

Genome Sequencing & Assembly

Michael Schatz

April 21, 2013

CSHL Genome Access





Outline

I. Assembly theory

1. Assembly by analogy
2. De Bruijn and Overlap graph
3. Coverage, read length, errors, and repeats

2. Genome assemblers

1. ALLPATHS-LG, SOAPdenovo, Celera Assembler
2. Assemblathon

3. Applications

1. Whole Genome Alignment with MUMmer
2. Gene Finding



Outline

I. Assembly theory

1. Assembly by analogy
2. De Bruijn and Overlap graph
3. Coverage, read length, errors, and repeats

2. Genome assemblers

1. ALLPATHS-LG, SOAPdenovo, Celera Assembler
2. Assemblathon

3. Applications

1. Whole Genome Alignment with MUMmer
2. Gene Finding

Shredded Book Reconstruction

- Dickens accidentally shreds the first printing of A Tale of Two Cities
 - Text printed on 5 long spools

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

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

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

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

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

- How can he reconstruct the text?
 - $5 \text{ copies} \times 138,656 \text{ words} / 5 \text{ words per fragment} = 138k \text{ fragments}$
 - 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

Greedy Reconstruction

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

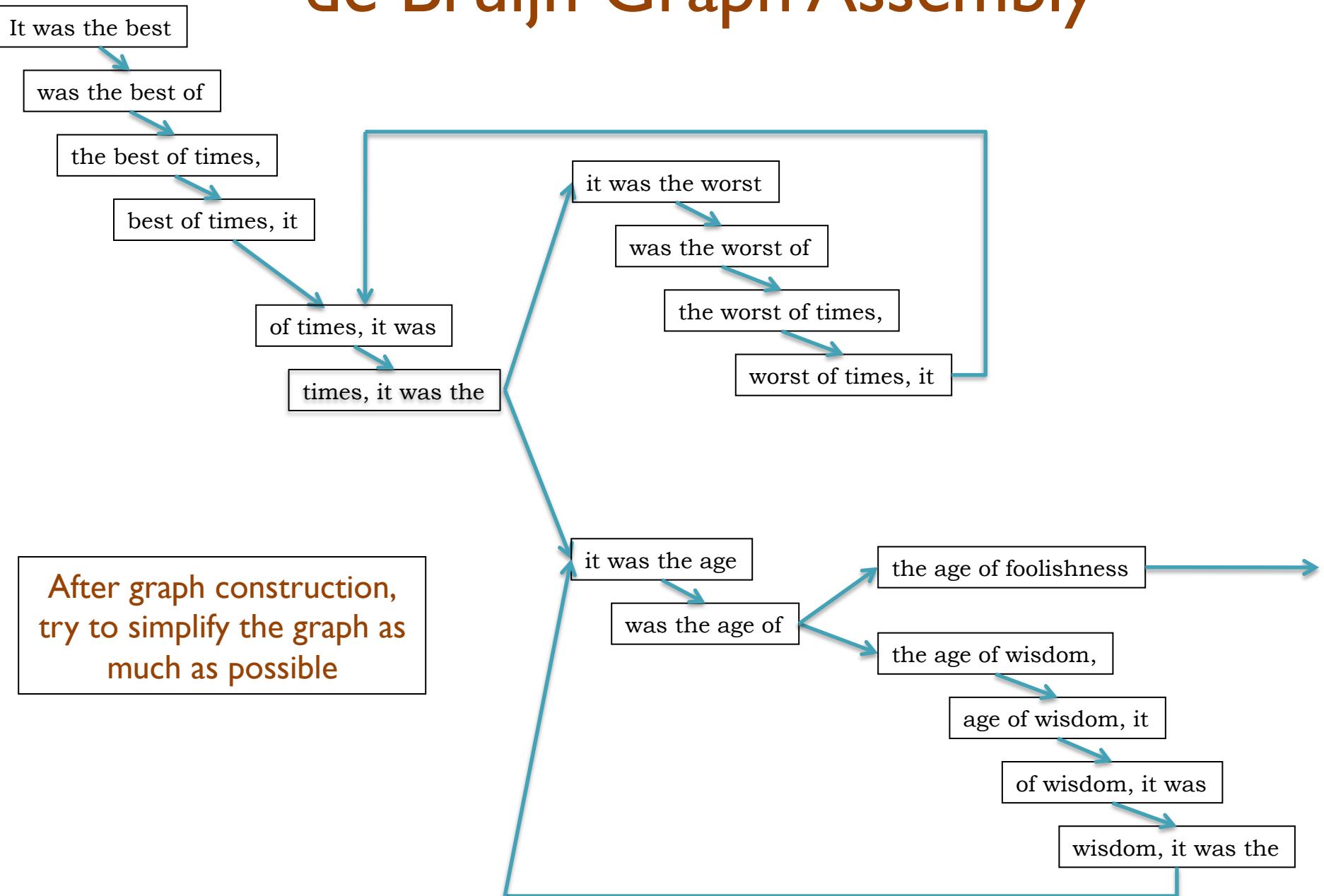
- Locally constructed graph reveals the global sequence structure
 - Overlaps between sequences implicitly computed

de Bruijn, 1946

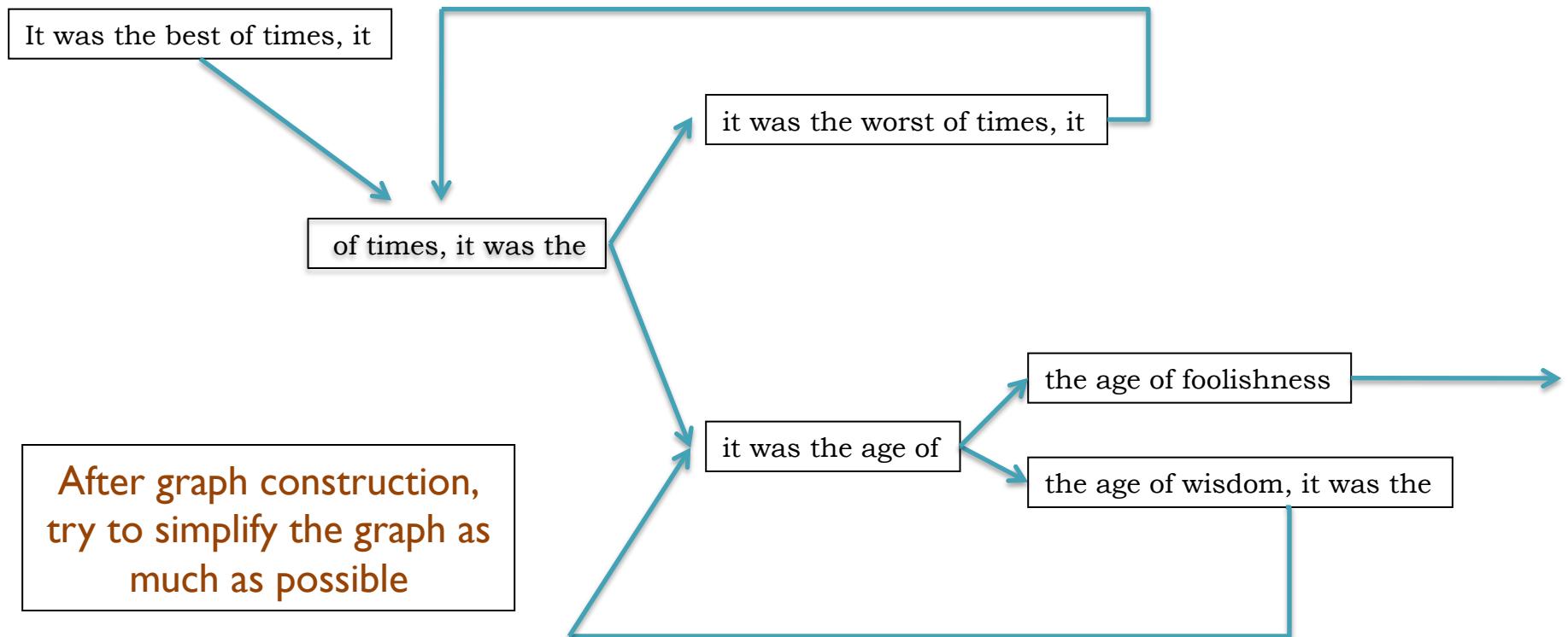
Idury and Waterman, 1995

Pevzner, Tang, Waterman, 2001

de Bruijn Graph Assembly



de Bruijn Graph Assembly



The full tale

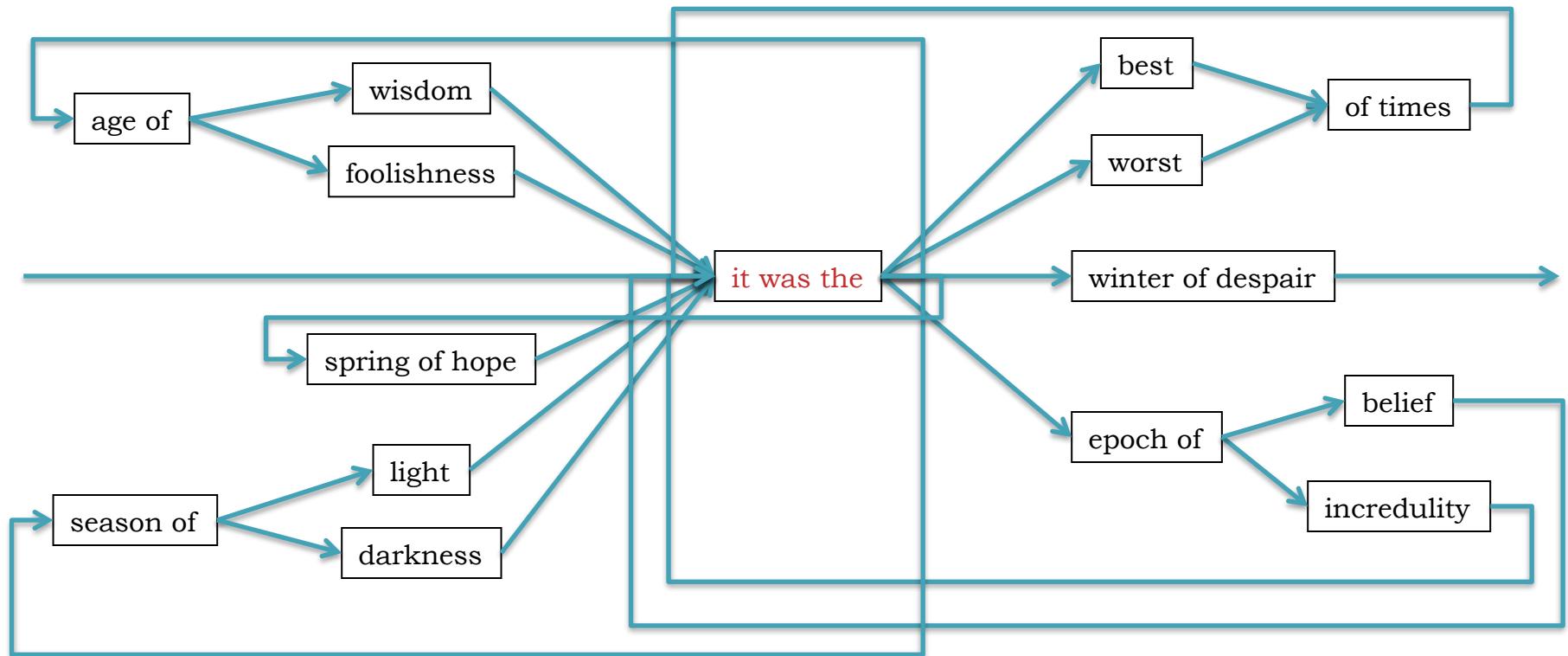
... it was the best of times it was the worst of times ...

... it was the age of wisdom it was the age of foolishness ...

... it was the epoch of belief it was the epoch of incredulity ...

... it was the season of light it was the season of darkness ...

... it was the spring of hope it was the winter of despair ...



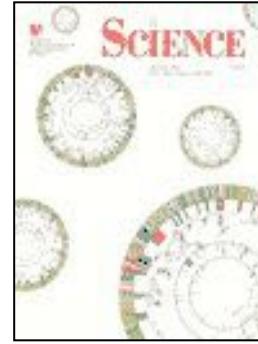
Milestones in Genome Assembly

Science Vol. 207 February 20, 1980
articles

Nucleotide sequence of bacteriophage Φ X174 DNA
E.Sanger, G.M.Air, R.G.Bonell, N.E.Browne, A.R.Coulson, J.C.Fiddes,
C.A.Hinchliffe, B.P., P.M.Shenstone & M.Smith
Add: Laboratory of Molecular Biology, 96 Royal Cambridge CB3 0EX, UK

A DNA sequence for the genome of bacteriophage Φ X174 of approximately 5375 base pairs has been determined by the rapid and simple 'linkage and recombination' method. The minimum molecular mass of the genome responsible for the production of the three types of infectious particles is estimated to be 4.5 x 10⁶ daltons. The linkage and recombination data for the proteins and RNA ds. Two pairs of genes are located by the same methods of Φ X174 using different methods. Three

The genome of bacteriophage Φ X174 is a single-stranded, non-recombinant molecule. The code of these genes as determined by genetic techniques¹ is as follows: C-D-A-L-D-E-K-N-E-T. It is not yet clear whether the genes are ordered in the same direction as they follow the sequential order of the DNA strand. It is not known if the sequential order of the genes is also linear.



1977. Sanger et al.
1st Complete Organism
5375 bp



1998. C.elegans SC
1st Multicellular Organism
BAC-by-BAC Phrap. 97Mbp



2000. Myers et al.
1st Large WGS Assembly.
Celera Assembler. 116 Mbp



2001. Venter et al., IHGSC
Human Genome
Celera Assembler/GigaAssembler. 2.9 Gbp



2010. Li et al.
1st Large SGS Assembly.
SOAPdenovo 2.2 Gbp

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

Assembly Applications

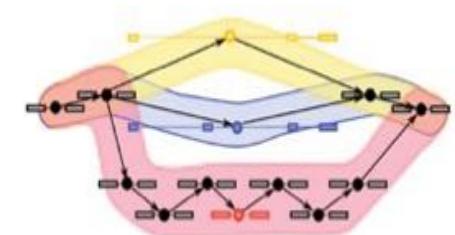
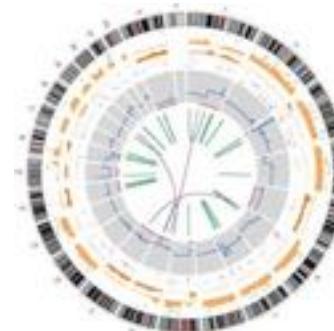
- Novel genomes



- Metagenomes

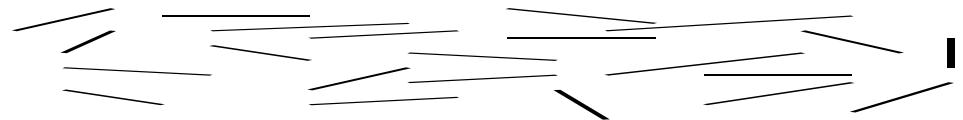


- Sequencing assays
 - Structural variations
 - Transcript assembly
 - ...



Assembling a Genome

1. Shear & Sequence DNA



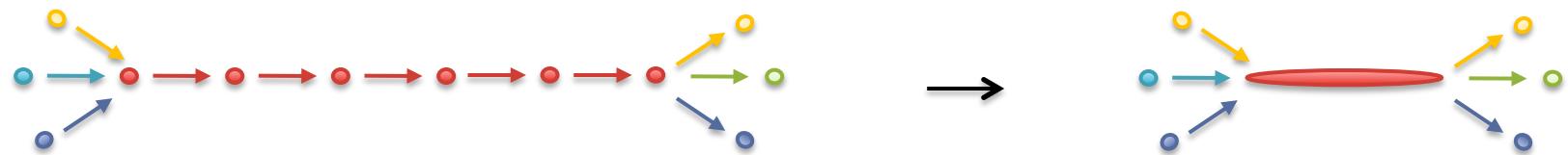
2. Construct assembly graph from overlapping reads

...AGCCTAGGGATGCGCGACACGT

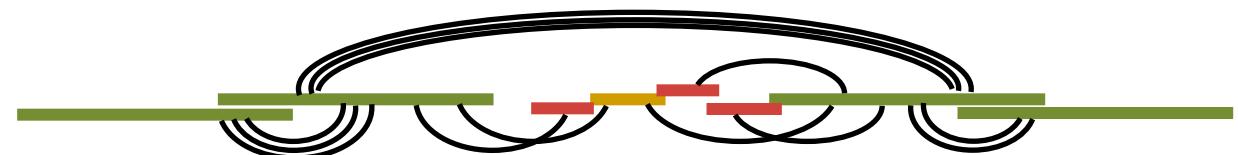
GGATGCGCGACACGT CGCATATCCGGTTGGT CAACCTCGGACGGAC

CAACCTCGGACGGAC CTCAGCGAA...

3. Simplify assembly graph



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



Why are genomes hard to assemble?

1. **Biological:**

- (Very) High ploidy, heterozygosity, repeat content



2. **Sequencing:**

- (Very) large genomes, imperfect sequencing

3. **Computational:**

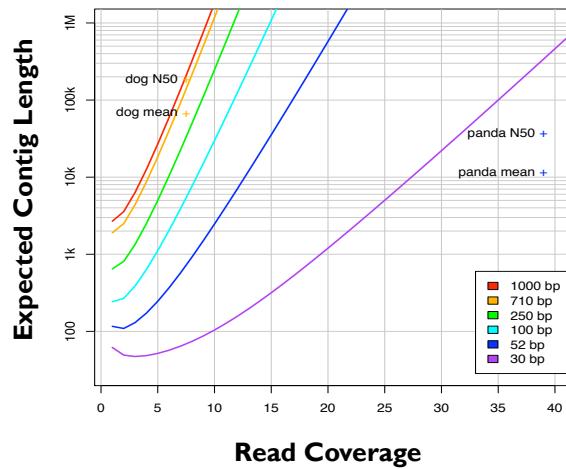
- (Very) Large genomes, complex structure

4. **Accuracy:**

- (Very) Hard to assess correctness

Ingredients for a good assembly

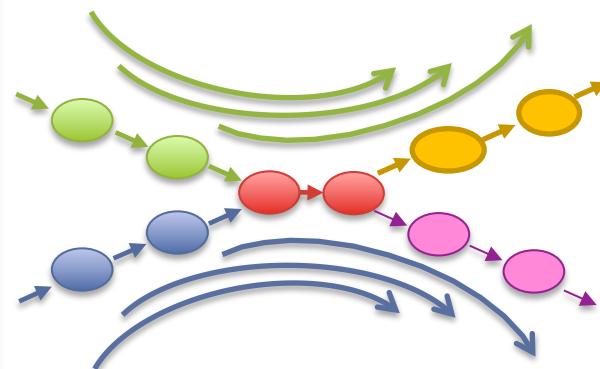
Coverage



High coverage is required

- Oversample the genome to ensure every base is sequenced with long overlaps between reads
- Biased coverage will also fragment assembly

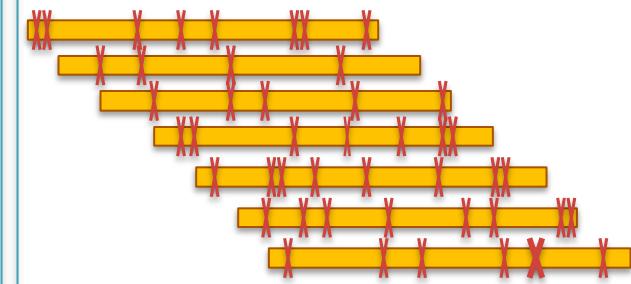
Read Length



Reads & mates must be longer than the repeats

- Short reads will have **false overlaps** forming hairball assembly graphs
- With long enough reads, assemble entire chromosomes into contigs

Quality



Errors obscure overlaps

- Reads are assembled by finding kmers shared in pair of reads
- High error rate requires very short seeds, increasing complexity and forming assembly hairballs

Current challenges in *de novo* plant genome sequencing and assembly
Schatz MC, Witkowski, McCombie, WR (2012) *Genome Biology*. 12:243

Second Generation Sequencing



2004
454/Roche
Pyrosequencing
Current Specs (Titanium):
1M 400bp reads / run =
1 Gbp / day

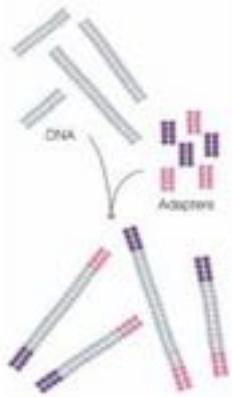


2007
Illumina
Sequencing by Synthesis
Current Specs (HiSeq 2000):
2.5B 100bp reads / run =
60Gbp / day

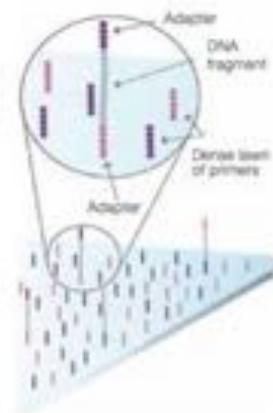


2008
ABI / Life Technologies
SOLiD Sequencing
Current Specs (5500xl):
5B 75bp reads / run =
30Gbp / day

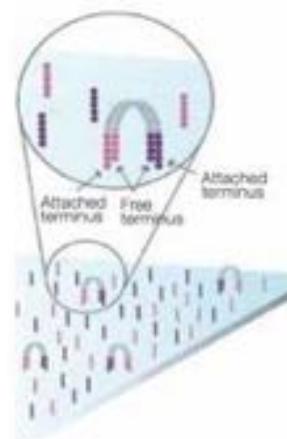
Illumina Sequencing by Synthesis



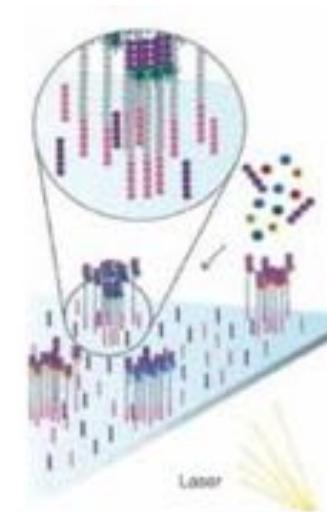
1. Prepare



2. Attach



3. Amplify



4. Image



5. Basecall

Metzker (2010) Nature Reviews Genetics 11:31-46
<http://www.youtube.com/watch?v=l99aKKHcxC4>

Paired-end and Mate-pairs

Paired-end sequencing

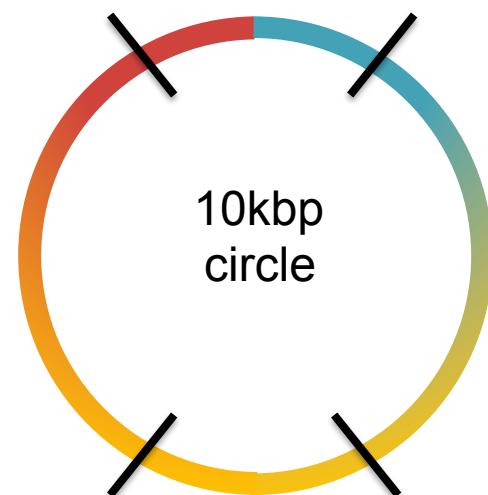
- Read one end of the molecule, flip, and read the other end
- Generate pair of reads separated by up to 500bp with inward orientation



Mate-pair sequencing

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

10kbp



2x100 @ ~10kbp (outies)

