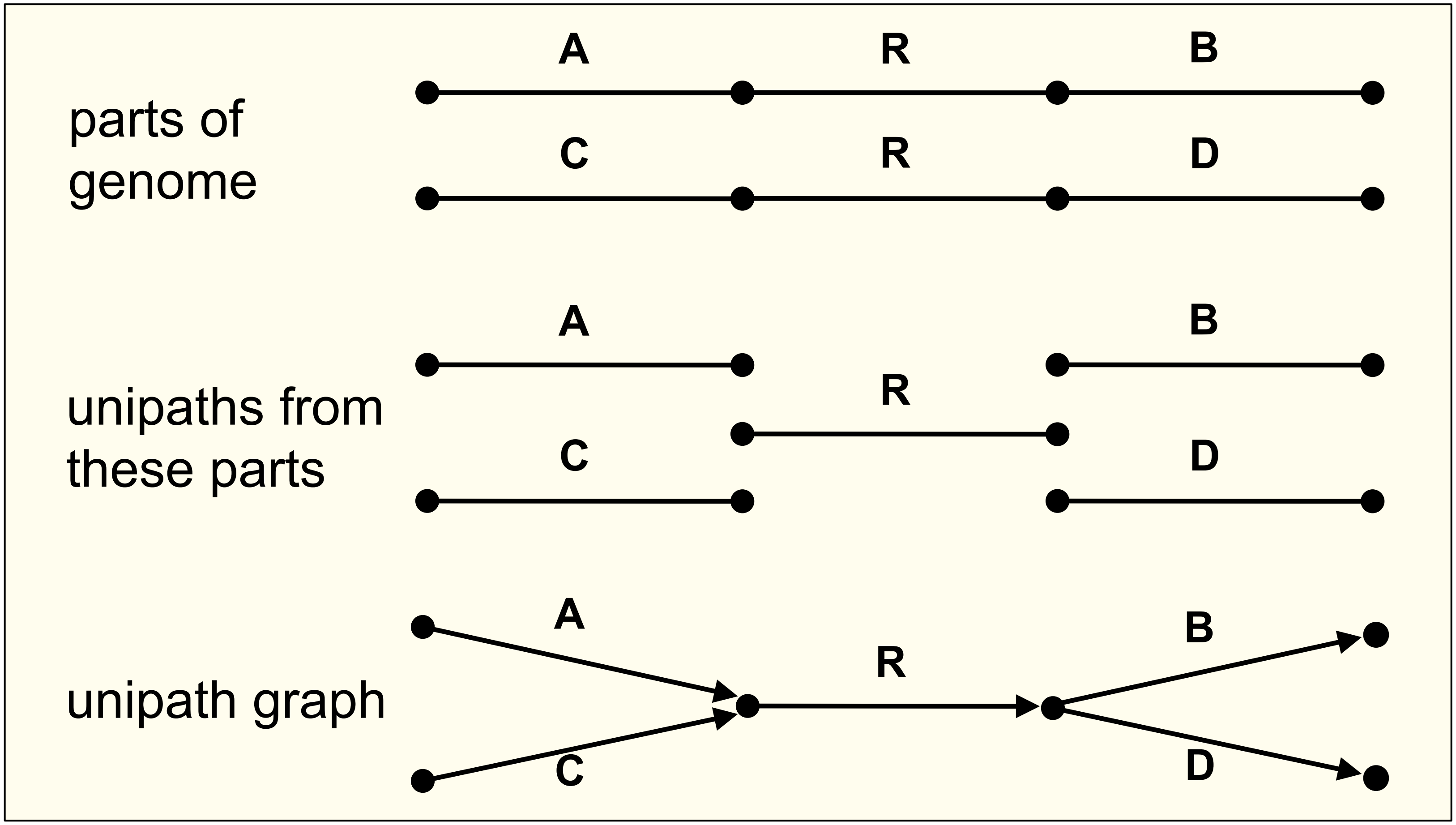
To close a read pair (red), we require the existence of another read pair (blue), overlapping perfectly like this:



More than one closure allowed (but rare).

Unipaths

Unipath: unbranched part of genome – squeeze together perfect repeats of size ≥ K



parts of genome

unipaths from these parts

unipath graph

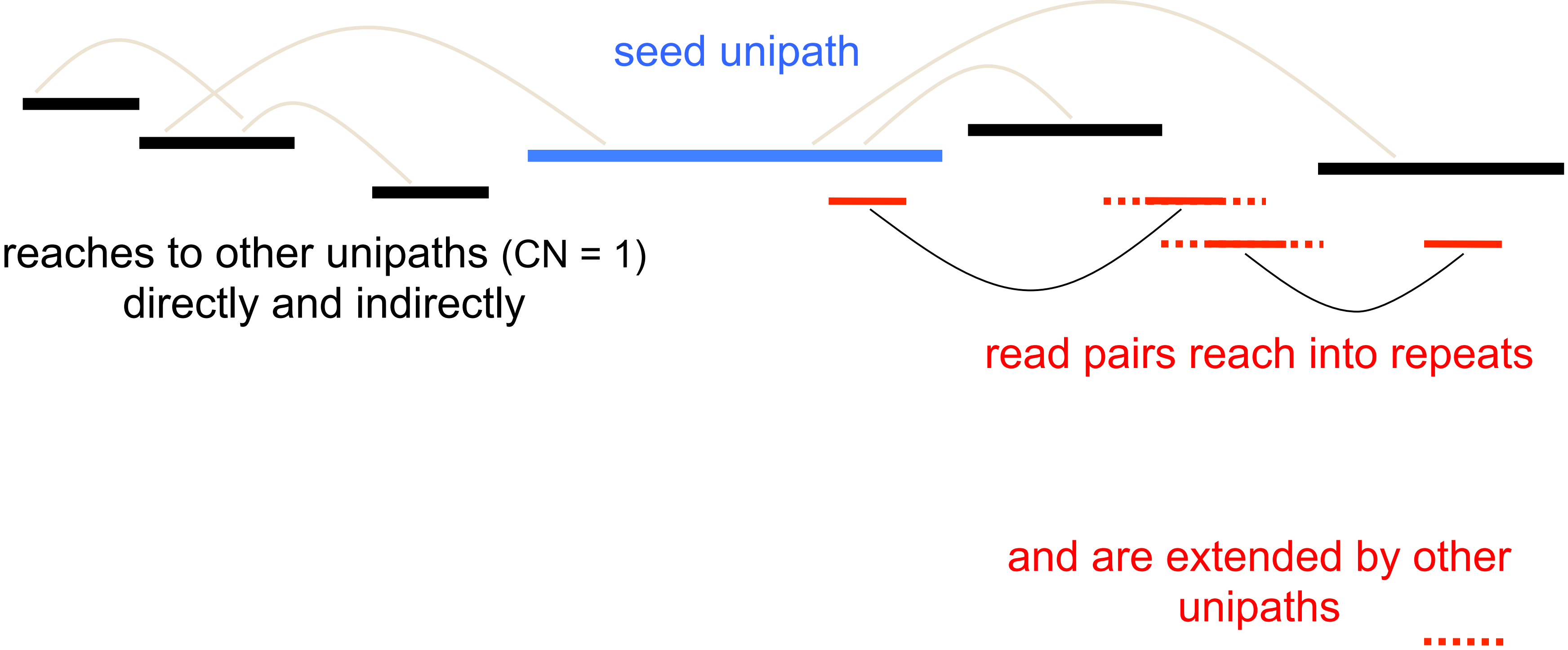
Adjacent unipaths overlap by K-1 bases

Localization

I. Find ‘seed’ unipaths, evenly spaced across genome

(ideally long, of copy number CN = 1)

