Graphs and Genomes

Michael Schatz

Bioinformatics Lecture 3

Quantitative Biology 2014

Dynamic Programming Matrix

Compute the optimal alignment of ABC…XY..N and DEF…UV…M

	0	A	B	C	…	X	Y	…	N
0									
D									
E									
F									
…									
U									
V									
…									
M									

Dynamic Programming Matrix

Compute the optimal alignment of ABC…XY..N and DEF…UV…M

	0	**A**	**B**	**C**	**…**	**X**	**Y**	**…**	**N**
0	0	1	2	3		X	X+1		N
D	1								
E	2								
F	3								
…									
U	U								
V	U+1								
…									
M	M								

Top row and first column are easy: it takes L-edits to transform and empty string into a length L string

Dynamic Programming Matrix

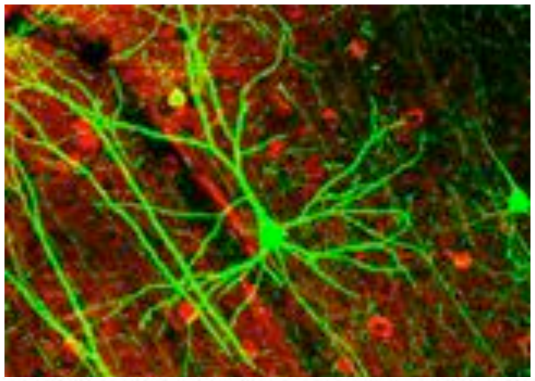
Compute the optimal alignment of “ABC…XY..N” and “DEF…UV…M”

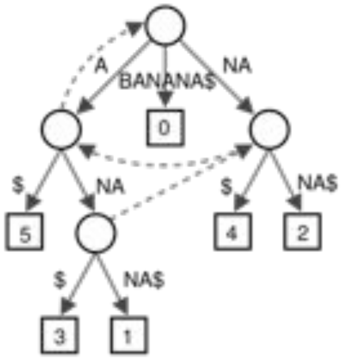
	0	**A**	**B**	**C**	**…**	**X**	**Y**	**…**	**N**
0	0	1	2	3		X	X+1		N
D	1								
E	2								
F	3								
…									
U	U					γ	α		
V	U+1					β	Ω		
…									
M	M								

$$\Omega = \min \begin{cases} \text{“Up” + 1} & \alpha+1 \\ \text{“Left+ + 1} & \beta+1 \\ \text{“Diagonal” +0/1} & \gamma+1 \end{cases}$$

Up	Left	Diagonal
`ABC...XY-`	`ABC....X`**`Y`**	`ABC...X`**`Y`**
`DEF....U`**`V`**	`DEF...UV-`	`DEF...U`**`V`**
α	β	γ

Biological Networks





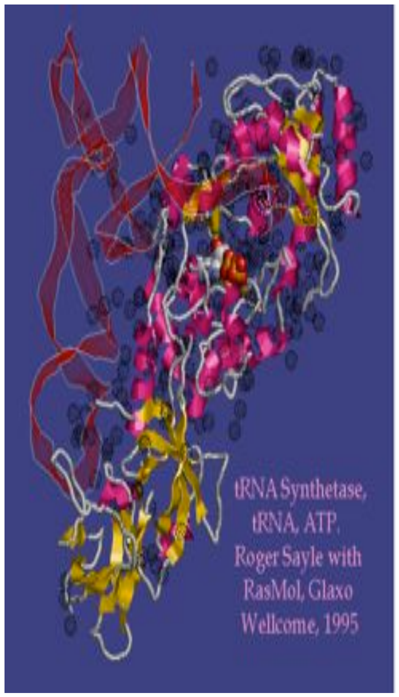
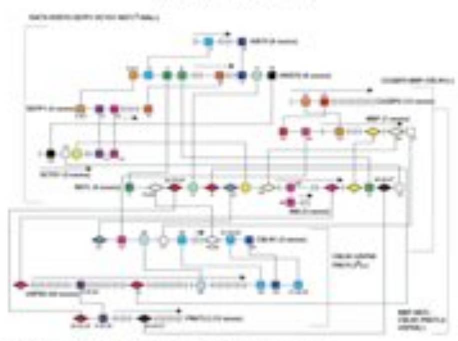
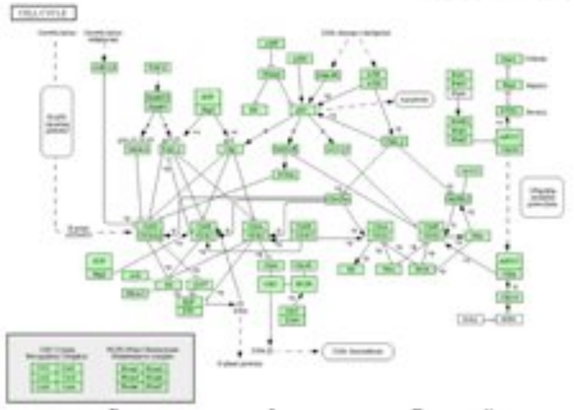

tRNA Synthetase, tRNA, ATP. Roger Sayle with RasMol, Glaxo Wellcome, 1995

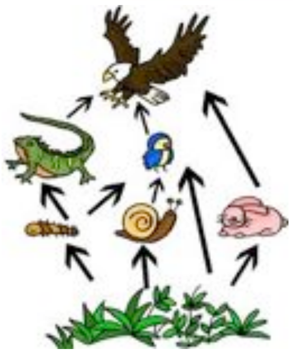
Figure 5 Putative regulatory elements shared between groups of correlated and anticorrelated genes

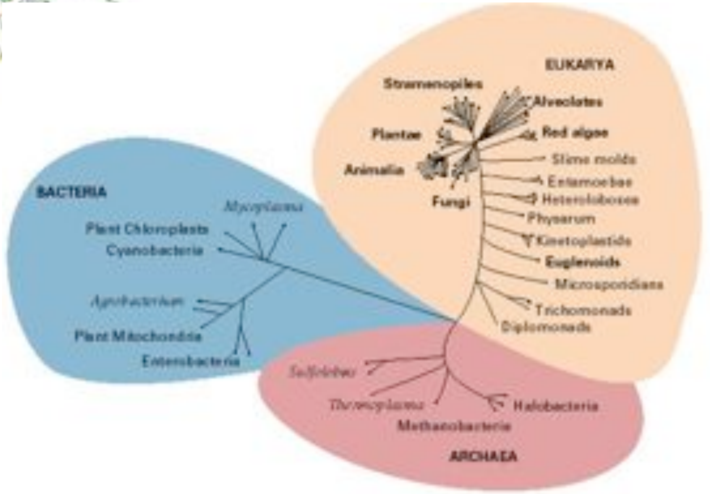


Vanessa M. Brown et al. Genome Res. 2002; 12: 868-884

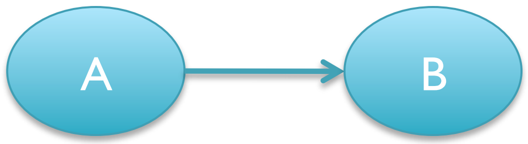
Cold Spring Harbor Laboratory Press





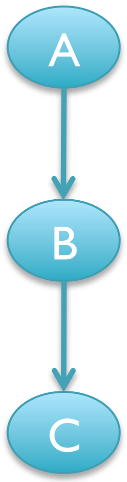


Graphs

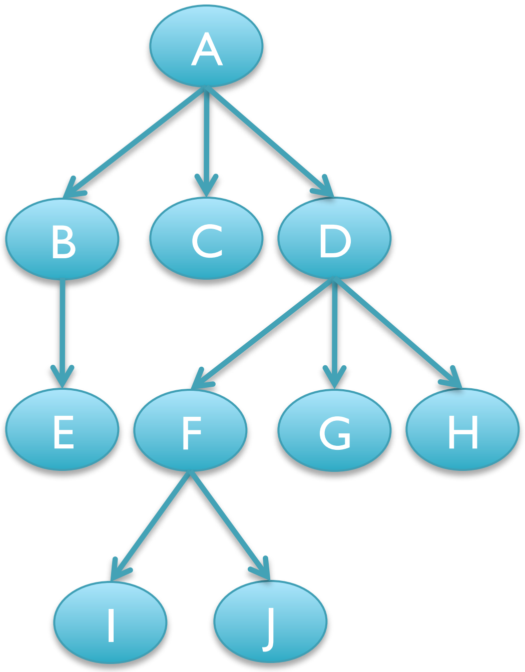


- Nodes
  - People, Proteins, Genes, Neurons, Sequences, Numbers, …
- Edges
  - A is connected to B
  - A is related to B
  - A regulates B
  - A precedes B
  - A interacts with B
  - A activates B
  - …

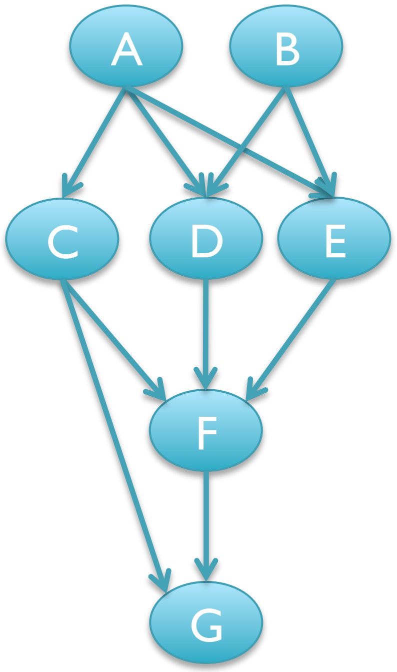
Graph Types



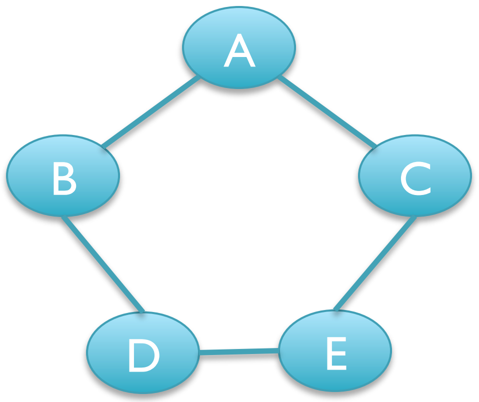
List



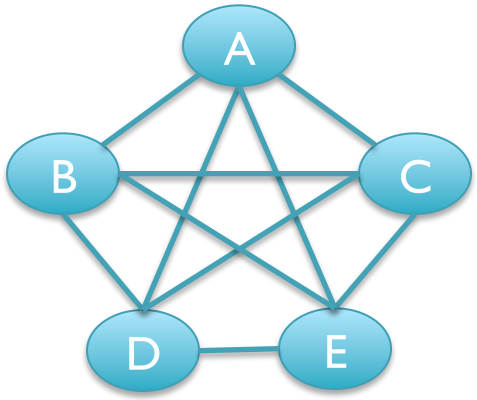
Tree



Directed
Acyclic
Graph

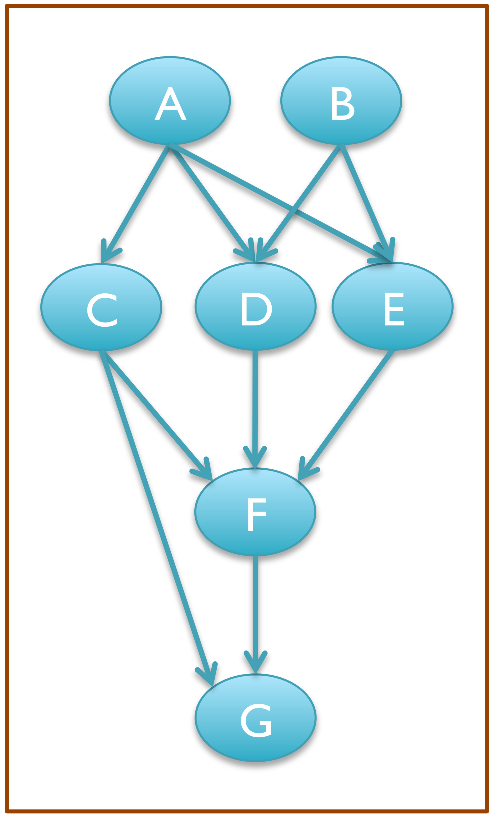


Cycle



Complete

Representing Graphs



Adjacency Matrix

Good for dense graphs
Fast, Fixed storage: N^2 bits

	A	B	C	D	E	F	G
A			1	1	1		
B				1	1		
C						1	1
D						1	
E						1	
F							1
G							

Adjacency List

Good for sparse graphs
Compact storage: 4 bytes/edge

A: C, D, E
B: D, E
C: F, G
D: F
E: F
G:

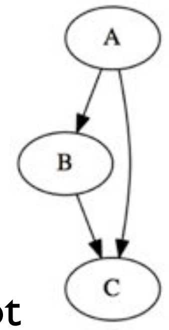
Edge List

Easy, good if you (mostly) need to iterate through the edges
8 bytes / edge

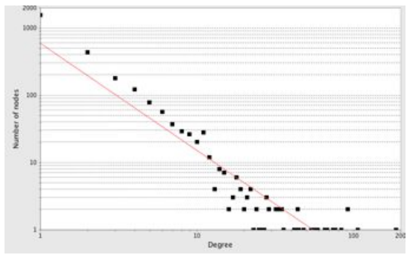
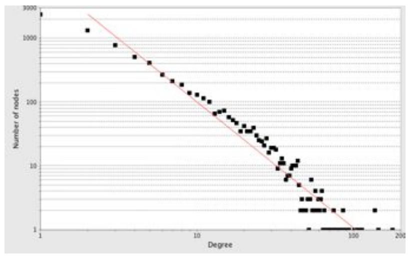
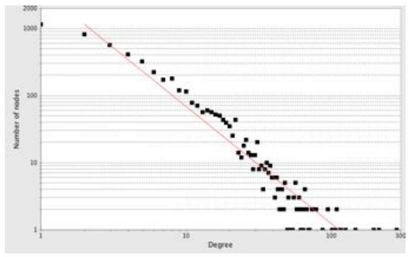
A,C	B,C	C,F
A,D	B,D	C,G
A,E	B,E	D,F
E,F	F,G	

Tools
Matlab: http://www.mathworks.com/
Graphviz: http://www.graphviz.org/
Gephi: https://gephi.org/
Cytoscape: http://www.cytoscape.org/

```
digraph G {
	A->B
	B->C
	A->C
}
dot –Tpdf -og.pdf g.dot
```



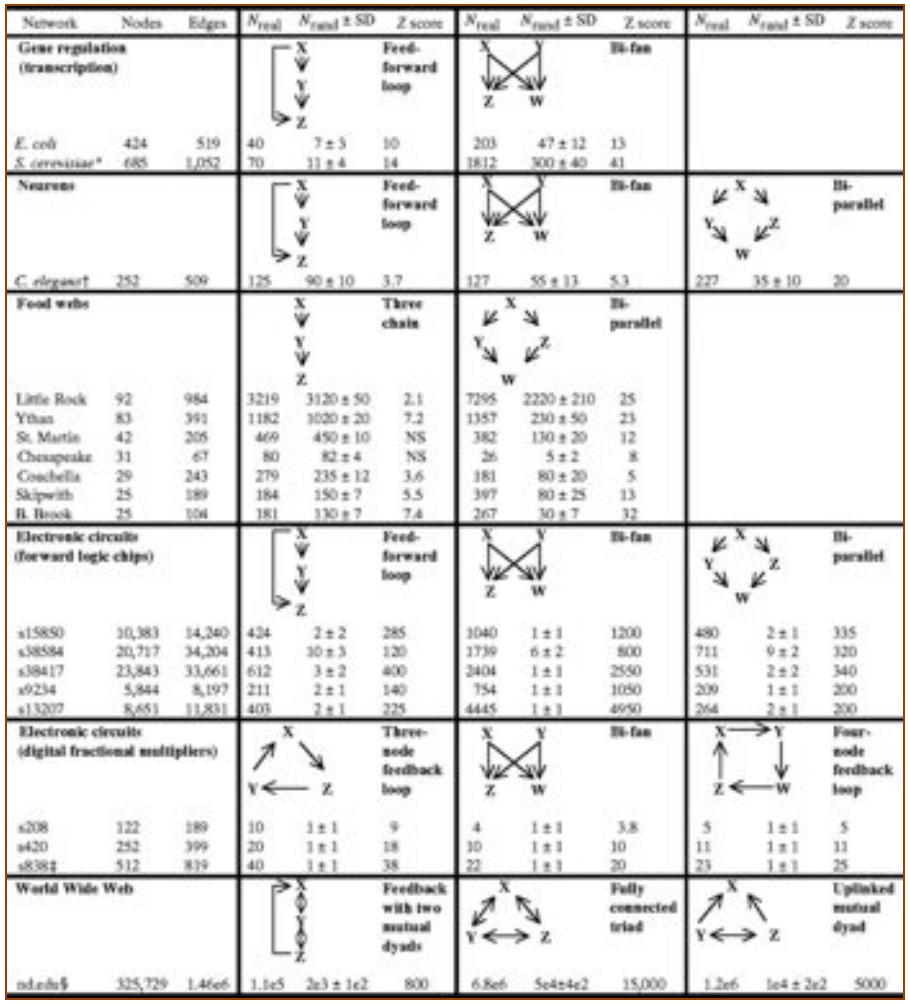
Network Characteristics

	C. elegans	*D. melanogaster*	*S. cerevisiae*
# Nodes	2646	7464	4965
# Edges	4037	22831	17536
Avg. / Max Degree	3.0 / 187	6.1 / 178	7.0 / 283
# Components	109	66	32
Largest Component	2386	7335	4906
Diameter	14	12	11
Avg. Shortest Path	4.8	4.4	4.1
Data Sources	2H	2x2H, TAP-MS	8x2H, 2xTAP, SUS
Degree Distributions			

Small World: Avg. Shortest Path between nodes is small
Scale Free: Power law distribution of degree – preferential attachment

Network Motifs

- Network Motif
  - Simple graph of connections
  - Exhaustively enumerate all possible 1, 2, 3, … k node motifs
- Statistical Significance
  - Compare frequency of a particular network motif in a real network as compared to a randomized network
- Certain motifs are "characteristic features" of the network