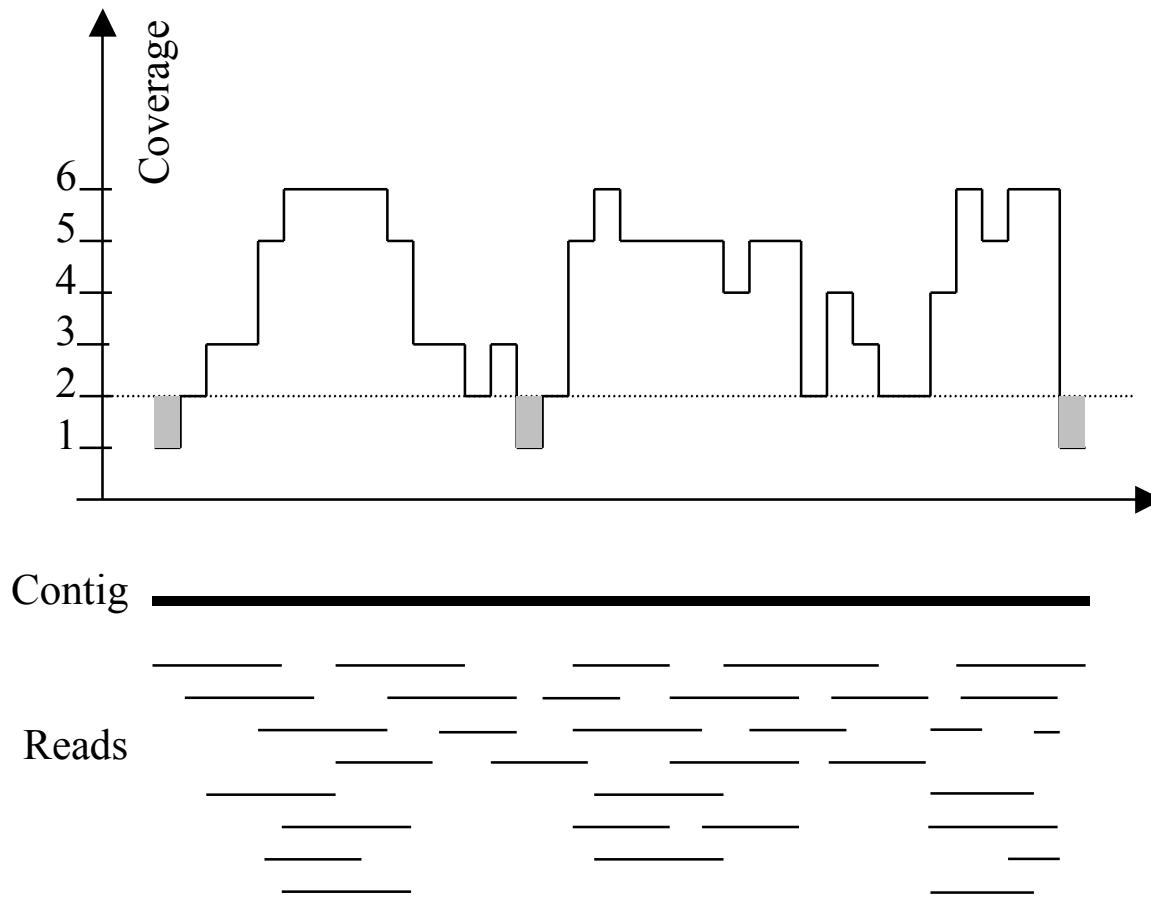


2x100 @ 300bp (innies)



Coverage

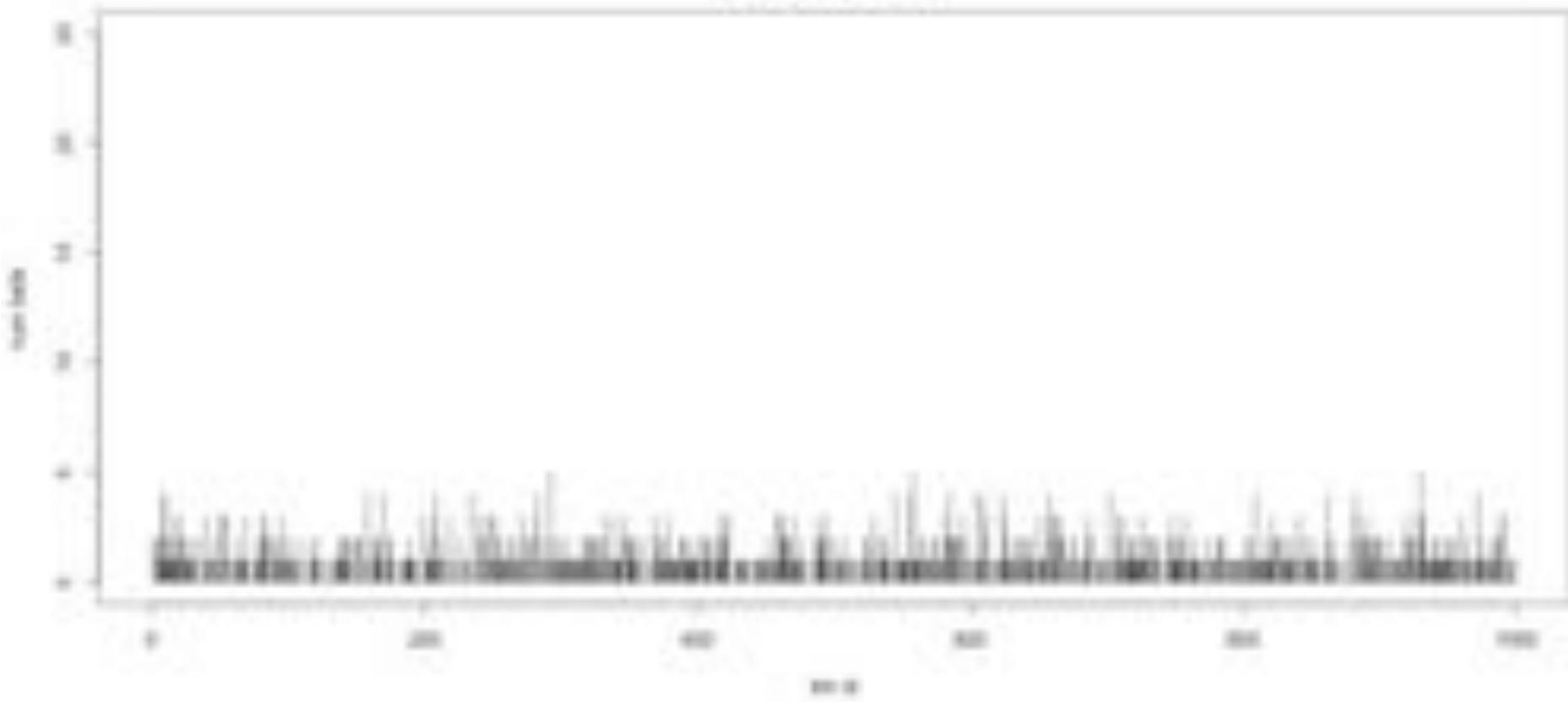
Typical contig coverage



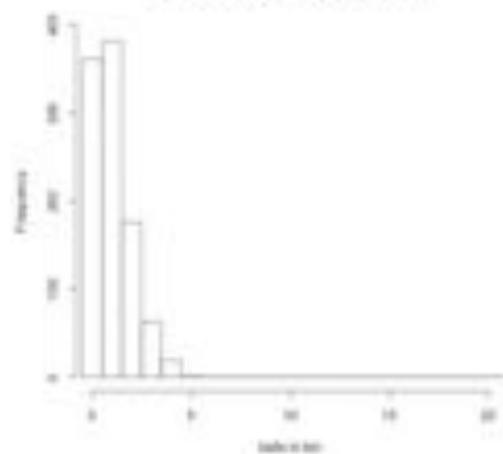
Imagine raindrops on a sidewalk

Balls in Bins IX

Balls in Bins
Total balls: 1000

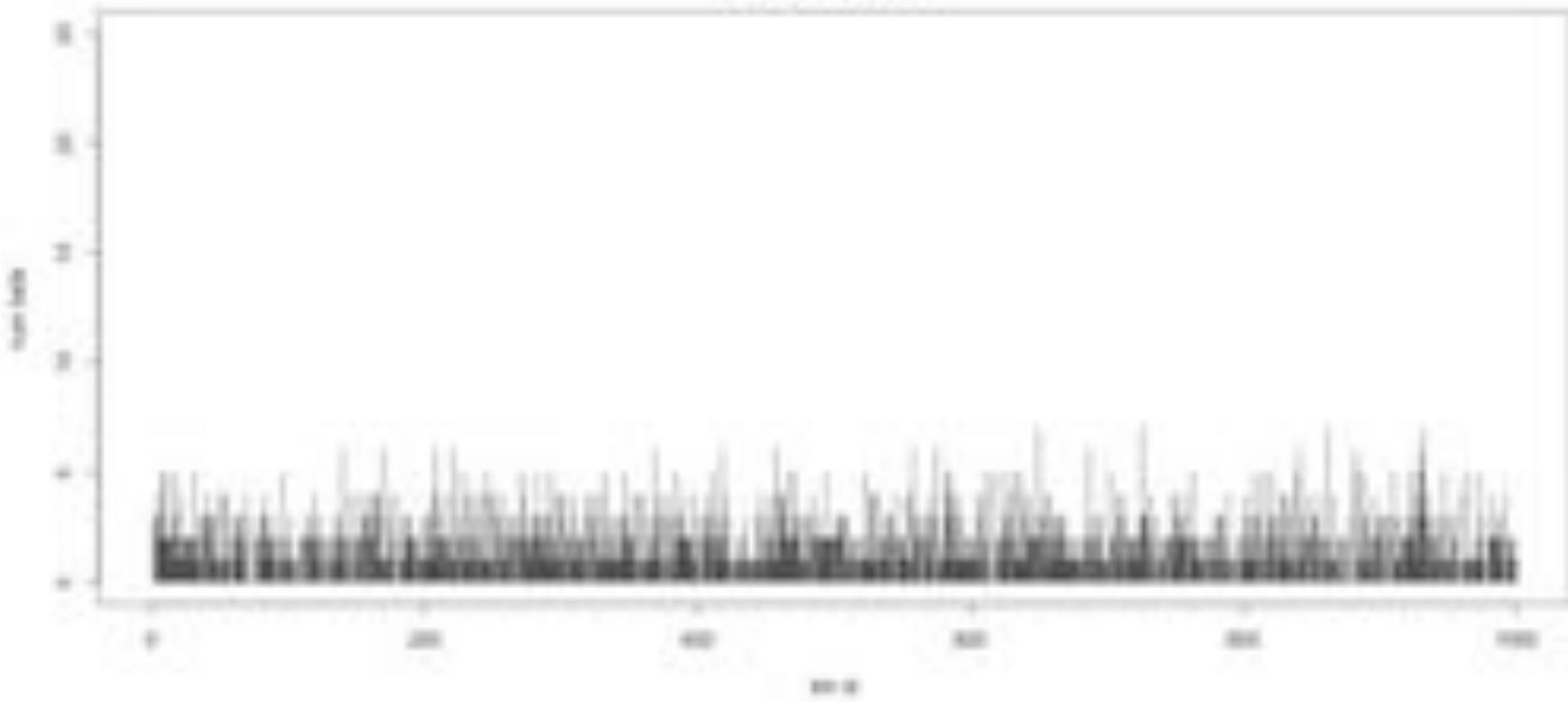


Histogram of balls in each bin
Total balls: 1000. Empty bins: 301

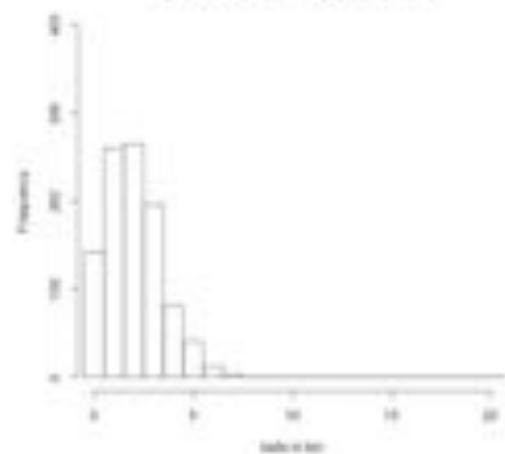


Balls in Bins 2x

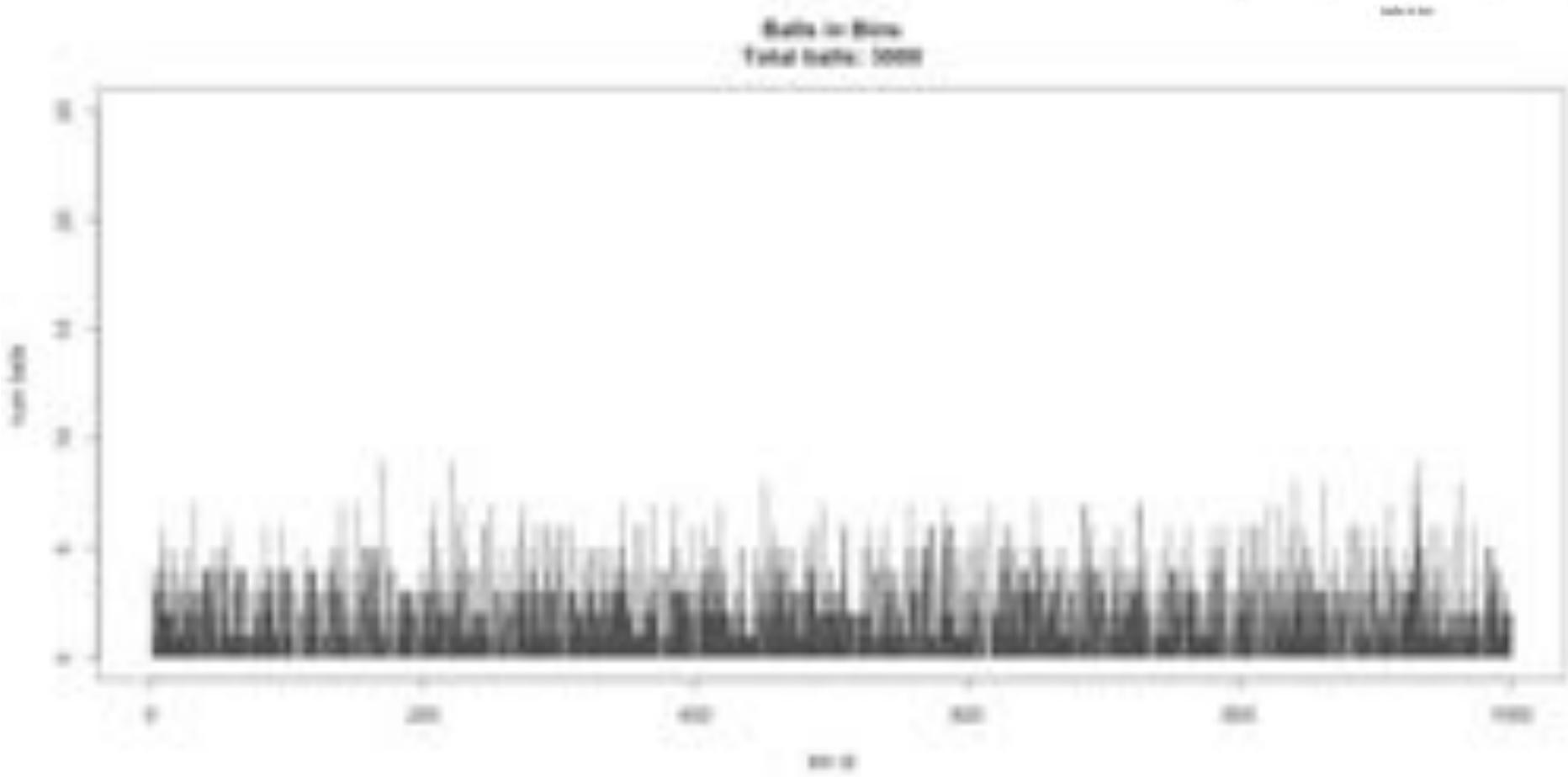
Balls in Bins
Total balls: 2000



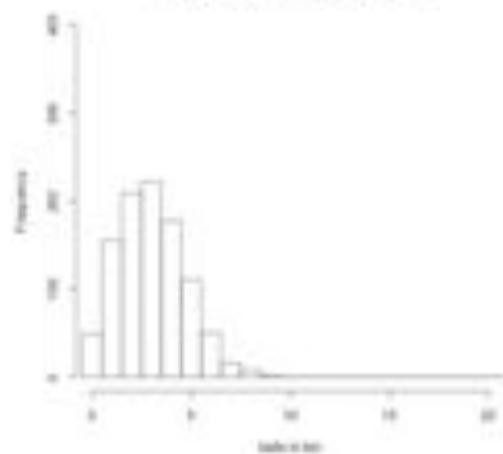
Histogram of balls in each bin
Total balls: 2000. Empty bins: 142



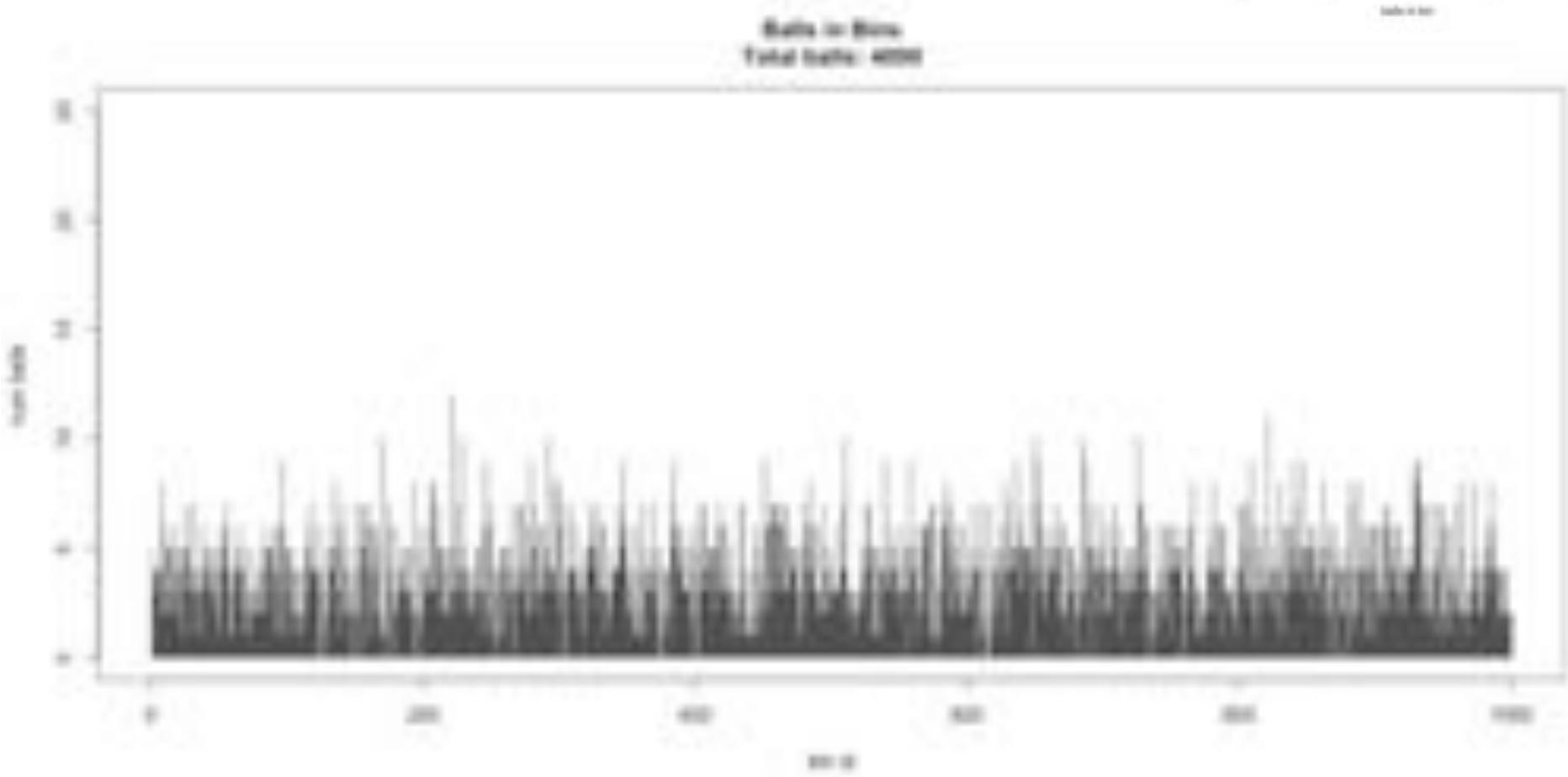
Balls in Bins 3x



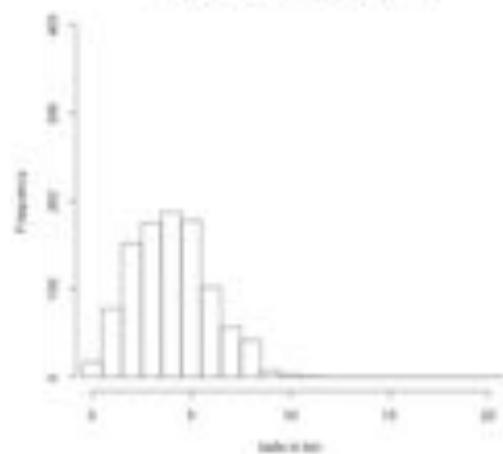
Histogram of balls in each bin
Total balls: 3000 Empty bins: 49



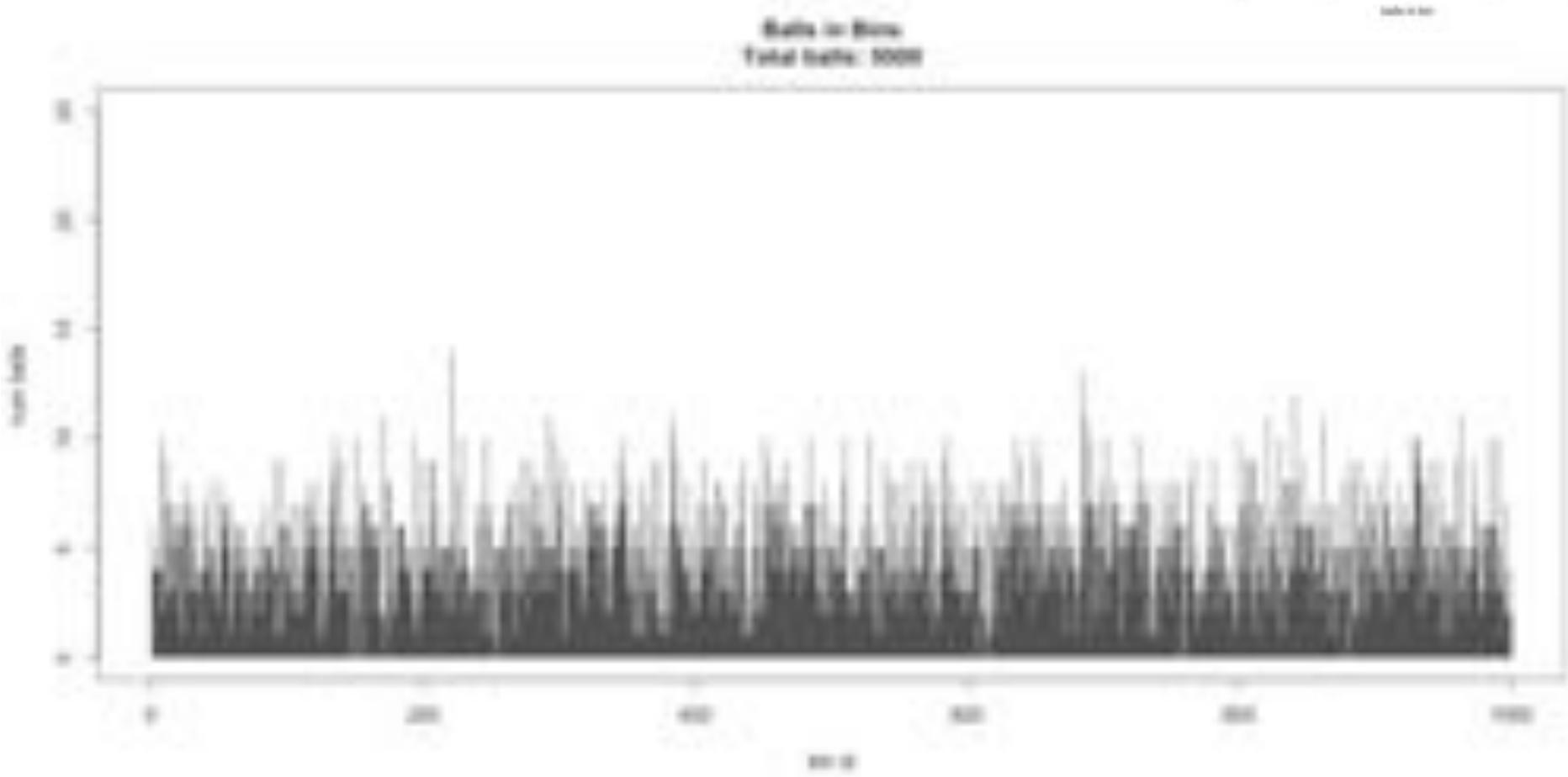
Balls in Bins 4x



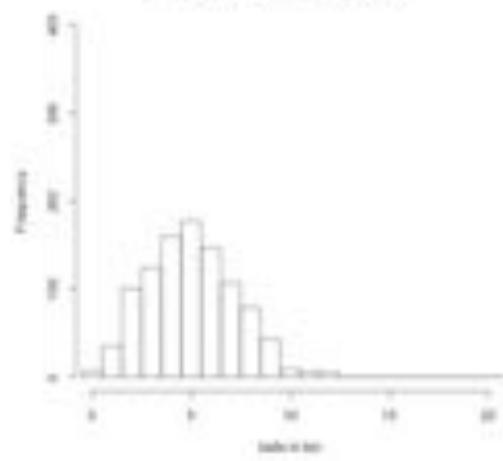
Histogram of balls in each bin
Total balls: 4000 Empty bins: 17