II. Form neighborhood around each seed



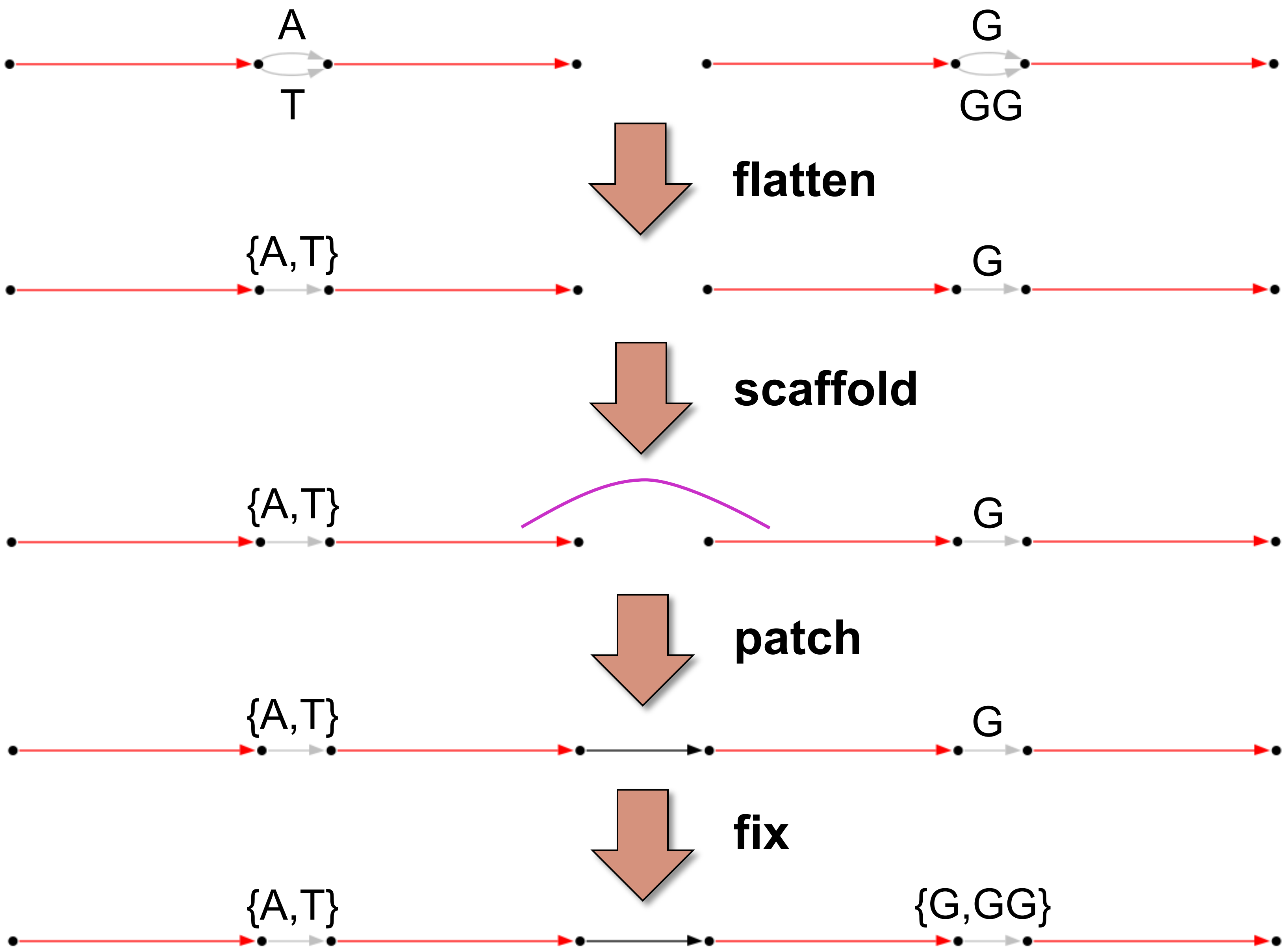
seed unipath

reaches to other unipaths (CN = 1)
directly and indirectly

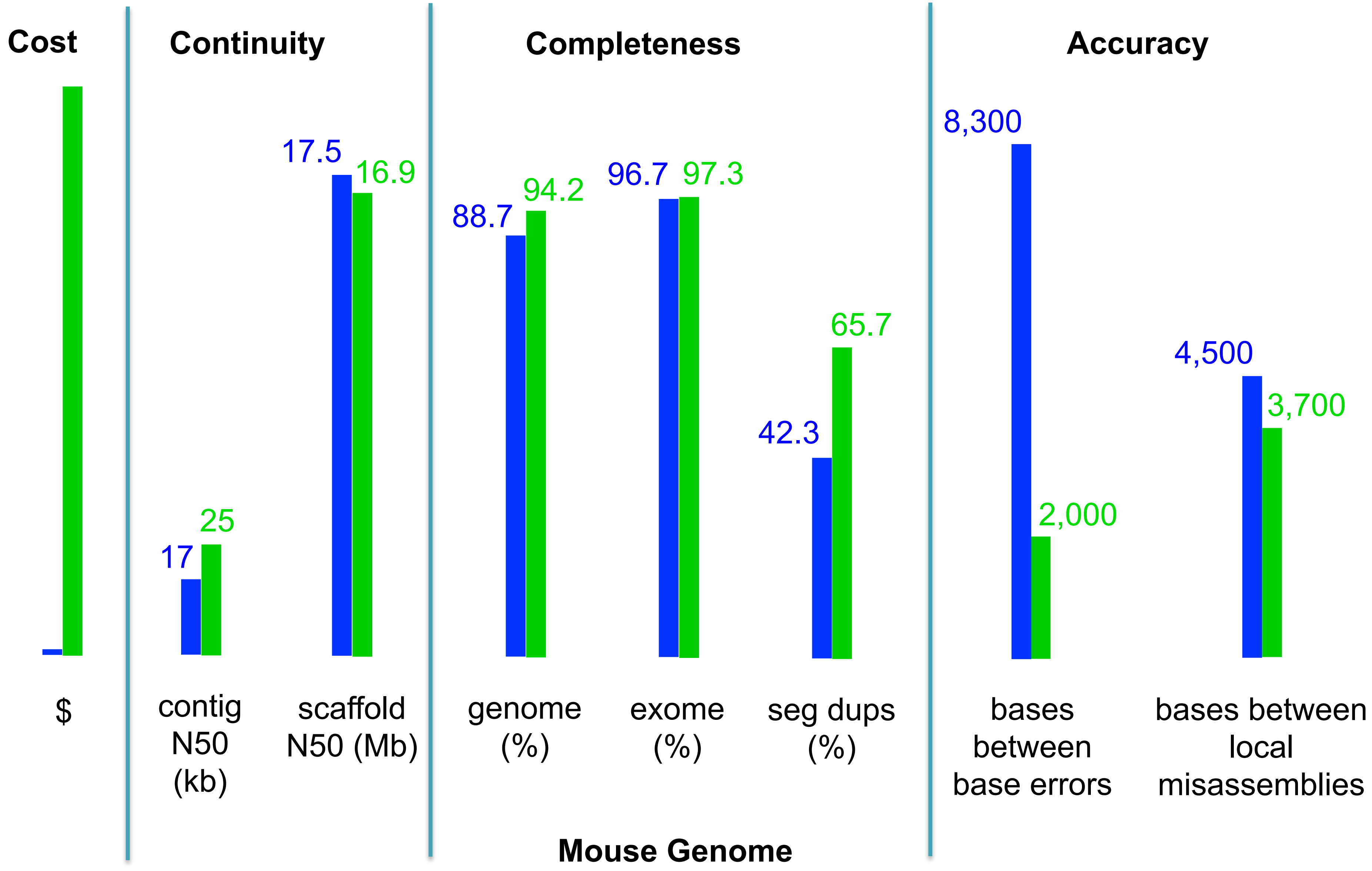
read pairs reach into repeats

and are extended by other
unipaths

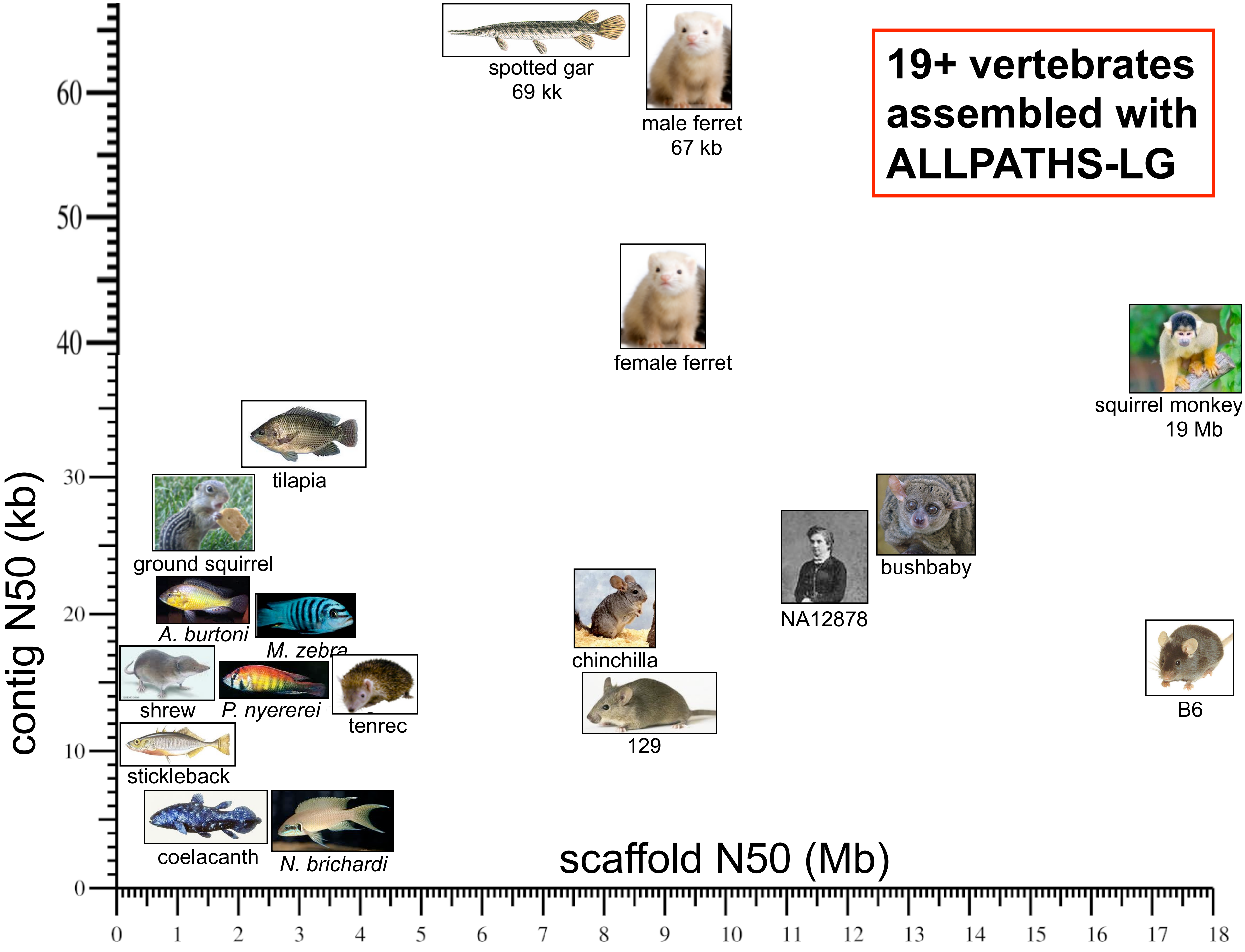
Create assembly from global assembly graph

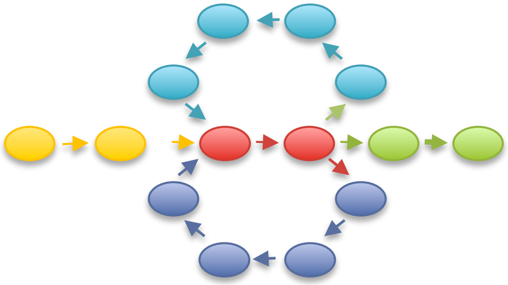


Large genome recipe: ALLPATHS-LG vs capillary



19+ vertebrates assembled with ALLPATHS-LG





Genome assembly with SOAPdenovo

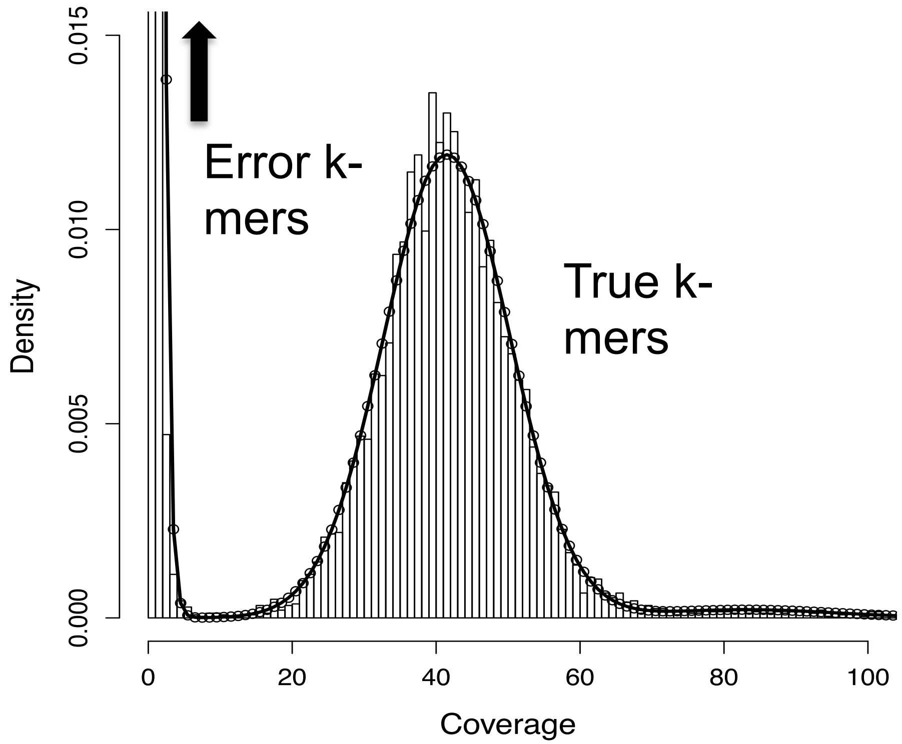
Error Correction with Quake

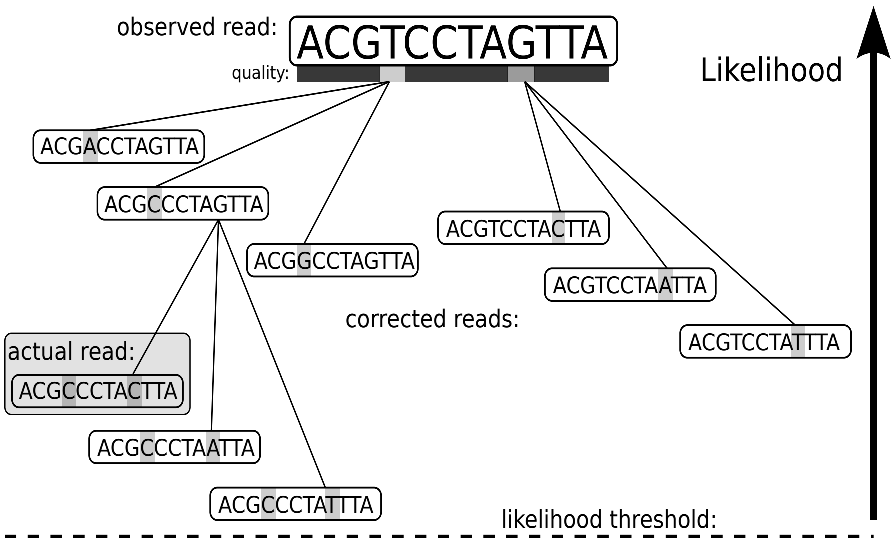
1. Count all "Q-mers" in reads

- Fit coverage distribution to mixture model of errors and regular coverage
- Automatically determines threshold for trusted k-mers