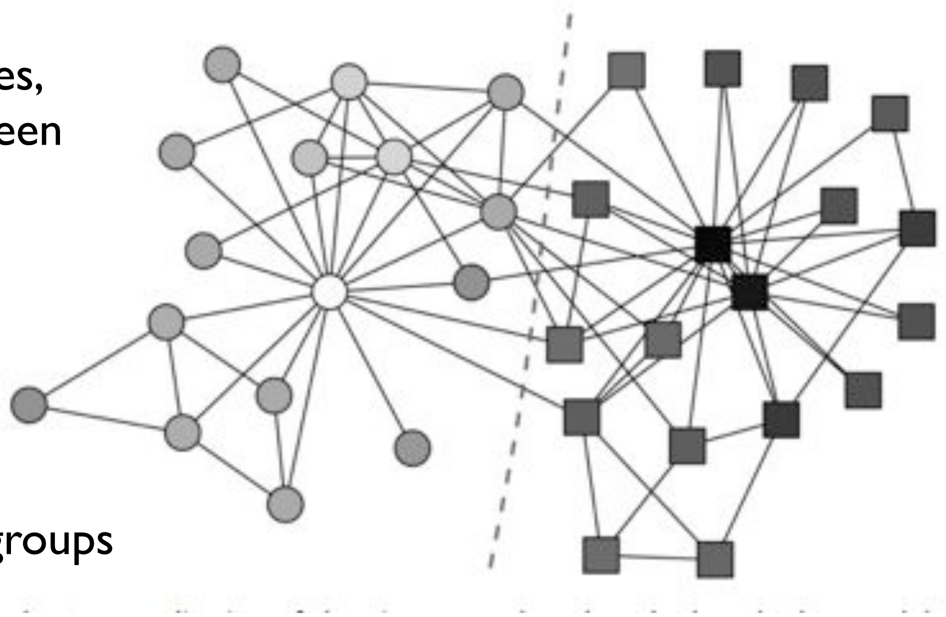


Network	Nodes	Edges	N_{real}	$N_{rand} \pm SD$	Z score	N_{real}	$N_{rand} \pm SD$	Z score	N_{real}	$N_{rand} \pm SD$	Z score
Gene regulation (transcription)					**Feed-forward loop**			**Bi-fan**			
E. coli	424	519	40	7 ± 3	10	203	47 ± 12	13			
*S. cerevisiae**	685	1,052	70	11 ± 4	14	1812	300 ± 40	41			
Neurons					**Feed-forward loop**			**Bi-fan**			**Bi-parallel**
C. elegans†	252	509	125	90 ± 10	3.7	127	55 ± 13	5.3	227	35 ± 10	20
Food webs					**Three chain**			**Bi-parallel**			
Little Rock	92	984	3219	3120 ± 50	2.1	7295	2220 ± 210	25			
Ythan	83	391	1182	1020 ± 20	7.2	1357	230 ± 50	23			
St. Martin	42	205	469	450 ± 10	NS	382	130 ± 20	12			
Chesapeake	31	67	80	82 ± 4	NS	26	5 ± 2	8			
Coachella	29	243	279	235 ± 12	3.6	181	80 ± 20	5			
Skipwith	25	189	184	150 ± 7	5.5	397	80 ± 25	13			
B. Brook	25	104	181	130 ± 7	7.4	267	30 ± 7	32			
Electronic circuits (forward logic chips)					**Feed-forward loop**			**Bi-fan**			**Bi-parallel**
s15850	10,383	14,240	424	2 ± 2	285	1040	1 ± 1	1200	480	2 ± 1	335
s38584	20,717	34,204	413	10 ± 3	120	1739	6 ± 2	800	711	9 ± 2	320
s38417	23,843	33,661	612	3 ± 2	400	2404	1 ± 1	2550	531	2 ± 2	340
s9234	5,844	8,197	211	2 ± 1	140	754	1 ± 1	1050	209	1 ± 1	200
s13207	8,651	11,831	403	2 ± 1	225	4445	1 ± 1	4950	264	2 ± 1	200
Electronic circuits (digital fractional multipliers)					**Three-node feedback loop**			**Bi-fan**			**Four-node feedback loop**
s208	122	189	10	1 ± 1	9	4	1 ± 1	3.8	5	1 ± 1	5
s420	252	399	20	1 ± 1	18	10	1 ± 1	10	11	1 ± 1	11
s838‡	512	819	40	1 ± 1	38	22	1 ± 1	20	23	1 ± 1	25
World Wide Web					**Feedback with two mutual dyads**			**Fully connected triad**			**Uplinked mutual dyad**
nd.edu§	325,729	1.46e6	1.1e5	2e3 ± 1e2	800	6.8e6	5e4±4e2	15,000	1.2e6	1e4 ± 2e2	5000

Network Motifs: Simple Building Blocks of Complex Networks
Milo et al (2002) *Science. 298:824-827*

Modularity

- Community structure
  - Densely connected groups of vertices, with only sparser connections between groups
  - Reveals the structure of large-scale network data sets



- Modularity
  - The number of edges falling within groups minus the expected number in an equivalent network with edges placed at random
  - Larger positive values => Stronger community structure
  - Optimal assignment determined by computing the eigenvector of the modularity matrix

$$Q = \boxed{\frac{1}{4m}} \sum_{ij} \left(\boxed{A_{ij}} - \boxed{\frac{k_i k_j}{2m}} \right) \boxed{(s_i s_j + 1)}$$

Normalization factor

Adjacency matrix

Random Prob. (product of degrees)

Indicates same group

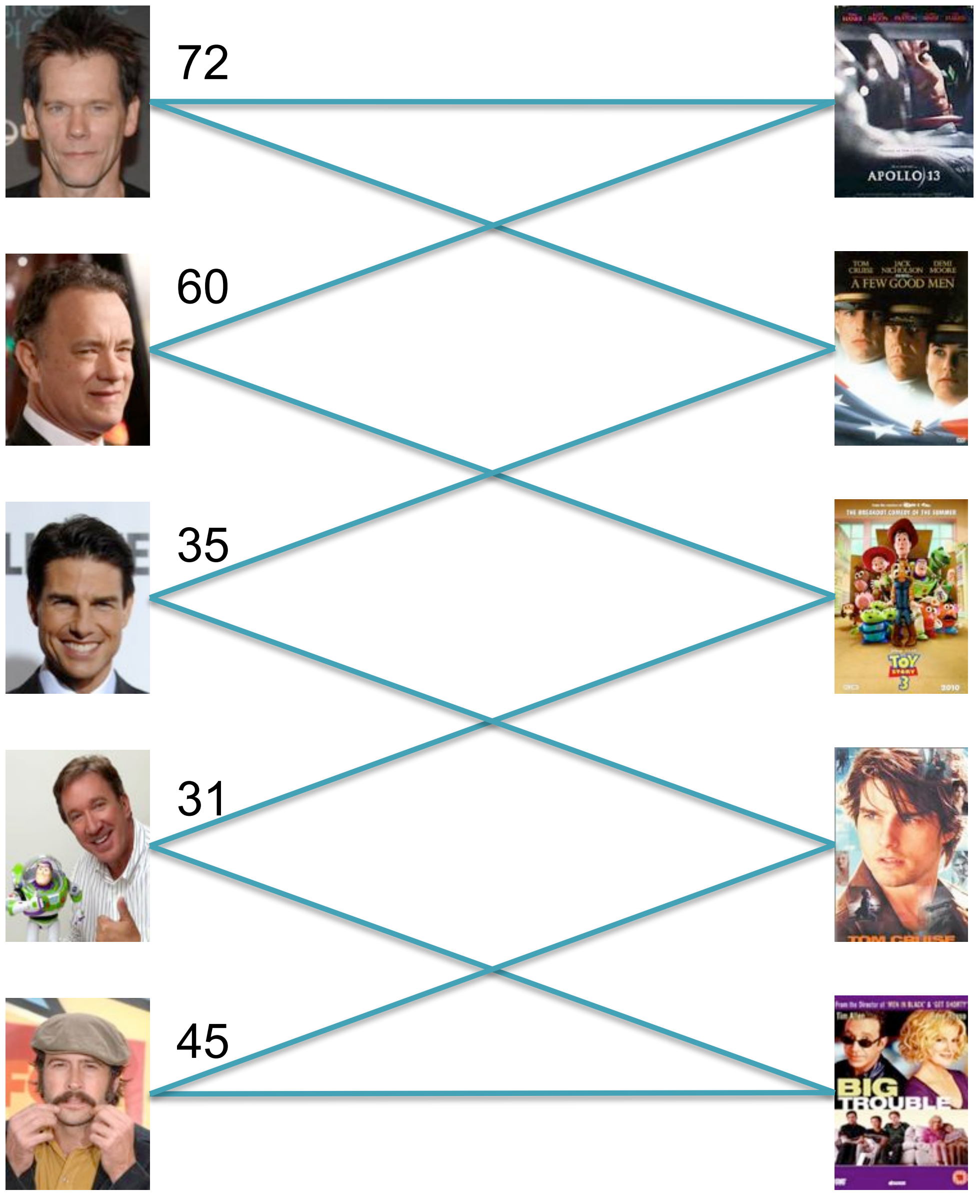
Modularity and community structure in networks. Newman ME (2006) *PNAS. 103(23) 8577-8582*

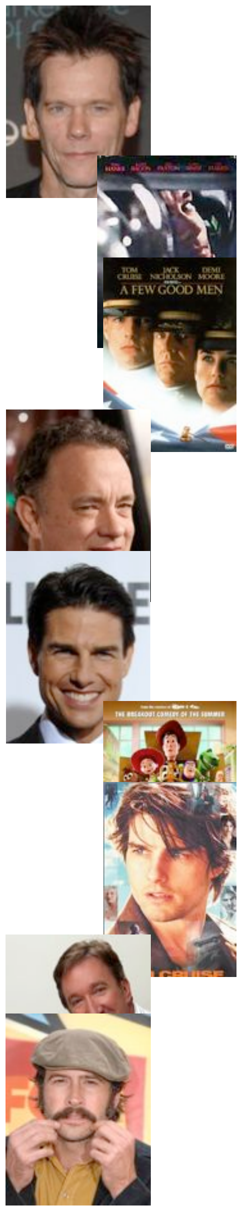
Kevin Bacon and Bipartite Graphs

Find the ***shortest*** path from Kevin Bacon to Jason Lee

Breadth First Search: 4 hops

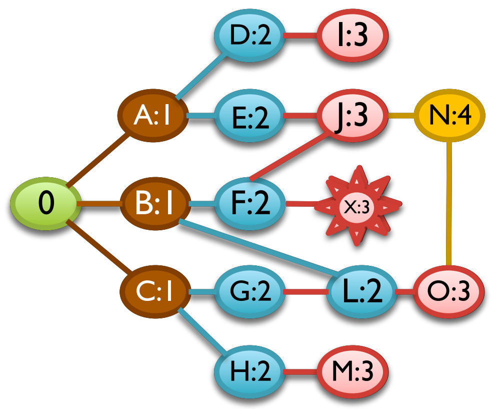
Bacon Distance: 2





BFS

```
BFS(start, stop)
// initialize all nodes dist = -1
start.dist = 0
list.addEnd(start)
while (!list.empty())
   cur = list.begin()
   if (cur == stop)
      print cur.dist;
   else
      foreach child in cur.children
         if (child.dist == -1)
            child.dist = cur.dist+1
            list.addEnd(child)
```



0

A,B,C
B,C,D,E
C,D,E,F,L

D,E,F,L,G,H
E,F,L,G,H,I
F,L,G,H,I,J
L,G,H,I,J,X
G,H,I,J,X,O
H,I,J,X,O

I,J,X,O,M
J,X,O,M
X,O,M,N
O,M,N
M,N

N

[How many nodes will it visit?]

[What's the running time?]

[What happens for disconnected components?]

BFS

```
BFS(start, stop)
// initialize all nodes dist = -1
start.dist = 0
list.addEnd(start)
while (!list.empty())
    cur = list.begin()
    if (cur == stop)
        print cur.dist;
    else
        foreach child in cur.children
            if (child.dist == -1)
                child.dist = cur.dist+1
                list.addEnd(child)
```

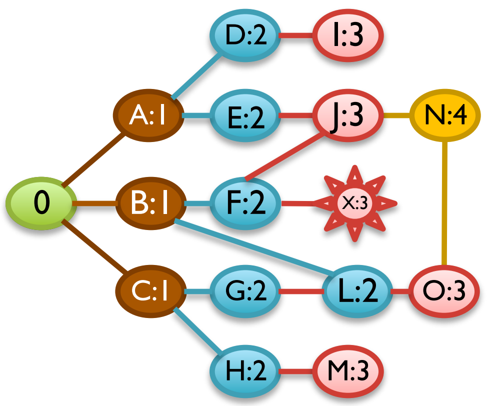
0

A,B,C
B,C,D,E
C,D,E,F,L

D,E,F,L,G,H
E,F,L,G,H,I
F,L,G,H,I,J
L,G,H,I,J,X
G,H,I,J,X,O
H,I,J,X,O

I,J,X,O,M
J,X,O,M
X,O,M,N
O,M,N
M,N

N



DFS

```
DFS(start, stop)
// initialize all nodes dist = -1
start.dist = 0
list.addEnd(start)
while (!list.empty())
    cur = list.end()
    if (cur == stop)
        print cur.dist;
    else
        foreach child in cur.children
            if (child.dist == -1)
                child.dist = cur.dist+1
                list.addEnd(child)
```

0

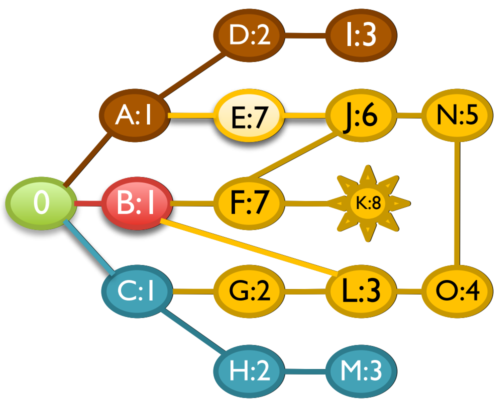
A,B,C

A,B,G,H
A,B,G,M

A,B,G
A,B,L
A,B,O
A,B,N
A,B,J
A,B,E,F
A,B,E,K
A,B,E

A,B

A
D
I



BFS and TSP

- BFS computes the shortest path between a pair of nodes in O(|E|) = O(|N|^2)
- What if we wanted to compute the shortest path visiting every node once?
  - Traveling Salesman Problem

ABDCA: 4+2+5+3 = 14

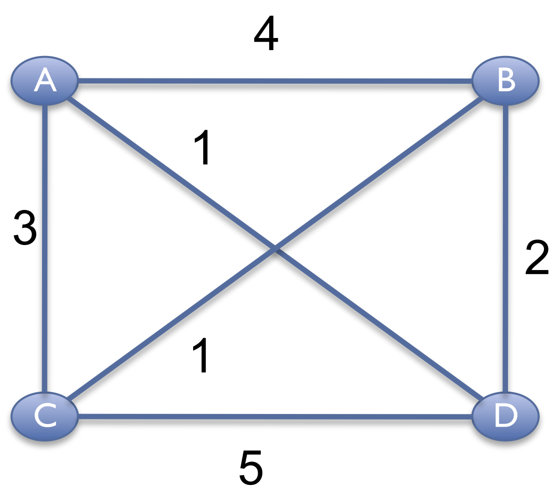
ACDBA: 3+5+2+4 = 14*

ABCDA: 4+1+5+1 = 11

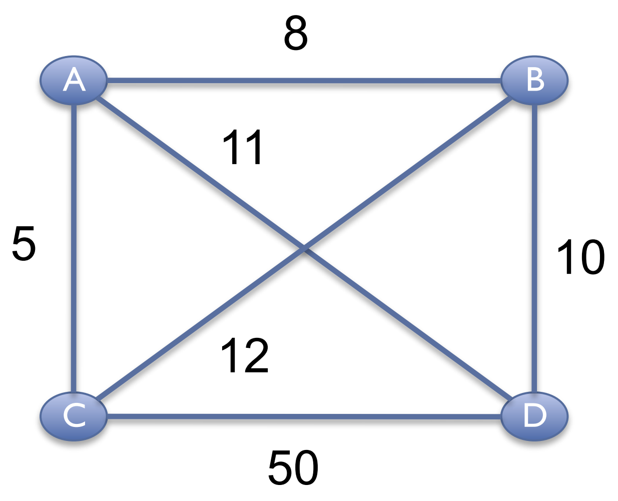
ADCBA: 1+5+1+4 = 11*

ACBDA: 3+1+2+1 = 7

ADBCA: 1+2+1+3= 7 *

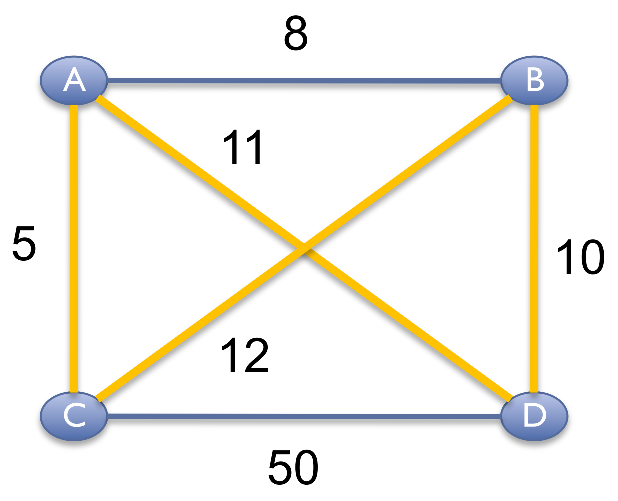


Greedy Search



Greedy Search

```
Greedy Search
cur=graph.randNode()
while (!done)
    next=cur.getNextClosest()
```



Greedy: ABDCA = 5+8+10+50= 73

Optimal: ACBDA = 5+11+10+12 = 38

Greedy finds the global optimum only when

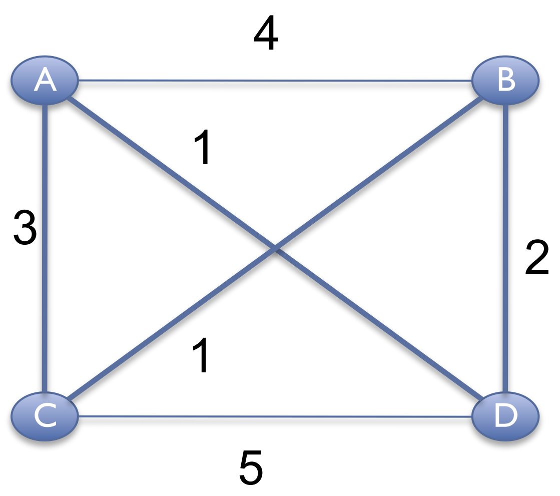
1. Greedy Choice: Local is correct without reconsideration
2. Optimal Substructure: Problem can be split into subproblems

Optimal Greedy: Making change with the fewest number of coins

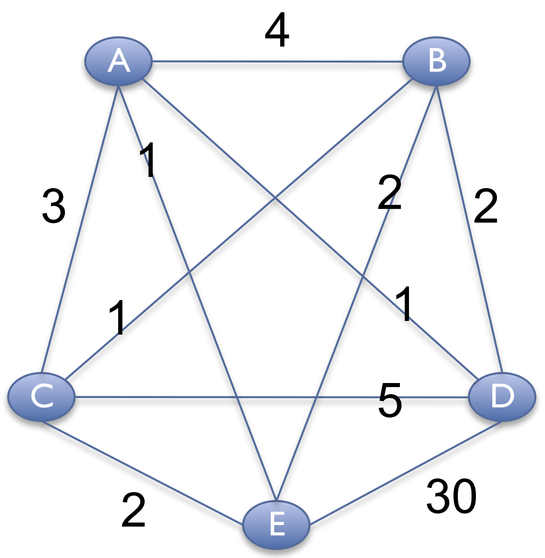
TSP Complexity

- No fast solution
  - Knowing optimal tour through n cities doesn't seem to help much for n+1 cities

[How many possible tours for n cities?]