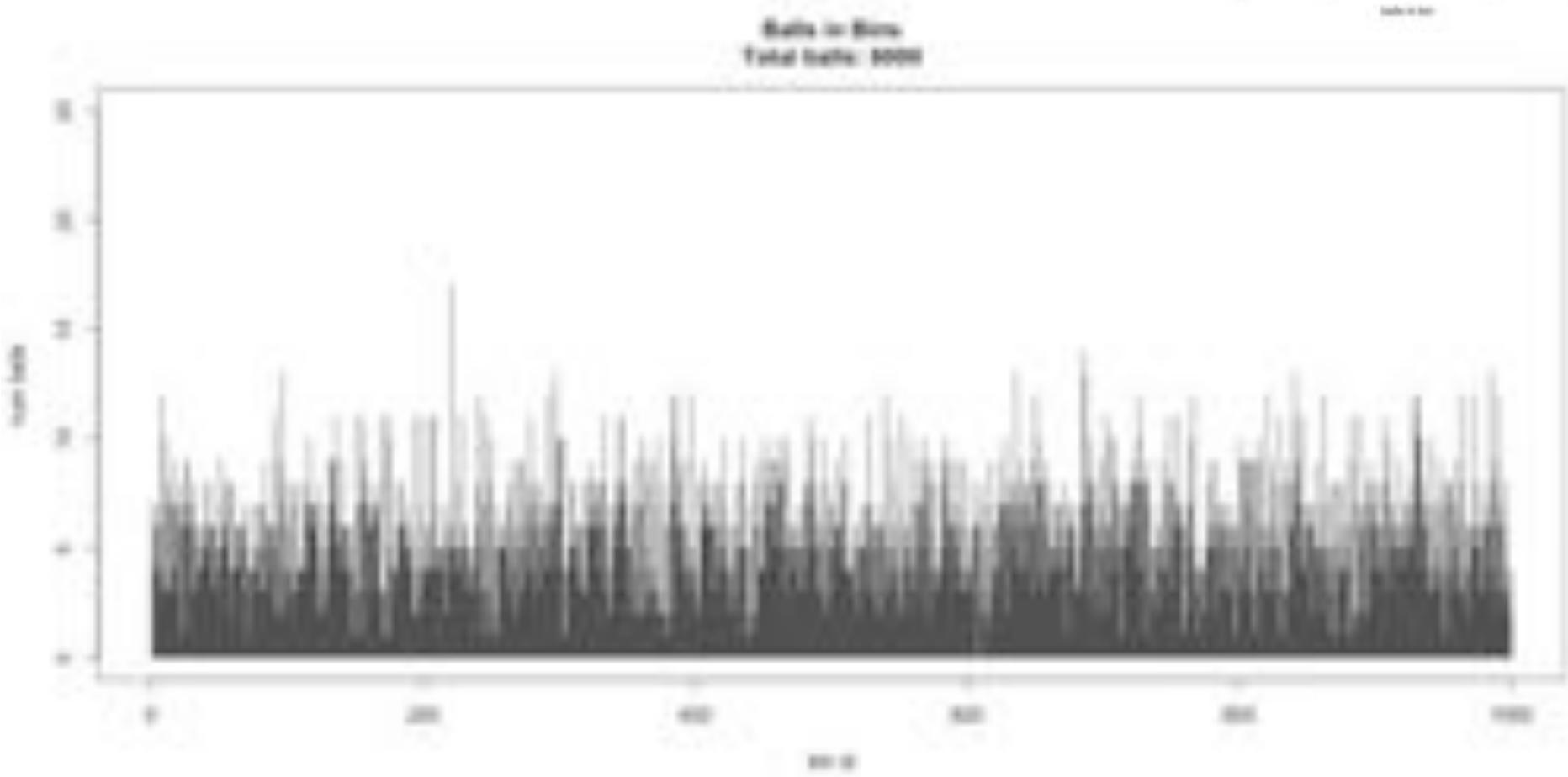
Balls in Bins 5x



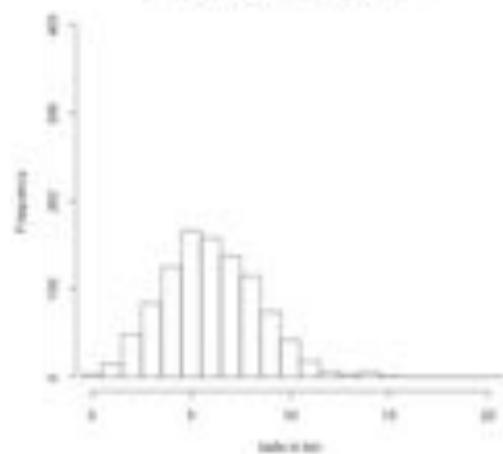
Histogram of balls in each bin
Total balls: 5000 Empty bins: 7



Balls in Bins 6x

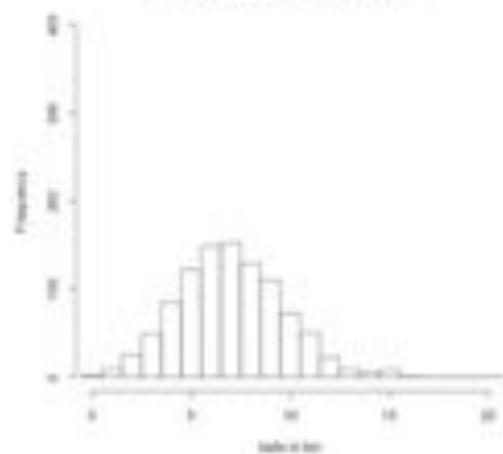


Histogram of balls in each bin
Total balls: 6000 Empty bins: 2

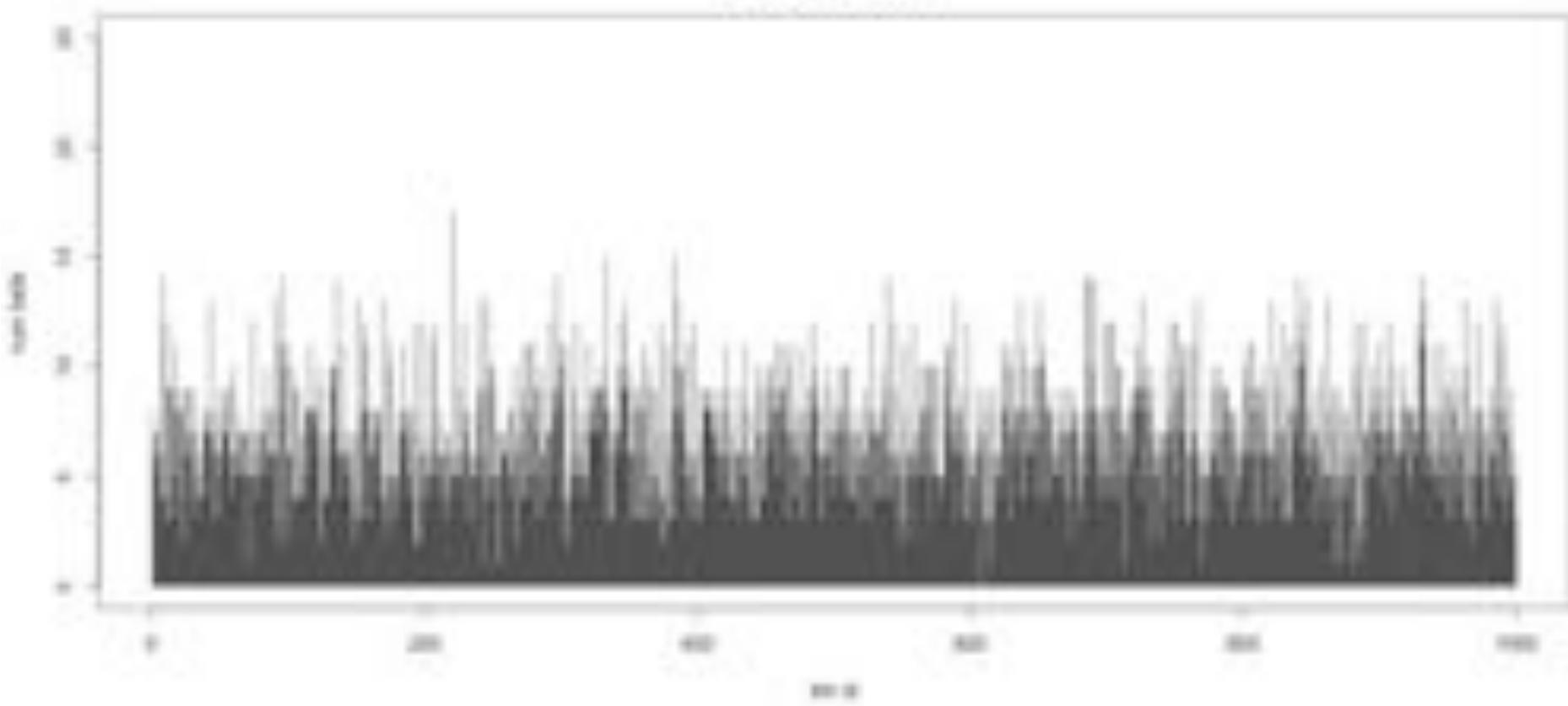


Balls in Bins 7x

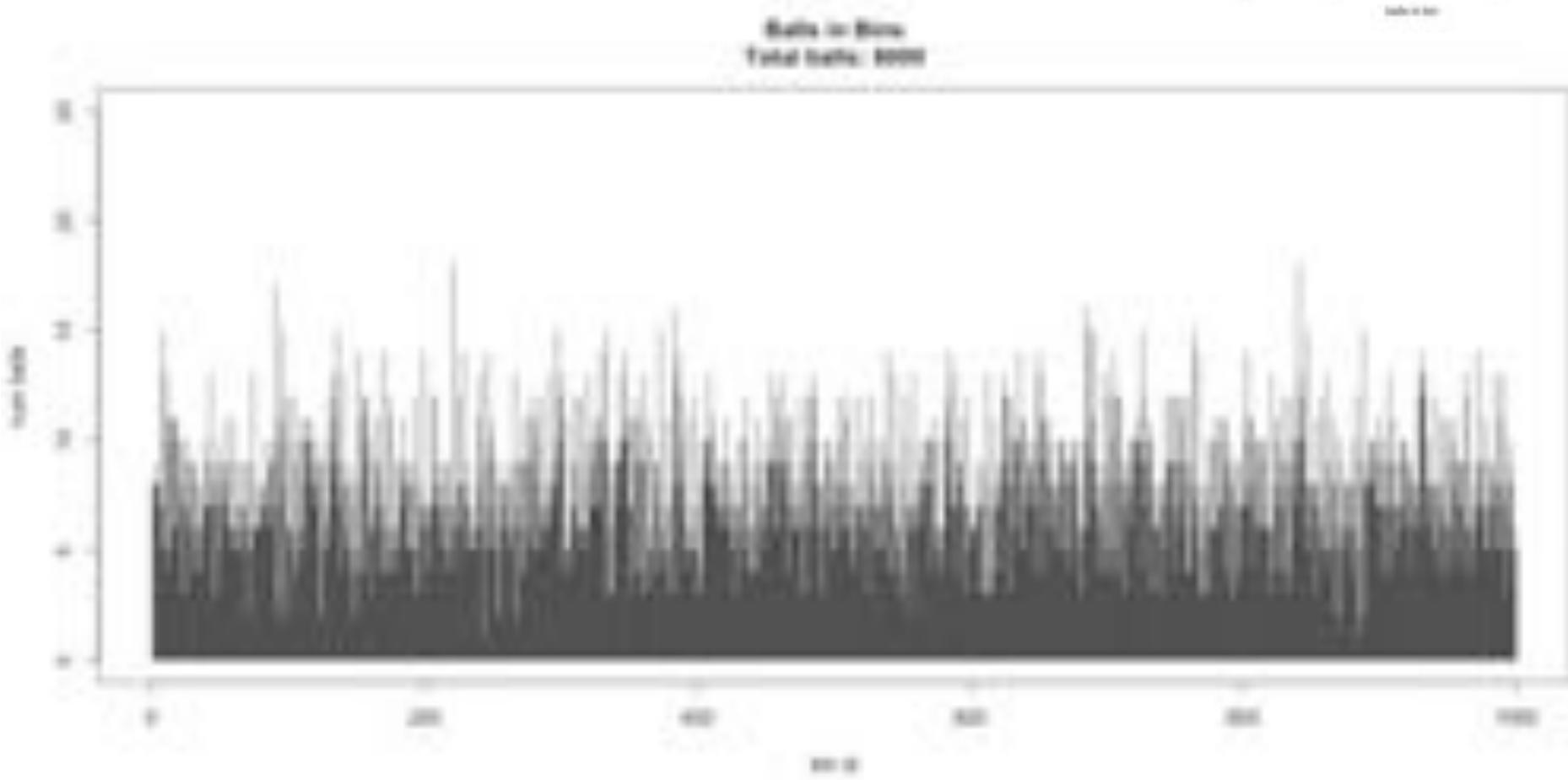
Histogram of balls in each bin
Total balls: 7000 Empty bins: 2



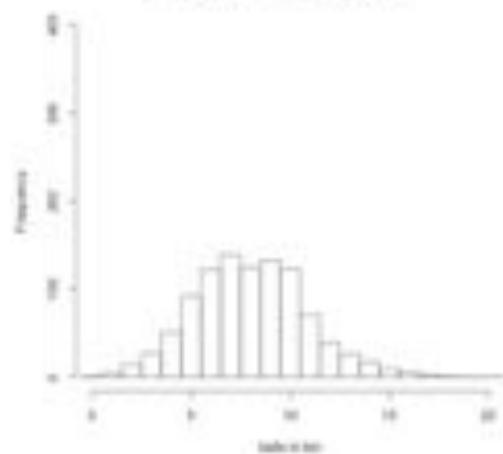
Balls in Bins
Total balls: 7000



Balls in Bins 8x



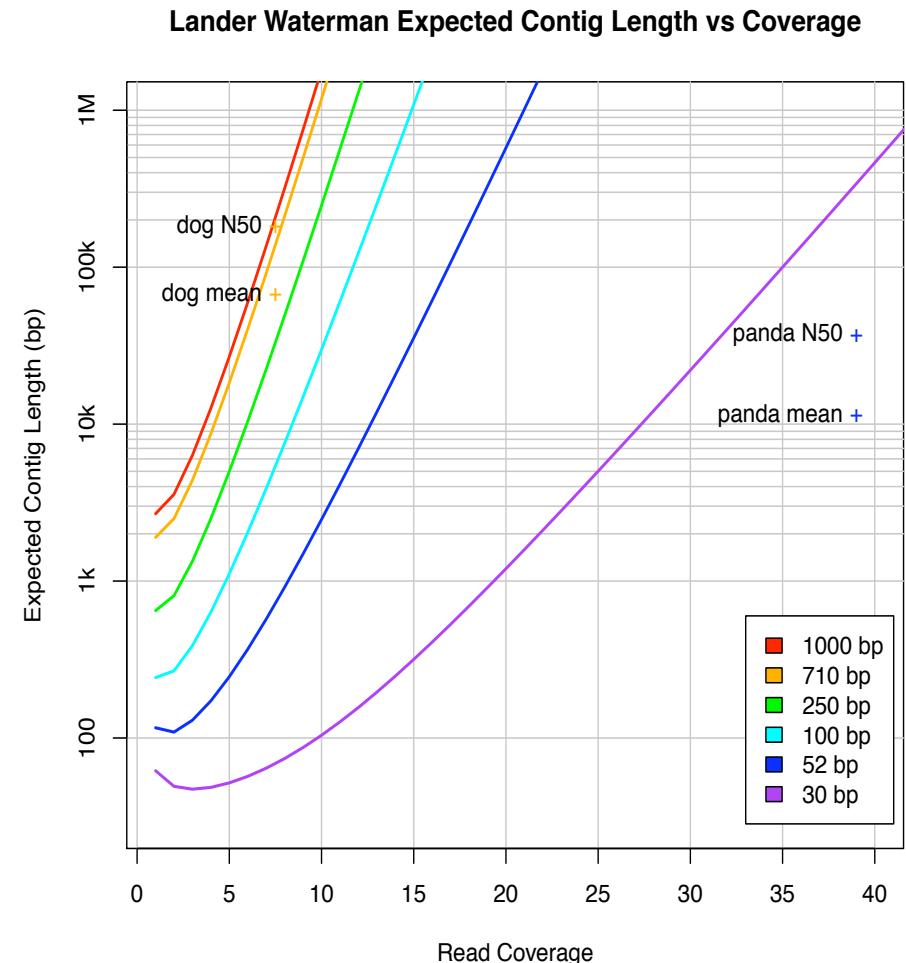
Histogram of balls in each bin
Total balls: 8000 Empty bins: 0



Coverage and Read Length

Idealized Lander-Waterman model

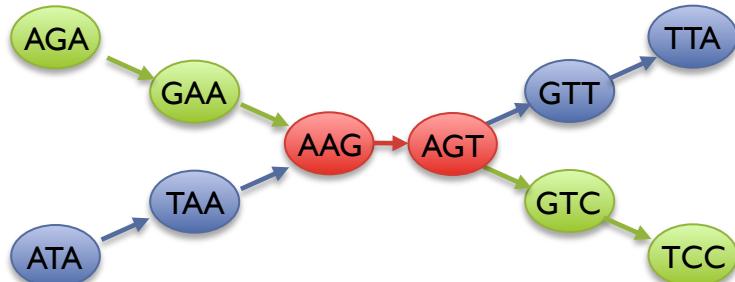
- Reads start at perfectly random positions
- Contig length is a function of coverage and read length
 - Short reads require much higher coverage to reach same expected contig length
- Need even high coverage for higher ploidy, sequencing errors, sequencing biases
 - Recommend 100x coverage



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

Two Paradigms for Assembly

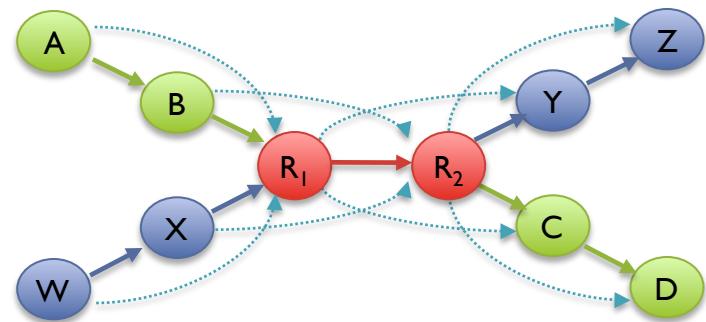
de Bruijn Graph



Short read assemblers

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

Overlap Graph



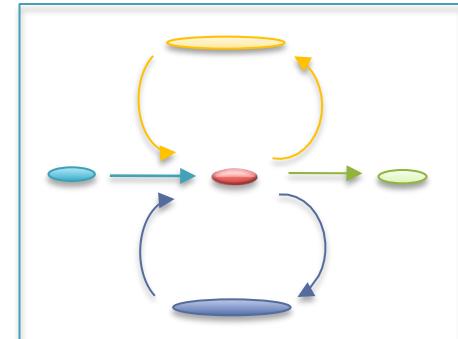
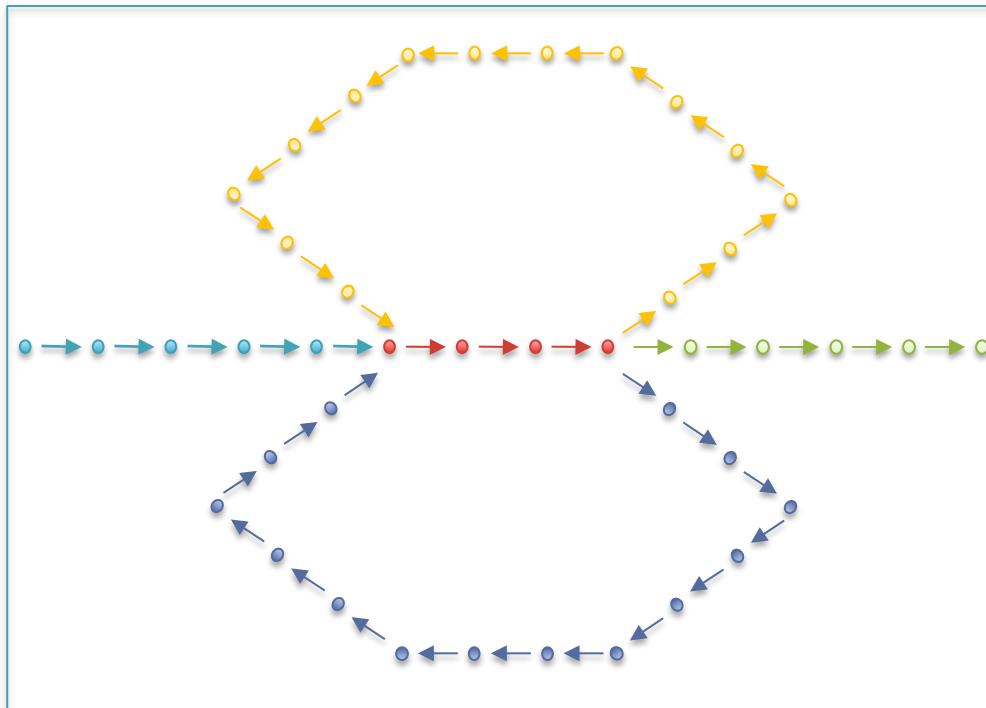
Long read assemblers

- 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.

Unitigging / Unipathing

- After simplification and correction, compress graph down to its non-branching initial contigs
 - Aka “unitigs”, “unipaths”
 - Unitigs end because of (1) lack of coverage, (2) errors, and (3) repeats



Errors in the graph



(Chaisson, 2009)

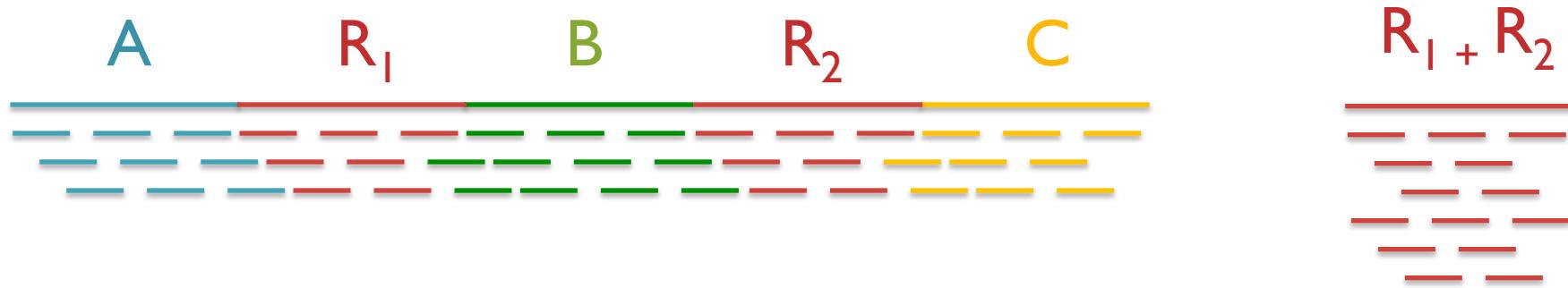
Clip Tips	Pop Bubbles
<p>was the worst of times,</p> <p>was the worst of tymes,</p> <p>the worst of times, it</p>	<p>was the worst of times,</p> <p>was the worst of tymes,</p> <p>times, it was the age</p> <p>tymes, it was the age</p>
<p>the worst of tymes,</p> <p>was the worst of</p> <p>the worst of times,</p> <p>worst of times, it</p>	<p>tymes,</p> <p>was the worst of</p> <p>it was the age</p> <p>times,</p>

Repetitive regions

Repeat Type	Definition / Example	Prevalence
Low-complexity DNA / Microsatellites	$(b_1 b_2 \dots b_k)^N$ where $1 \leq k \leq 6$ CACACACACACACACACACA	2%
SINEs (Short Interspersed Nuclear Elements)	<i>Alu</i> 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%

- Over 50% of mammalian genomes are repetitive
 - Large plant genomes tend to be even worse
 - Wheat: 16 Gbp; Pine: 24 Gbp