2. Correction Algorithm

- Considers editing erroneous kmers into trusted kmers in decreasing likelihood
- Includes quality values, nucleotide/nucleotide substitution rate



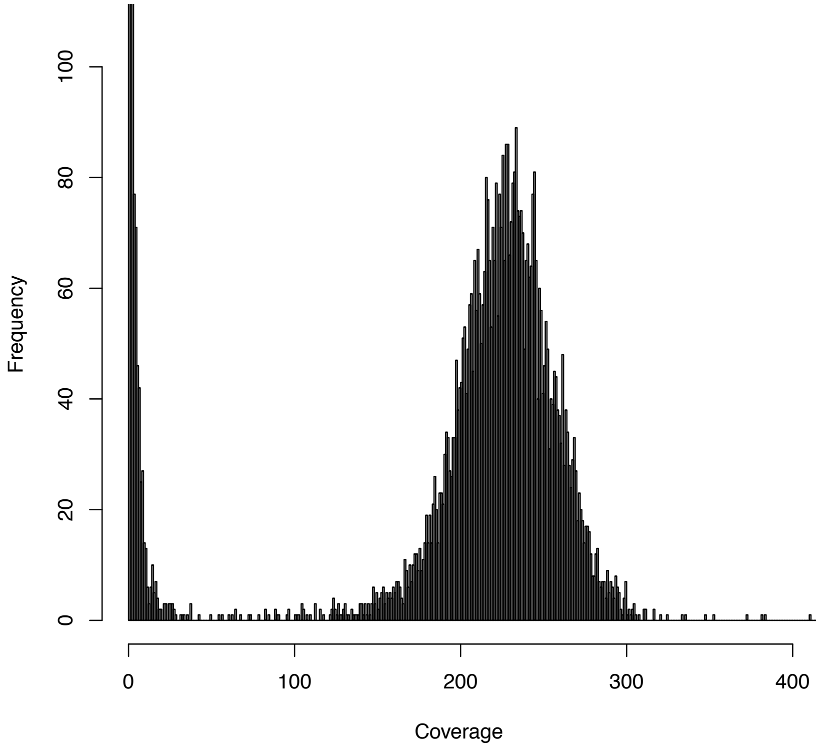


Quake: quality-aware detection and correction of sequencing reads.
Kelley, DR, Schatz, MC, Salzberg SL (2010) *Genome Biology.* 11:R116

Illumina Sequencing & Assembly

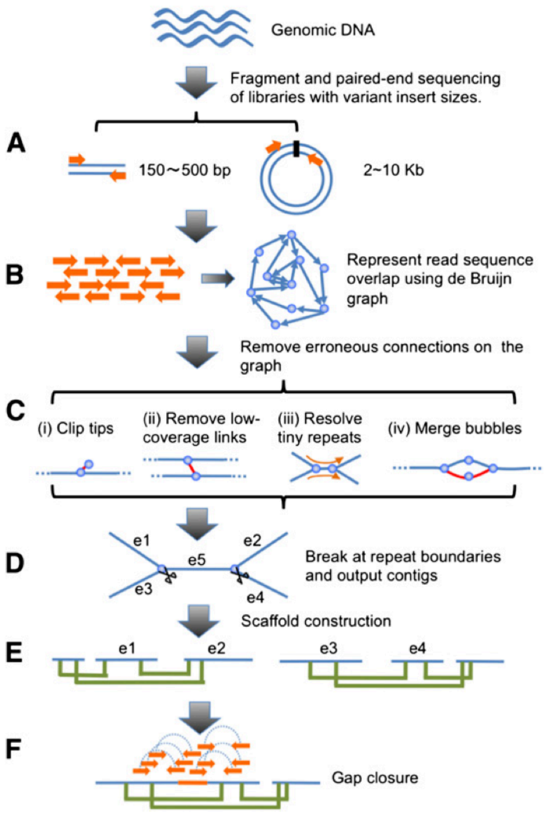
Quake Results

2x76bp @ 275bp
2x36bp @ 3400bp

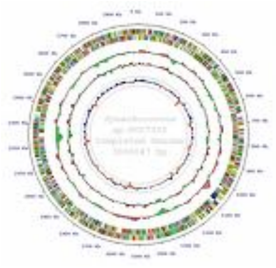


Validated	51,243,281	88.5%
Corrected	2,763,380	4.8%
Trim Only	3,273,428	5.6%
Removed	606,251	1.0%

SOAPdenovo Results


Genomic DNA
Fragment and paired-end sequencing of libraries with variant insert sizes.
A
150~500 bp
2~10 Kb
B
Represent read sequence overlap using de Bruijn graph
Remove erroneous connections on the graph
C
(i) Clip tips
(ii) Remove low-coverage links
(iii) Resolve tiny repeats
(iv) Merge bubbles
D
e1
e2
e5
e3
e4
Break at repeat boundaries and output contigs
Scaffold construction
E
e1
e2
e3
e4
F
Gap closure

	# ≥ 100bp	N50 (bp)
Scaffolds	2,340	253,186
Contigs	2,782	56,374
Unitigs	4,151	20,772

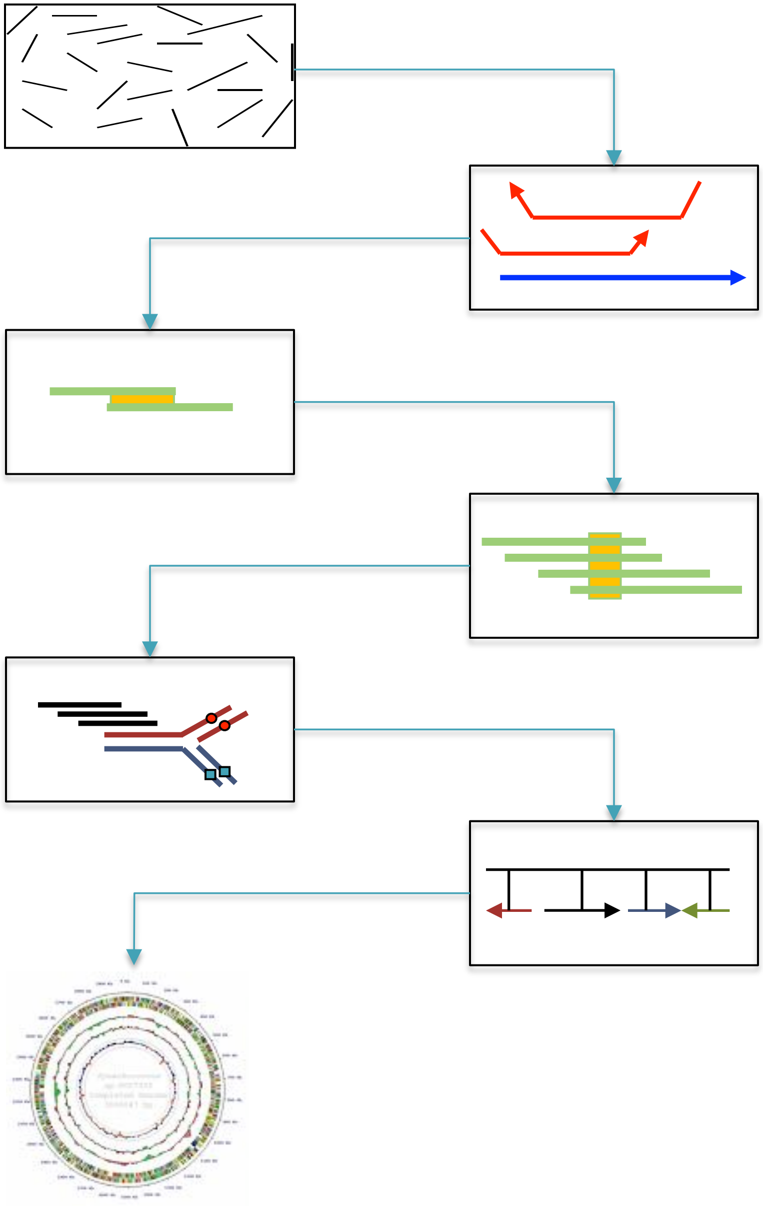


Genome assembly with the Celera Assembler

Celera Assembler

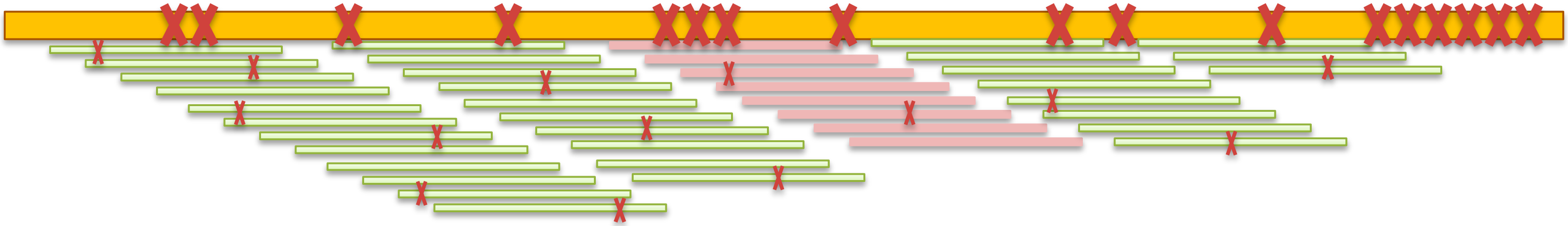
http://wgs-assembler.sf.net

1. Pre-overlap
   - Consistency checks
2. Trimming
   - Quality trimming & partial overlaps
3. Compute Overlaps
   - Find high quality overlaps
4. Error Correction
   - Evaluate difference in context of overlapping reads
5. Unitigging
   - Merge consistent reads
6. Scaffolding
   - Bundle mates, Order & Orient
7. Finalize Data
   - Build final consensus sequences