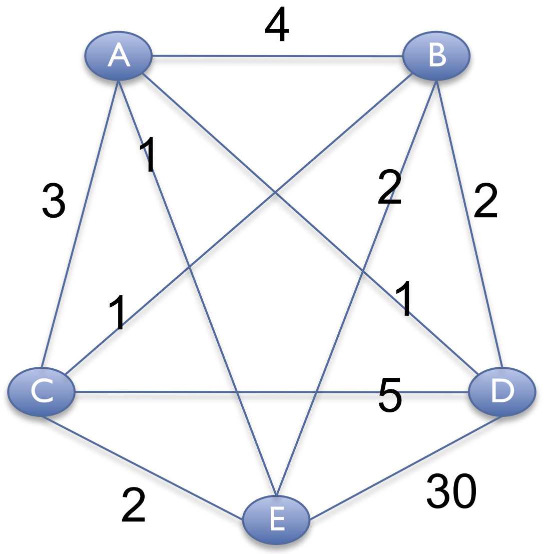


- Extensive searching is the only provably correct algorithm
  - Brute Force: O(n!)
    - ~20 cities max
    - 20! = 2.4 x 10^{18}



Branch-and-Bound

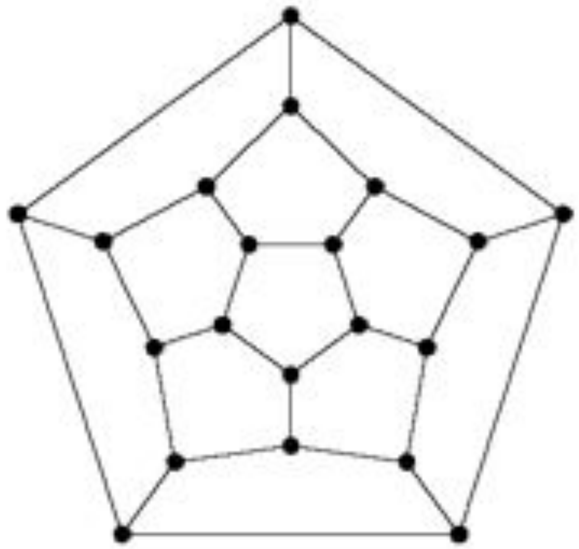
- Abort on suboptimal solutions as soon as possible
  - ADBECA = 1+2+2+2+3 = 10
  - ABDE = 4+2+30 > 10
  - ADE = 1+30 > 10
  - AED = 1+30 > 10
  - …



- Performance Heuristic
  - Always gives the optimal answer
  - Doesn't always help performance, but often does
  - Current TSP record holder:
    - 85,900 cities
    - $85900! = 10^{386526}$

[When not?]

TSP and NP-complete



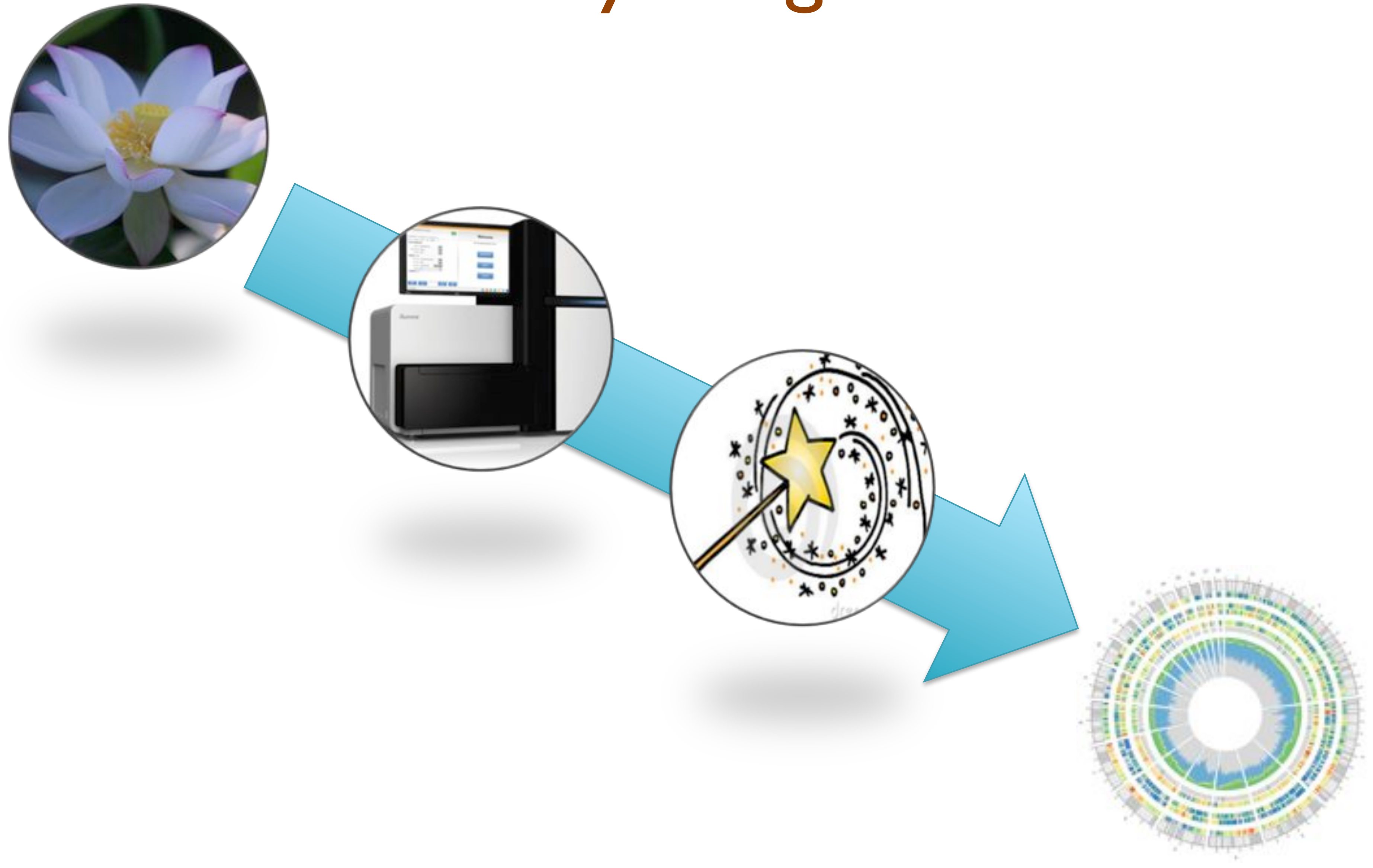
- TSP is one of many extremely hard problems of the class NP-complete
  - Extensive searching is the only way to find an exact solution
  - Often have to settle for approx. solution

- WARNING: Many biological problems are in this class
  - Find a tour the visits every node once (Genome Assembly)
  - Find the smallest set of vertices covering the edges (Essential Genes)
  - Find the largest clique in the graph (Protein Complexes)
  - Find the highest mutual information encoding scheme (Neurobiology)
  - Find the best set of moves in tetris
  - …
  - http://en.wikipedia.org/wiki/List_of_NP-complete_problems

Break

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

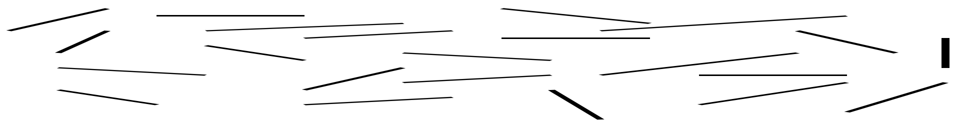
What is your genome?



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

Sequencing a Genome

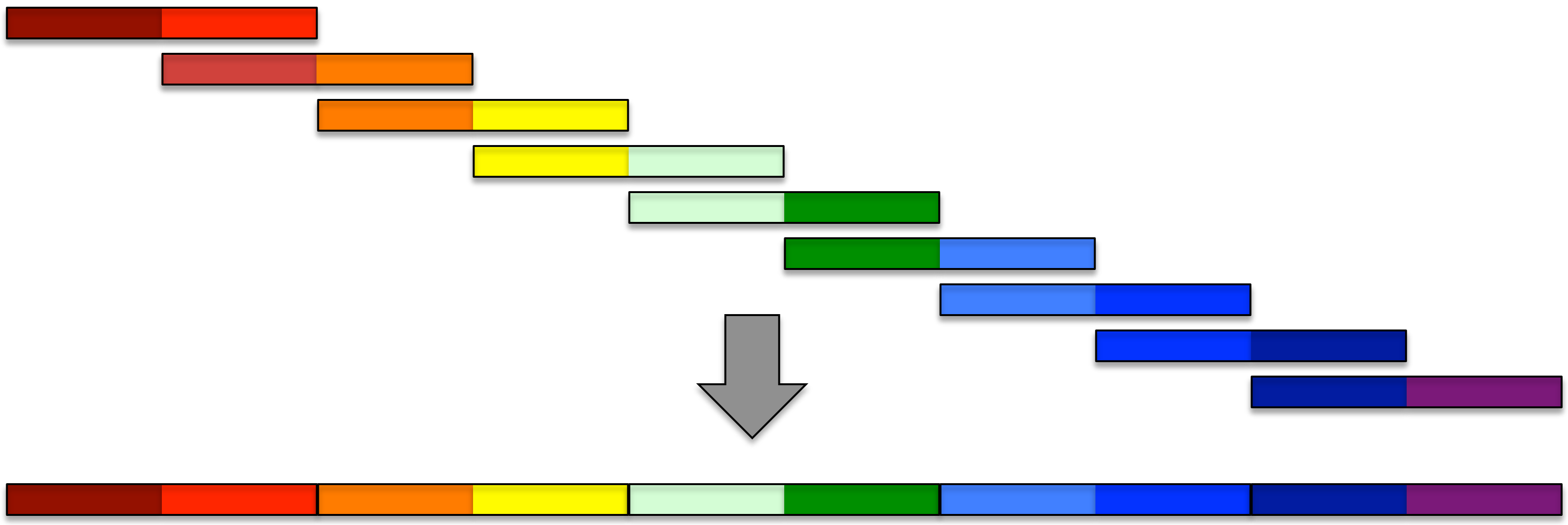
1. Shear & Sequence DNA



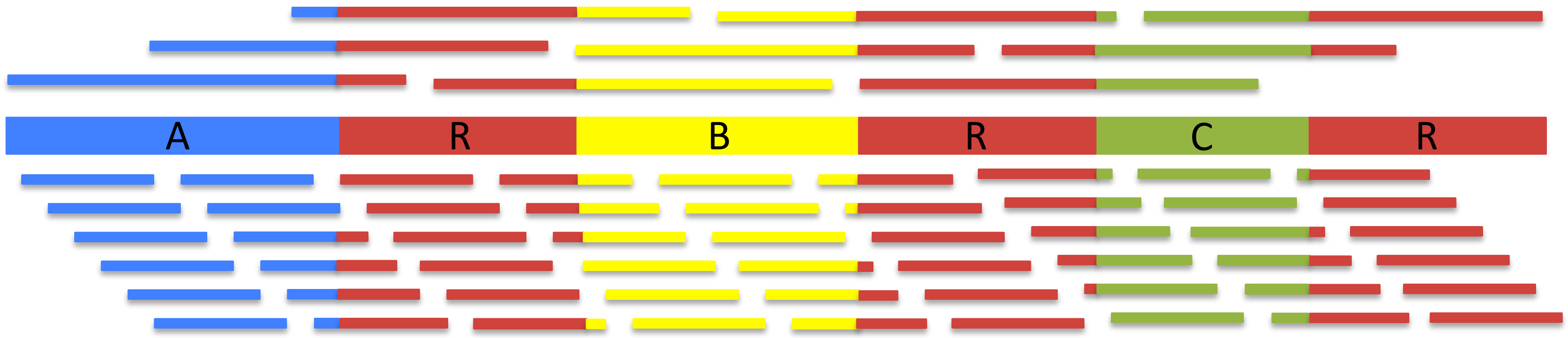
2. Construct assembly graph from overlapping reads

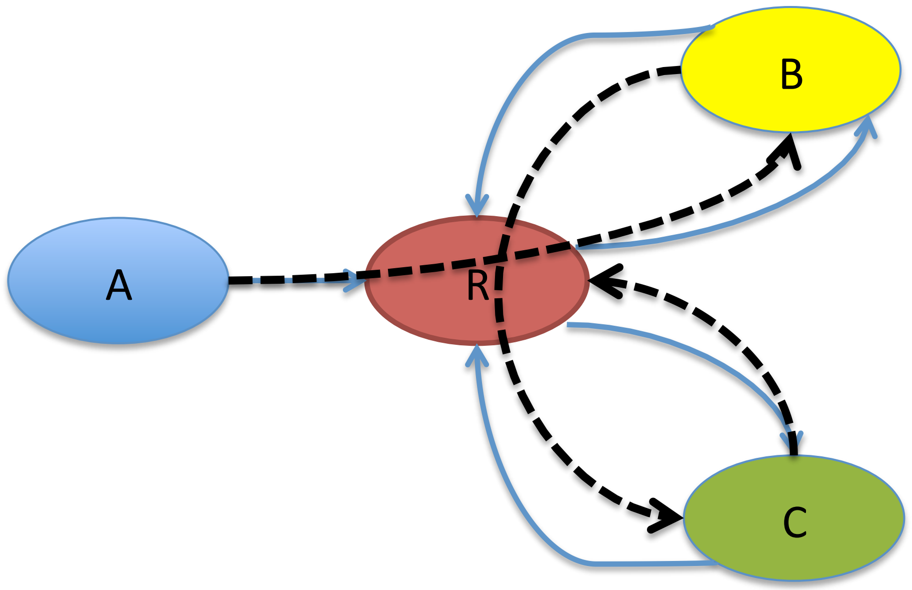
```
...AGCCTAGGGATGCGCGACACGT
          GGATGCGCGACACGTCGCATATCCGGTTTGGTCAACCTCGGACGGAC
                                          CAACCTCGGACGGACCTCAGCGAA...
```

3. Simplify assembly graph

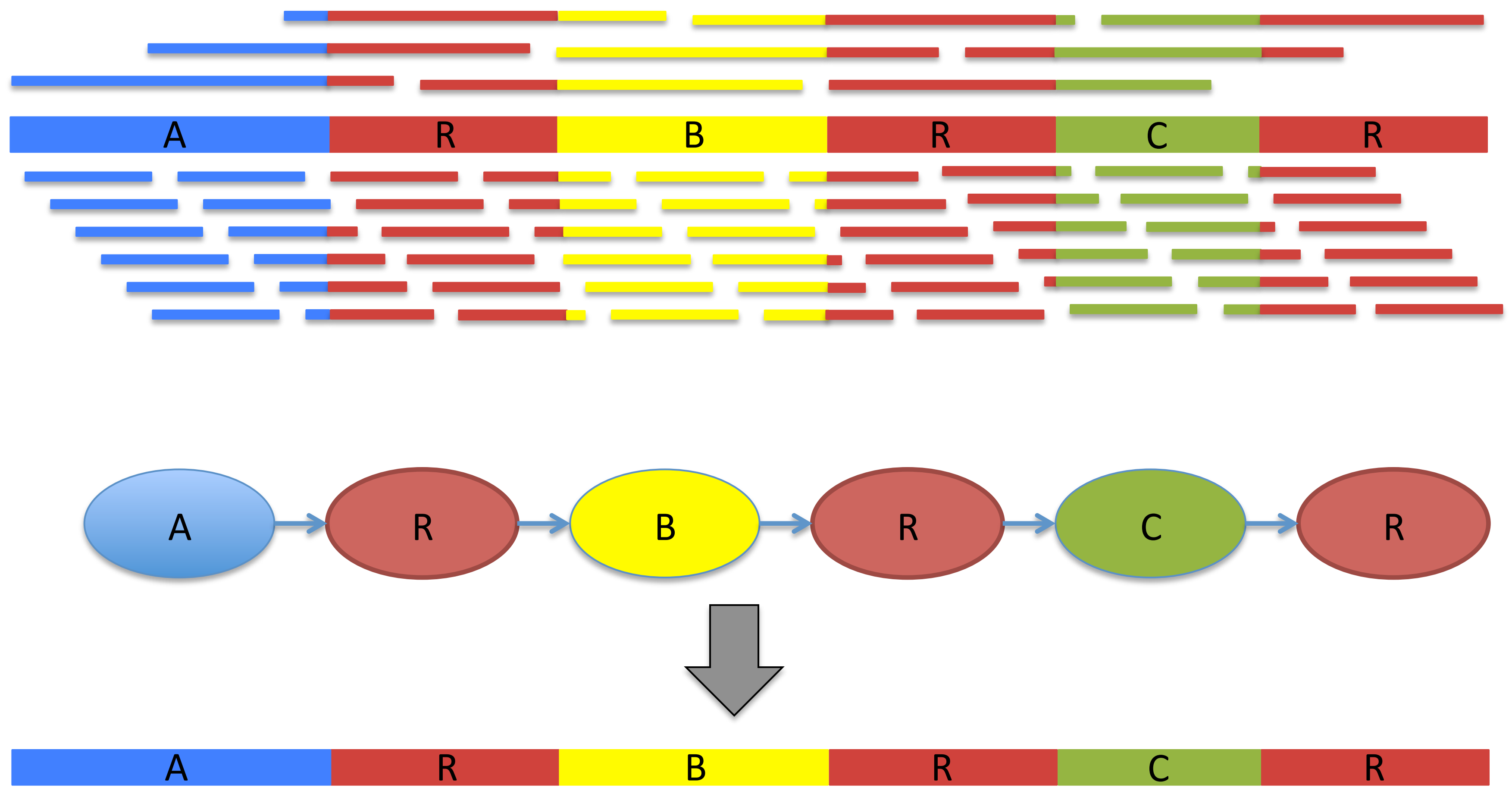


Assembly Complexity





Assembly Complexity



The advantages of SMRT sequencing
Roberts, RJ, Carneiro, MO, Schatz, MC (2013) *Genome Biology.* 14:405