Repeats and Coverage Statistics

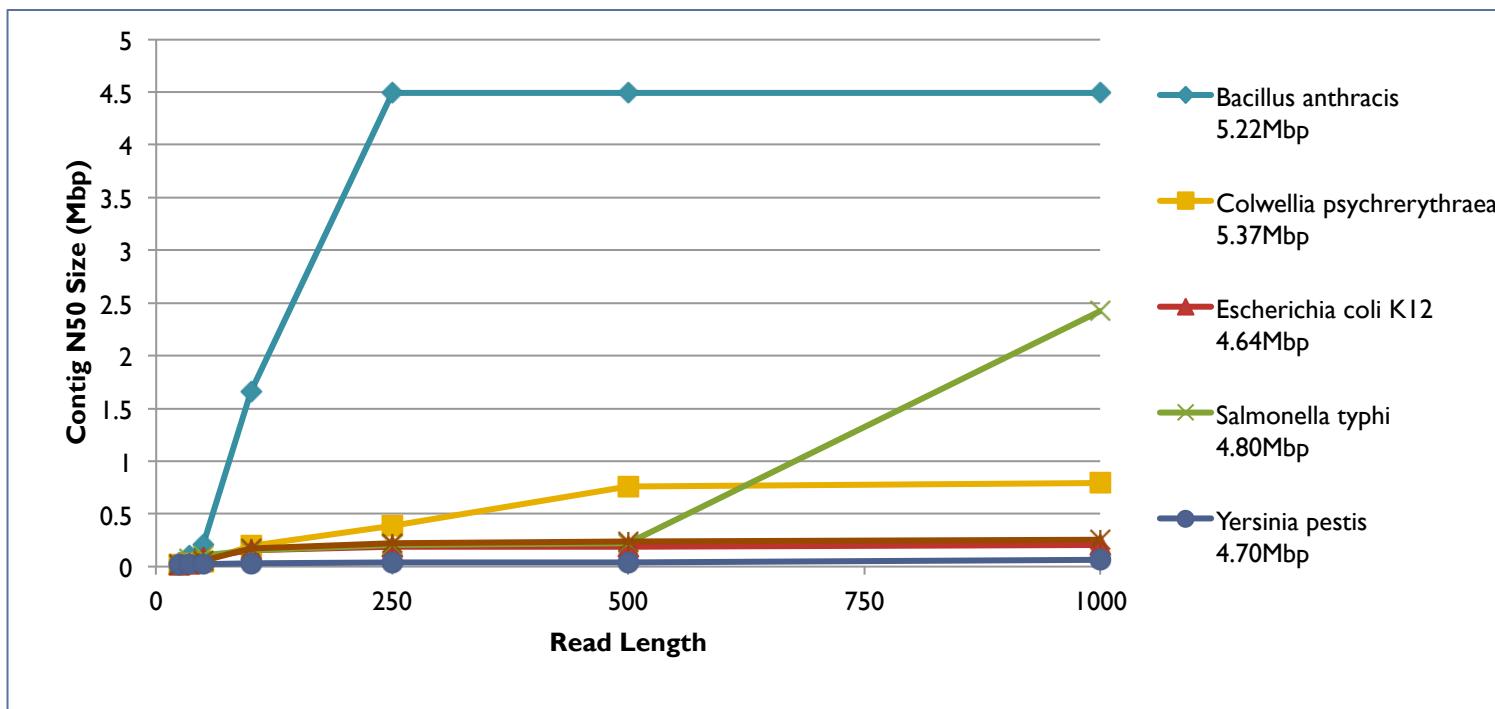


- If n reads are a uniform random sample of the genome of length G , we expect $k = n \Delta/G$ reads to start in a region of length Δ .
 - If we see many more reads than k (if the arrival rate is $> A$), it is likely to be a collapsed repeat
 - Requires an accurate genome size estimate

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

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

Repeats and Read Length

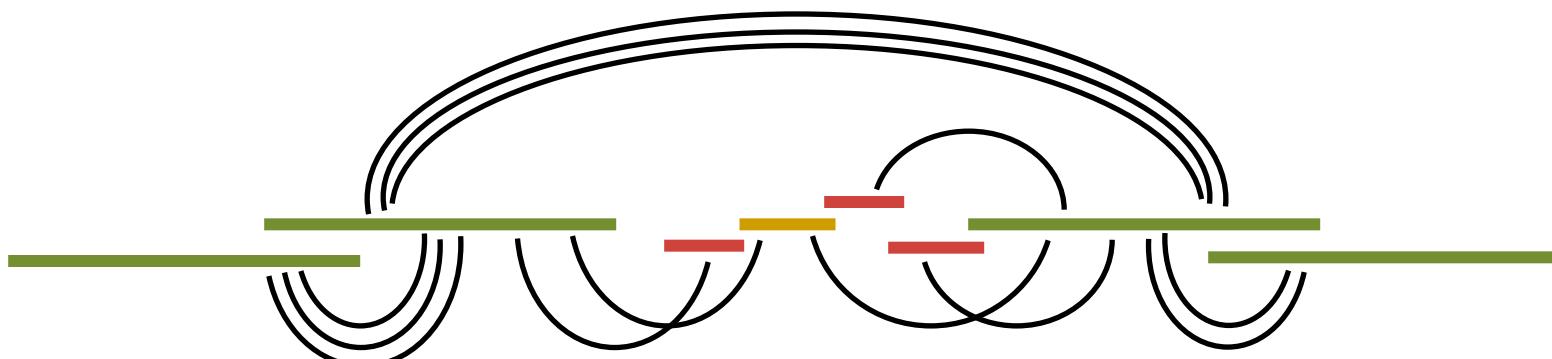


- Explore the relationship between read length and contig N50 size
 - Idealized assembly of read lengths: 25, 35, 50, 100, 250, 500, 1000
 - Contig/Read length relationship depends on specific repeat composition

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

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 as large as the N50 value

Example: 1 Mbp genome



N50 size = 30 kbp

$$(300k+100k+45k+45k+30k = 520k \geq 500\text{ kbp})$$

Note:

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

Break





Outline

I. Assembly theory

1. Assembly by analogy
2. De Bruijn and Overlap graph
3. Coverage, read length, errors, and repeats

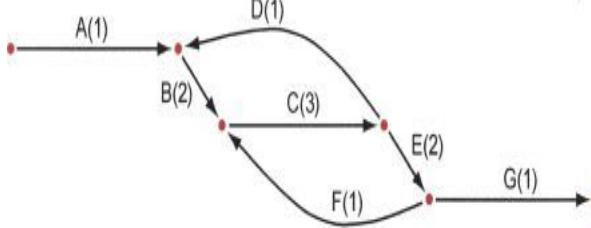
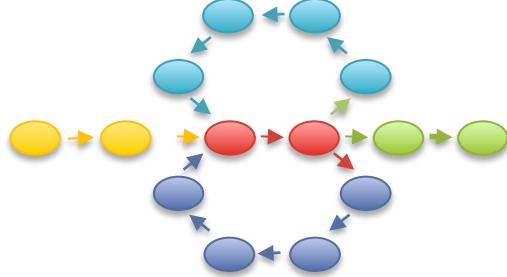
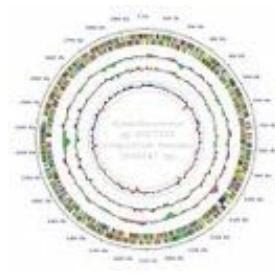
2. Genome assemblers

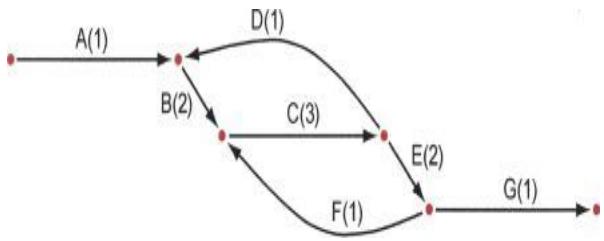
1. ALLPATHS-LG, SOAPdenovo, Celera Assembler
2. Assemblathon

3. Applications

1. Whole Genome Alignment with MUMmer
2. Gene Finding

Assembly Algorithms

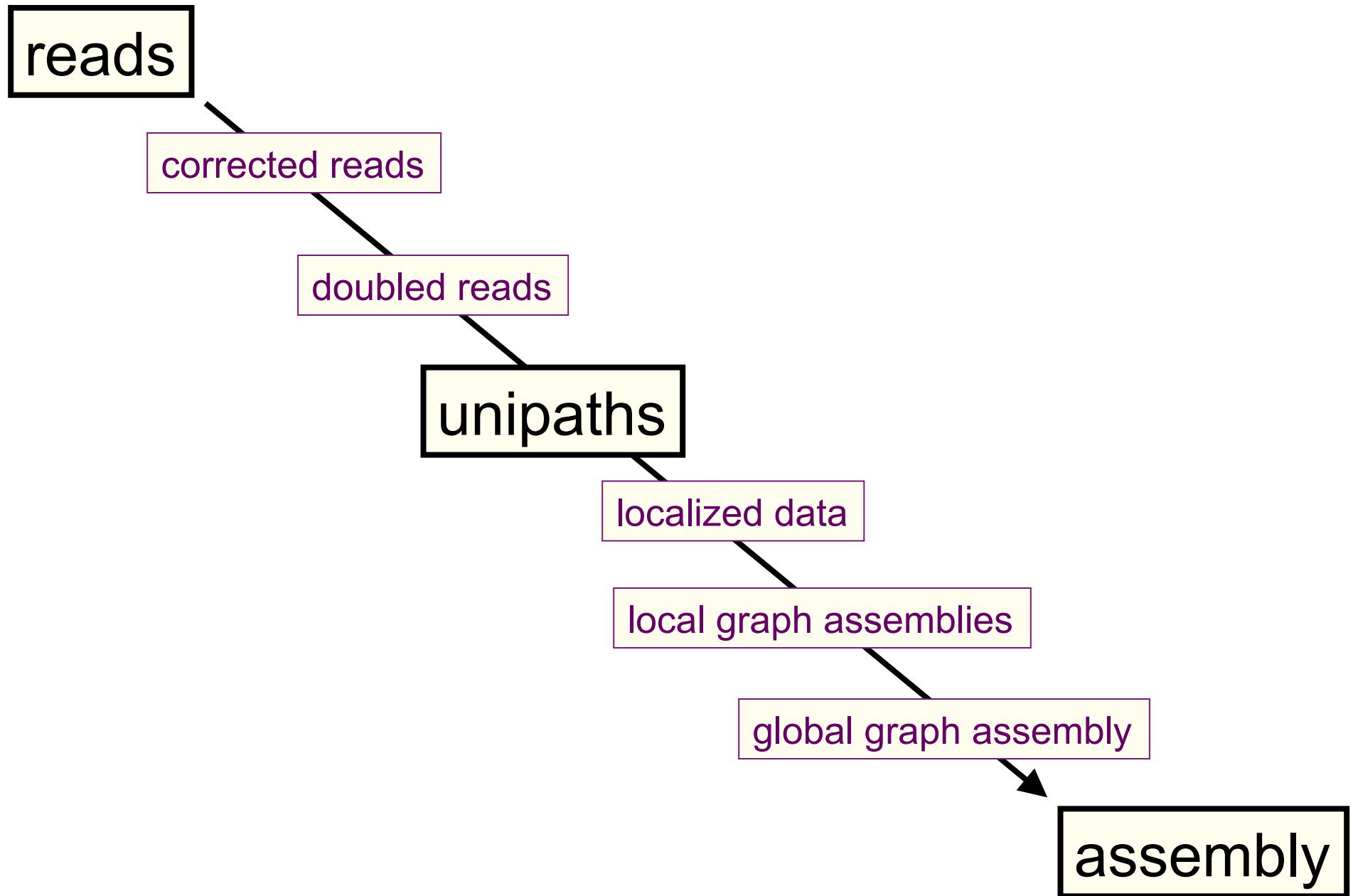
ALLPATHS-LG	SOAPdenovo	Celera Assembler
		
Broad's assembler (Gnerre et al. 2011)	BGI's assembler (Li et al. 2010)	JCVI's assembler (Miller et al. 2008)
De bruijn graph Short + PacBio (patching)	De bruijn graph Short reads	Overlap graph Medium + Long reads
Easy to run if you have compatible libraries	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



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.

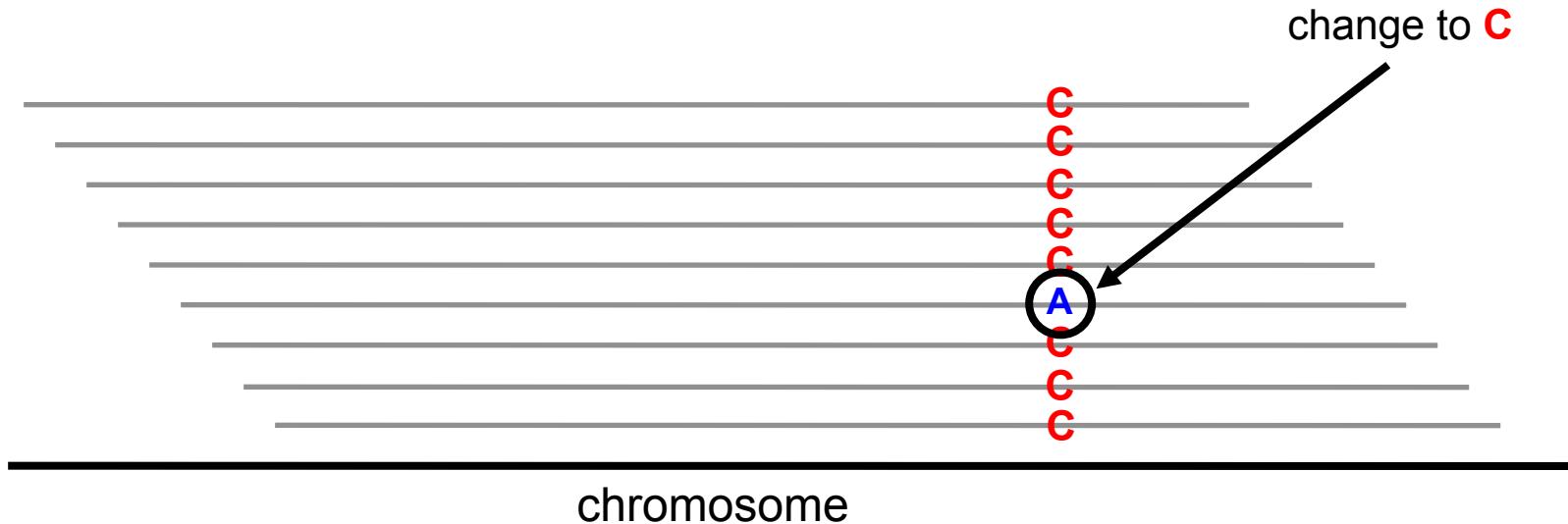
**For best results. Normally not used for small genomes.
However essential to assemble long repeats or duplications.

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

All: protocols are either available, or in progress.

Error correction

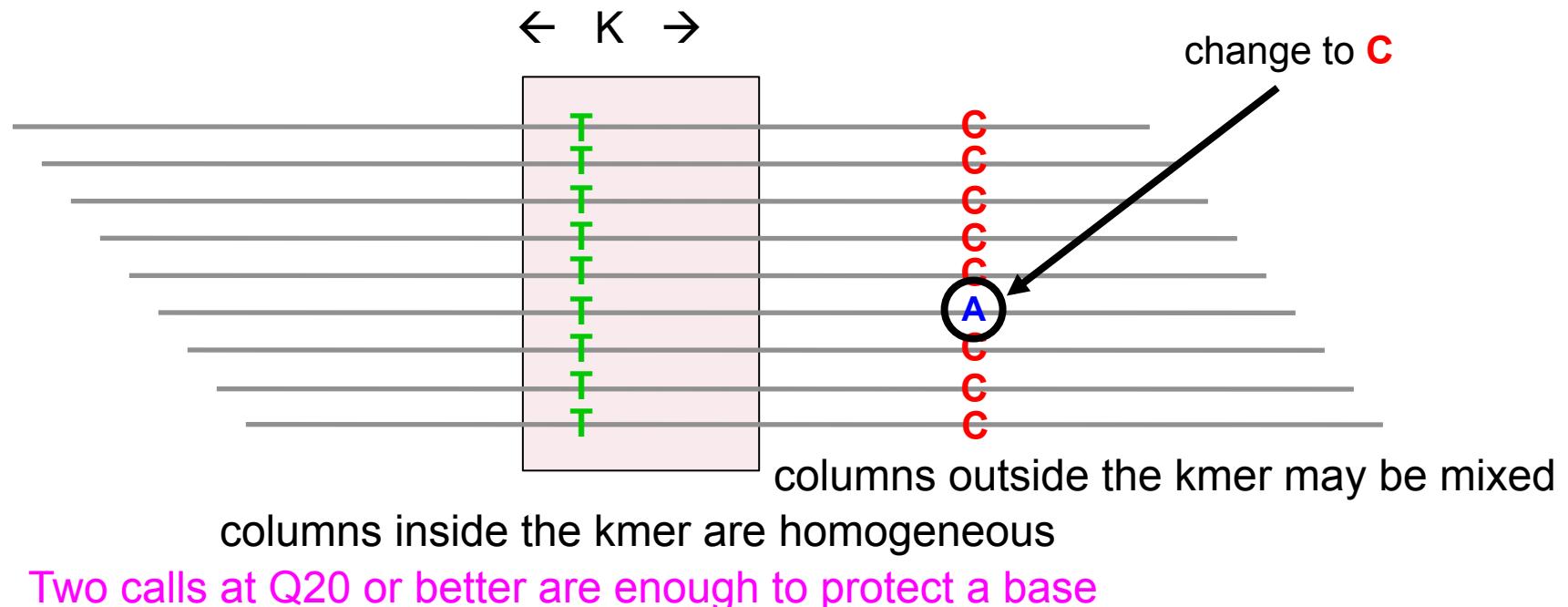
Given a crystal ball, we could stack reads on the chromosomes they came from (with homologous chromosomes separate), then let each column ‘vote’:



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

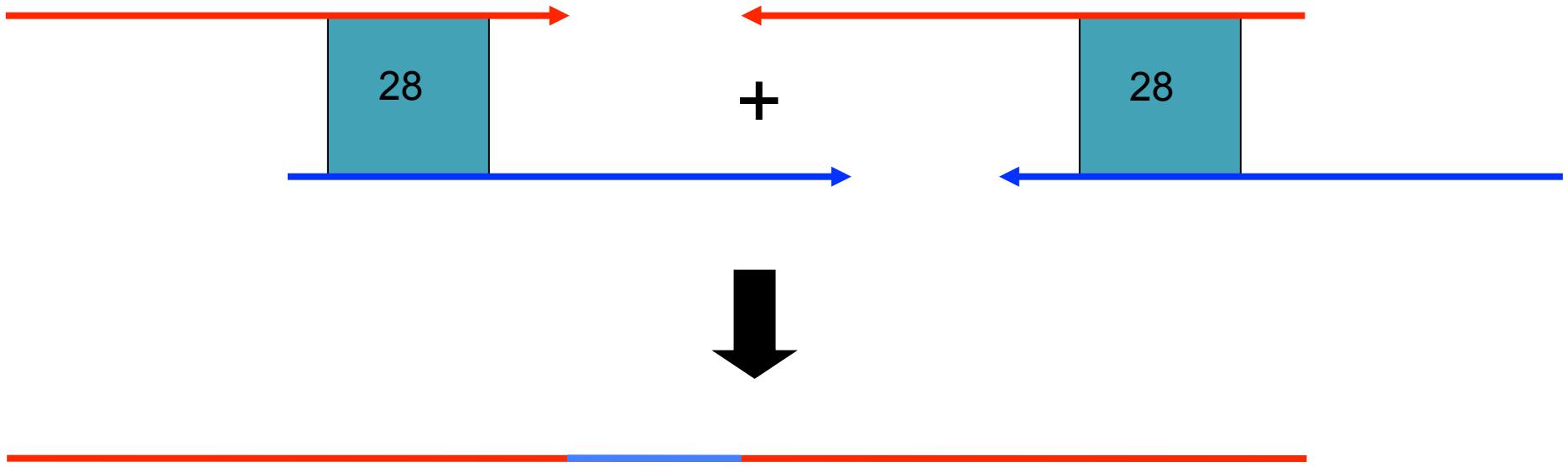
Error correction

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)



Read doubling

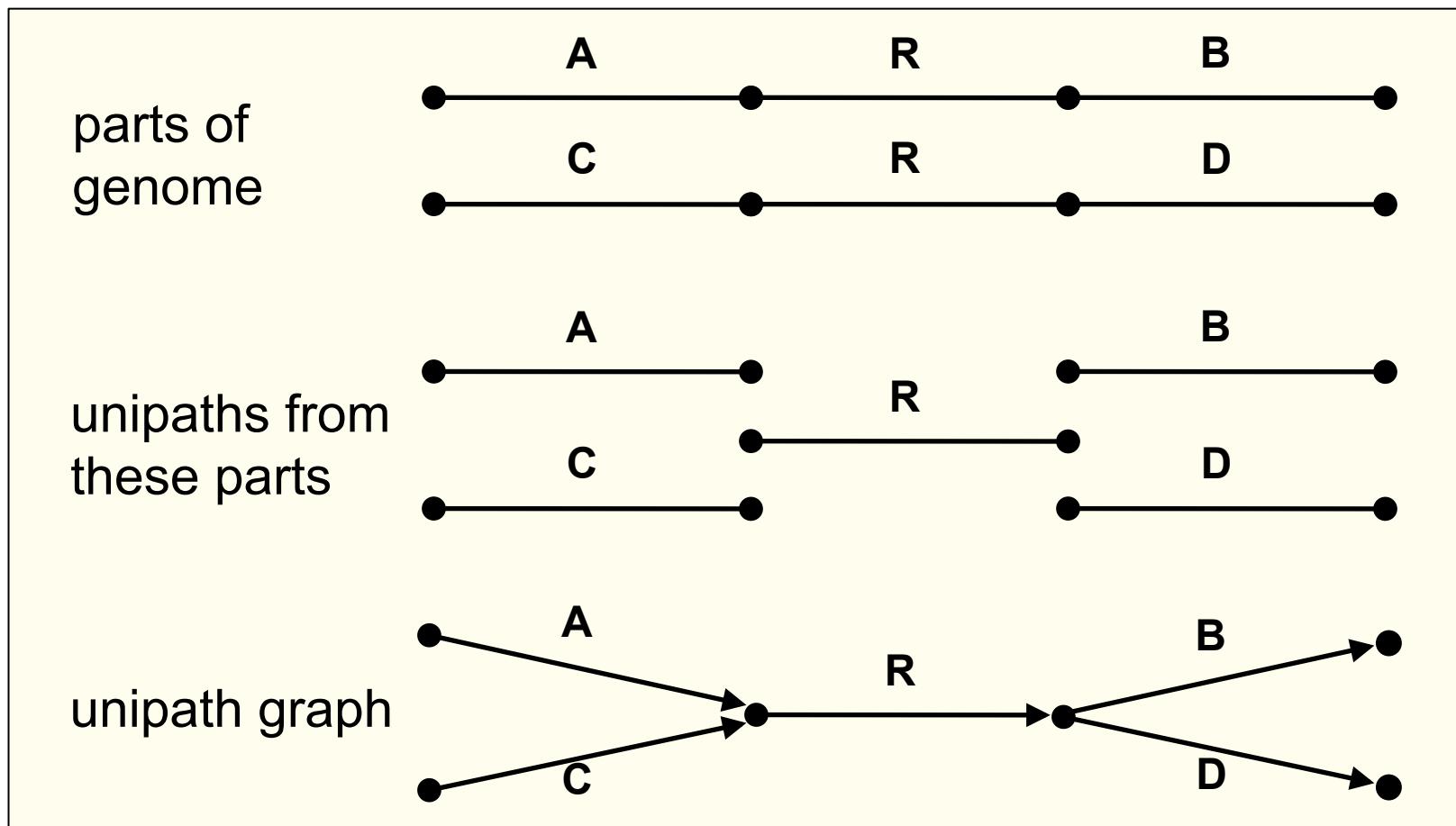
To close a read pair (red), we require the existence of another read pair (blue), overlapping perfectly like this:



More than one closure allowed (but rare).