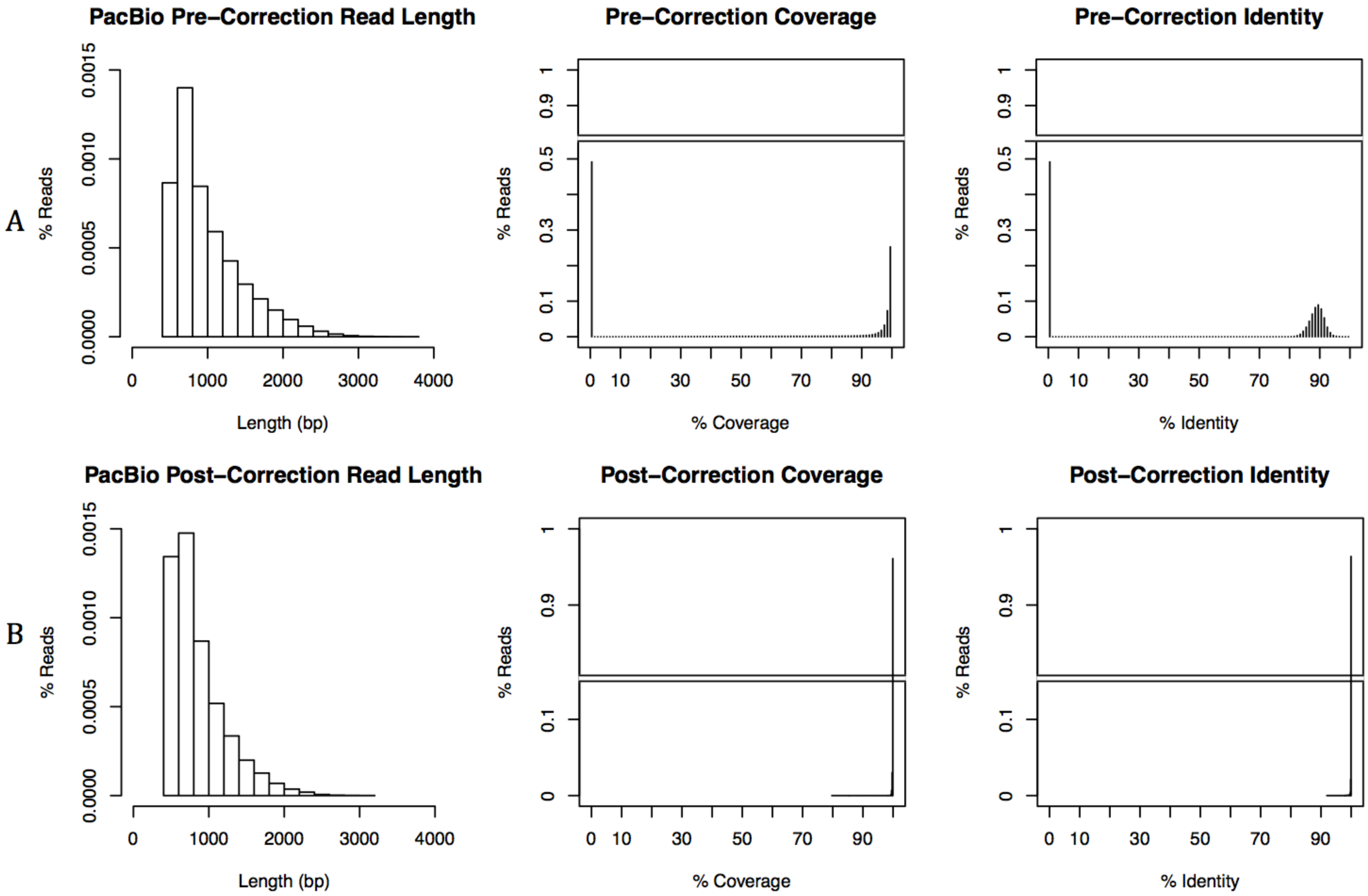
SMRT-hybrid Error Correction & Assembly

1. Trim/correct SR sequence
2. Compute an SR layout for each LR
    1. Map SRs to LRs
    2. Trim LRs at coverage gaps
    3. Compute consensus for each LR
3. Co-assemble corrected LRs and SRs
    - Celera Assembler enhanced to support 32 Kbp reads



Hybrid error correction and de novo assembly of single-molecule sequencing reads. Koren, S, Schatz, MC, Walenz, BP, Martin, J, Howard, J, Ganapathy, G, Wang, Z, Rasko, DA, McCombie, WR, Jarvis, ED, Phillippy, AM. (2011) *In preparation.*

Error Correction Results



PacBio Pre–Correction Read Length

Pre–Correction Coverage

Pre–Correction Identity

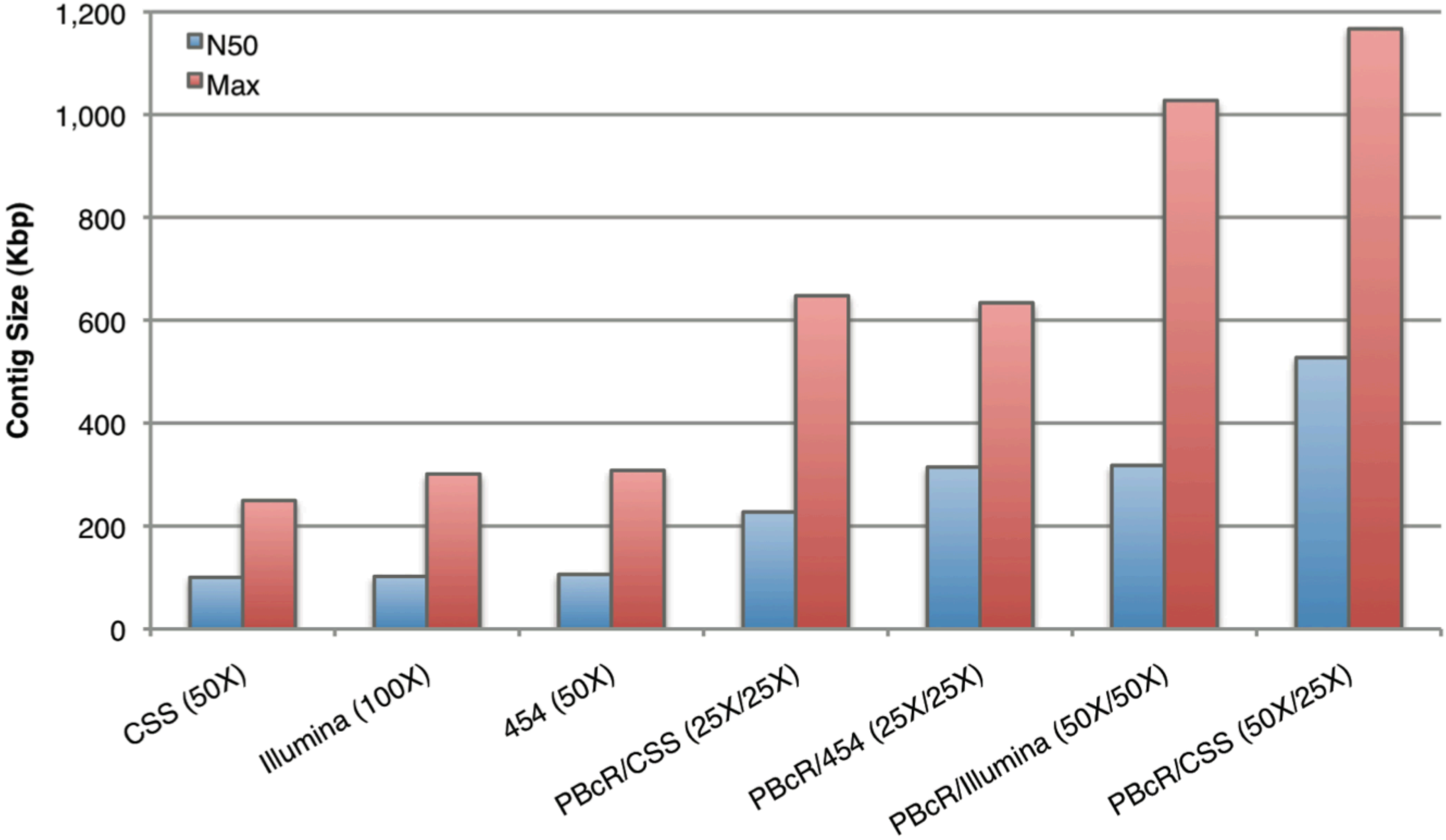
PacBio Post–Correction Read Length

Post–Correction Coverage

Post–Correction Identity

Correction results of 20x PacBio coverage of E. coli K12 corrected using 50x Illumina

Hybrid Assembly Results



SMRT-hybrid assembly results of 50x PacBio corrected coverage of E. coli K12
Long reads lead to ***contigs*** over 1Mbp

THE ASSEMBLATHON

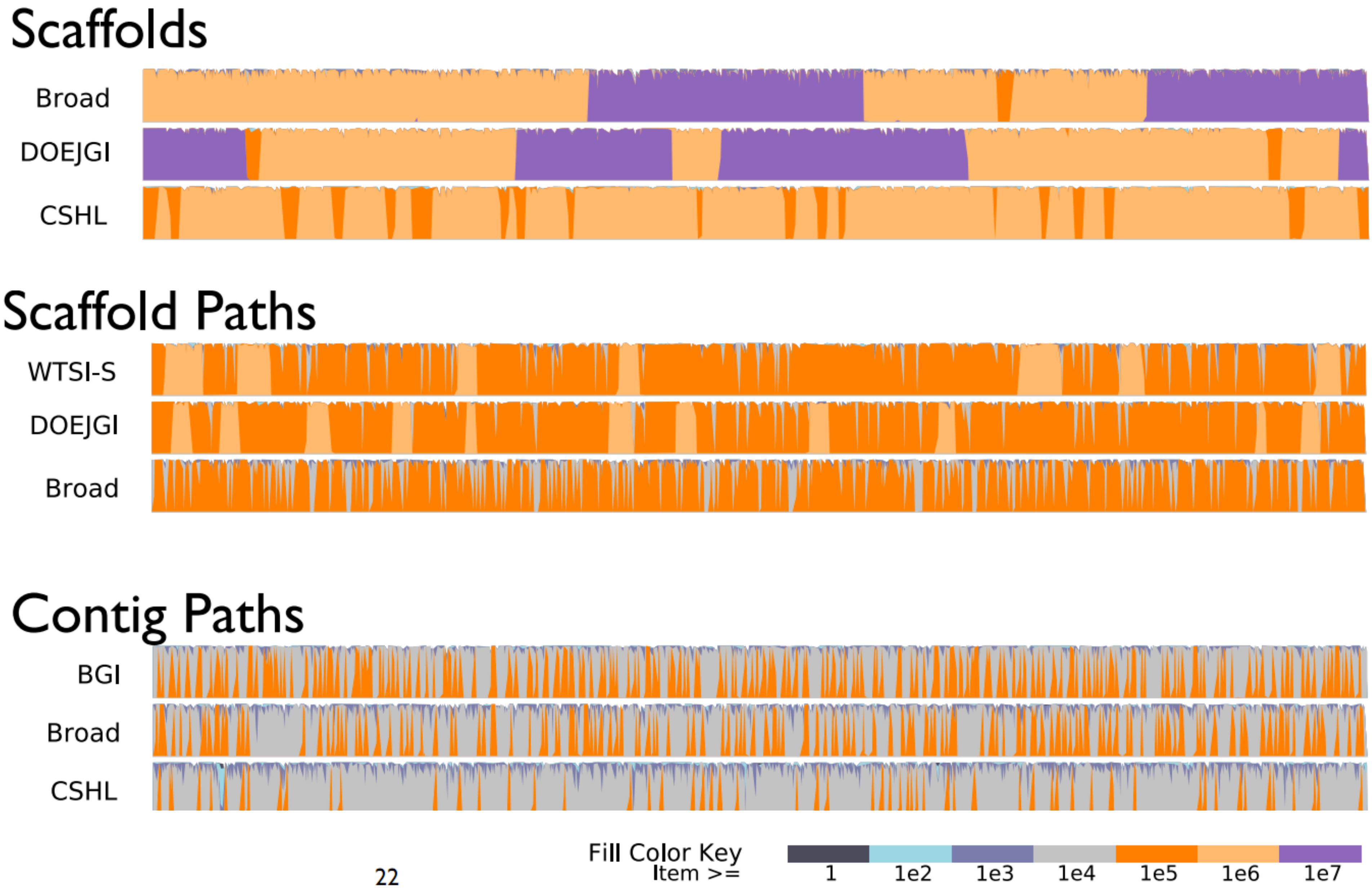
- Attempt to answer the question:
  “What makes a good assembly?”