Milestones in Genome Assembly

articles

Nucleotide sequence of bacteriophage ΦX174 DNA

F. Sanger, G. M. Air, B. G. Barrell, N. L. Brown, A. R. Coulson, J. C. Fiddes, C. A. Hutchison III, P. M. Slocombe & M. Smith

[illegible]

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

1995. Fleischmann *et al.*
1st Free Living Organism
TIGR Assembler. 1.8Mbp

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

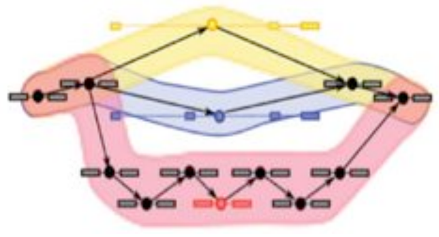
2000. Myers *et al.*
1st Large WGS Assembly.
Celera Assembler. 116 Mbp

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

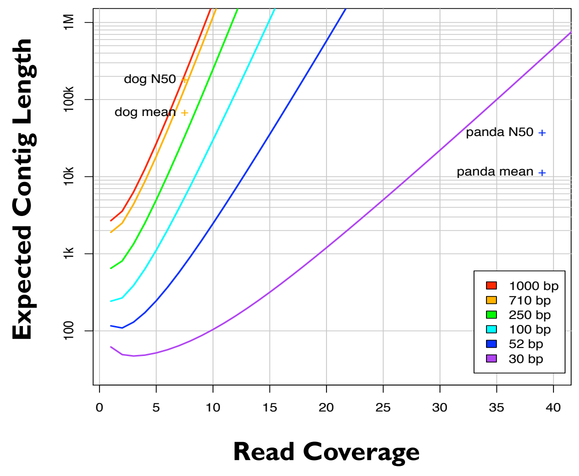
Assembly Applications

- Novel genomes
- Metagenomes
- Sequencing assays
  - Structural variations
  - Transcript assembly
  - …



Ingredients for a good assembly

Coverage

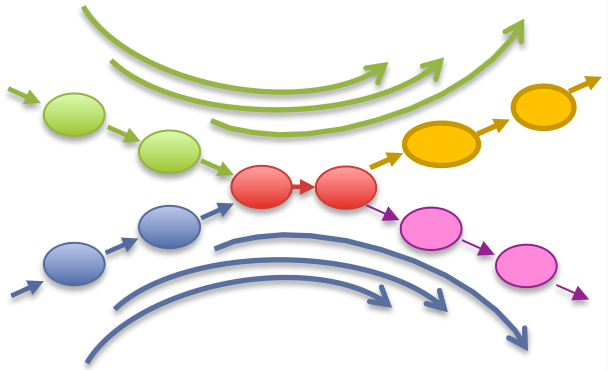


High coverage is required

– Oversample the genome to ensure every base is sequenced with long overlaps between reads

– Biased coverage will also fragment assembly

Read Length

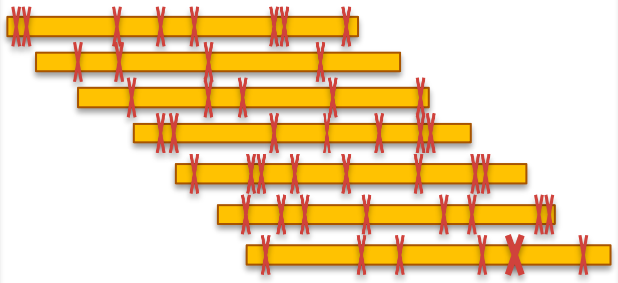


Reads & mates must be longer than the repeats

– Short reads will have ***false overlaps*** forming hairball assembly graphs

– With long enough reads, assemble entire chromosomes into contigs

Quality



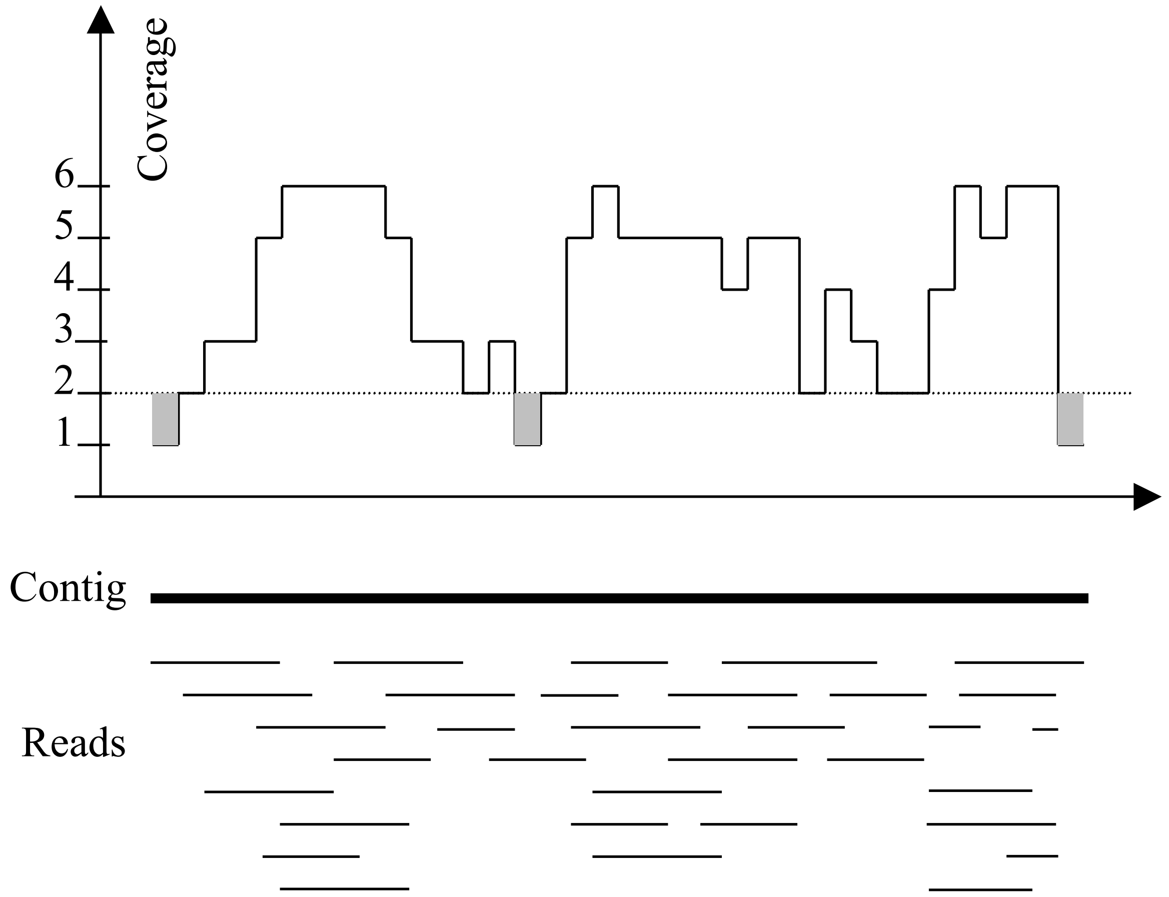
Errors obscure overlaps

– 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

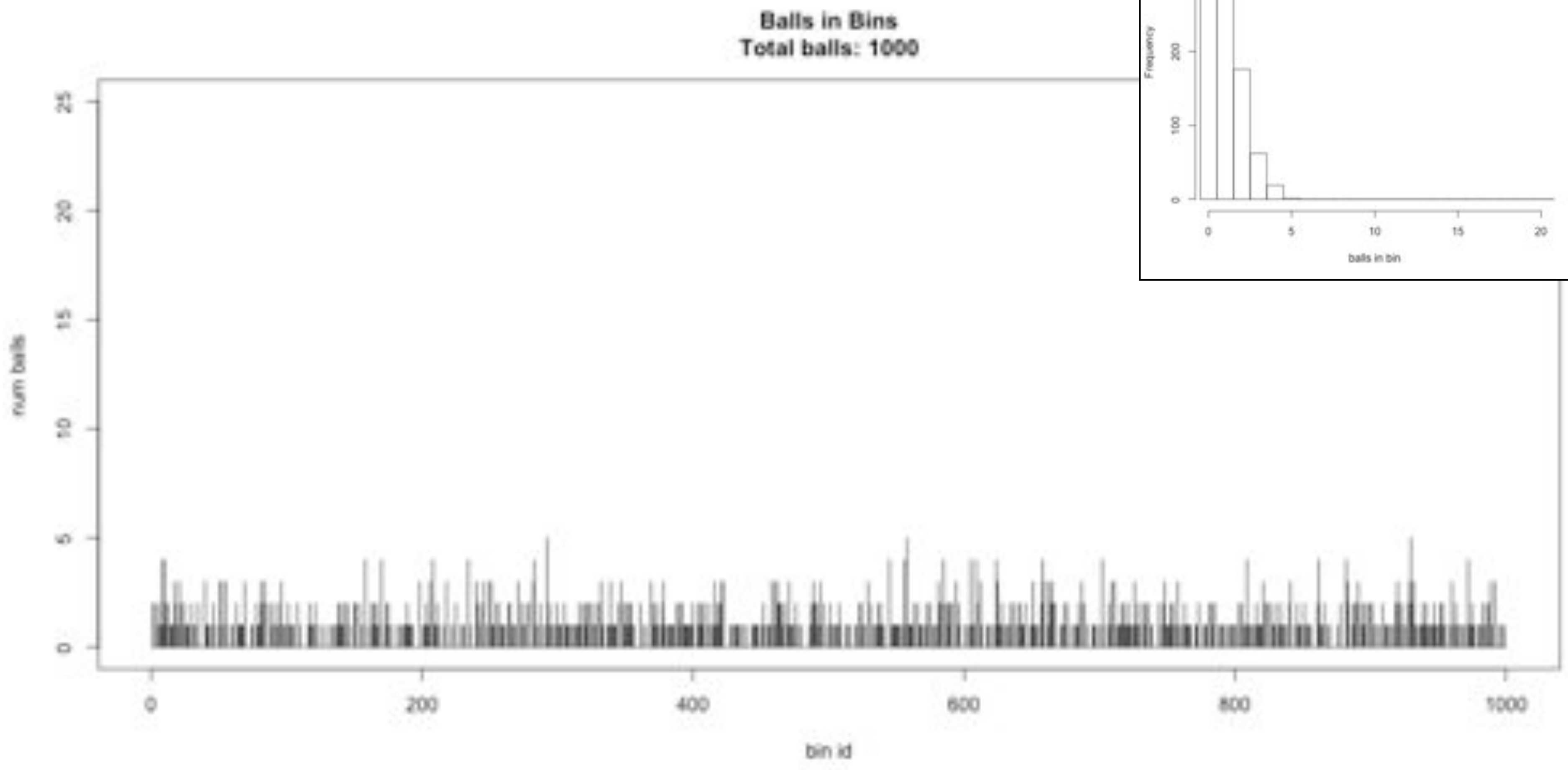
Typical sequencing coverage

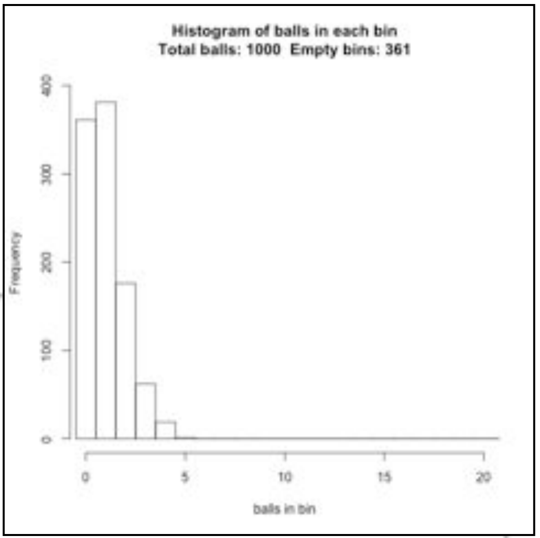


Imagine raindrops on a sidewalk

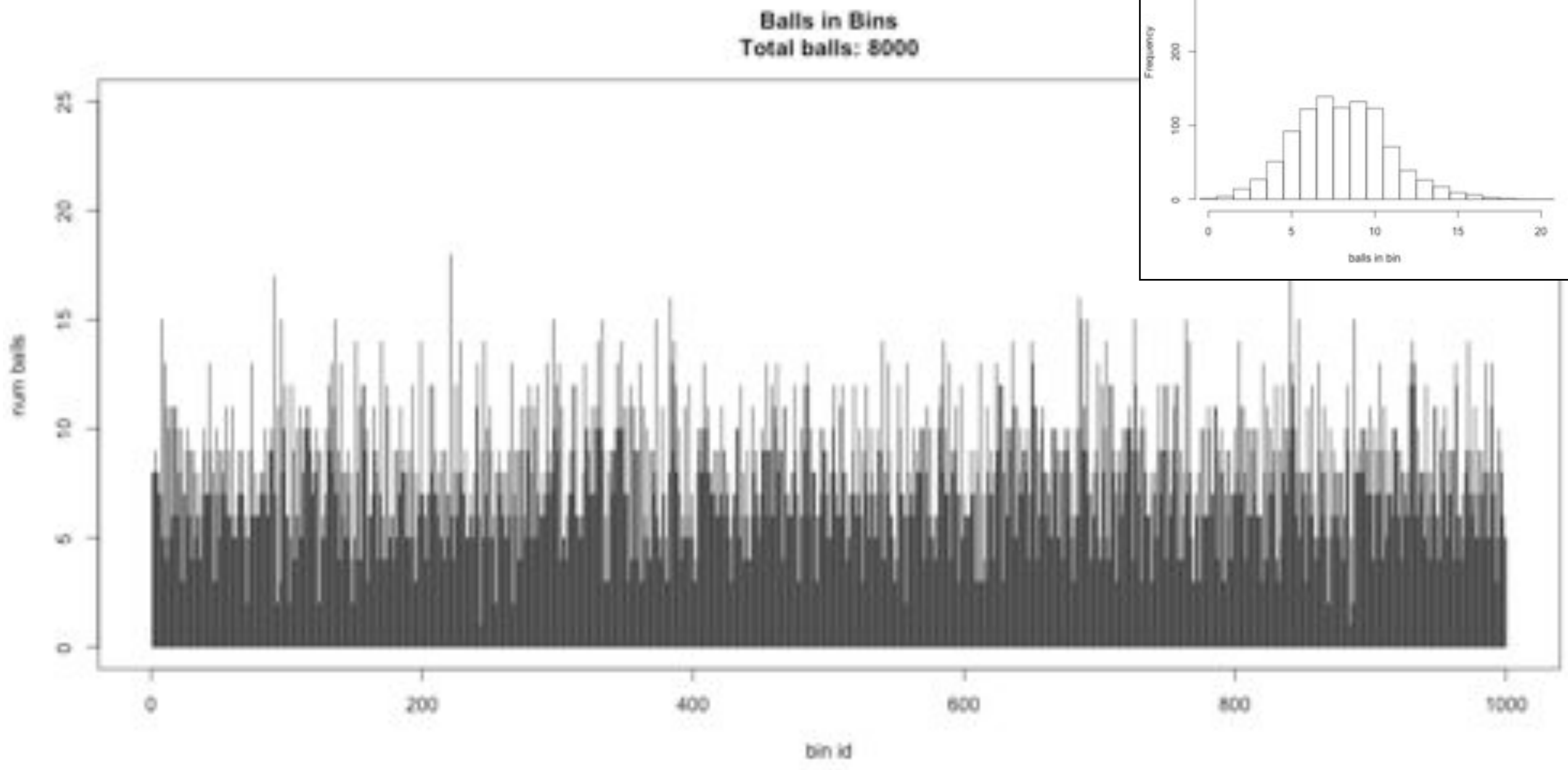
We want to cover the entire sidewalk but each drop costs $1

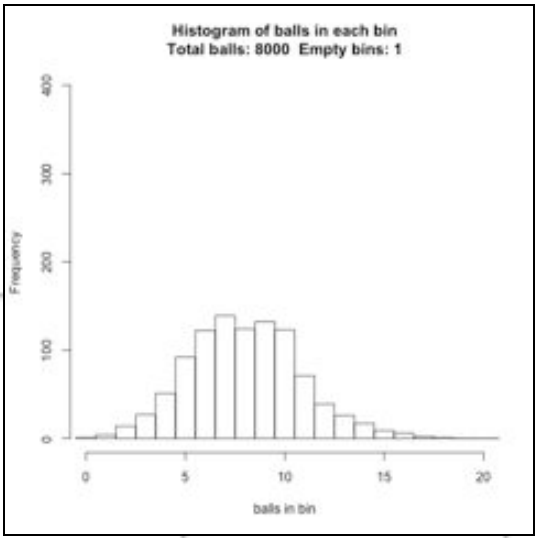
1x sequencing





8x sequencing





Poisson Distribution

The probability of a given number of events occurring in a fixed interval of time and/or space if these events occur with a known average rate and independently of the time since the last event.