Unipaths

Unipath: unbranched part of genome – squeeze together perfect repeats of size $\geq K$



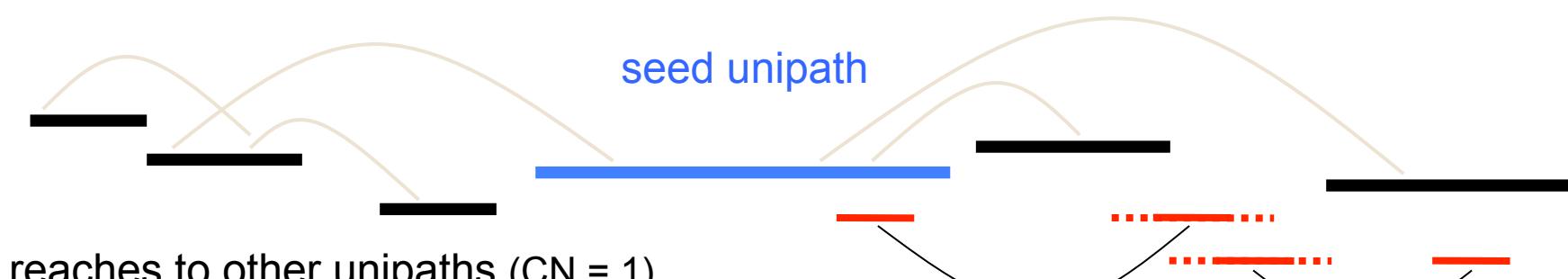
Adjacent unipaths overlap by $K-1$ bases

Localization

- I. Find ‘seed’ unipaths, evenly spaced across genome**
(ideally long, of copy number CN = 1)



- II. Form neighborhood around each seed**



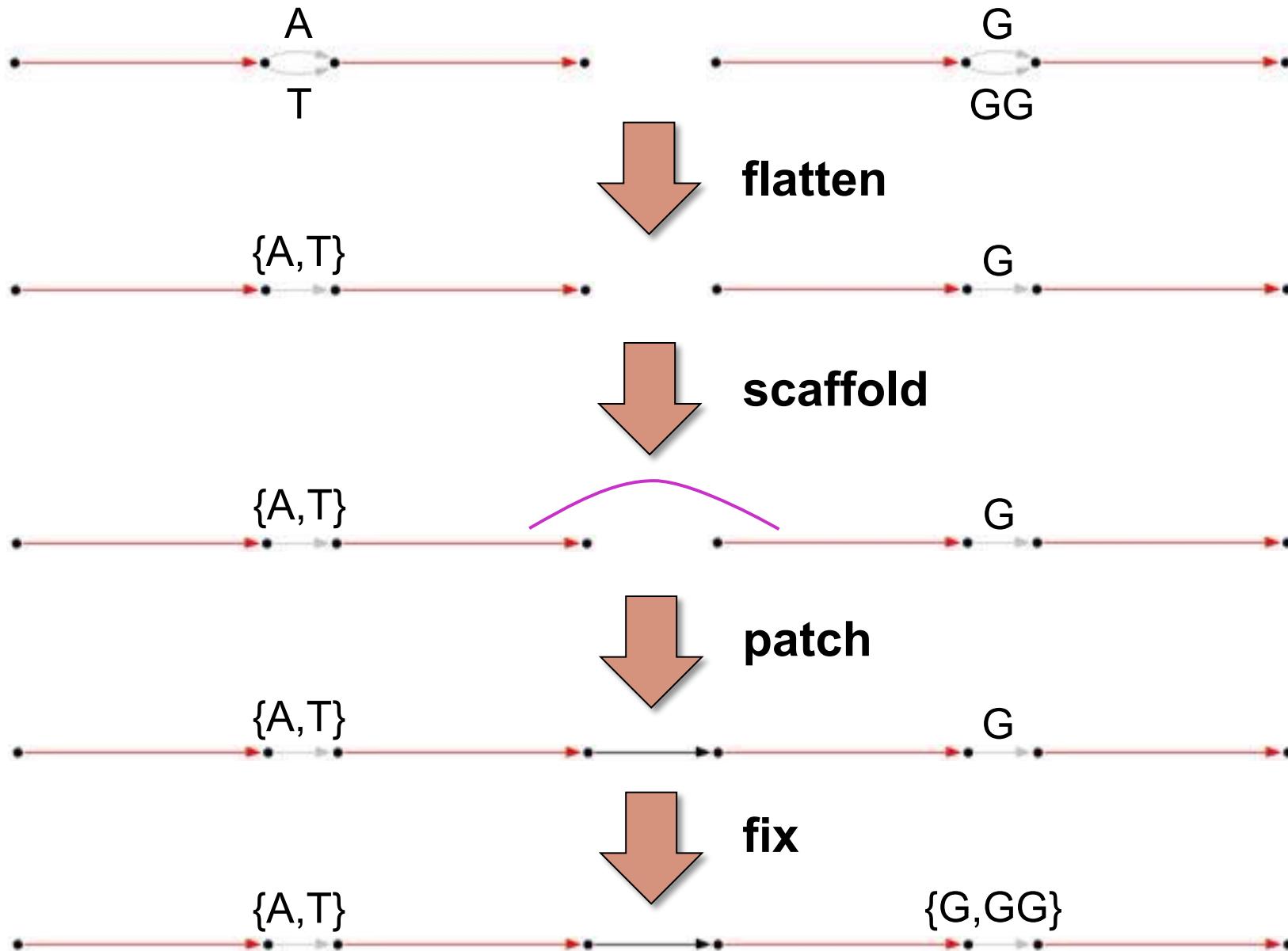
reaches to other unipaths (CN = 1)
directly and indirectly

read pairs reach into repeats

and are extended by other
unipaths

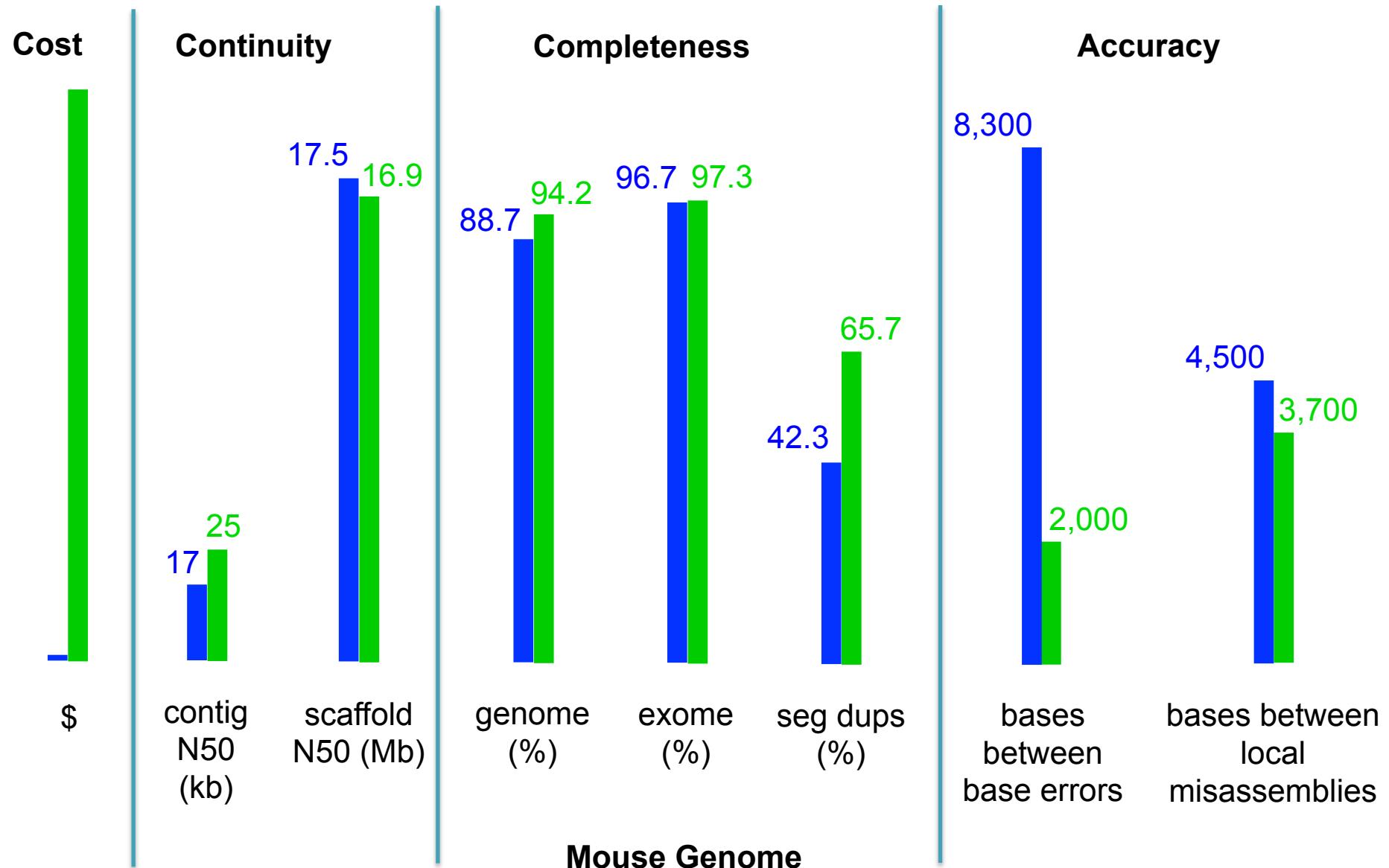
.....

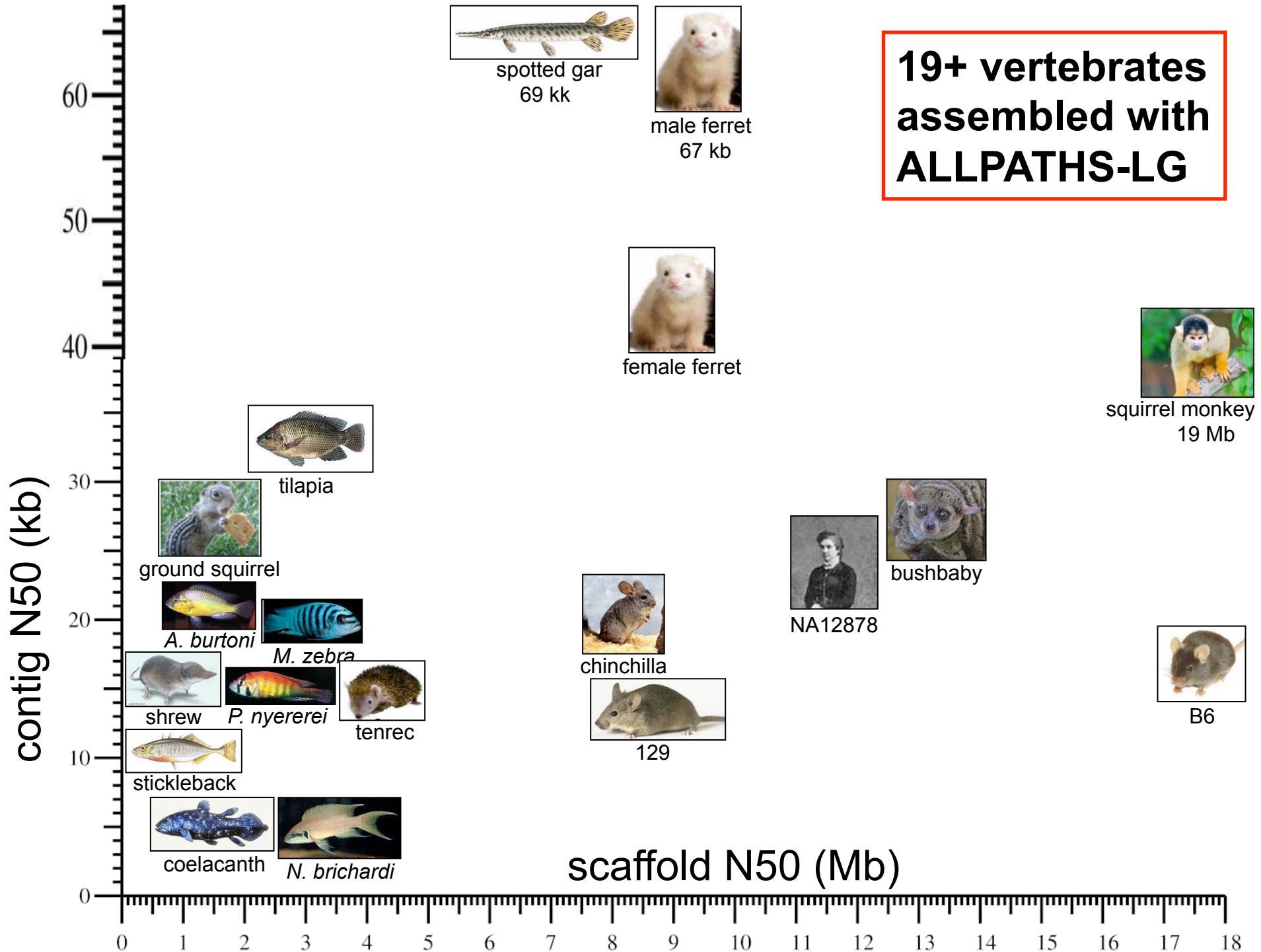
Create assembly from global assembly graph

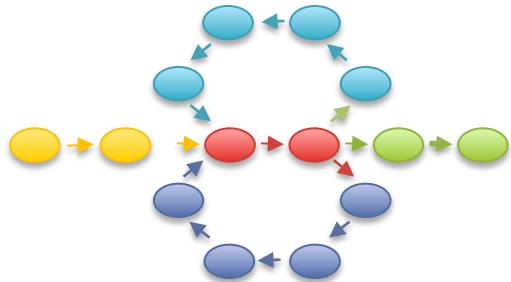




Large genome recipe: ALLPATHS-LG vs capillary







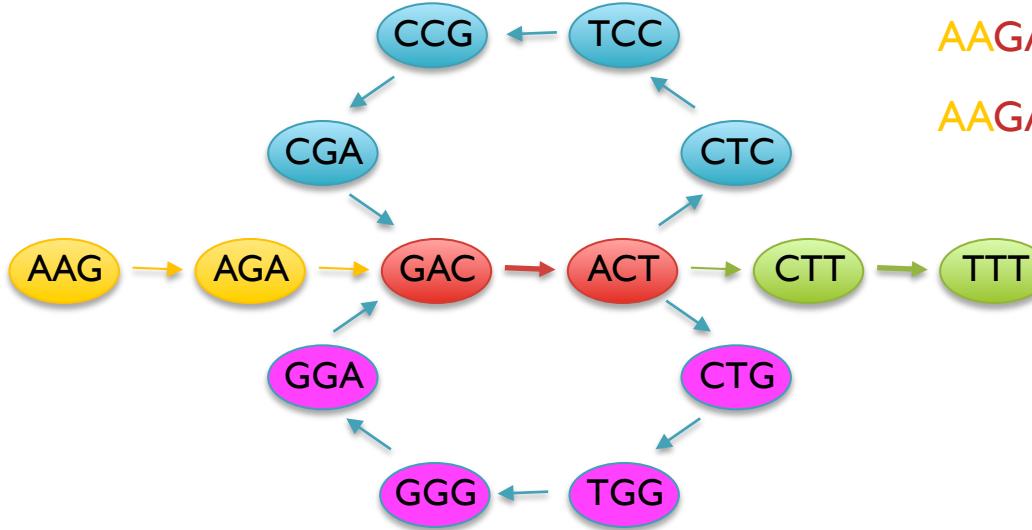
Genome assembly with SOAPdenovo

Short Read Assembly

Reads

AAGA
ACTT
ACTC
ACTG
AGAG
CCGA
CGAC
CTCC
CTGG
CTTT
...

de Bruijn Graph



Potential Genomes

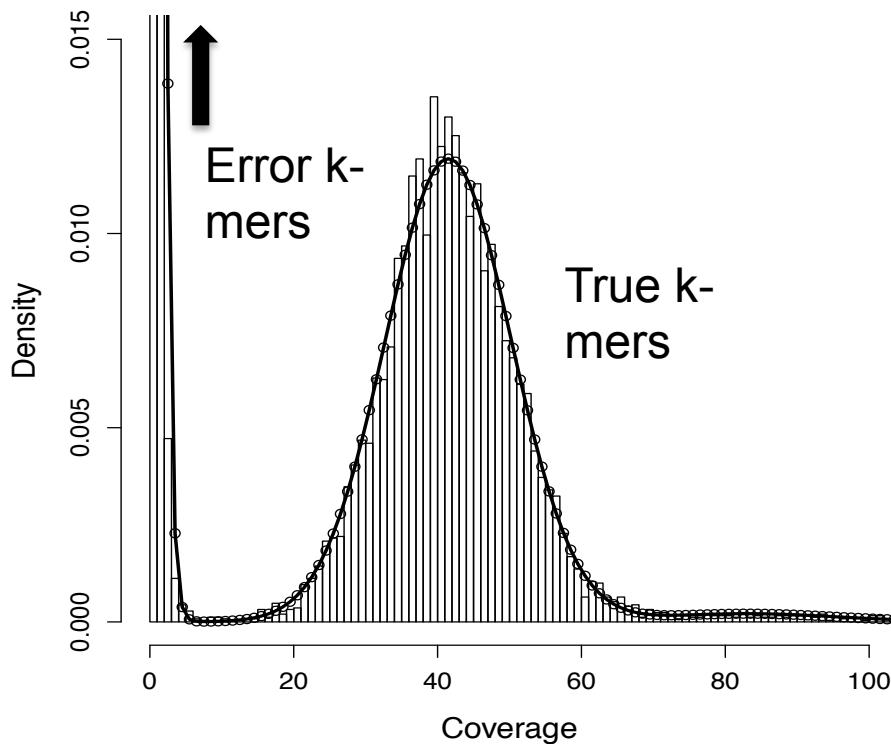
AAGACTCCGACTGGGACTTT
AAGACTGGGACTCCGACTTT

- Genome assembly as finding an Eulerian tour of the de Bruijn graph
 - Human genome: >3B nodes, >10B edges
- The new short read assemblers require tremendous computation
 - Velvet (Zerbino & Birney, 2008) serial: > 2TB of RAM
 - ABySS (Simpson et al., 2009) MPI: 168 cores x ~96 hours
 - SOAPdenovo (Li et al., 2010) pthreads: 40 cores x 40 hours, >140 GB RAM

Error Correction with Quake

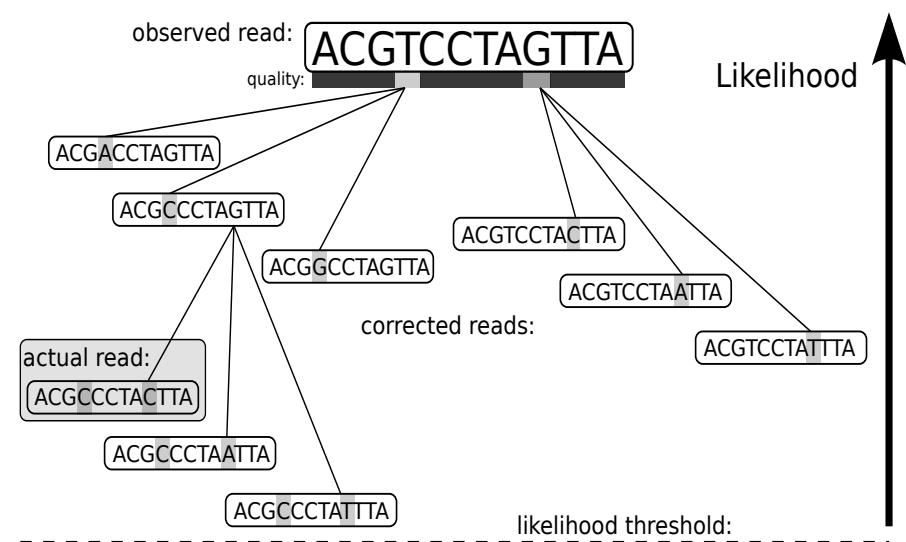
I. Count all “Q-mers” in reads

- 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

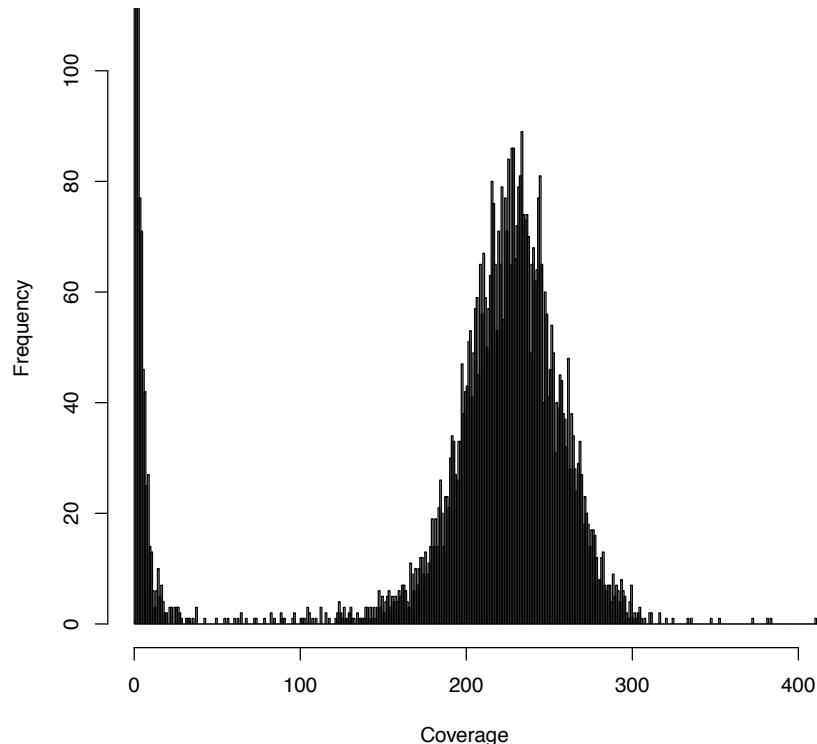


Quake: quality-aware detection and correction of sequencing reads.
Kelley, DR, Schatz, MC, Salzberg SL (2010) *Genome Biology*. 11:R116

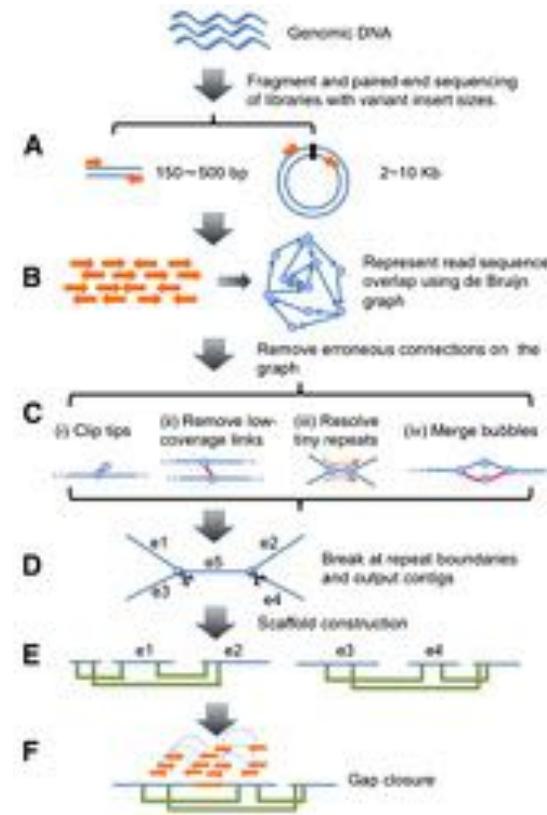
Illumina Sequencing & Assembly

Quake Results

2x76bp @ 275bp
2x36bp @ 3400bp

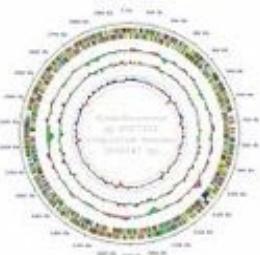


SOAPdenovo Results



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

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

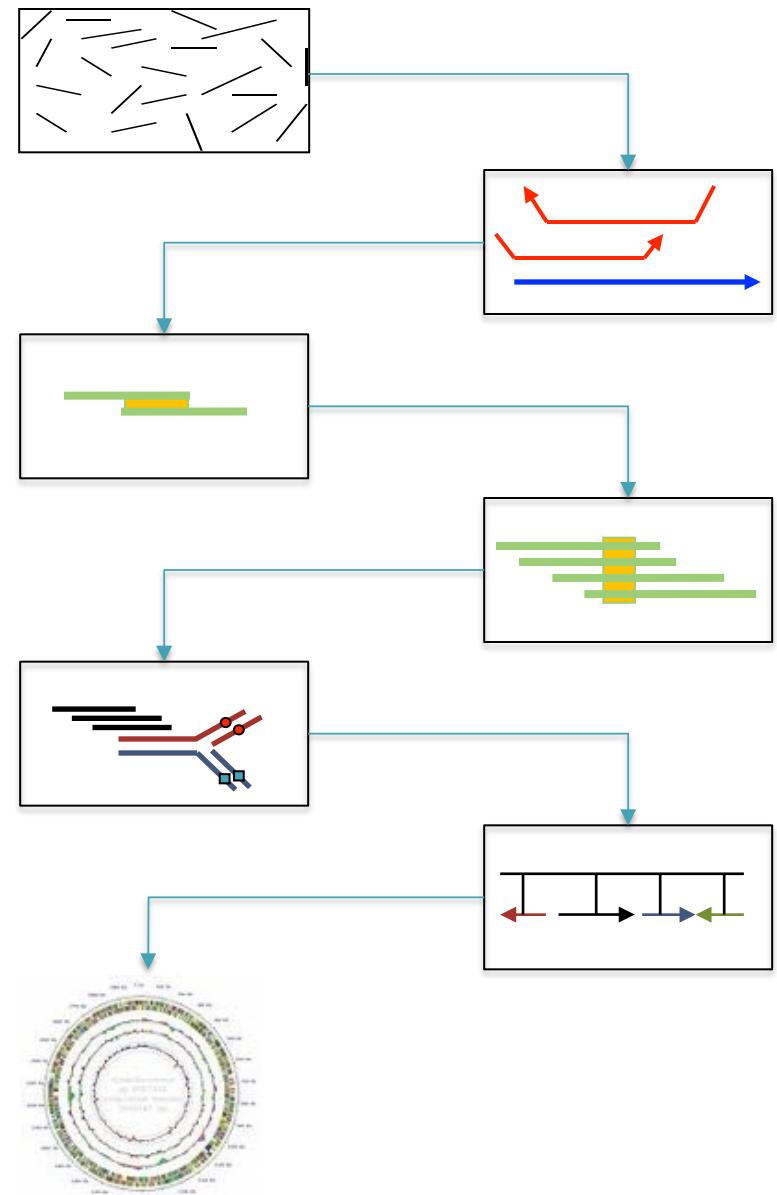


Genome assembly with the Celera Assembler

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



Hybrid Sequencing



Illumina
Sequencing by Synthesis

High throughput (60Gbp/day)
High accuracy (~99%)
Short reads (~100bp)