- Organizers provided simulated sequence data
  - Simulated 100 base pair Illumina reads from simulated diploid organism

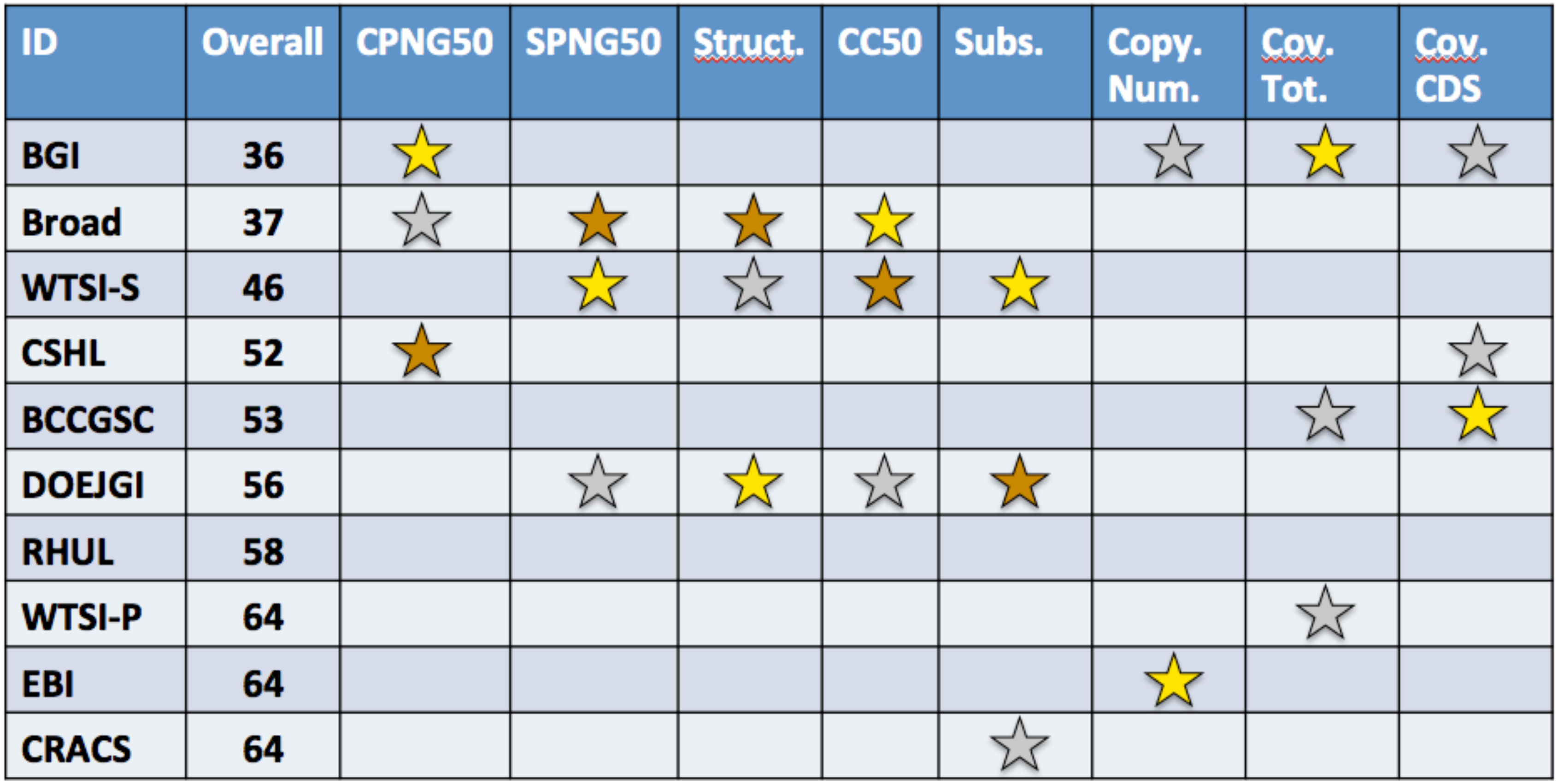
- 41 submissions from 17 groups

- Results demonstrate trade-offs assemblers must make

Assembly Results



Final Rankings

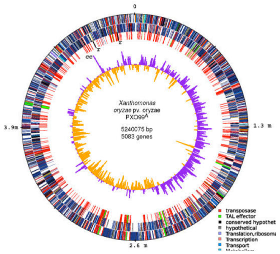


ID	Overall	CPNG50	SPNG50	Struct.	CC50	Subs.	Copy. Num.	Cov. Tot.	Cov. CDS
BGI	36	🥇					🥈	🥇	🥈
Broad	37	🥈	🥉	🥉	🥇				
WTSI-S	46		🥇	🥈	🥉	🥇			
CSHL	52	🥉							🥈
BCCGSC	53							🥈	🥇
DOEJGI	56		🥈	🥇	🥈	🥉			
RHUL	58								
WTSI-P	64							🥈	
EBI	64						🥇		
CRACS	64					🥈			

- SOAPdenovo and ALLPATHS came out neck-and-neck followed closely behind by SGA, Celera Assembler, ABySS

- My recommendation for "typical" short read assembly is to use ALLPATHS

Assembly Summary



Assembly quality depends on

1. ***Coverage***: low coverage is mathematically hopeless
2. ***Repeat composition***: high repeat content is challenging
3. ***Read length***: longer reads help resolve repeats
4. ***Error rate***: errors reduce coverage, obscure true overlaps

- Assembly is a hierarchical, starting from individual reads, build high confidence contigs/unitigs, incorporate the mates to build scaffolds
  - Extensive error correction is the key to getting the best assembly possible from a given data set

- Watch out for collapsed repeats & other misassemblies
  - Globally/Locally reassemble data from scratch with better parameters & stitch the 2 assemblies together

Break

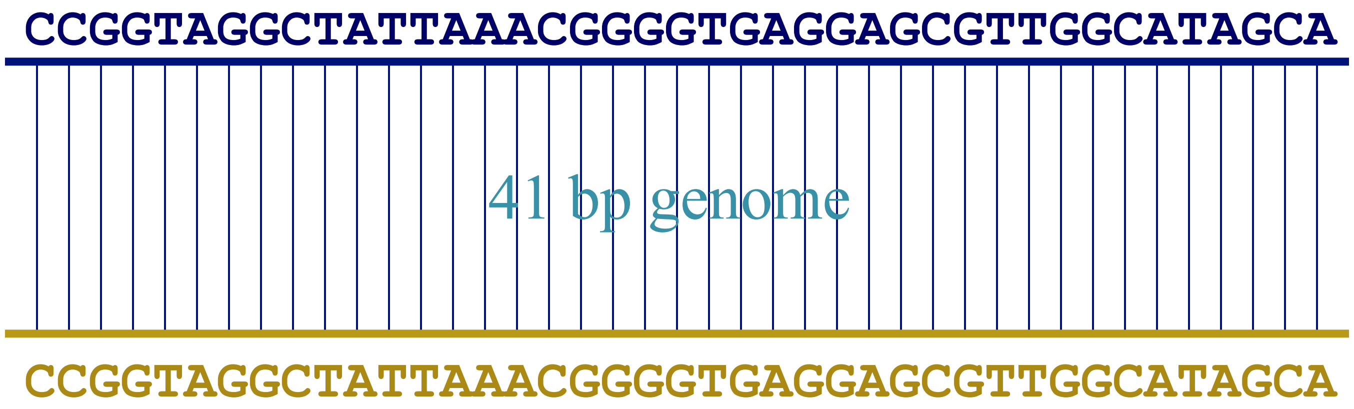
PHASE TWO : INTERPRETATION
I THINK I FOUND A CORNER PIECE.
3 BILLION PIECES
GENOME
SHENEMAN The Star-Ledger

Whole Genome Alignment with MUMmer

Slides Courtesy of Adam M. Phillippy
amp@umics.umd.edu

Goal of WGA

- For two genomes, *A* and *B*, find a mapping from each position in *A* to its corresponding position in *B*



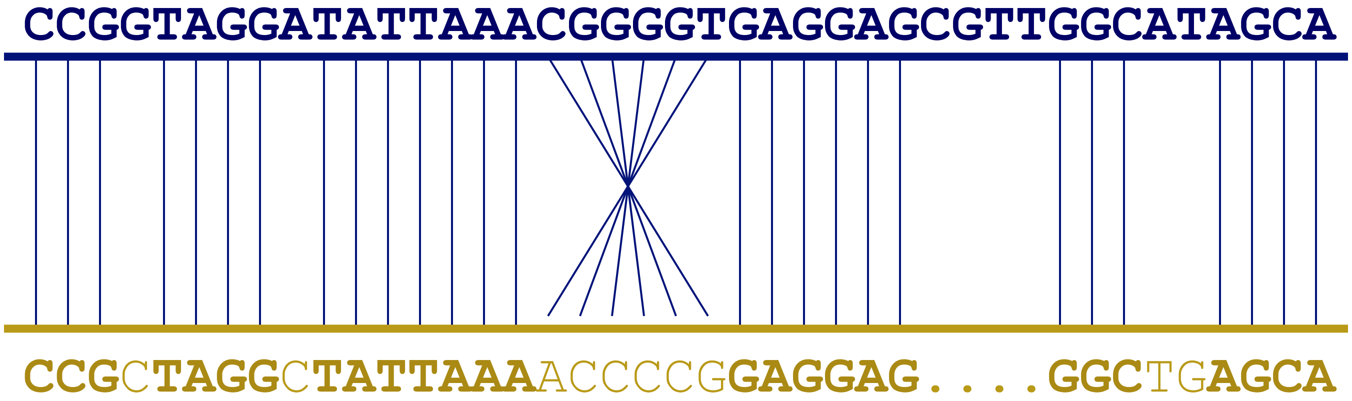
CCGGTAGGCTATTAAACGGGGTGAGGAGCGTTGGCATAGCA

41 bp genome

CCGGTAGGCTATTAAACGGGGTGAGGAGCGTTGGCATAGCA

Not so fast...

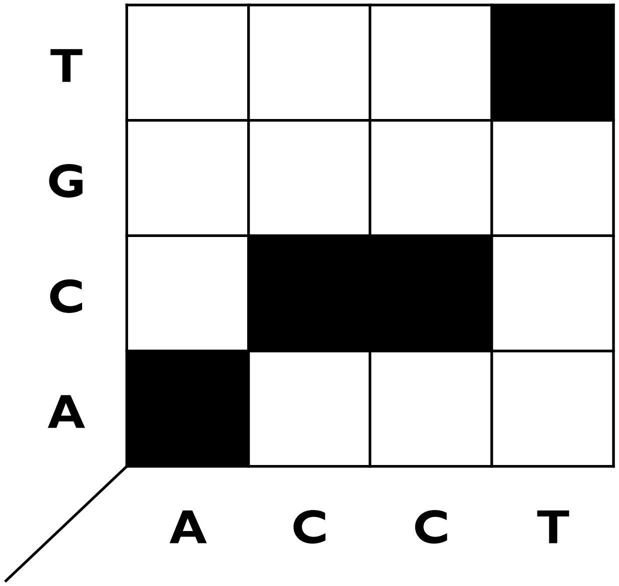
- Genome *A* may have insertions, deletions, translocations, inversions, duplications or SNPs with respect to *B* (sometimes all of the above)