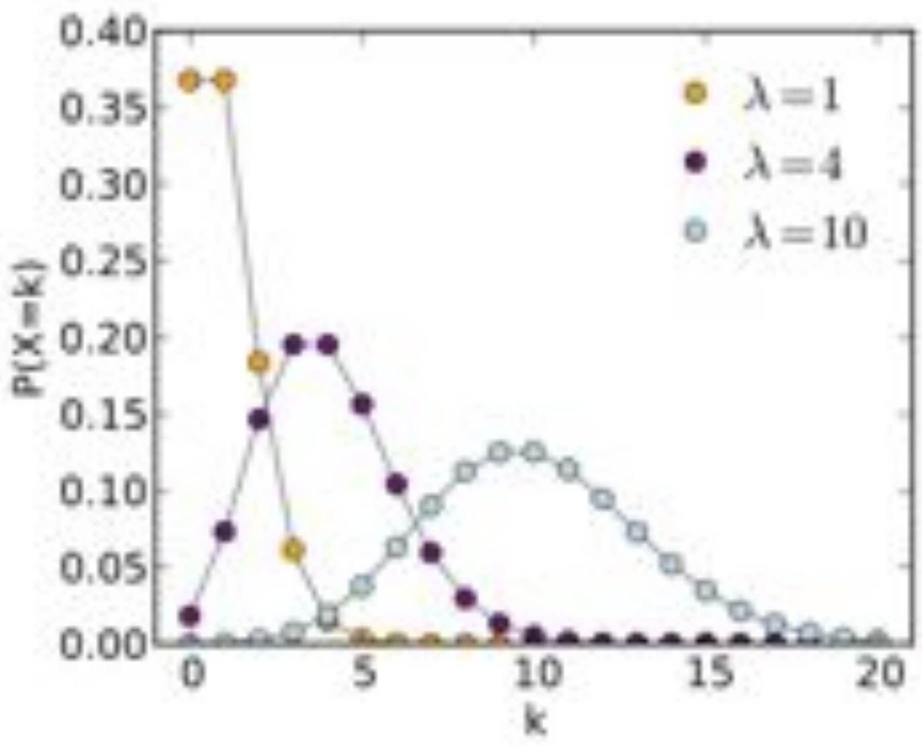
Formulation comes from the limit of the binomial equation

Resembles a normal distribution, but over the positive values, and with only a single parameter.

Key property:

- ***The standard deviation is the square root of the mean.***

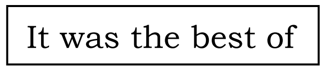
$$P(k) = \frac{\lambda^k}{k!}e^{-\lambda}$$



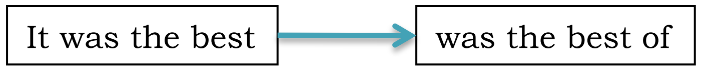
de Bruijn Graph Construction

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

Original Fragment



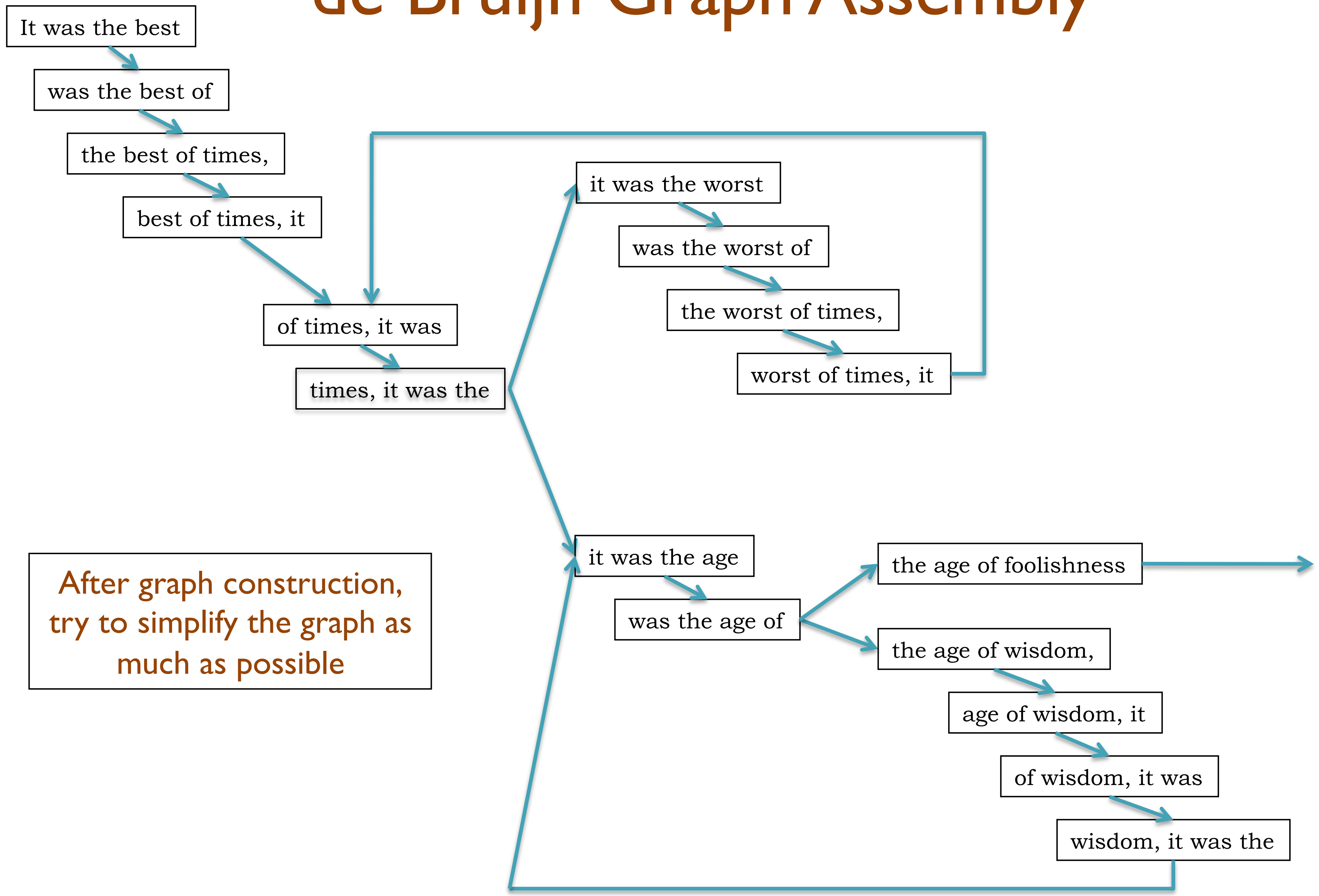
Directed Edge



- 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



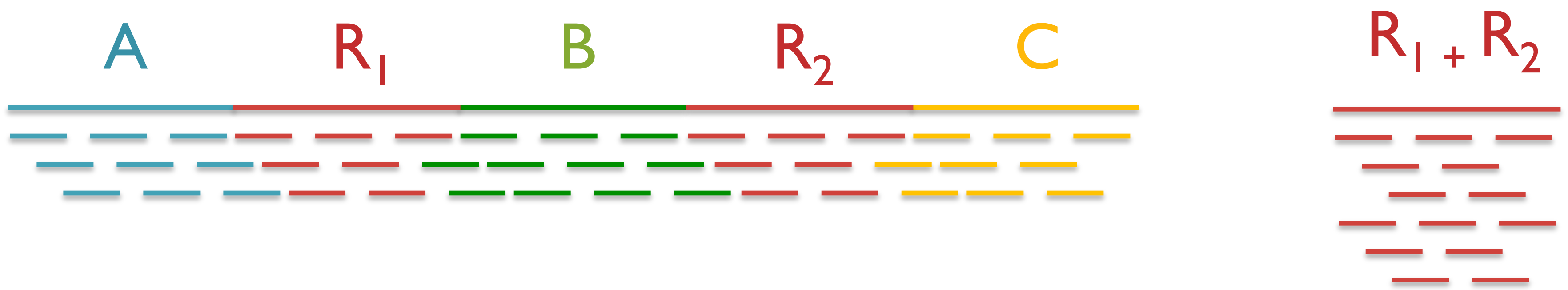
After graph construction, try to simplify the graph as much as possible

Repetitive regions

Repeat Type	Definition / Example	Prevalence
Low-complexity DNA / Microsatellites	$(b_1b_2\ldots b_k)^N$ where $1 \leq k \leq 6$ CACACACACACACACACACA	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%

- 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* *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

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

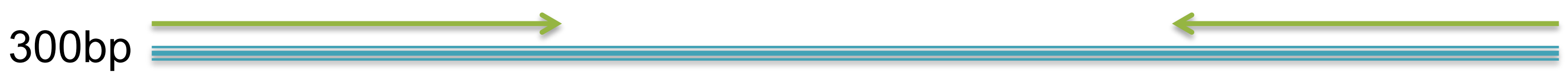
$$A(\Delta,k) = \ln\left(\frac{\Pr(1-copy)}{\Pr(2-copy)}\right) = \ln\left(\frac{\frac{(\Delta n/G)^{k}}{k!}e^{\frac{-\Delta n}{G}}}{\frac{(2\Delta n/G)^{k}}{k!}e^{\frac{-2\Delta n}{G}}}\right) = \frac{n\Delta}{G} - k\ln 2$$

The fragment assembly string graph
Myers, EW (2005) Bioinformatics. 21(suppl 2): ii79-85.

Paired-end and Mate-pairs

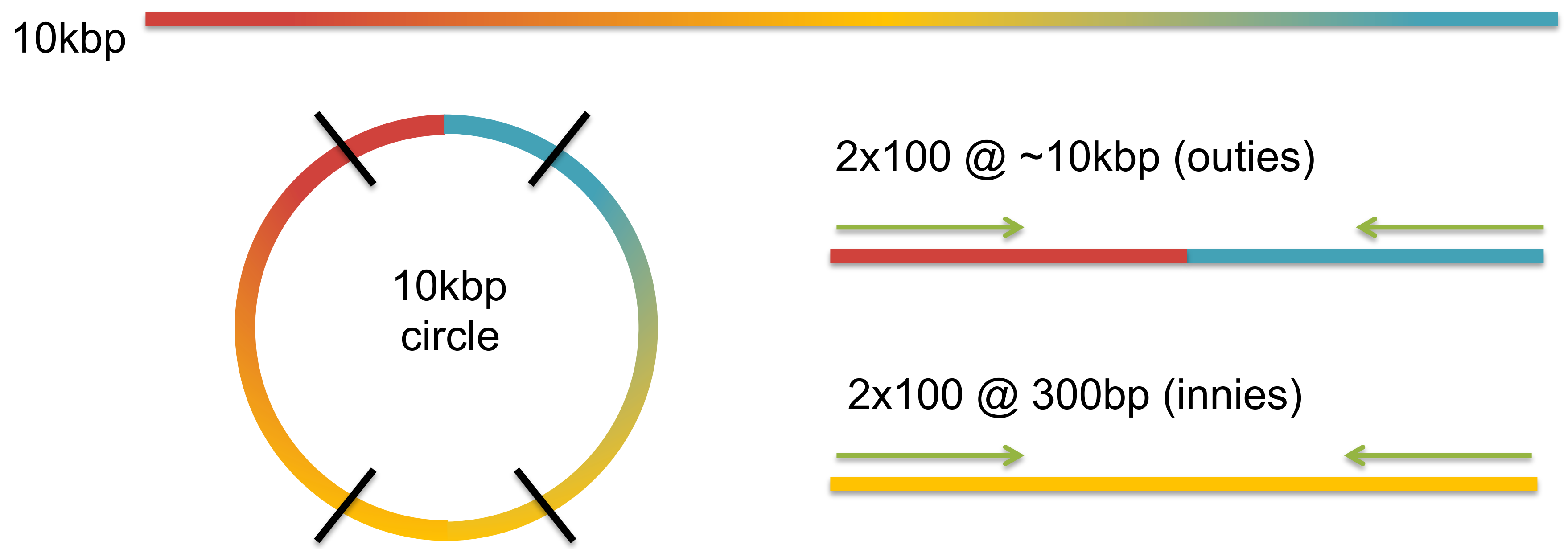
Paired-end sequencing

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



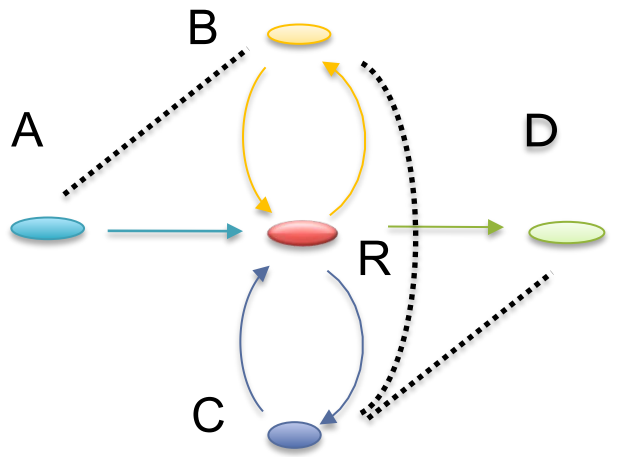
Mate-pair sequencing

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

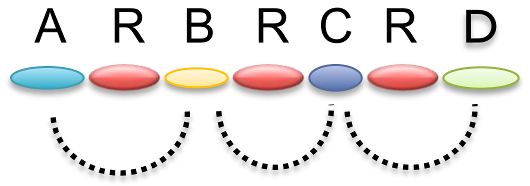


Scaffolding

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



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

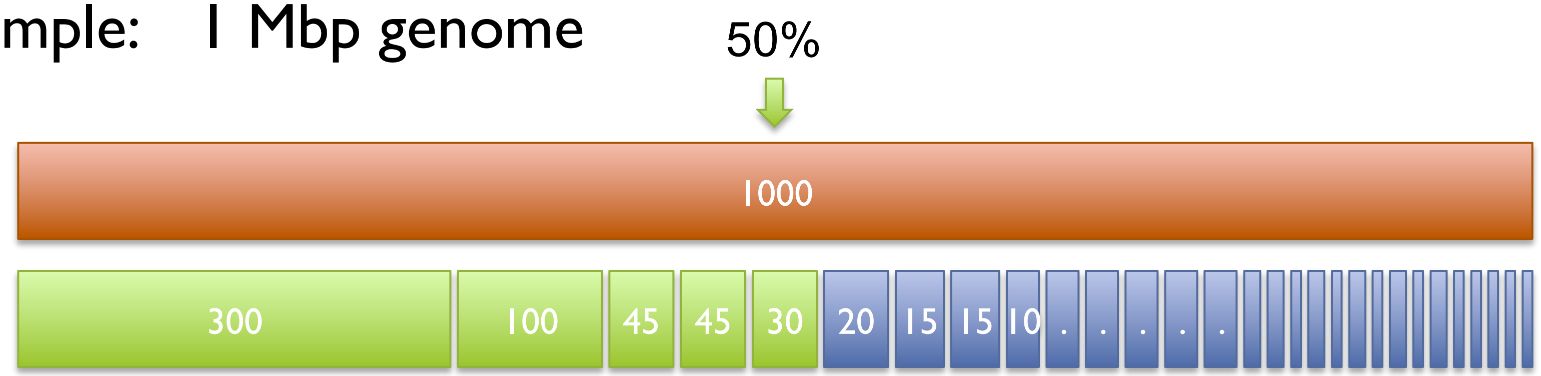


- Final scaffold may have internal gaps called sequencing gaps
  - We know the order, orientation, and spacing, but just not the bases. Fill with Ns instead

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 >= 500kbp)

Note:

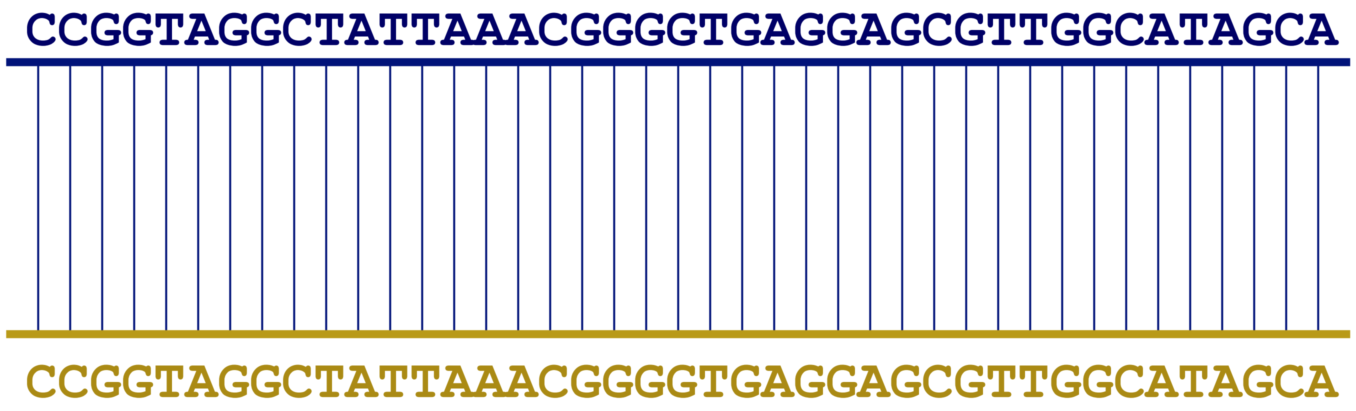
N50 values are only meaningful to compare when base genome size is the same in all cases

Whole Genome Alignment with MUMmer

Slides Courtesy of Adam M. Phillippy
University of Maryland

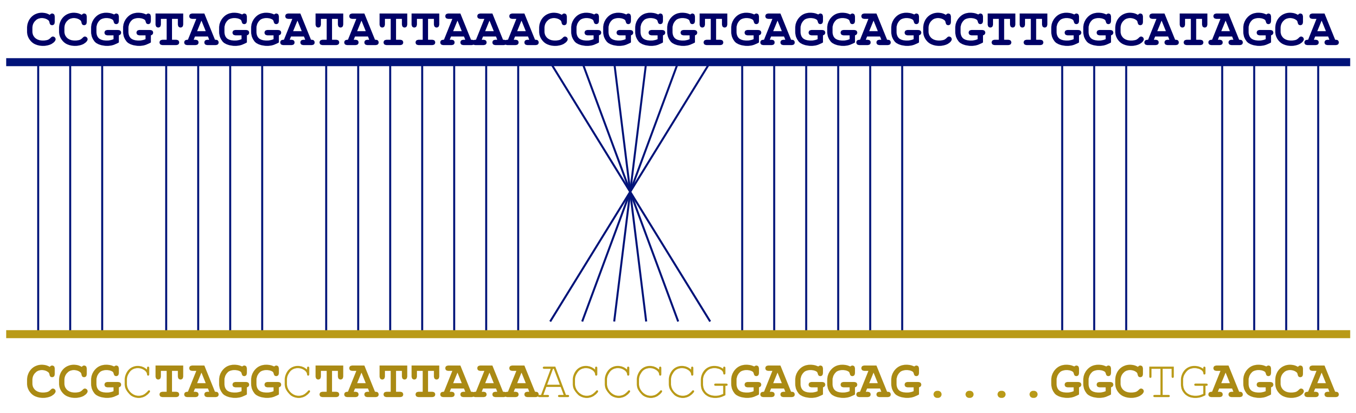
Goal of WGA

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


CCGGTAGGCTATTAAACGGGGTGAGGAGCGTTGGCATAGCA
CCGGTAGGCTATTAAACGGGGTGAGGAGCGTTGGCATAGCA

Not so fast...