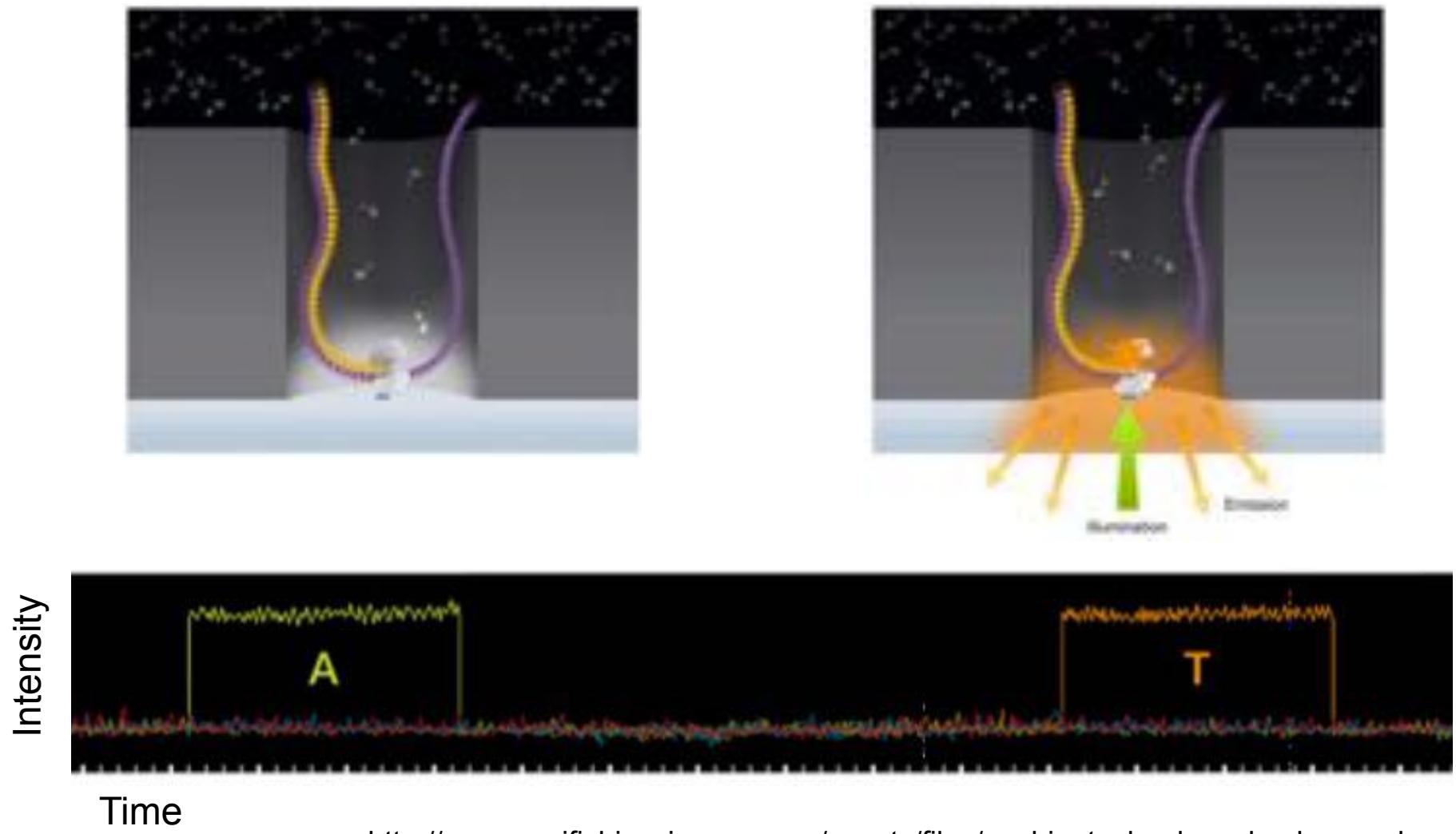


Pacific Biosciences
SMRT Sequencing

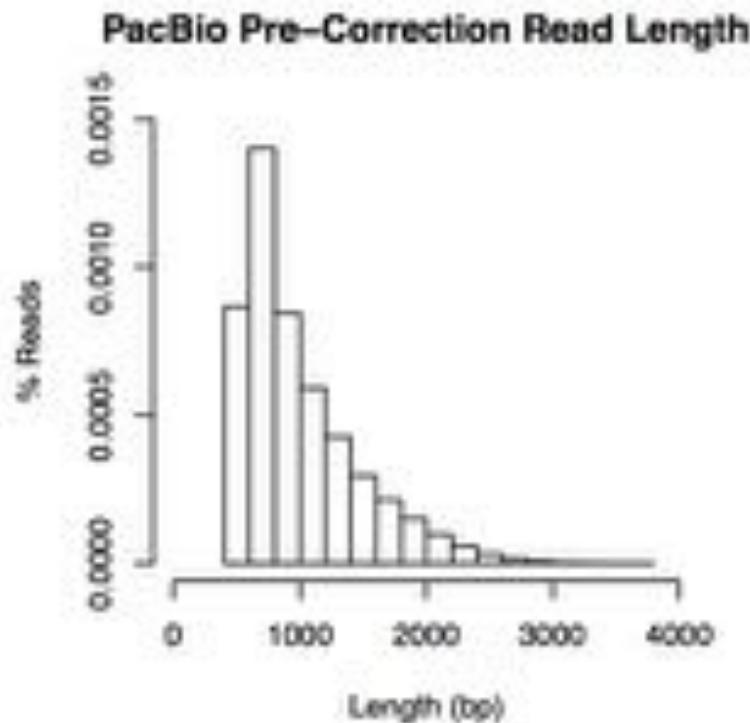
Lower throughput (600Mbp/day)
Lower accuracy (~85%)
Long reads (2-5kbp+)

SMRT Sequencing

Imaging of fluorescently phospholinked labeled nucleotides as they are incorporated by a polymerase anchored to a Zero-Mode Waveguide (ZMW).



SMRT Sequencing Data



Match	83.7%
Insertions	11.5%
Deletions	3.4%
Mismatch	1.4%

TTGTAAGCAGTTGAAAACATATGTGTGGATTAGATAAAGAACATGAAAG
||| ||| ||| ||| ||| ||| ||| ||| ||| ||| ||| ||| ||| ||| |||
TTGTAAGCAGTTGAAAACATATGTGT-GATTAG-ATAAAGAACATGGAAG

ATTATAAA-CAGTTGATCCATT-AGAAGA-AAACGCAGGCGCTAGG
| | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
A-TATAAAATCAGTTGATCCATTAGAA-AGAACGC-AAAGGC-GCTAGG

CAACCTTGAAATGTAATCGCACTGAGAACAGATTATTCCGCGCCCG
| | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
C-ACCTTG-ATGT-AT--CACTGAGAACAGATTATTCCGCGCCCG

TAACGAATCAAGATTCTGAAAACACAT-ATAACAACCTCCAAAA-CACAA
| | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
T-ACGAATC-AGATTCTGAAAACA-ATGAT---ACCTCCAAAAAGCACAA

-AGGAGGGGAAAGGGGGGAATATCT-ATAAAAGATTACAAATTAGA-TGA
| | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
GAGGAGG-AA-GAATATCTGAT-AAAGATTACAAATT-GAGTGA

ACT-AATTCACAA TA-AATAACACTTTA-ACAGAATTGAT-GGAA-GTT
| | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
ACTAATTCACAA-ATAATAACACTTTAGACAAATTGATGGGAAGGTT

TCGGAGAGATCCAACAAATGGGC-ATCGCCTTGAT-GTTAC-AATCAA
| | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
TC-GAGAGATCC-AAACAAAT-GGC GATCG-CTTGACGTTACAATCAA

ATCCAGTGGAAAATATAATTTATGCAATCCAGGAACCTATTACAATTAG
| | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
ATCCAGT-GAAAATATA- TTATGC-ATCCA-GAACTTATTACAATTAG

Sample of 100k reads aligned with BLASR requiring >100bp alignment

PacBio Error Correction

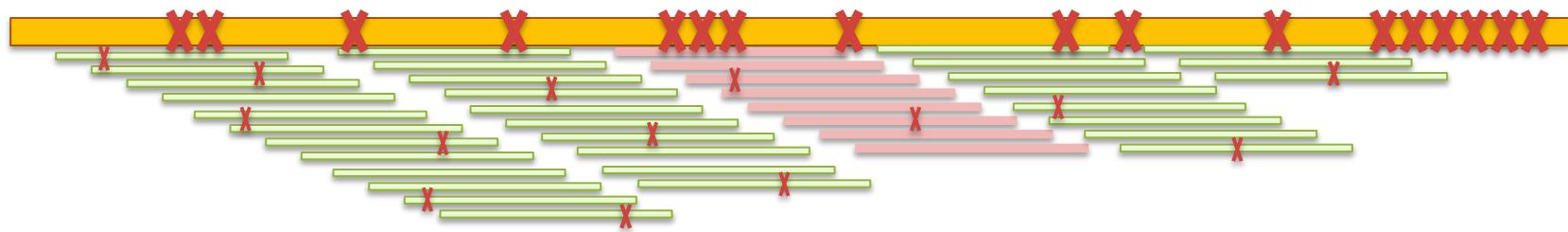
<http://wgs-assembler.sf.net>

I. Correction Pipeline

1. Map short reads to long reads
2. Trim long reads at coverage gaps
3. Compute consensus for each long read



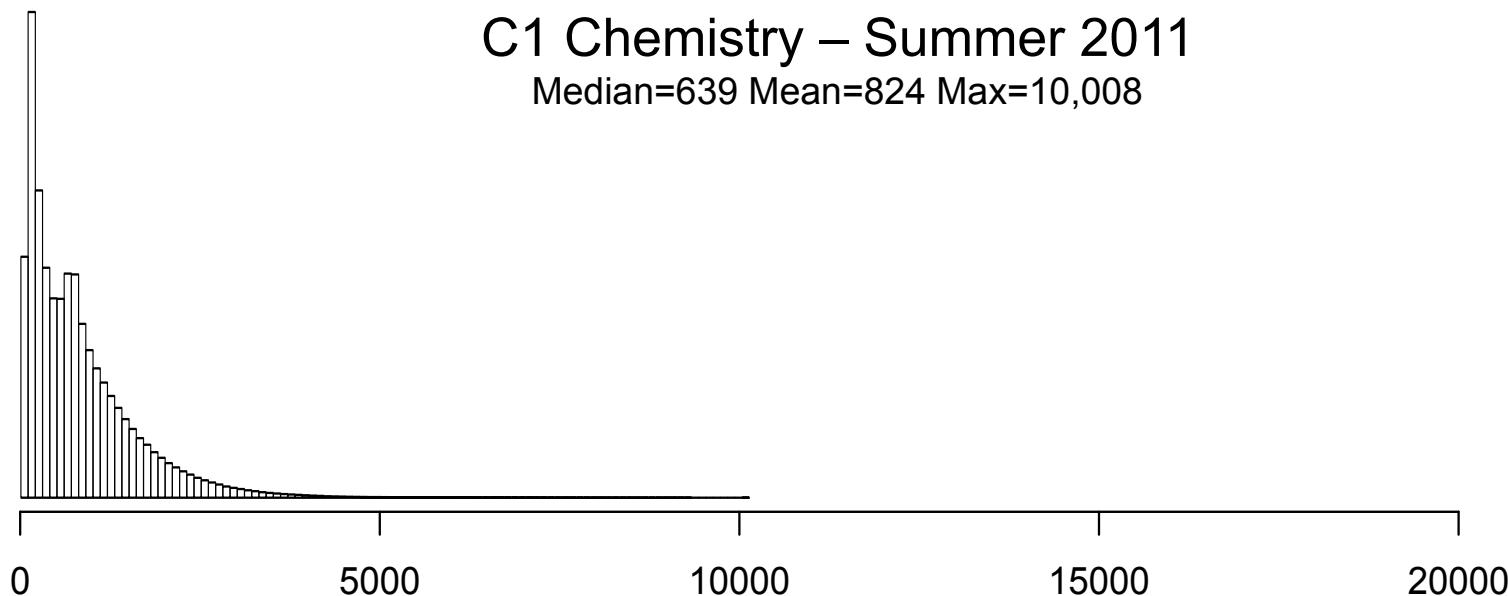
2. Error corrected reads can be easily assembled, aligned



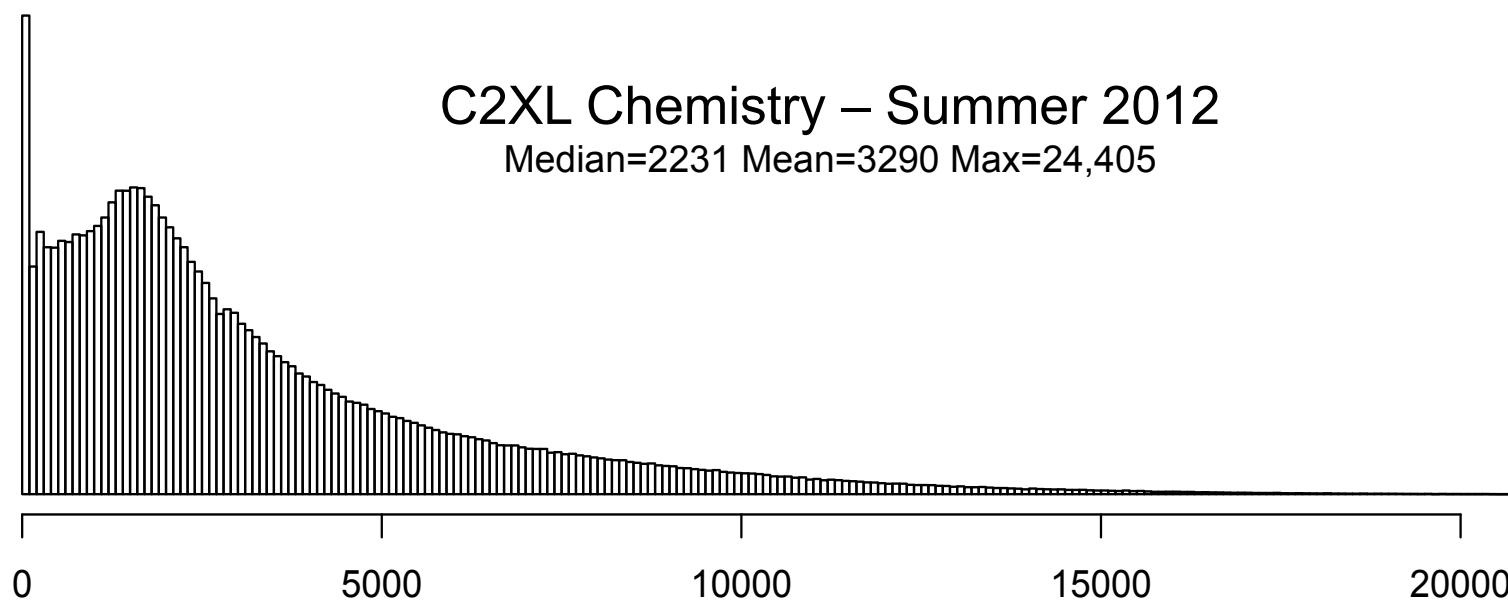
Hybrid error correction and de novo assembly of single-molecule sequencing reads.
Koren, S, Schatz, MC, et al. (2012) *Nature Biotechnology*. doi:10.1038/nbt.2280

PacBio Long Read Rice Sequencing

C1 Chemistry – Summer 2011
Median=639 Mean=824 Max=10,008

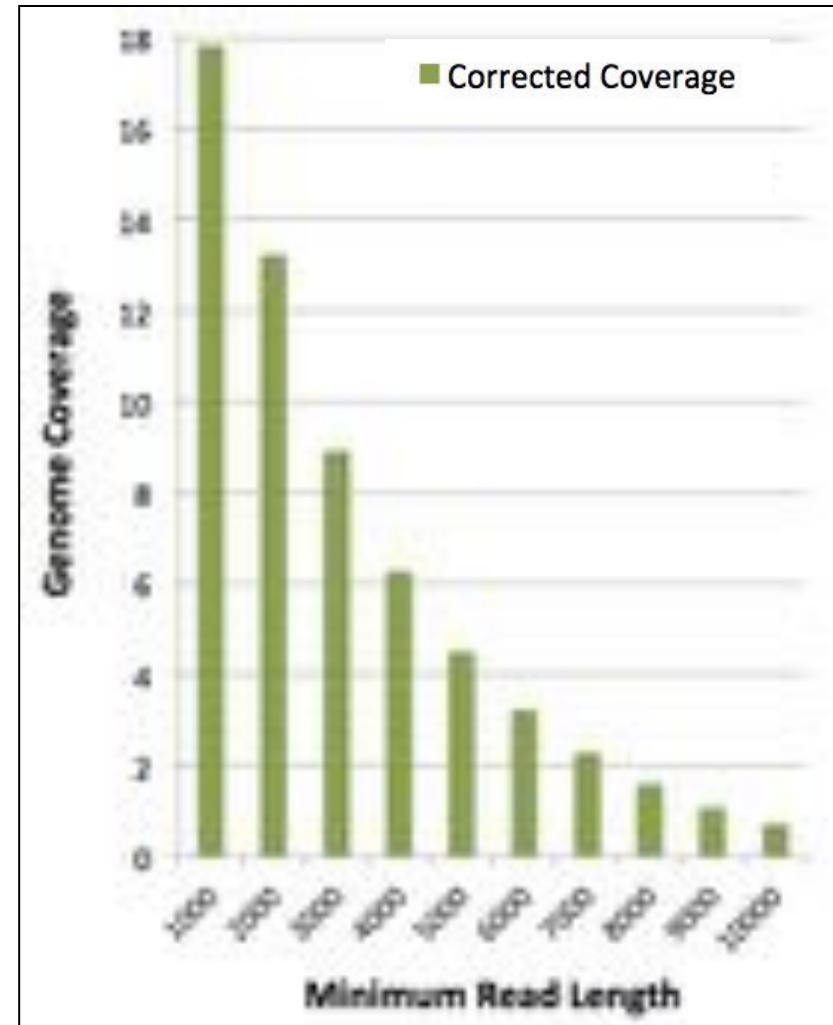


C2XL Chemistry – Summer 2012
Median=2231 Mean=3290 Max=24,405



Preliminary Rice Assemblies

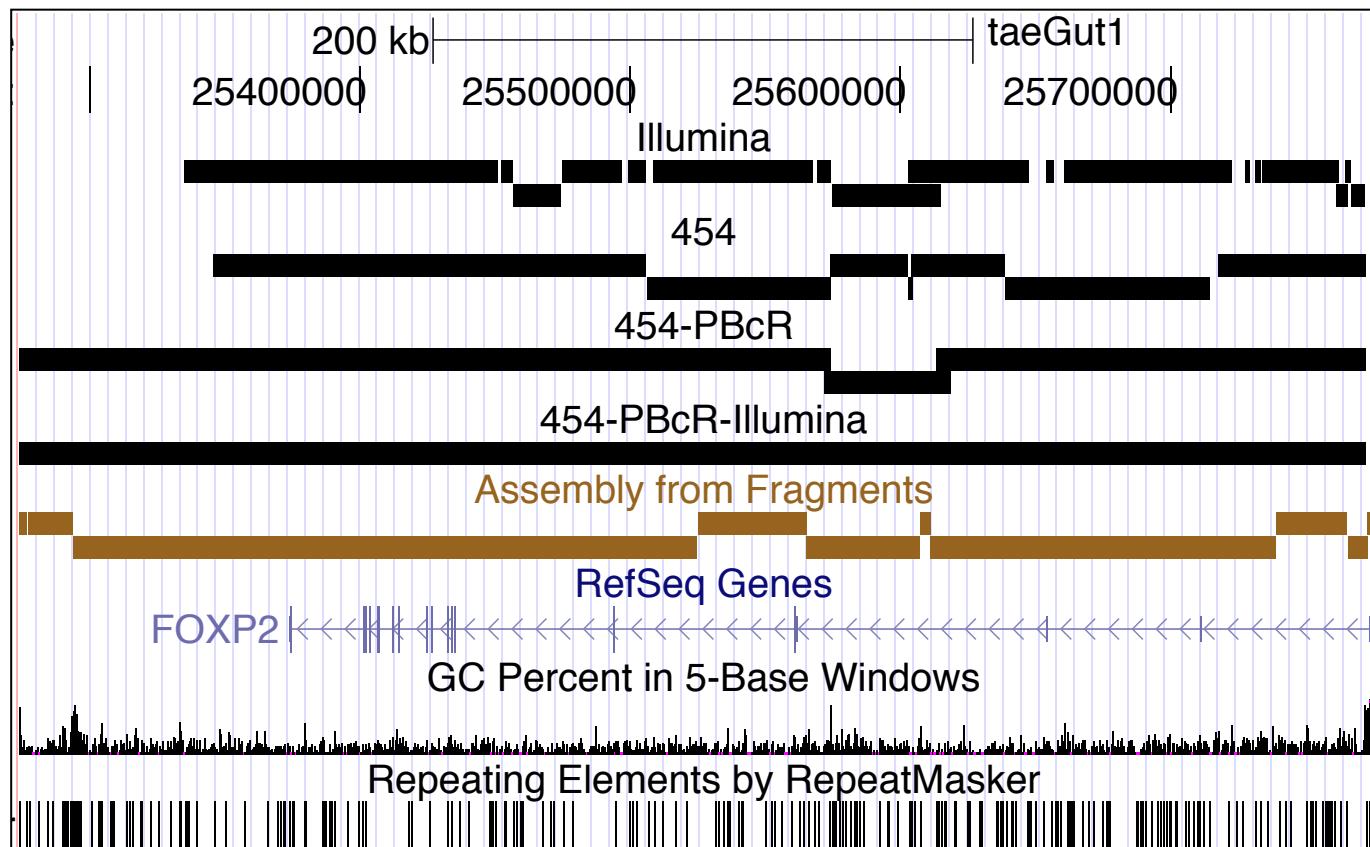
Assembly	Contig NG50
HiSeq Fragments 50x 2x100bp @ 180	3,925
MiSeq Fragments 23x 459bp 8x 2x251bp @ 450	6,332
“ALLPATHS-recipe” 50x 2x100bp @ 180 36x 2x50bp @ 2100 51x 2x50bp @ 4800	18,248
PBeCR Reads 7x @ 3500 ** MiSeq for correction	50,995
PBeCR + Illumina Shred 7x @ 3500 ** MiSeq for correction 5x @ 3000bp shred	59,695



In collaboration with McCombie & Ware labs @ CSHL

Improved Gene Reconstruction

FOXP2 assembled in a single contig in the PacBio parrot assembly



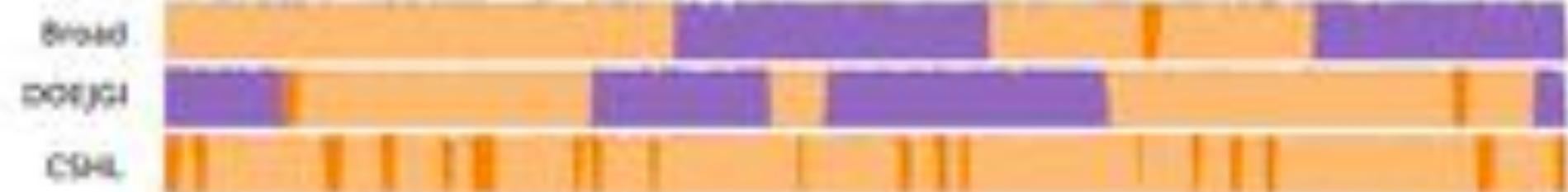
Hybrid error correction and de novo assembly of single-molecule sequencing reads.
Koren, S, Schatz, MC, et al. (2012) *Nature Biotechnology*. doi:10.1038/nbt.2280

THE ASSEMBLATHON

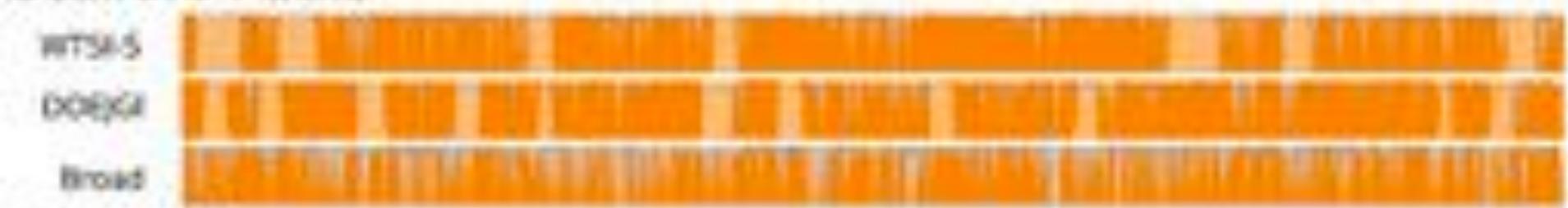
- Attempt to answer the question:
“What makes a good assembly?”
- Organizers provided simulated sequence data
 - Simulated 100 base pair Illumina reads from simulated diploid organism
- 41 submissions from 17 groups
- Results demonstrate trade-offs assemblers must make