CCGGTAGGATATTAAACGGGGTGAGGAGCGTTGGCATAGCA
CCGCTAGGCTATTAAAACCCCGGAGGAG....GGCTGAGCA

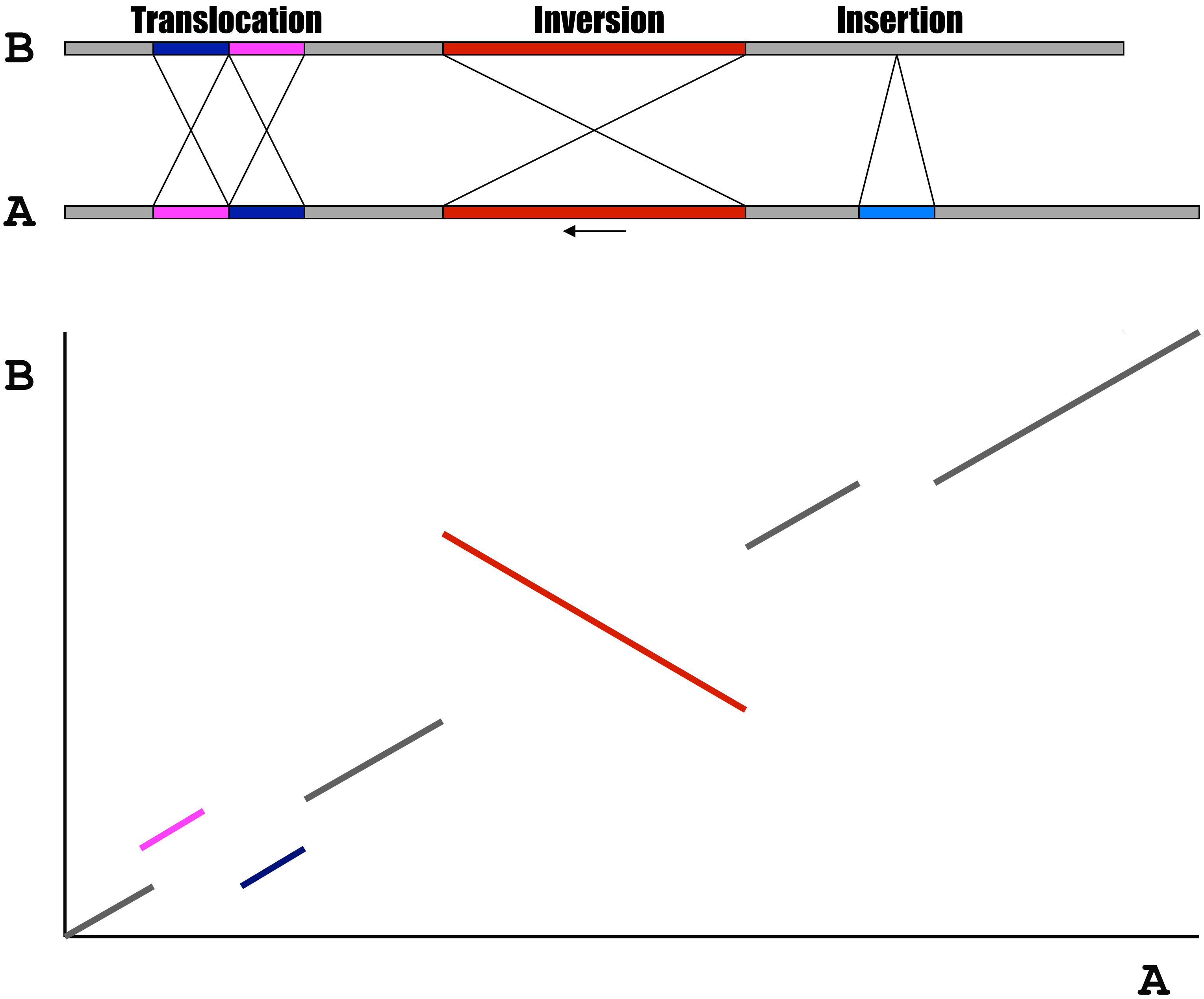
WGA visualization

- How can we visualize *whole* genome alignments?

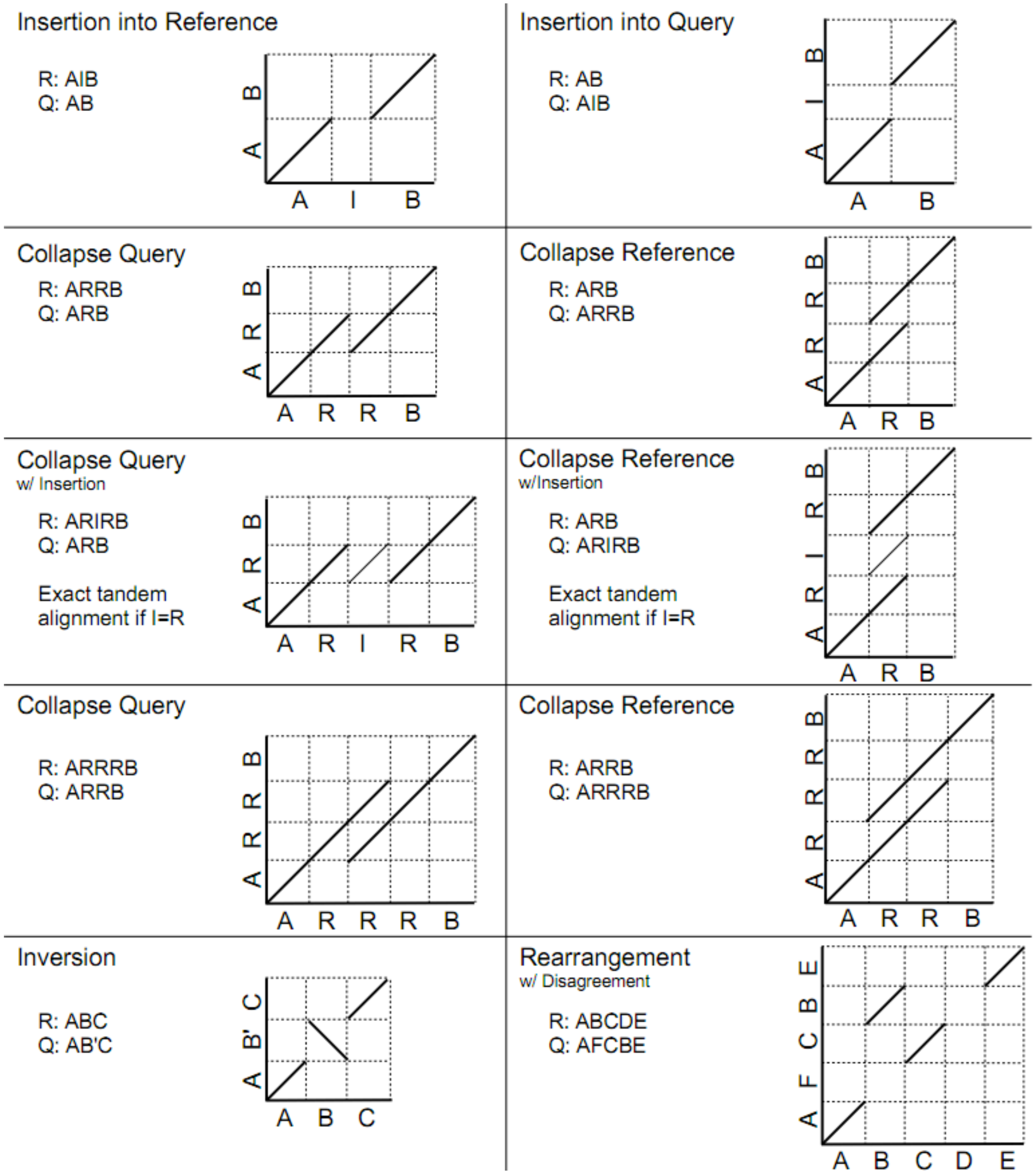
- With an alignment dot plot
  - *N* x *M* matrix
    - Let *i* = position in genome *A*
    - Let *j* = position in genome *B*
    - Fill cell *(i,j)* if A_i shows similarity to B_j



  - A perfect alignment between *A* and *B* would completely fill the positive diagonal


Translocation
Inversion
Insertion
B
A
B
A

SV Types



- Different structural variation types / misassemblies will be apparent by their pattern of breakpoints

- Most breakpoints will be at or near repeats

- Things quickly get complicated in real genomes

http://mummer.sf.net/manual/AlignmentTypes.pdf

MUMmer

- Maximal Unique Matcher (MUM)
  - match
    - exact match of a minimum length
  - maximal
    - cannot be extended in either direction without a mismatch
  - *unique*
    - occurs only once in both sequences (MUM)
    - occurs only once in a single sequence (MAM)
    - occurs one or more times in either sequence (MEM)

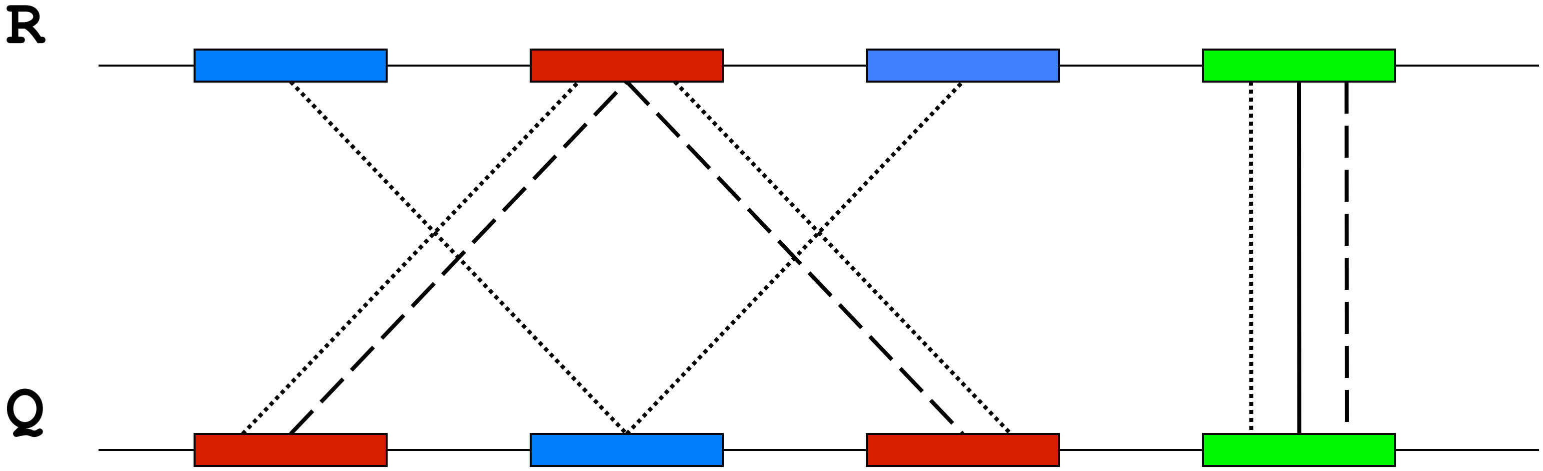
Fee Fi Fo Fum,

is it a MAM, MEM or MUM?

MUM : maximal unique match ———

MAM : maximal almost-unique match – – – – –

MEM : maximal exact match ··········



Seed and Extend

How can quickly find large MUMs?