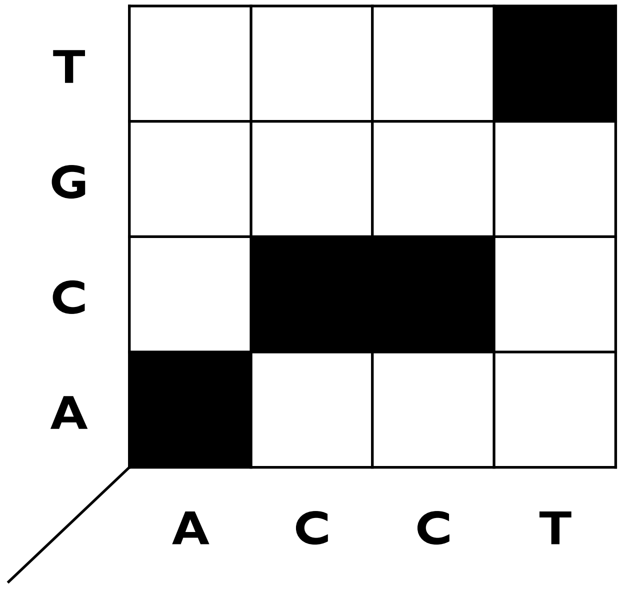
- Genome *A* may have insertions, deletions, translocations, inversions, duplications or SNPs with respect to *B* (sometimes all of the above)


CCGGTAGGATATTAAACGGGGTGAGGAGCGTTGGCATAGCA
CCGCTAGGCTATTAAAACCCCGGAGGAG....GGCTGAGCA

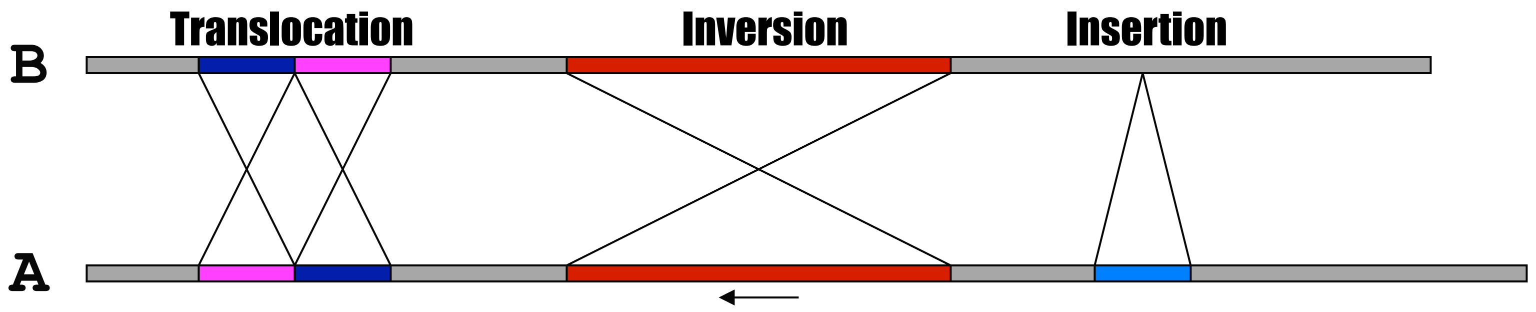
WGA visualization

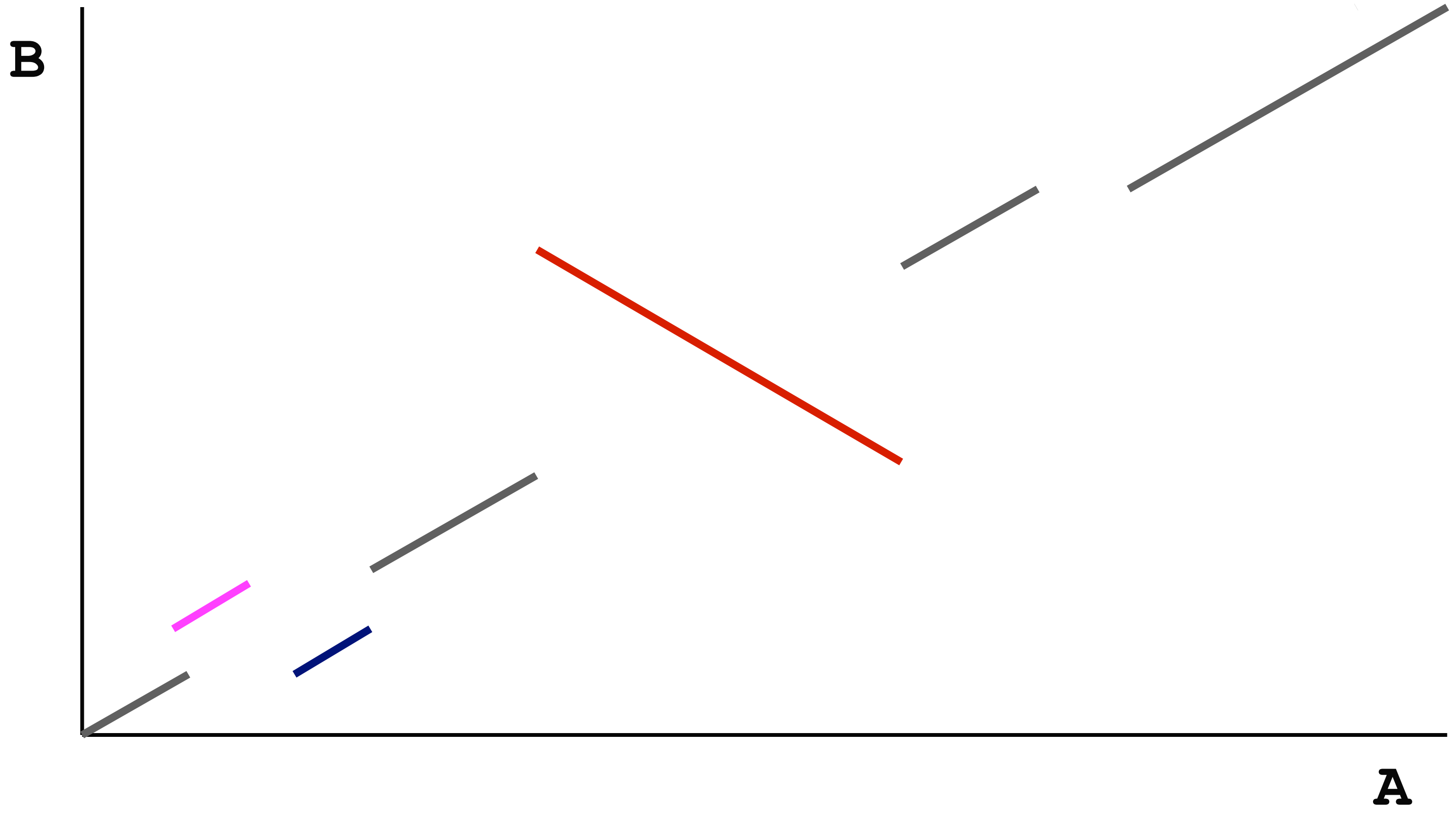
- How can we visualize *whole* genome alignments?

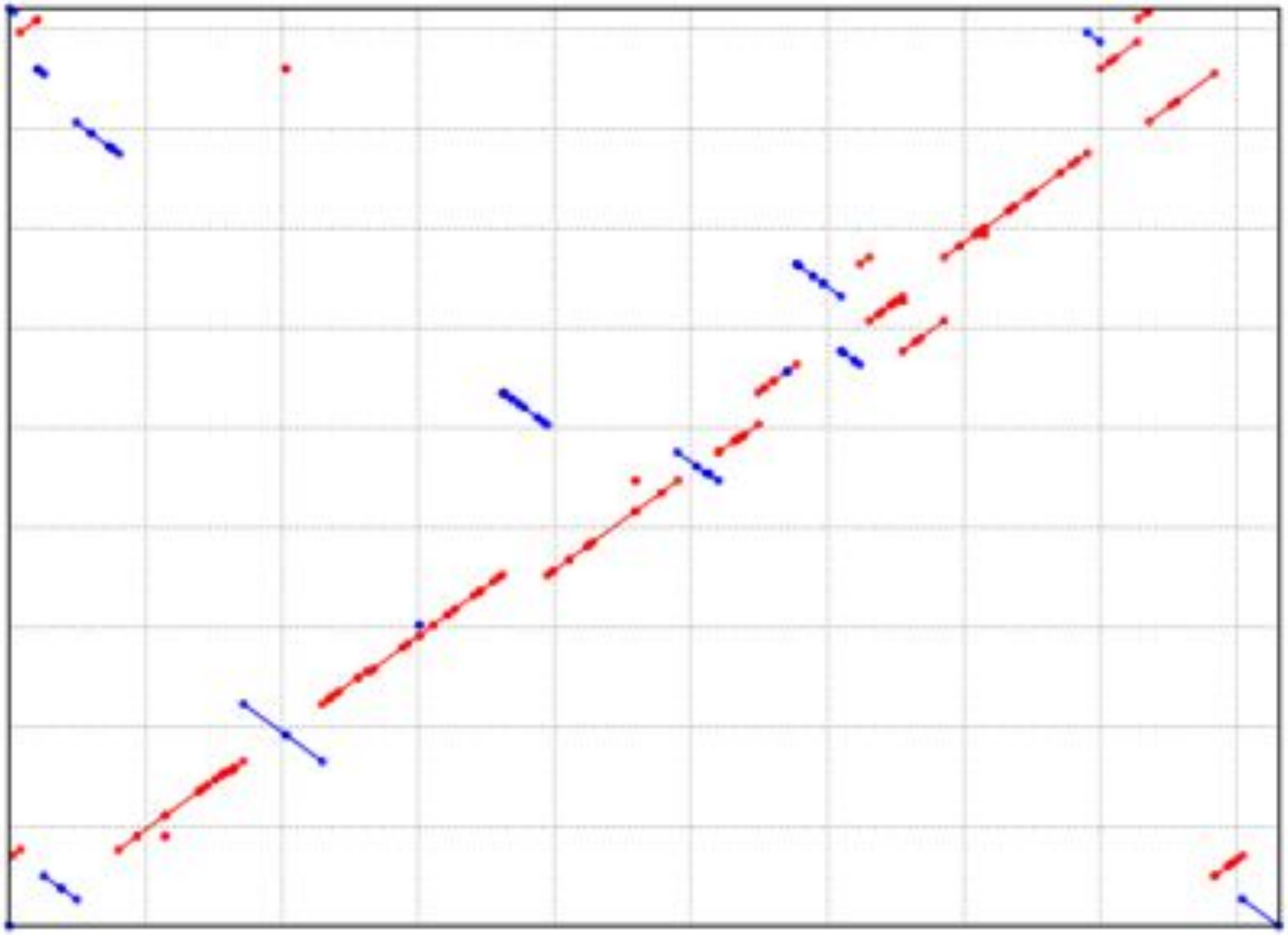
- With an alignment dot plot
  - *N* x *M* matrix
    - Let *i* = position in genome *A*
    - Let *j* = position in genome *B*
    - Fill cell *(i,j)* if A_i shows similarity to B_j



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


Translocation
Inversion
Insertion
B
A

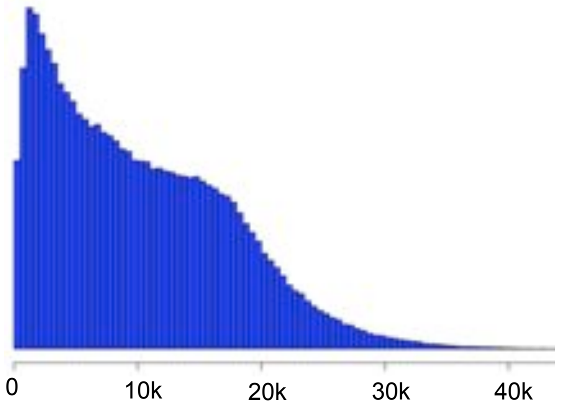

B
A



Alignment of 2 strains of *Y. pestis*
http://mummer.sourceforge.net/manual/

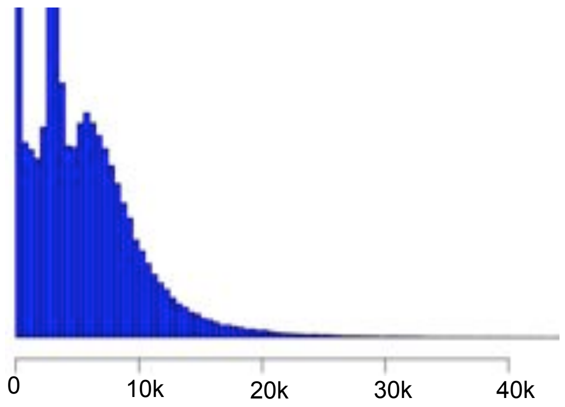
3rd Gen Long Read Sequencing

PacBio RS II



CSHL/PacBio

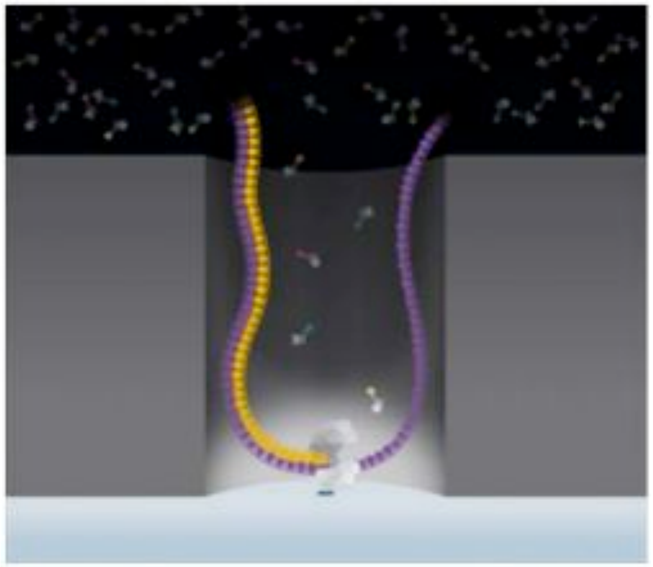
Oxford Nanopore

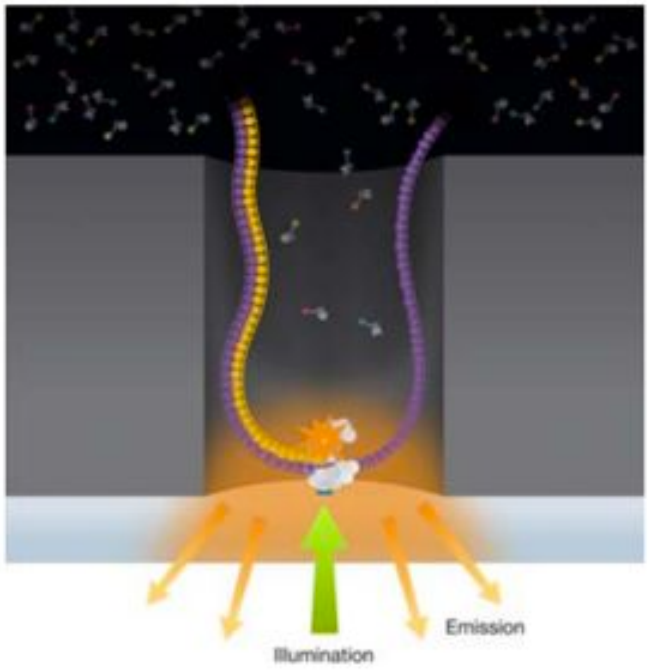


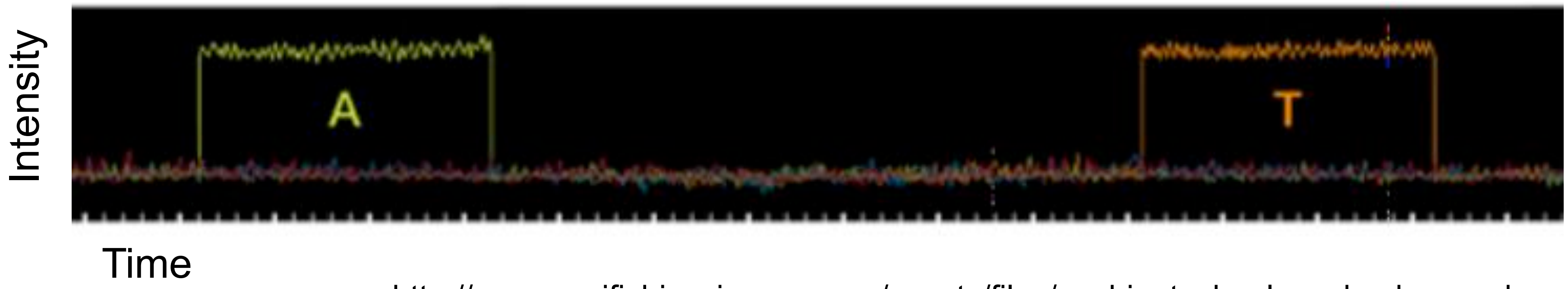
CSHL/ONT

PacBio SMRT Sequencing

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

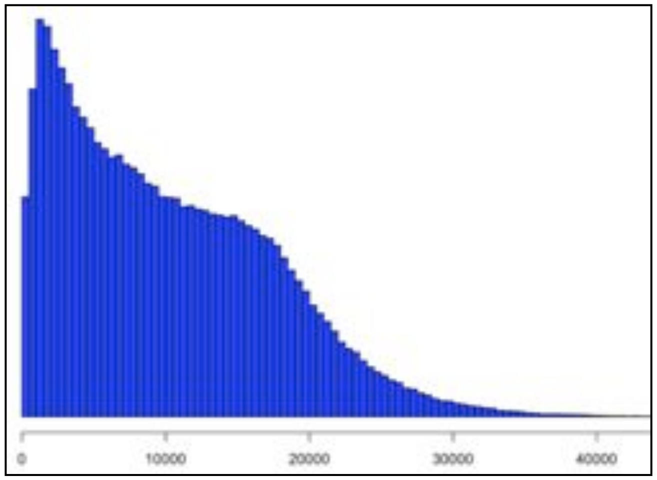






http://www.pacificbiosciences.com/assets/files/pacbio_technology_backgrounder.pdf