Assembly Results

Scaffolds



Scaffold Paths



Contig Paths



Fill Color Key
Base 44



Final Rankings

ID	Overall	OPNG50	SPN450	Struct.	CDS	Sufs.	Copy. Num.	Cov. Tot.	Cov. CDS
BGI	36	★					★	★	★
Broad	37	★	★	★	★				
WTSI-S	46		★	★	★	★			
CSHL	52	★							★
BCGSC	53						★		★
DOEJGI	56		★	★	★	★			
RHUL	58								
WTSI-P	64							★	
EBI	64						★		
CRAES	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
 - Celera Assembler if you have 454 or PacBio reads

Break





Outline

I. Assembly theory

1. Assembly by analogy
2. De Bruijn and Overlap graph
3. Coverage, read length, errors, and repeats

2. Genome assemblers

1. ALLPATHS-LG, SOAPdenovo, Celera Assembler
2. Assemblathon

3. Applications

1. Whole Genome Alignment with MUMmer
2. Gene Finding with Glimmer

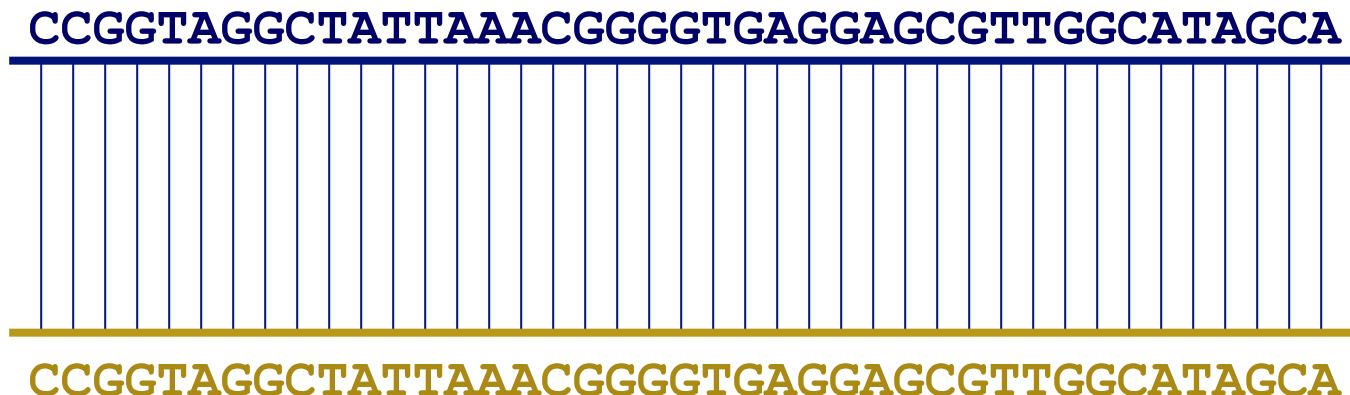


Whole Genome Alignment with MUMmer

Slides Courtesy of Adam M. Phillippy

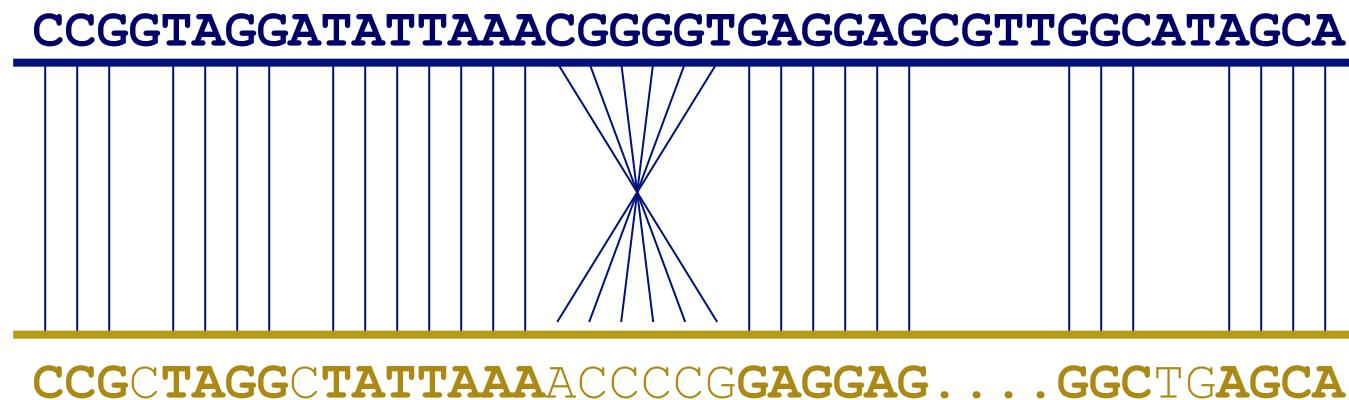
Goal of WGA

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



Not so fast...

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



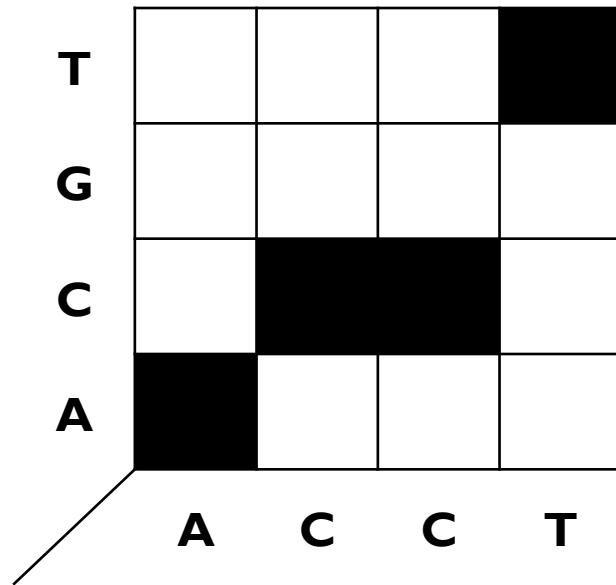
WGA visualization

- How can we visualize *whole genome* alignments?

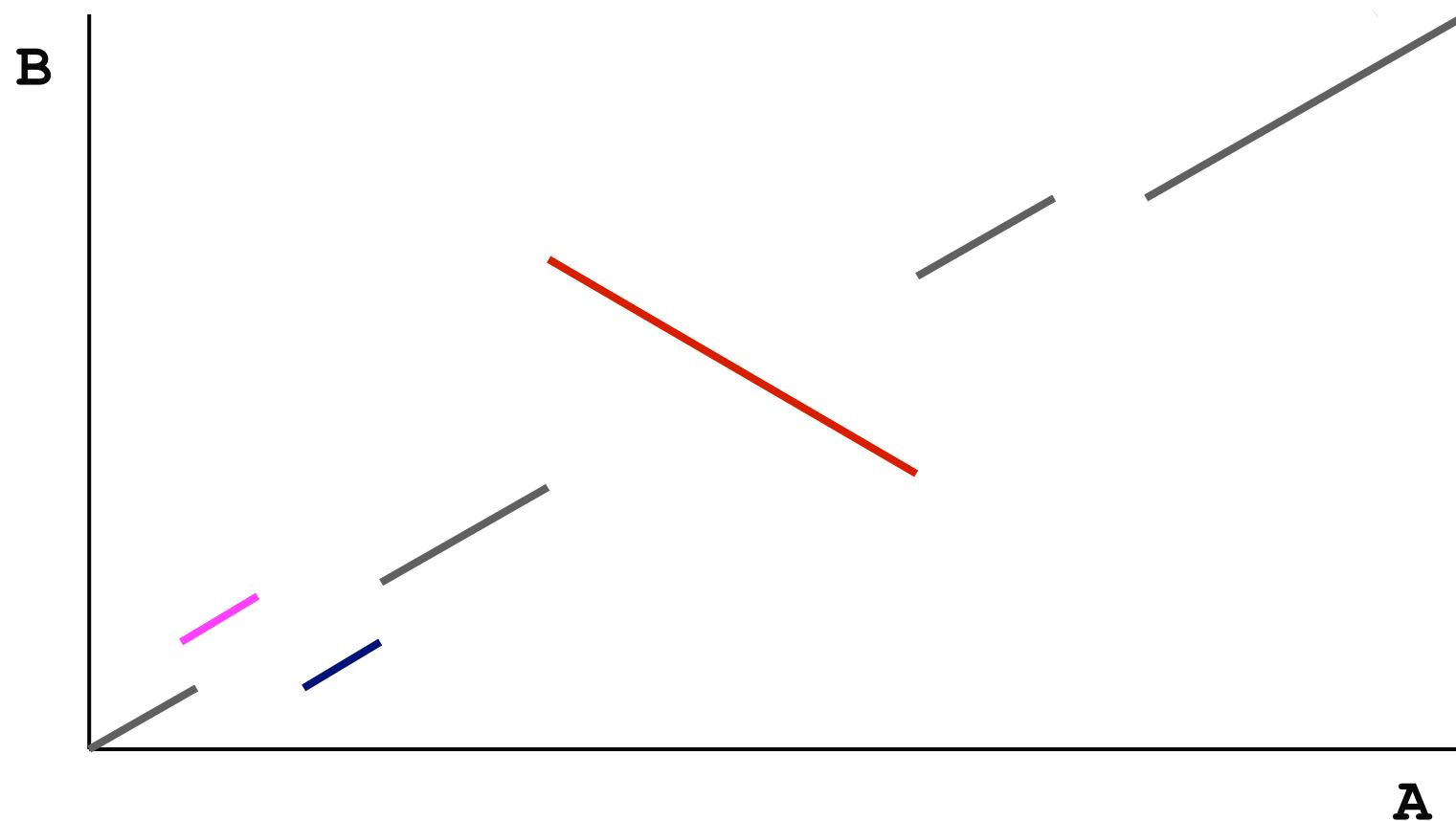
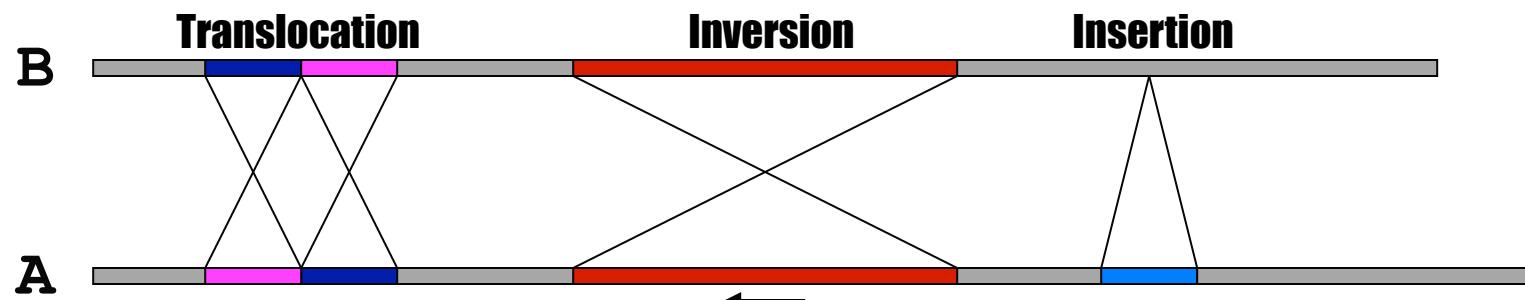
- With an alignment dot plot

- $N \times M$ matrix

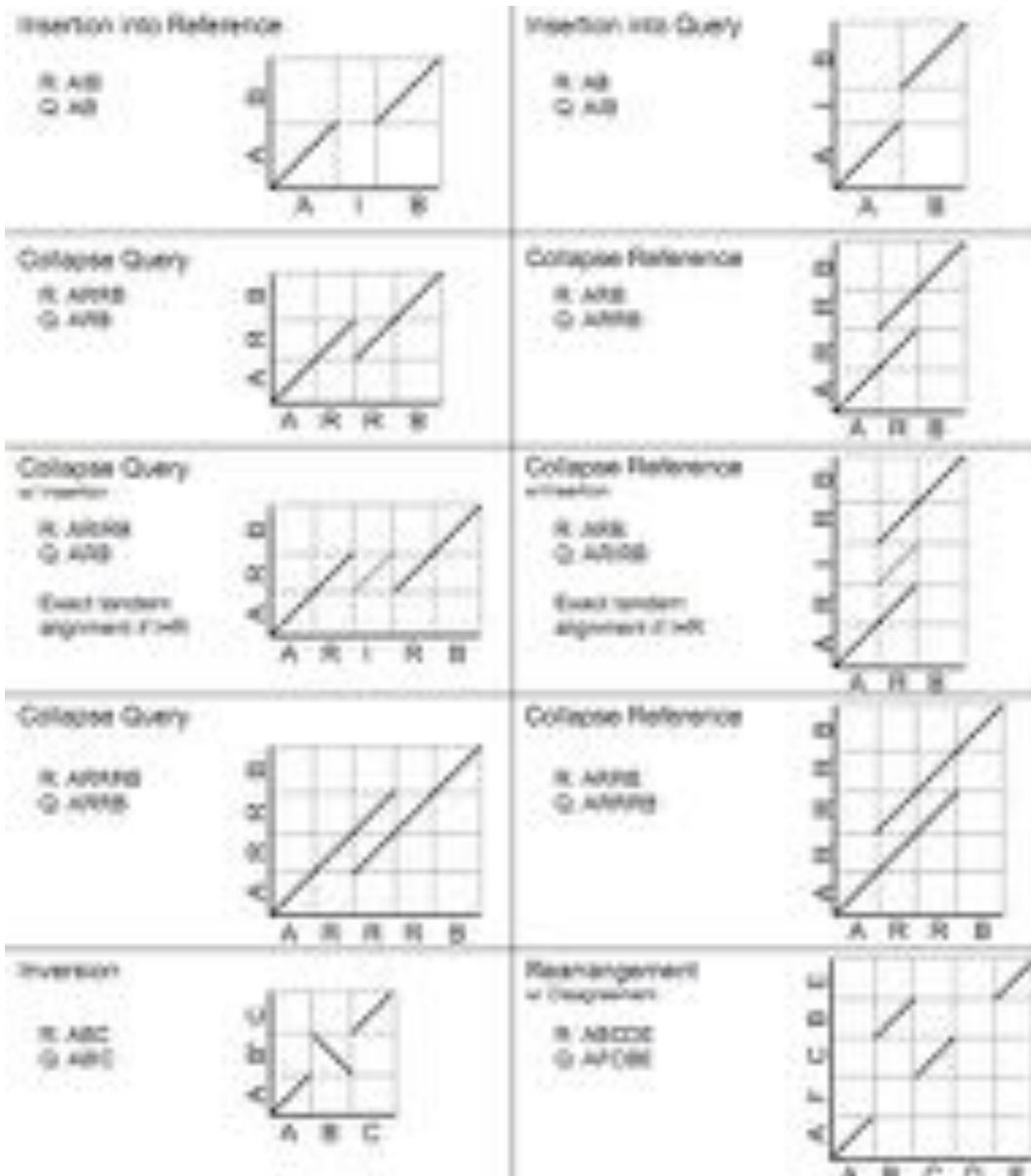
- Let i = position in genome A
 - Let j = position in genome B
 - Fill cell (i,j) if A_i shows similarity to B_j



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



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](http://mummer.sf.net/manual/AlignmentTypes.pdf)

Seed-and-extend with MUMmer

How can quickly align two genomes?

I. Find maximal-unique-matches (MUMs)

- ◆ 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)

2. Cluster MUMs

- ◆ using size, gap and distance parameters

3. Extend clusters

- ◆ using modified Smith-Waterman algorithm

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

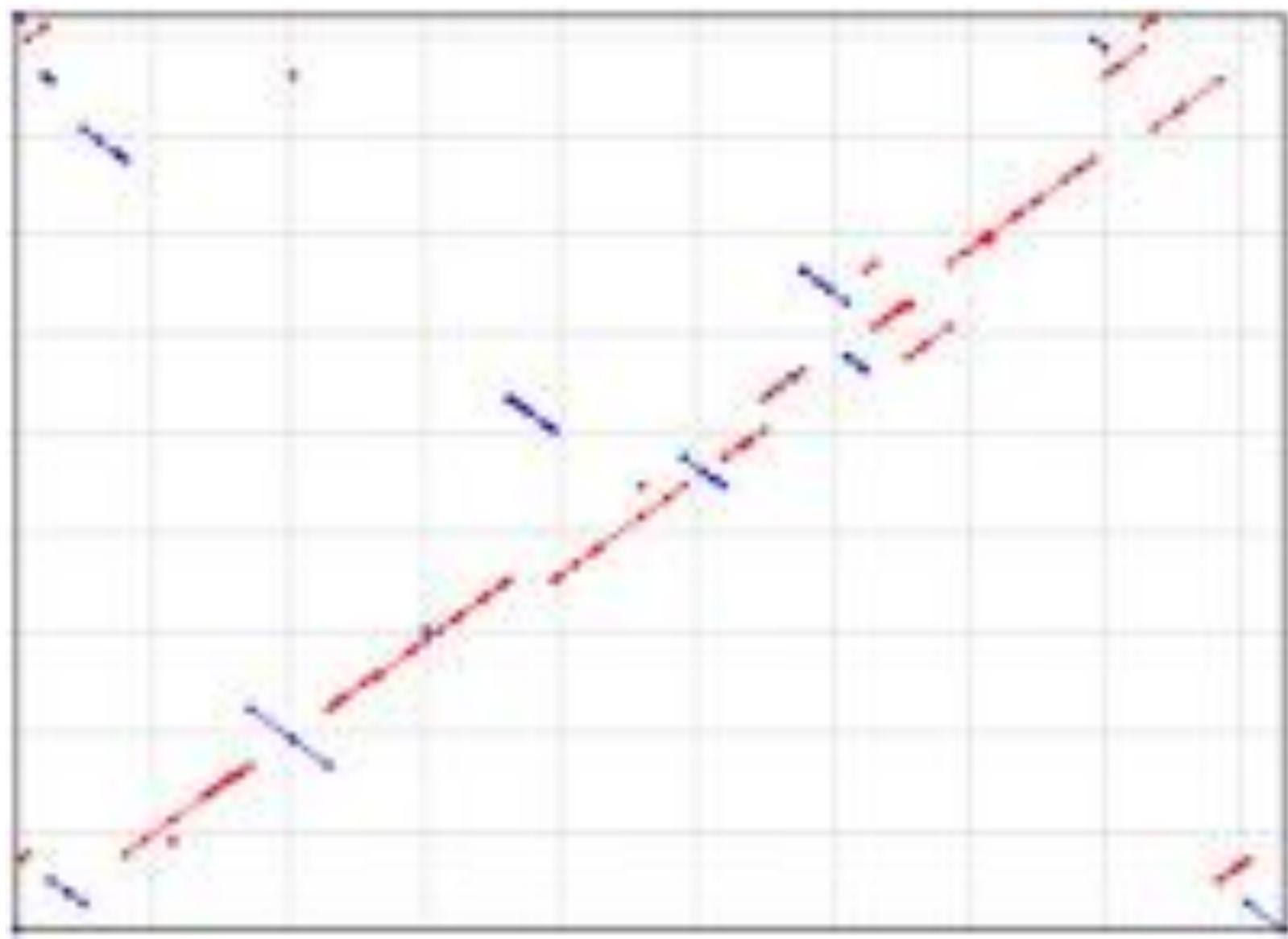
mummerplot --large --layout out.delta.m

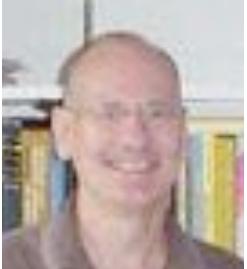
--large Large plot

--layout Nice layout for multi-fasta files

--x11 Default, draw using x11 (--postscript, --png)

*requires gnuplot





Bacterial Gene Finding with Glimmer

(also Archaeal and viral gene finding)

Arthur L. Delcher and Steven Salzberg
Center for Bioinformatics and Computational Biology
Johns Hopkins University School of Medicine

Gene Prediction: Computational Challenge

aatgcatgcggctatgcta atgcatgcggctatgcta agctggatccgat gaca atgcatgcggctatgctaa
tgcatgcggctatgca agctggatccgat gactatgcta agctggatccgat gaca atgcatgcggctatgc
taatgaatggtcttggatt taccttggaa atgcta agctggatccgat gaca atgcatgcggctatgcta atg
atggtcttggatt taccttggaa atatgcta atgcatgcggctatgcta agctggatccgat gaca atgcatgc
ggctatgcta atgcatgcggctatgca agctggatccgat gactatgcta agctgcggctatgcta atgcatg
cggtatgcta agctggatccgat gaca atgcatgcggctatgcta atgcatgcggctatgca agctggatc
ctgcggctatgcta atgcatggatccgat gaca atgcatgcggctatgcta atgcatgcggctatgca agctggatc
ctatgcta atgcatggatccgat gaca atgcatgcggctatgcta atgcatgcggctatgca agctggatc
cggtatgcta agctggatccgat gaca atgcatgcggctatgcta atgcatgcggctatgca agctggatc
cgat gactatgcta agctgcggctatgcta atgcatgcggctatgcta agctcatgcggctatgcta agctgg
aatgcatgcggctatgcta agctggatccgat gaca atgcatgcggctatgcta atgcatgcggctatgca ag
ctggatccgat gactatgcta agctgcggctatgcta atgcatgcggctatgcta agctcggtatgcta atg
atggtcttggatt taccttggaa atgcta agctggatccgat gaca atgcatgcggctatgcta atgcatgg
ttggatt taccttggaa atatgcta atgcatgcggctatgcta agctggatgcatgcggctatgcta agctgg
gatccgat gaca atgcatgcggctatgcta atgcatgcggctatgca agctggatccgat gactatgcta agc
tgccgctatgcta atgcatgcggctatgcta agctcatgcgg

Gene Prediction: Computational Challenge

aatgcatgcggctatgctaattgcattgcggctatgctaagctggatccgatgacaatgcattgcggctatgctaa
tgcatgcggctatgcaaggctggatccgatgactatgctaagctggatccgatgacaatgcattgcggctatgc
taatgaatggtcttggatttaccttggaatatgctaagctggatccgatgacaatgcattgcggctatgctaattga
atggtcttggatttaccttggaatatgctaattgcattgcggctatgctaagctggatccgatgacaatgcattgc
ggctatgctaattgcattgcggctatgcaaggctggatccgatgactatgctaagctggatccgatgacaatgcattgc
ggctatgctaagctggatccgatgacaatgcattgcggctatgctaattgcattgcggctatgcaagctggatc
ctggatccgatgctaattgcattgcggctatgctaattgcattgcggctatgcaagctggatccgatgacaatgcattgc
ctatgctaattgcattgcggctatgctaattgcattgcggctatgctaattgcattgcggctatgcaagctggatc
cggtatgctaagctggatccgatgacaatgcattgcggctatgctaattgcattgcggctatgcaagctggatc
ctggatccgatgctaattgcattgcggctatgctaattgcattgcggctatgctaattgcattgcggctatgcaagctgg
aatgcatgcggctatgctaagctggatccgatgacaatgcattgcggctatgctaattgcattgcggctatgcaag
ctggatccgatgactatgctaagctggatccgatgacaatgcattgcggctatgctaattgcattgcggctatgctaattga
atggtcttggatttaccttggaatatgctaagctggatccgatgacaatgcattgcggctatgctaattgcattgcggctatg
ttggatccgatgacaatgcattgcggctatgctaagctggatccgatgacaatgcattgcggctatgctaagctgg
gatccgatgacaatgcattgcggctatgctaagctggatccgatgacaatgcattgcggctatgctaagctggatccgatg
tcggctatgctaattgcattgcggctatgctaagctcatgcgg

Gene!

Step One

- Find open reading frames (ORFs).

The diagram illustrates a DNA sequence with a green rectangular box highlighting a specific segment. Within this segment, a red arrow points to the first codon, labeled "Start codon". At the far right end of the highlighted segment, two red arrows point to consecutive codons, both labeled "Stop codon". The sequence itself is composed of various nucleotides, some of which are highlighted in yellow or orange, indicating specific base pairs or motifs.