1. Find MUMs
    - using a suffix tree
2. Cluster MUMs
    - using size, gap and distance parameters
3. Extend clusters
    - using modified Smith-Waterman algorithm

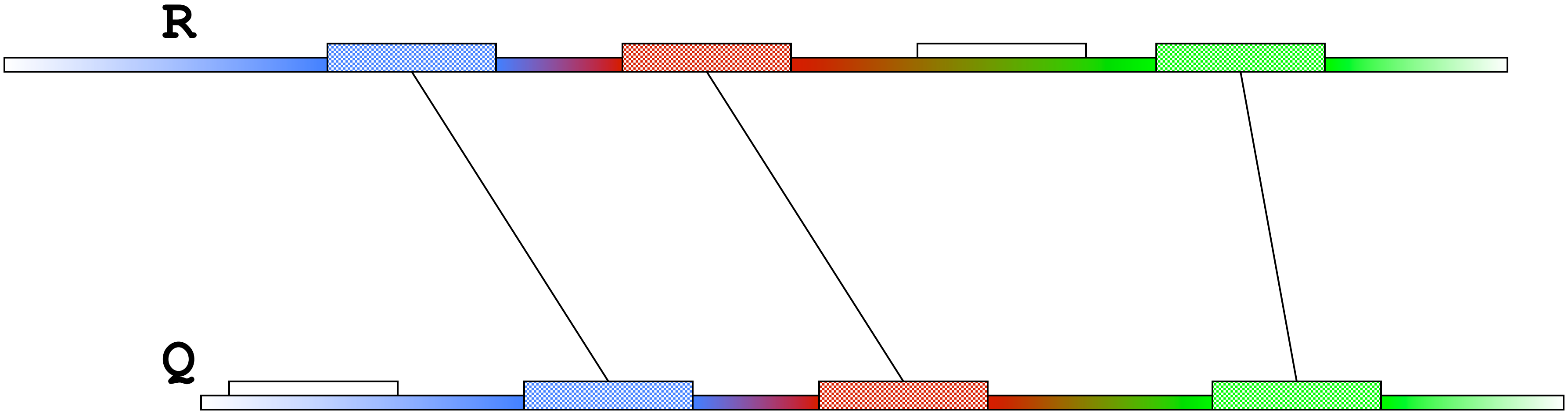
Seed and Extend

visualization

FIND all MUMs

CLUSTER consistent MUMs

EXTEND alignments



WGA example with **nucmer**

- *Yersina pestis CO92* vs. *Yersina pestis KIM*
  - High nucleotide similarity, 99.86%
    - Two strains of the same species
  - Extensive genome shuffling
    - Global alignment will not work
  - Highly repetitive
    - Many local alignments

WGA Alignment

```
nucmer -maxmatch CO92.fasta KIM.fasta
-maxmatch    Find maximal exact matches (MEMs)
```

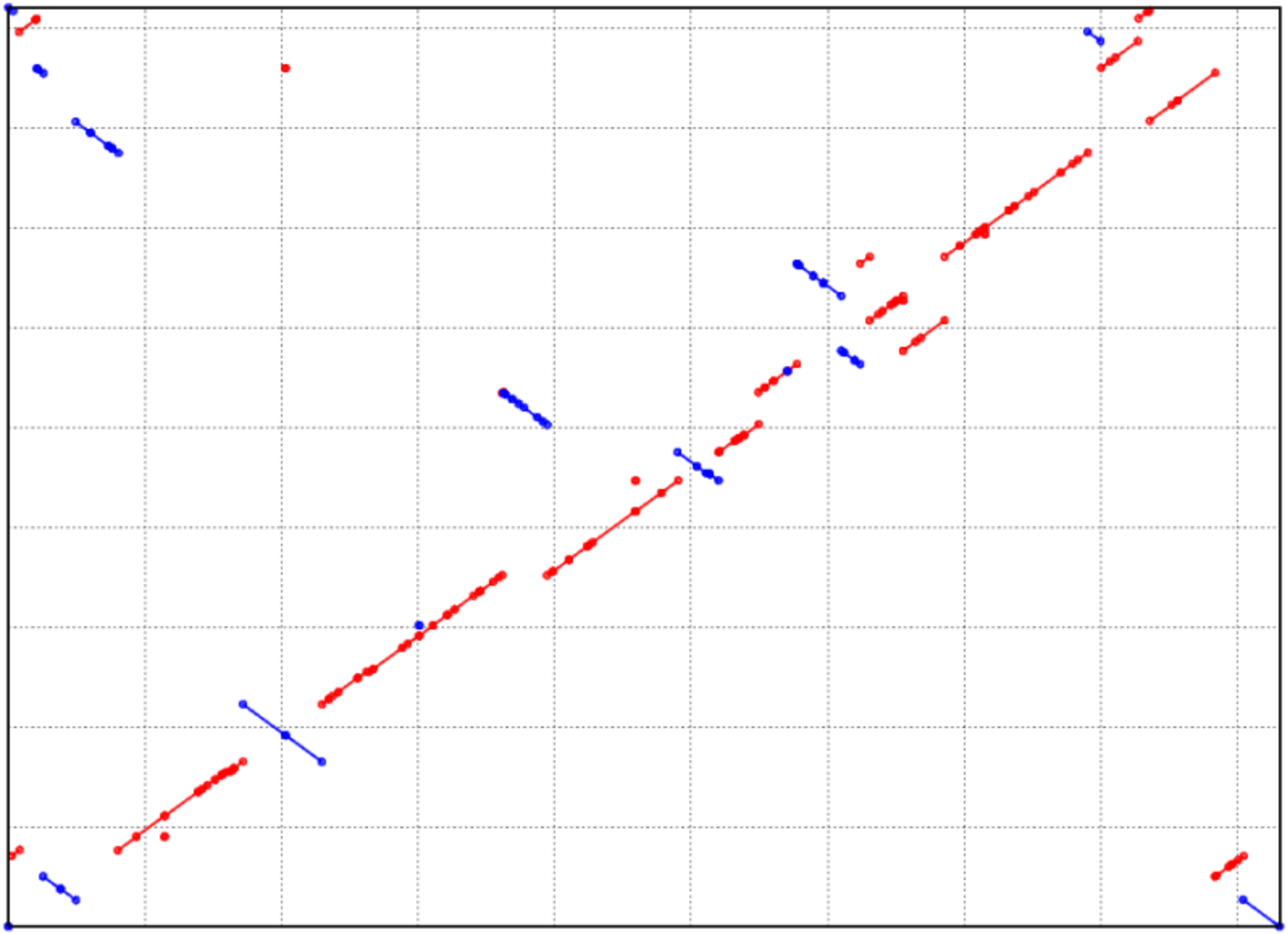
```
delta-filter -m out.delta > out.filter.m
-m   Many-to-many mapping
```

```
show-coords -r out.delta.m > out.coords
-r   Sort alignments by reference position
```

```
dnadiff out.delta.m
Construct catalog of sequence variations
```

```
mummerplot --large --layout out.delta.m
--large      Large plot
--layout Nice layout for multi-fasta files
--x11        Default, draw using x11 (--postscript, --png)
*requires gnuplot
```

See manual at http://mummer.sourceforge.net/manual



References

- Documentation
  - http://mummer.sourceforge.net
    - » publication listing
  - http://mummer.sourceforge.net/manual
    - » documentation
  - http://mummer.sourceforge.net/examples
    - » walkthroughs

- Email
  - mummer-help@lists.sourceforge.net
  - amp@umiacs.umd.edu

Acknowledgements

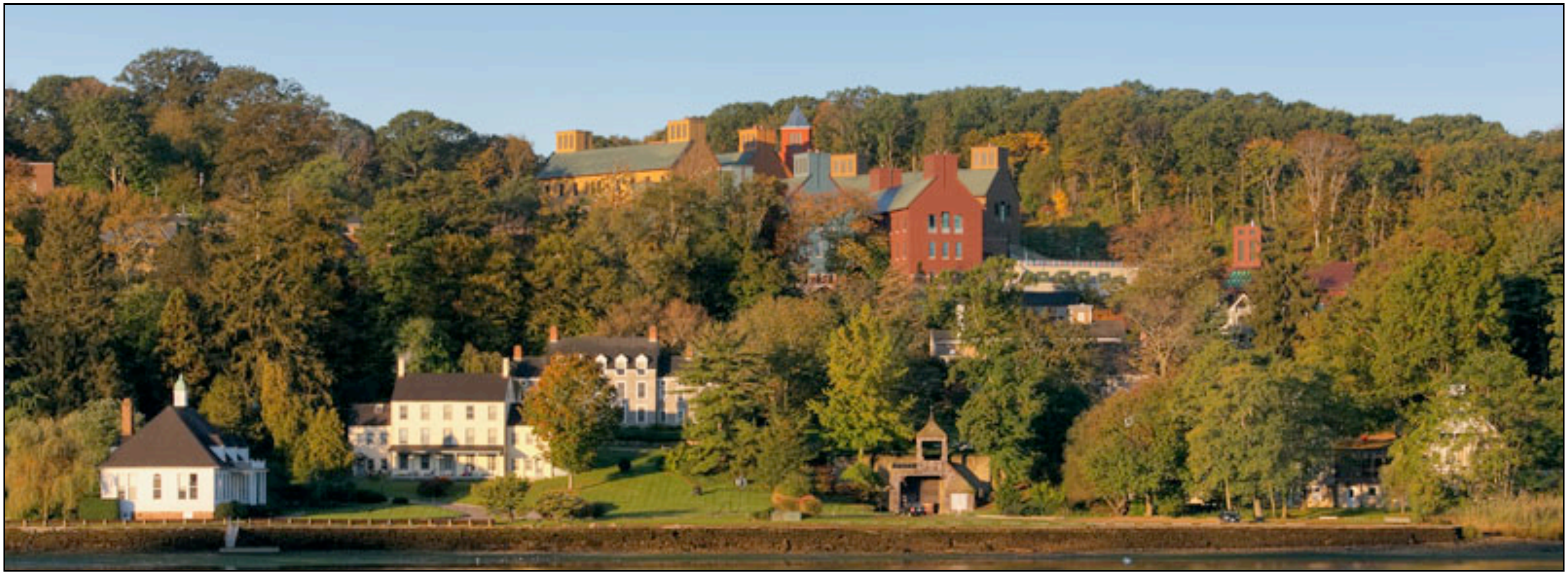
Schatzlab
Mitch Bekritsky
Matt Titmus
Hayan Lee
James Gurtowski
Anirudh Aithal
Rohith Menon
Goutham Bhat

CSHL
Dick McCombie
Melissa Kramer
Eric Antonio
Mike Wigler
Zach Lippman
Doreen Ware
Ivan Iossifov