SMRT Sequencing Data



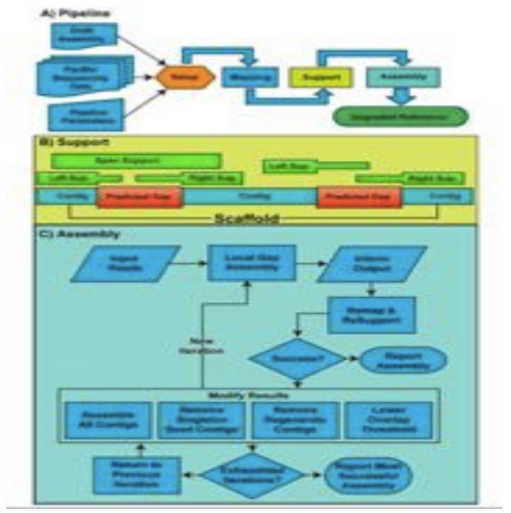
Match	83.7%
Insertions	11.5%
Deletions	3.4%
Mismatch	1.4%

```
TTGTAAGCAGTTGAAAACTATGTGTGGATTTAGAATAAAGAACATGAAAG
||||||||||||||||||||||||| ||||||| |||||||||||| |||
TTGTAAGCAGTTGAAAACTATGTGT-GATTTAG-ATAAAGAACATGGAAG

ATTATAAA-CAGTTGATCCATT-AGAAGA-AAACGCAAAAGGCGGCTAGG
| |||||| ||||||||||||| |||| | |||||| |||||| ||||||
A-TATAAATCAGTTGATCCATTAAGAA-AGAAACGC-AAAGGC-GCTAGG

CAACCTTGAATGTAATCGCACTTGAAGAACAAGATTTTATTCCGCGCCCG
| |||||| |||| ||  ||||||||||||||||||||||||||||||||
C-ACCTTG-ATGT-AT--CACTTGAAGAACAAGATTTTATTCCGCGCCCG

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

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

ACT-AATTCACAATA-AATAACACTTTTA-ACAGAATTGAT-GGAA-GTT
||| ||||||||| | ||||||||||||| ||| ||||||| |||| |||
ACTAAATTCACAA-ATAATAACACTTTTAGACAAAATTGATGGGAAGGTT

TCGGAGAGATCCAAAACAATGGGC-ATCGCCTTTGA-GTTAC-AATCAAA
|| ||||||||| ||||||| ||| |||| |||||| ||||| |||||||
TC-GAGAGATCC-AAACAAT-GGCGATCG-CTTTGACGTTACAAATCAAA

ATCCAGTGGAAAATATAATTTATGCAATCCAGGAACTTATTCACAATTAG
||||||| |||||||||  |||||| ||||| ||||||||||||||||||
ATCCAGT-GAAAATATA--TTATGC-ATCCA-GAACTTATTCACAATTAG
```

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

PacBio Assembly Algorithms

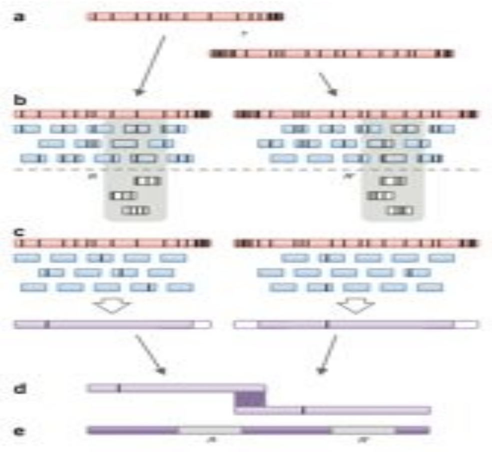
PBJelly



Gap Filling and Assembly Upgrade

English *et al* (2012)
PLOS One. 7(11): e47768

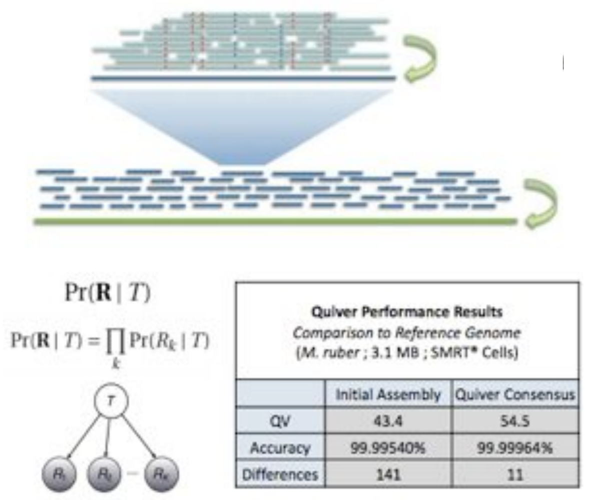
PacBioToCA & ECTools



Hybrid/PB-only Error Correction

Koren, Schatz, *et al* (2012)
Nature Biotechnology. 30:693–700

HGAP & Quiver



$\Pr(\mathbf{R} \mid T)$

$\Pr(\mathbf{R} \mid T) = \prod_k \Pr(R_k \mid T)$

Quiver Performance Results
Comparison to Reference Genome
(*M. ruber* ; 3.1 MB ; SMRT® Cells)

	Initial Assembly	Quiver Consensus
QV	43.4	54.5
Accuracy	99.99540%	99.99964%
Differences	141	11

PB-only Correction & Polishing

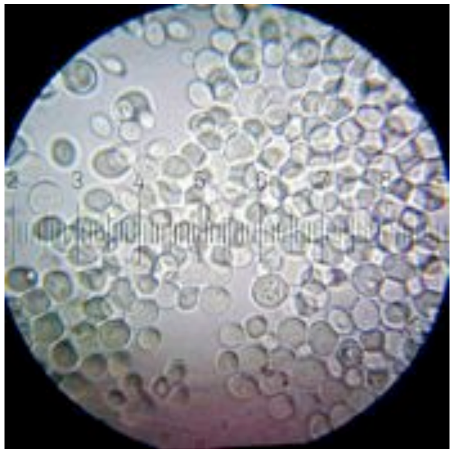
Chin *et al* (2013)
Nature Methods. 10:563–569

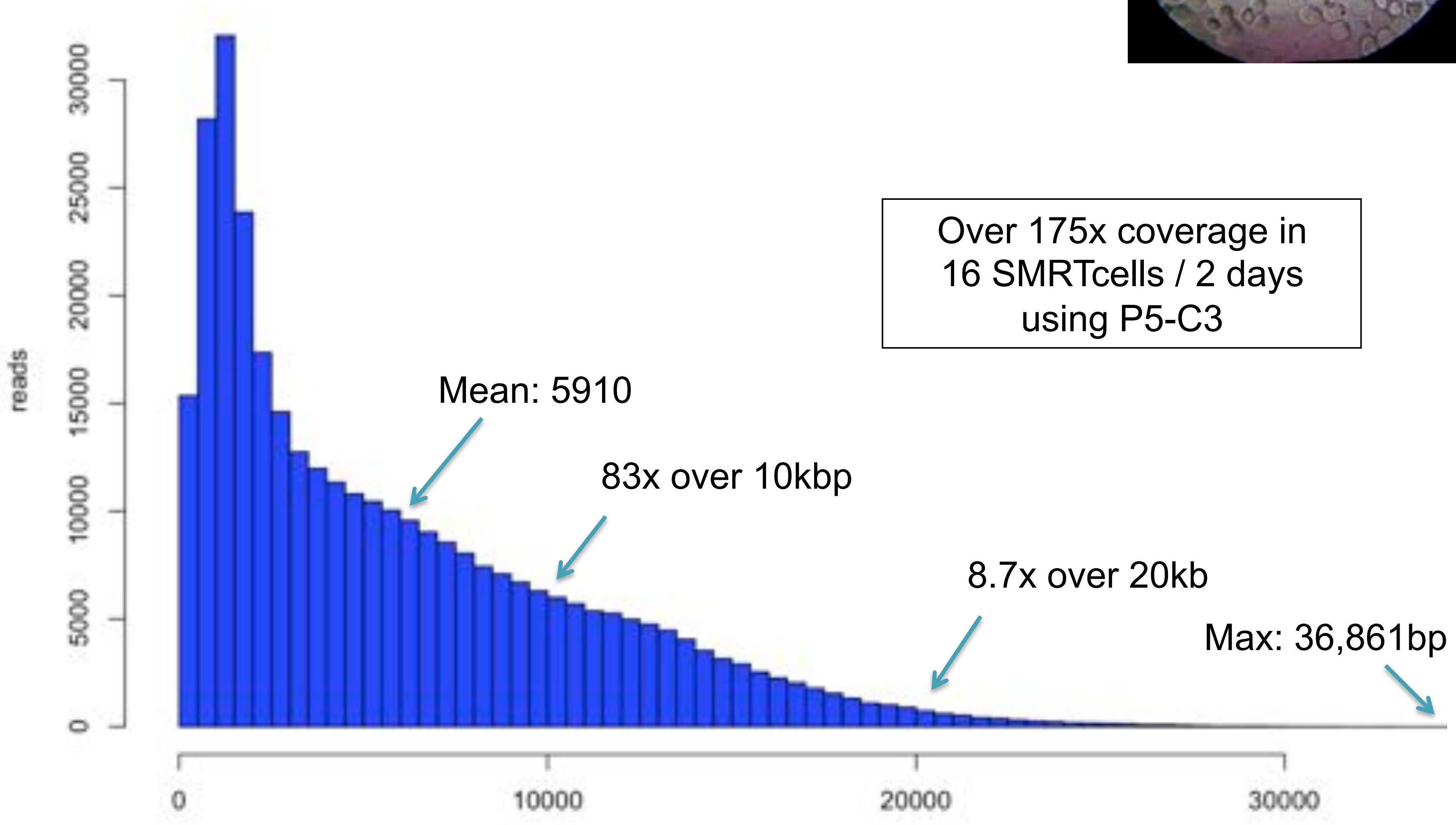
< 5x PacBio Coverage > 50x

S. cerevisiae W303

PacBio RS II sequencing at CSHL in the McCombie Lab

- Size selection using an 7 Kb elution window on a BluePippin™ device from Sage Science