Start codon

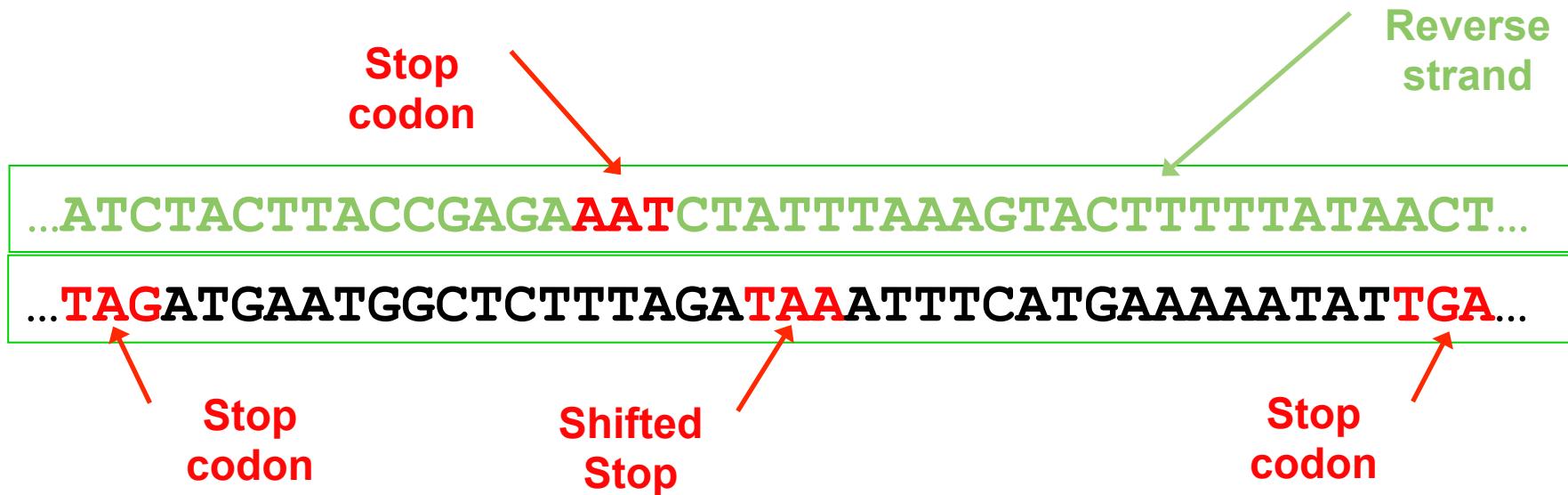
Stop codon

Stop codon

...TAGATGAATGGCTCTTTAGATAAATTTCATGAAAAATATTGA...

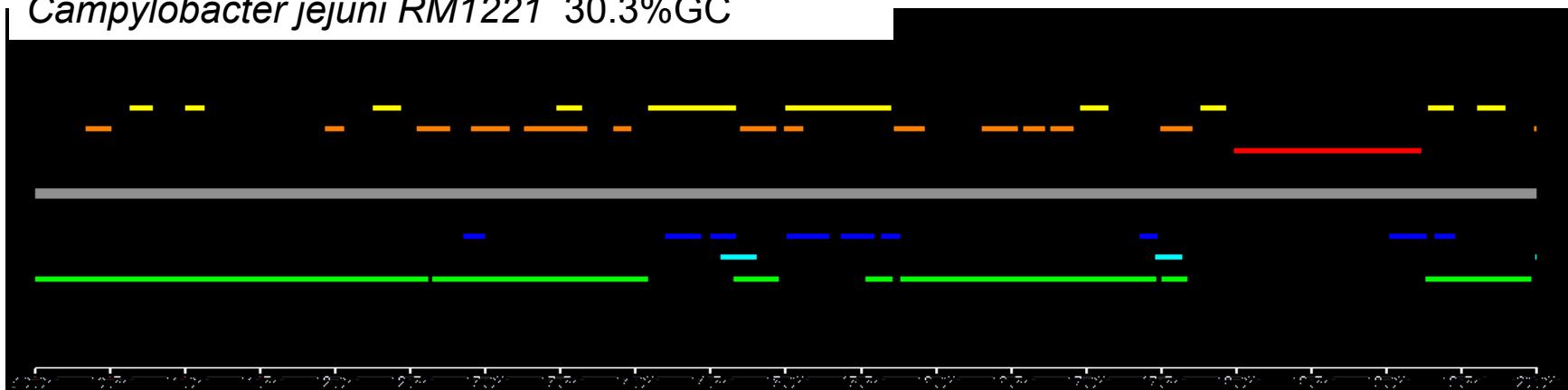
Step One

- Find open reading frames (ORFs).



- But ORFs generally overlap ...

Campylobacter jejuni RM1221 30.3%GC

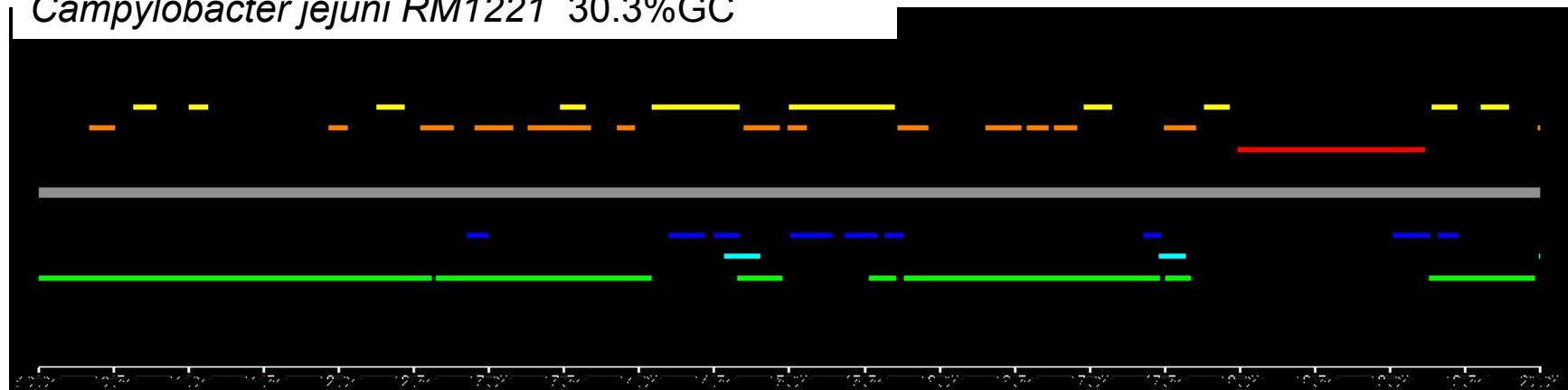


All ORFs longer than 100bp on both strands shown
- color indicates reading frame
Longest ORFs likely to be protein-coding genes

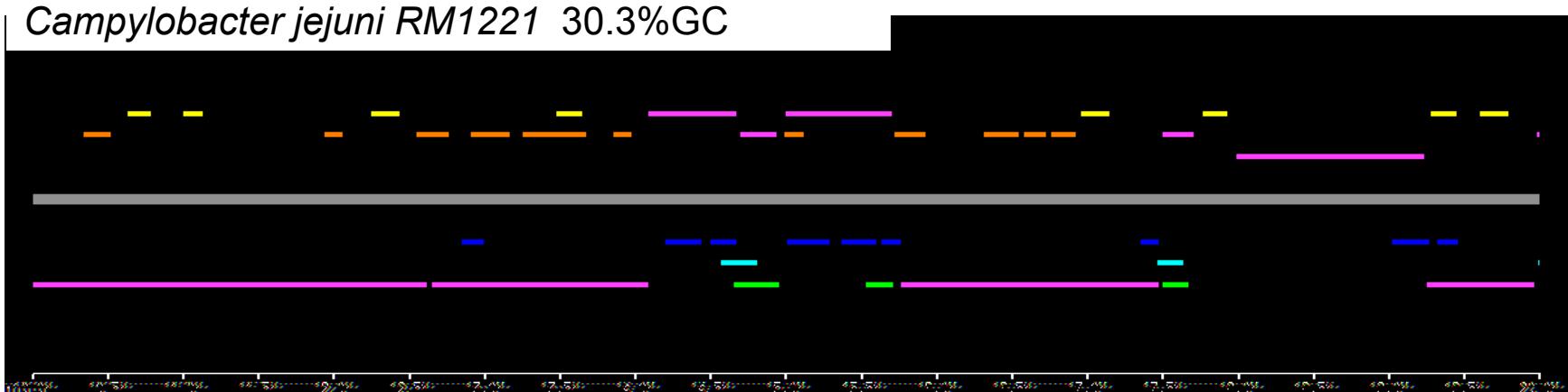
Note the low GC content

All genes are ORFs but not all ORFs are genes

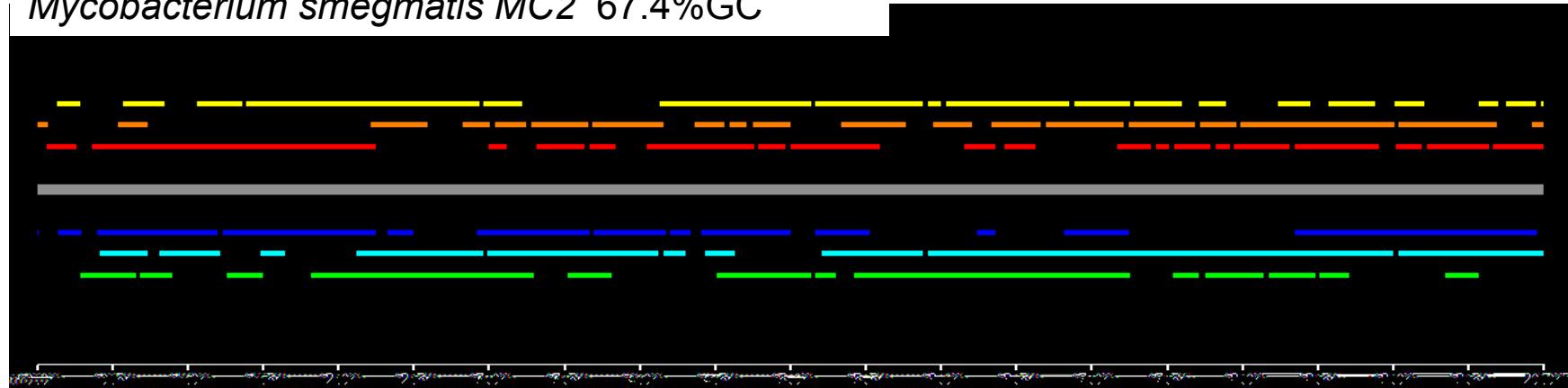
Campylobacter jejuni RM1221 30.3%GC



Campylobacter jejuni RM1221 30.3%GC

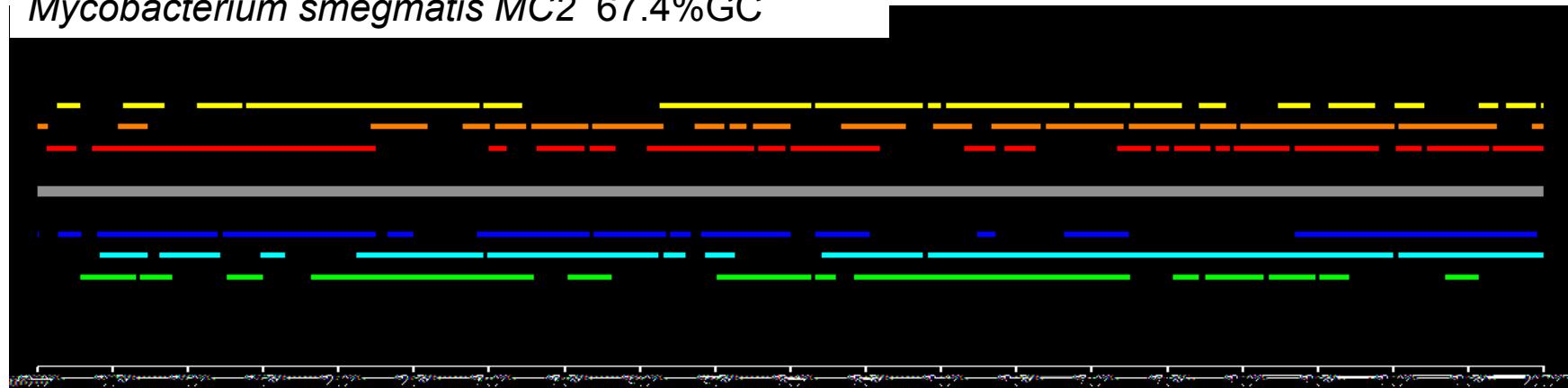


Mycobacterium smegmatis MC2 67.4%GC

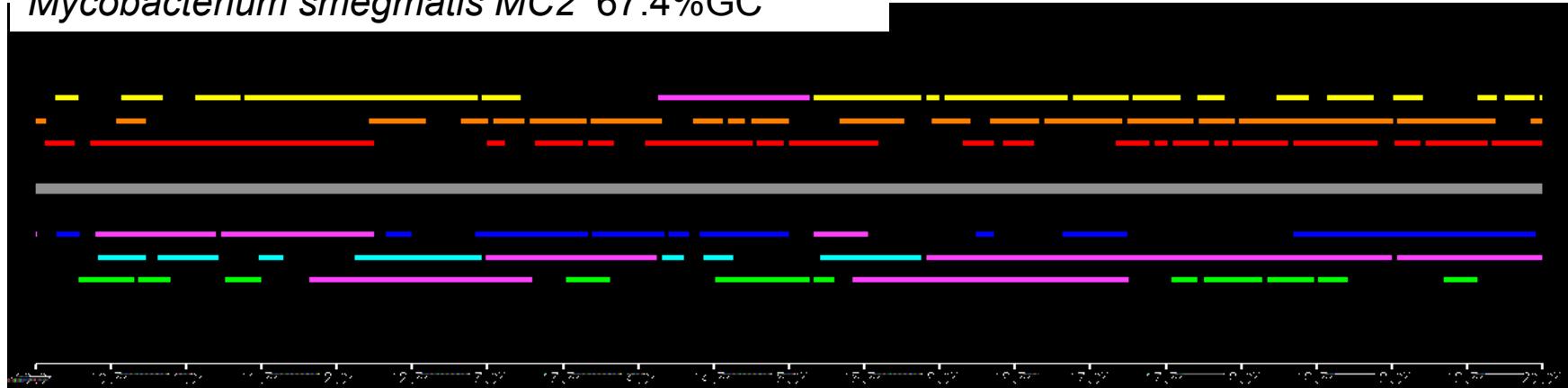


Note what happens in a high-GC genome

Mycobacterium smegmatis MC2 67.4%GC



Mycobacterium smegmatis MC2 67.4%GC



The Problem

- Need to decide which orfs are genes.
 - Then figure out the coding start sites
- Can do homology searches but that won't find novel genes
 - Besides, there are errors in the databases
- Generally can assume that there are some known genes to use as training set.
 - Or just find the obvious ones

Probabilistic Methods

- Create models that have a probability of generating any given sequence.
- Train the models using examples of the types of sequences to generate.
- The “score” of an orf is the probability of the model generating it.
 - Can also use a negative model (*i.e.*, a model of non-orphs) and make the score be the ratio of the probabilities (*i.e.*, the odds) of the two models.
 - Use logs to avoid underflow

Fixed-Order Markov Models

- k^{th} -order Markov model bases the probability of an event on the preceding k events.
- Example: With a 3rd-order model the probability of this sequence:

...C **TAG** AT ...

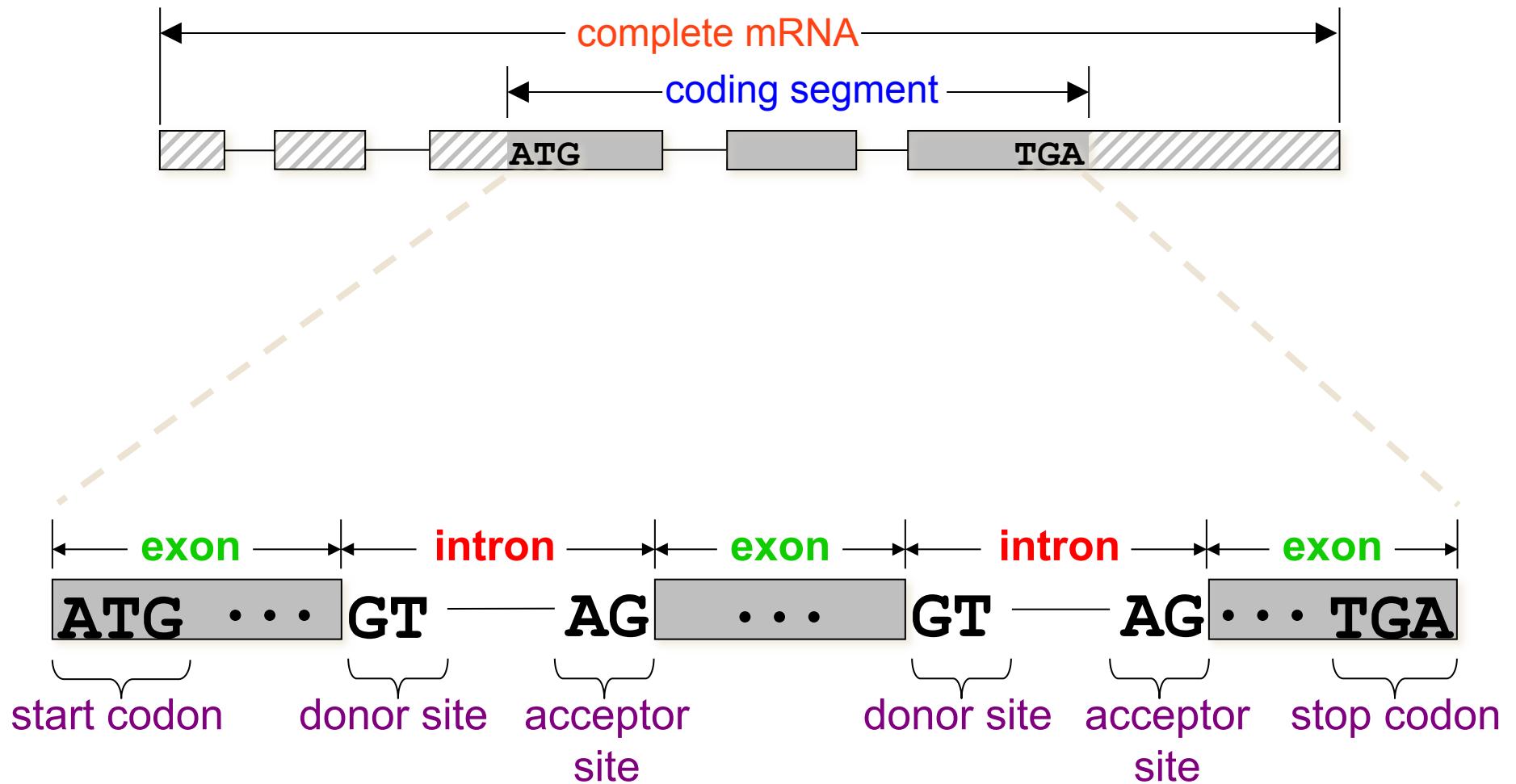
Context Target

- would be:

$$\dots P(G | CTA) \cdot P(A | TAG) \cdot P(T | AGA) \dots$$

Target Context

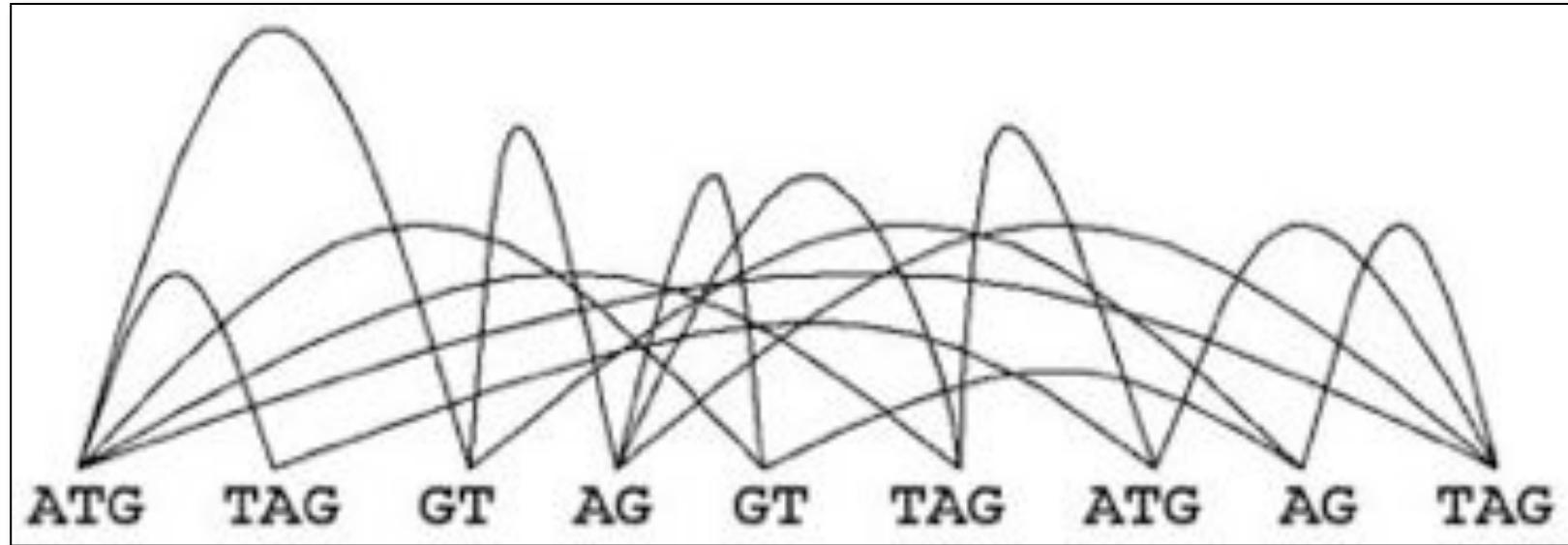
Eukaryotic Gene Syntax



Regions of the gene outside of the CDS are called **UTR's** (*untranslated regions*), and are mostly ignored by gene finders, though they are important for regulatory functions.

Representing Gene Syntax with ORF Graphs

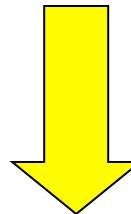
After identifying the most promising (i.e., highest-scoring) signals in an input sequence, we can apply the gene syntax rules to connect these into an *ORF graph*:



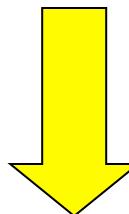
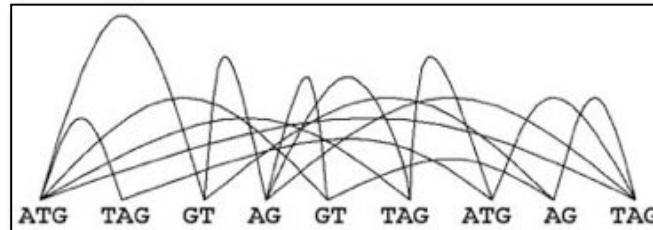
An ORF graph represents all possible *gene parses* (and their scores) for a given set of putative signals. A *path* through the graph represents a single gene parse.

Conceptual Gene-finding Framework

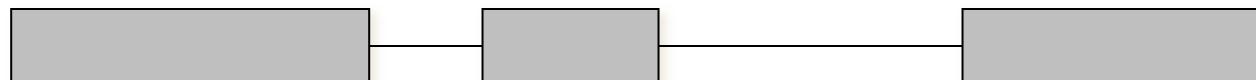
TATTCCGATCGATCGATCTCTCTAGCGTCTACG
CTATCATCGCTCTATTATCGCGCGATCGTCG
ATCGCGCGAGAGTATGCTACGTGATCGAATTG



identify most promising signals, score signals and content regions between them; induce an ORF graph on the signals



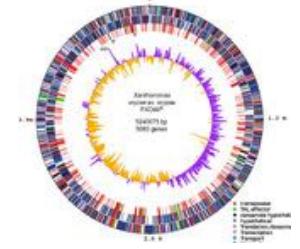
find highest-scoring path through ORF graph;
interpret path as a gene parse = gene structure



Other Resources

Resource	URL	Description
Google	http://www.google.com	Internet Search
Google Scholar	http://scholar.google.com/	Literature Searches
SeqAnswers	http://seqanswers.com/	Bioinformatics Forum
Wikipedia	http://www.wikipedia.org/	Overview on anything
Circos	http://circos.ca/	Circular Genome Plots
GraphViz	http://www.graphviz.org/	Graph Visualization
EndNote	http://endnote.com/	Citation Manager
R	http://www.r-project.org/	Stats & Visualizations
Weka	http://www.cs.waikato.ac.nz/ml/weka/	Data Mining
IGV	http://www.broadinstitute.org/igv/	Read Mapping Viz
Schatz Lab	http://schatzlab.cshl.edu/teaching/	Exercises and Lectures

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

Acknowledgements

Schatz Lab

Giuseppe Narzisi
Shoshana Marcus
James Gurtowski
Alejandro Wences
Hayan Lee
Rob Aboukhalil
Mitch Bekritsky
Charles Underwood
Rushil Gupta
Avijit Gupta
Shishir Horane
Deepak Nettem
Varrun Ramani
Piyush Kansal
Eric Biggers
Aspyn Palatnick

CSHL

Hannon Lab
Gingeras Lab
Iossifov Lab
Levy Lab
Lippman Lab
Lyon Lab
Martienssen Lab
McCombie Lab
Ware Lab
Wigler Lab

IT Department

NBACC

Adam Phillippy
Sergey Koren



Thank You!

 **Michael Schatz** @mike_schatz 26 Mar

Can you assemble genomes, find mutations, and decode secret messages? Get ready for the **#DNA60IFX** challenge! bit.ly/16VKqsG

[Expand](#)