JHU
Steven Salzberg
Ben Langmead
Jeff Leek

NBACC
Adam Phillipy
Sergey Koren

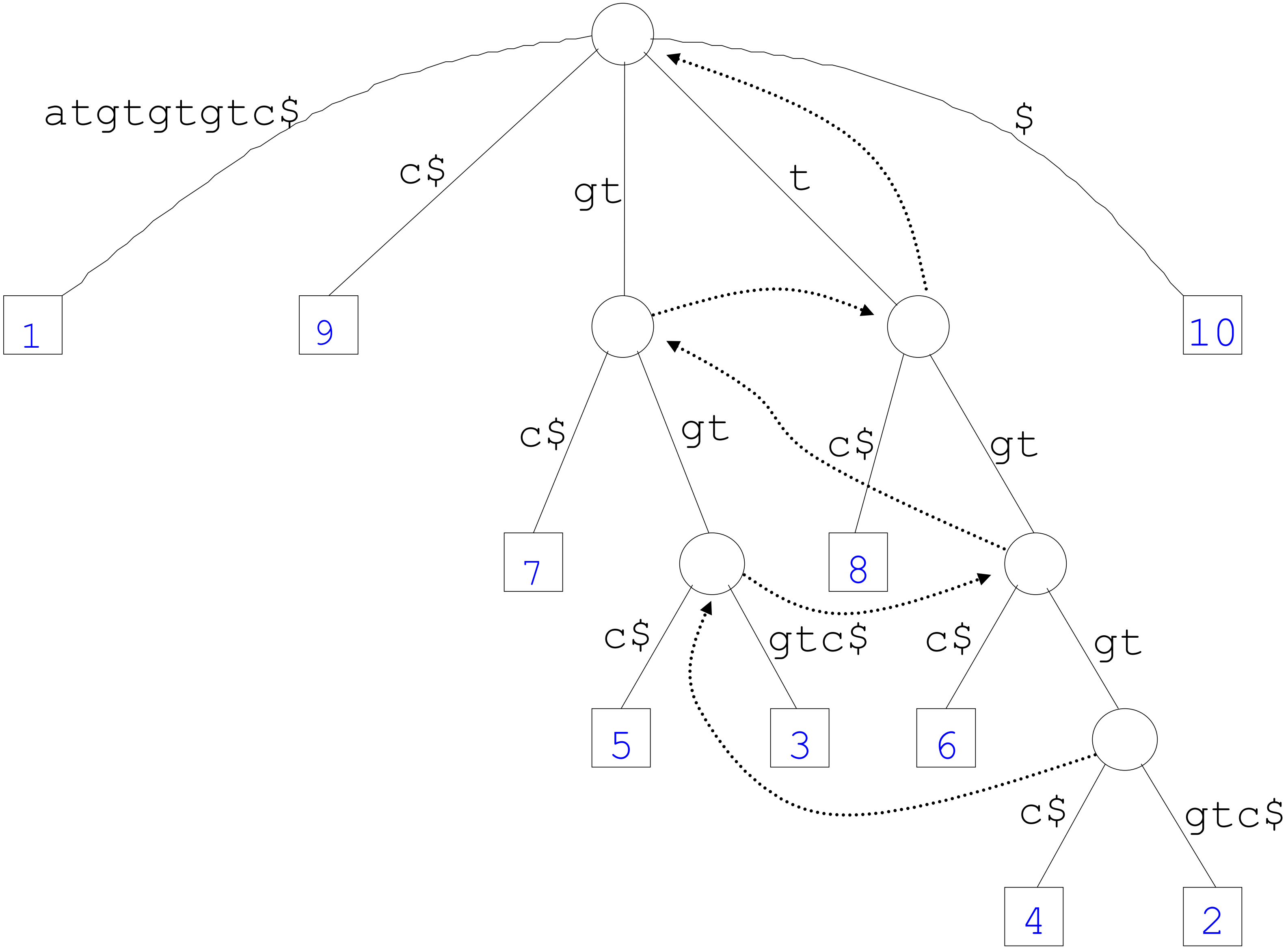
Univ. of Maryland
Mihai Pop
Art Delcher
Jimmy Lin
David Kelley
Dan Sommer
Cole Trapnell



Thank You

http://schatzlab.cshl.edu
@mike_schatz

Suffix Tree for atgtgtgtc$



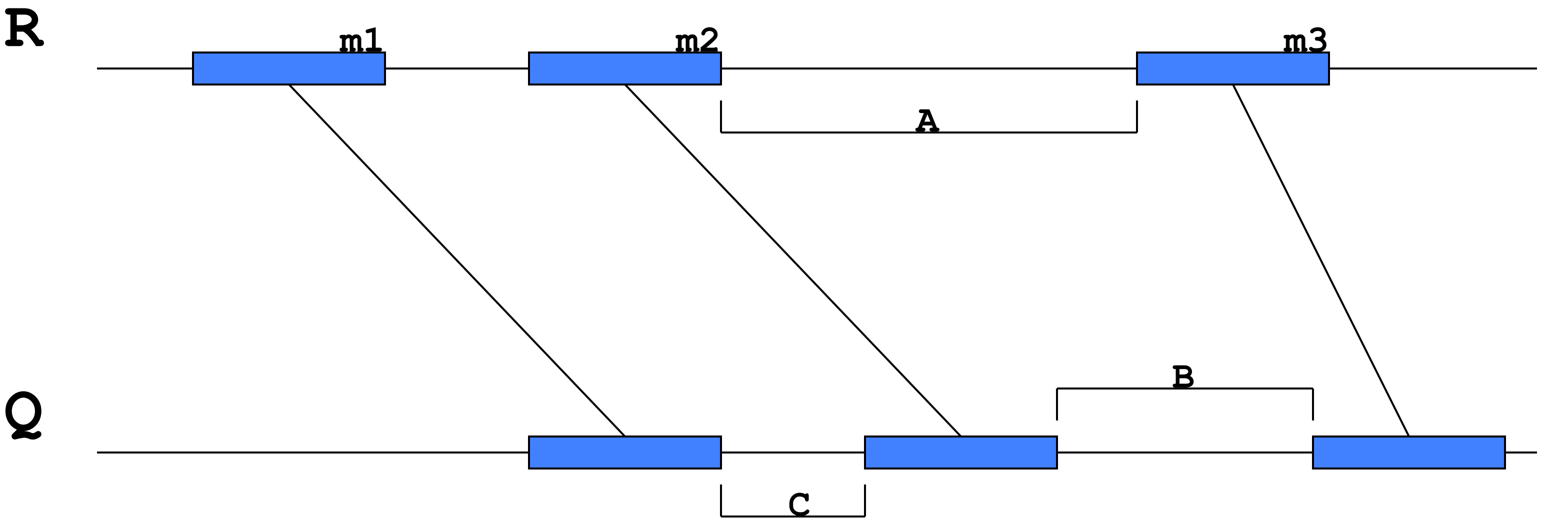
Drawing credit: Art Delcher

MUMmer Clustering

cluster length = Σm_i

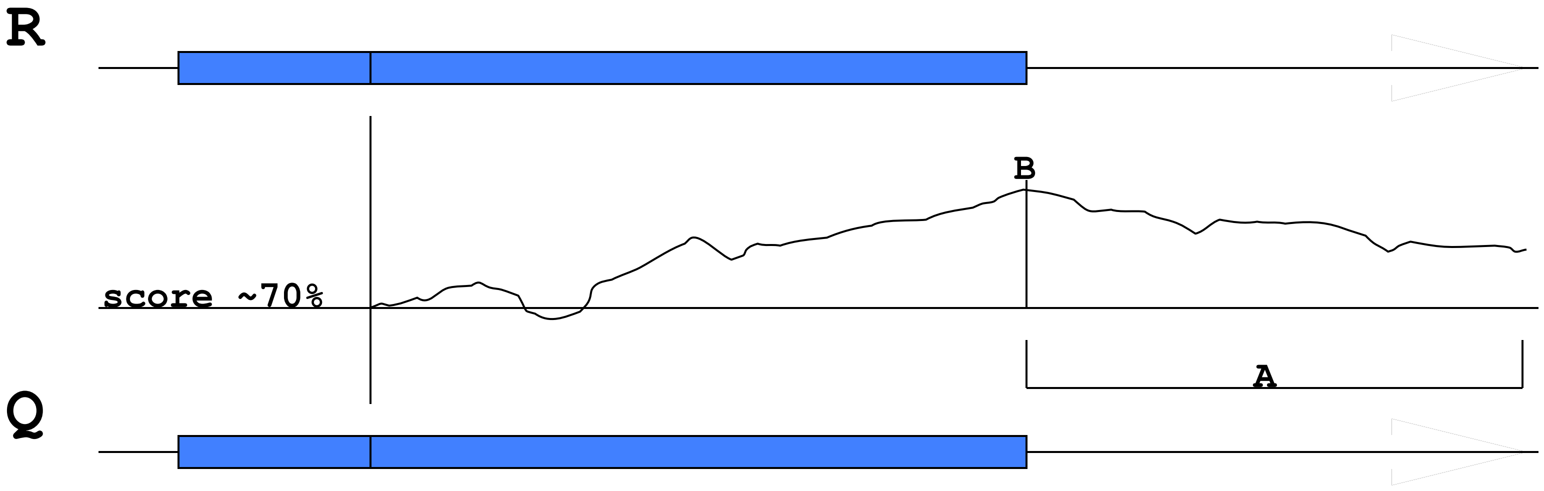
gap distance = C

indel factor = |B – A| / B *or* |B – A|



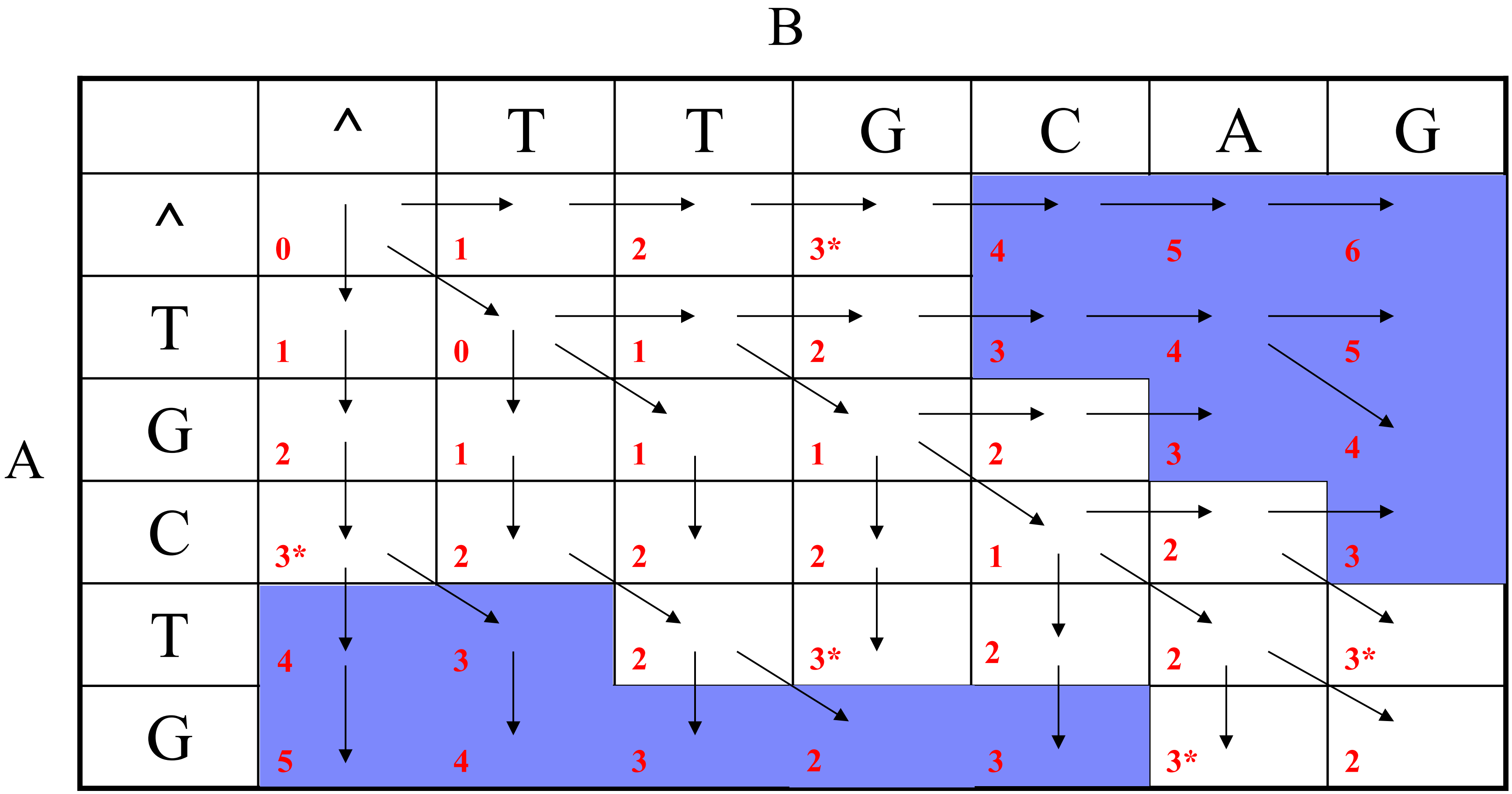
MUMmer Extending

break point = B



break length = A

MUMmer Banded Alignment



	^	T	T	G	C	A	G
^	0	1	2	3*	4	5	6
T	1	0	1	2	3	4	5
G	2	1	1	1	2	3	4
C	3*	2	2	2	1	2	3
T	4	3	2	3*	2	2	3*
G	5	4	3	2	3	3*	2