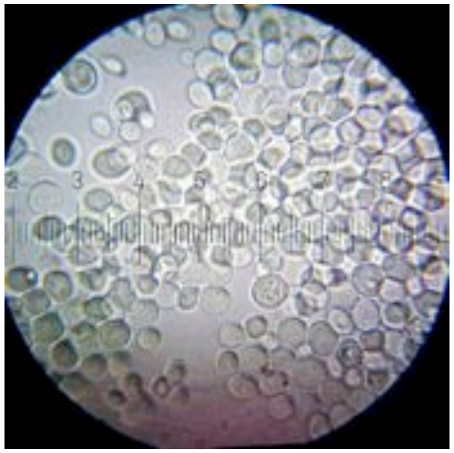
S. cerevisiae W303

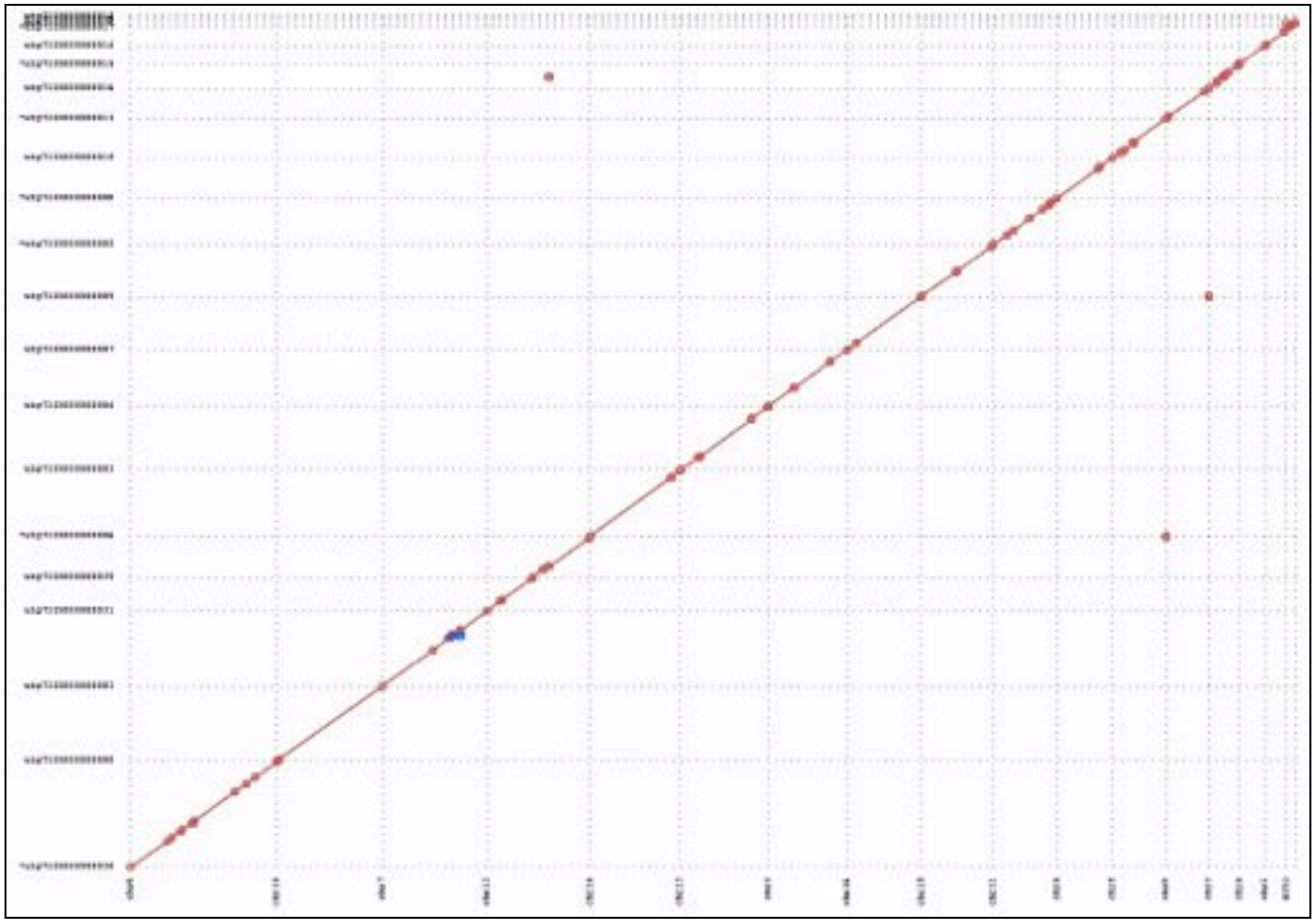
S288C Reference sequence

- 12.1Mbp; 16 chromo + mitochondria; N50: 924kbp

PacBio assembly using HGAP + Celera Assembler

- 12.4Mbp; 21 non-redundant contigs; N50: 811kbp; >99.8% id





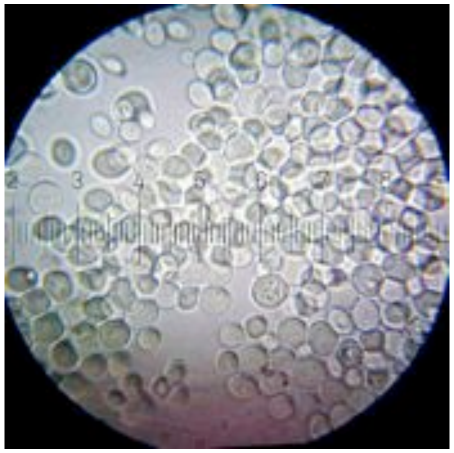
S. cerevisiae W303

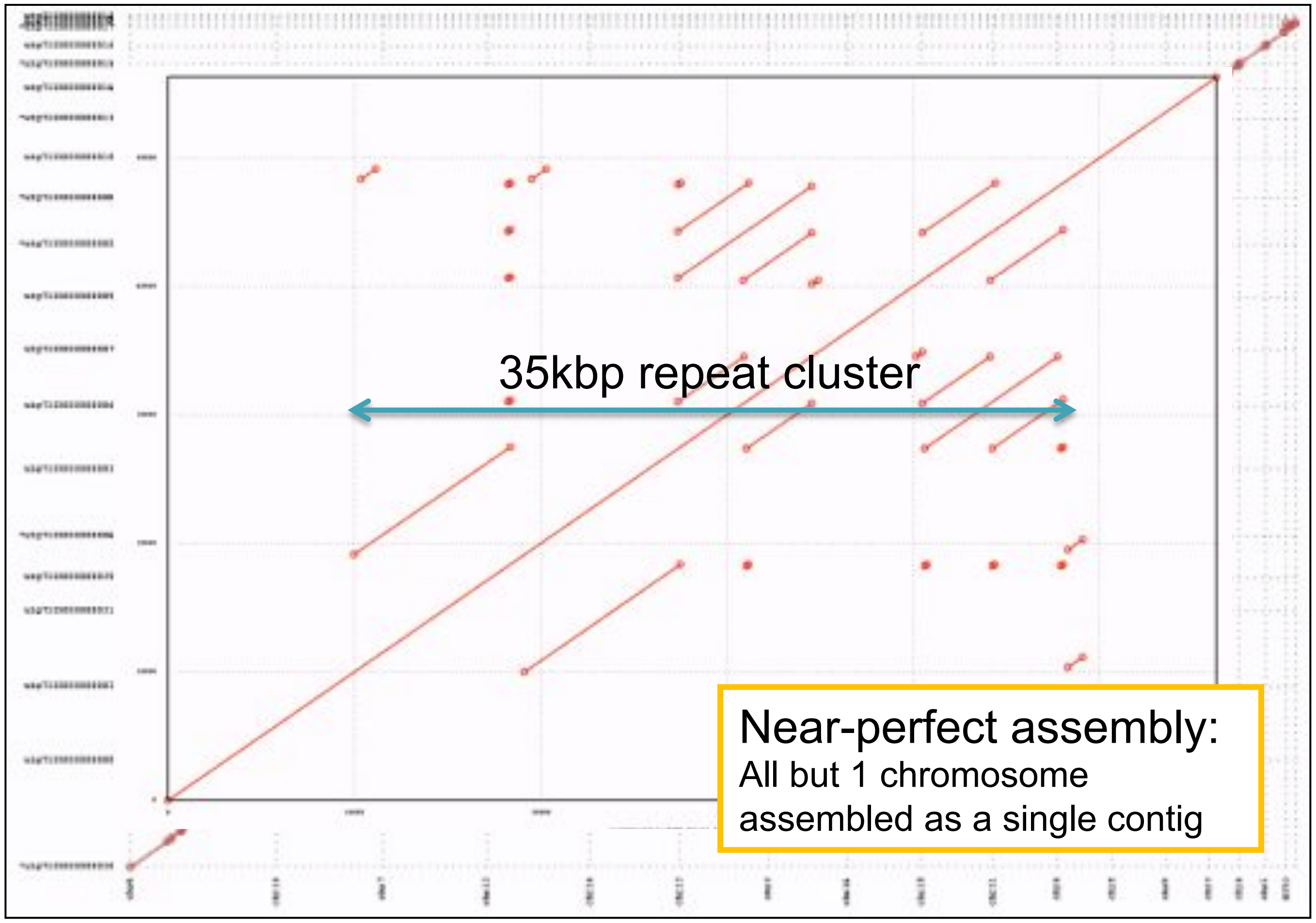
S288C Reference sequence

- 12.1Mbp; 16 chromo + mitochondria; N50: 924kbp

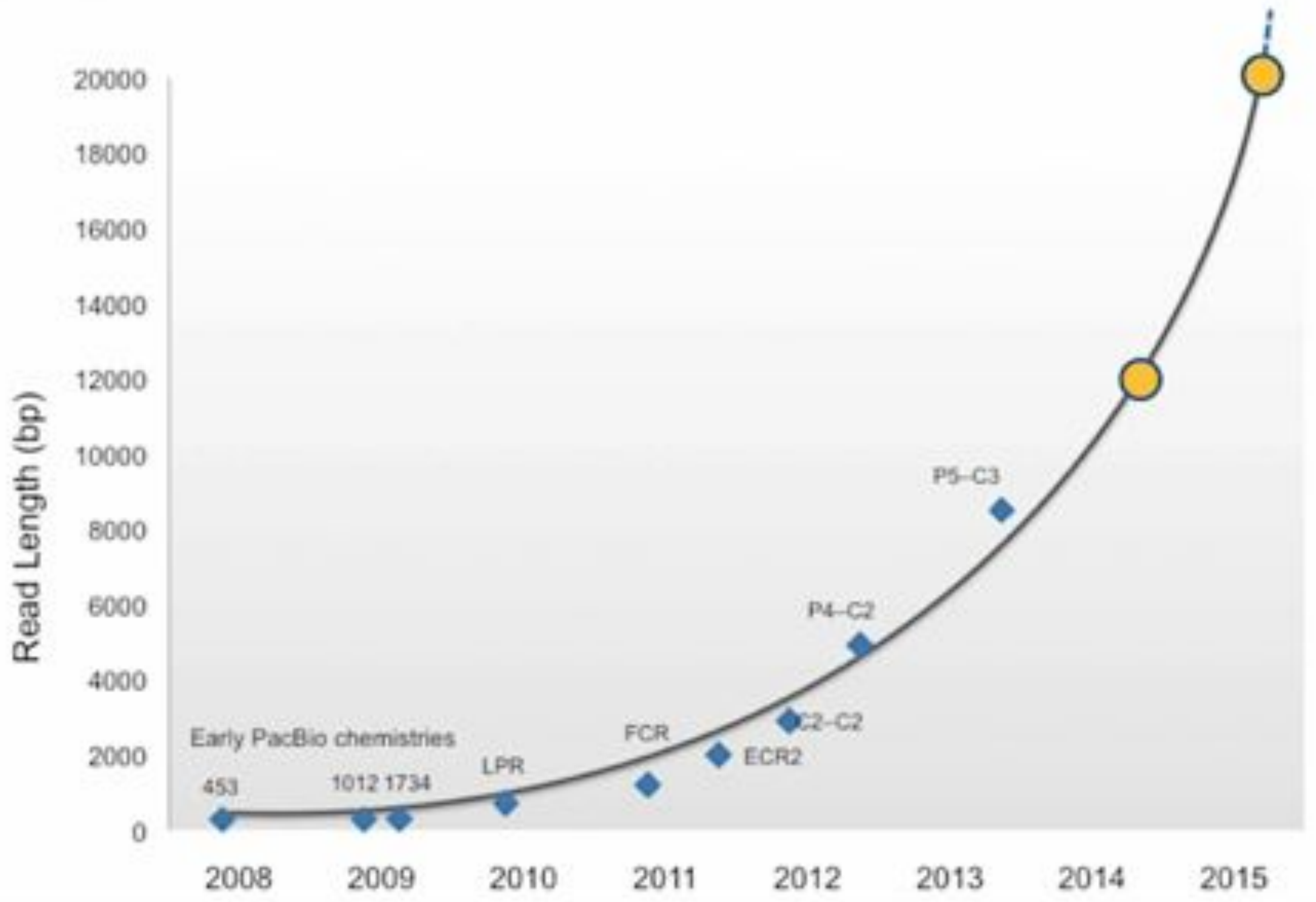
PacBio assembly using HGAP + Celera Assembler

- 12.4Mbp; 21 non-redundant contigs; N50: 811kbp; >99.8% id



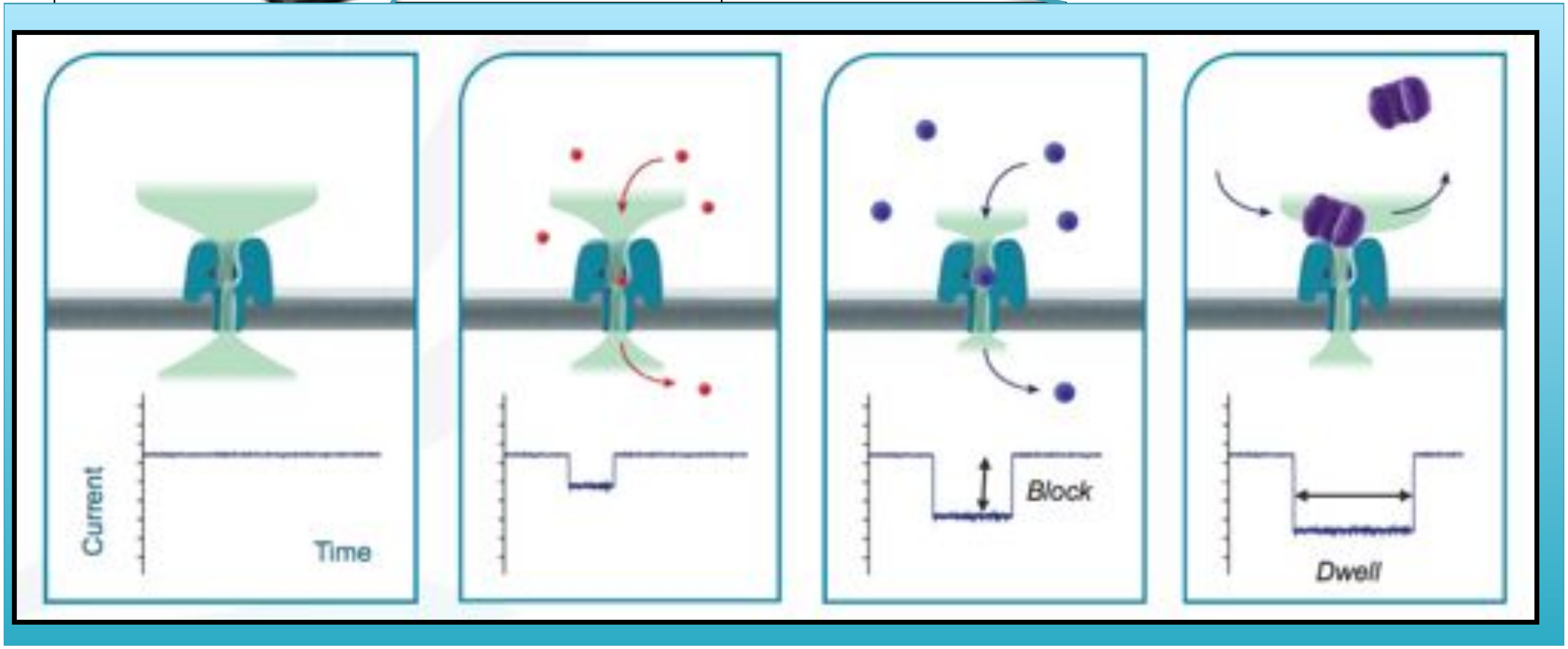


PacBio® Advances in Read Length

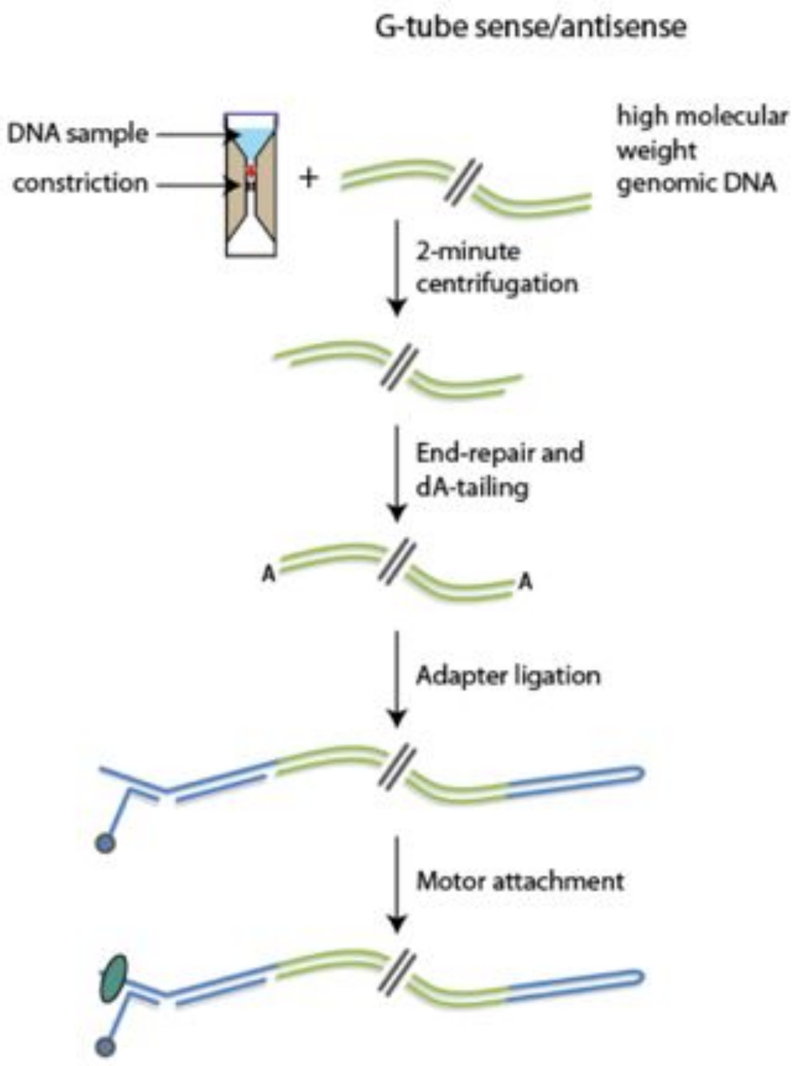


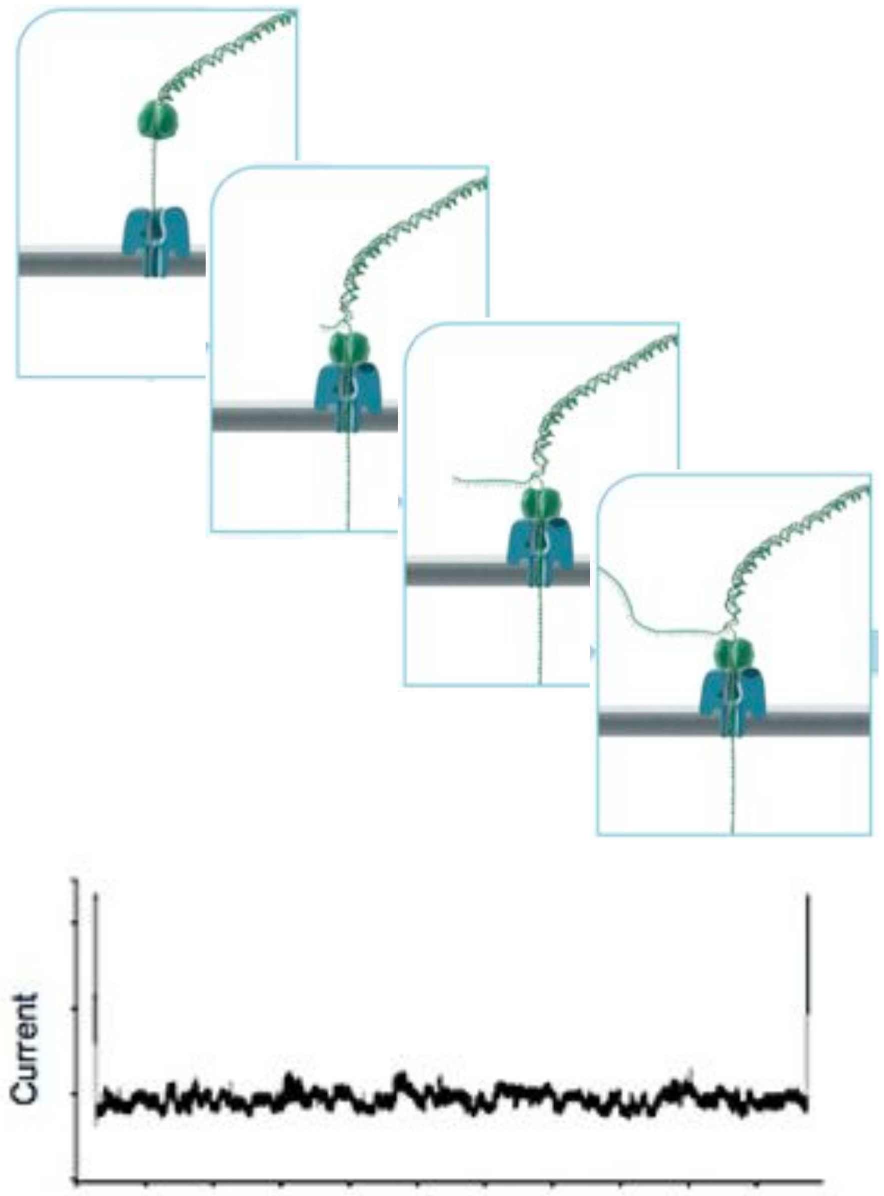
Oxford Nanopore MinION

- Thumb drive sized sequencer powered over USB
- Capacity for 512 reads at once
- Senses DNA by measuring changes to ion flow

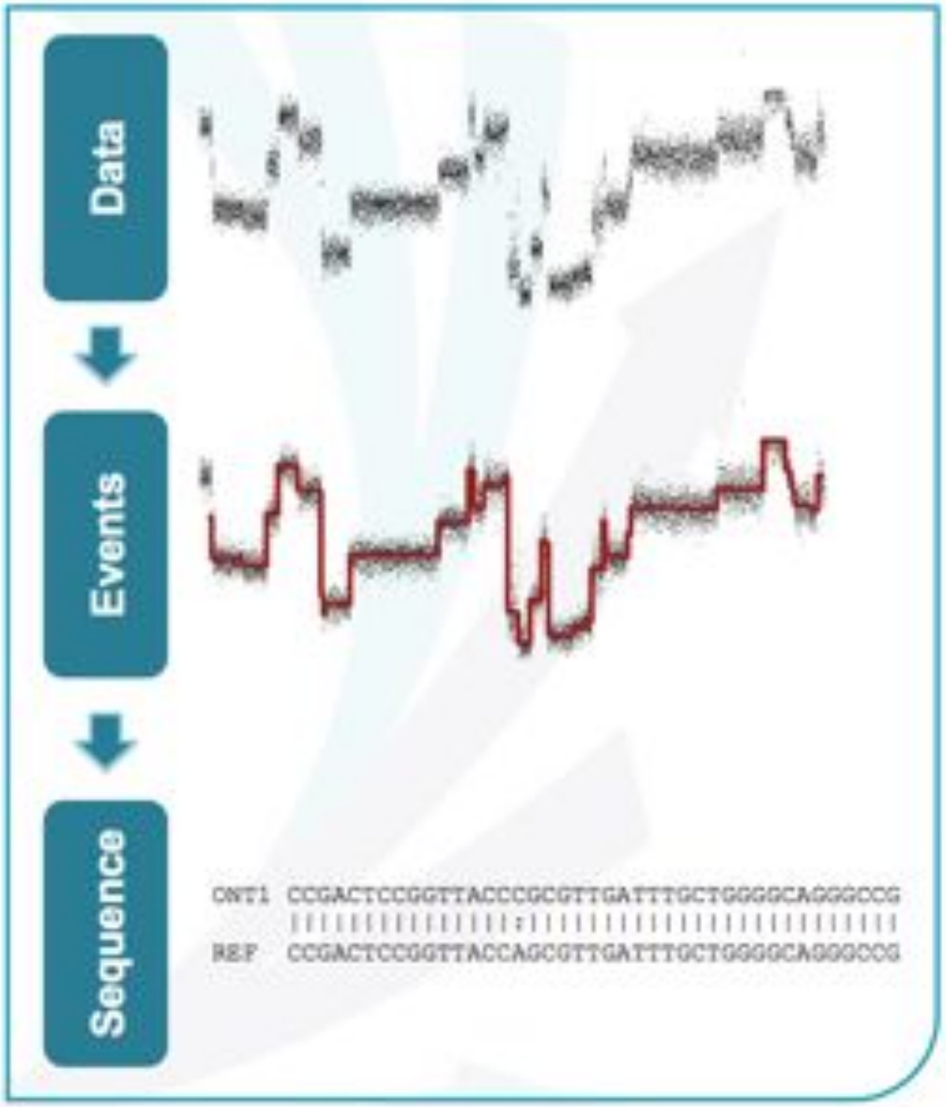


Nanopore Sequencing



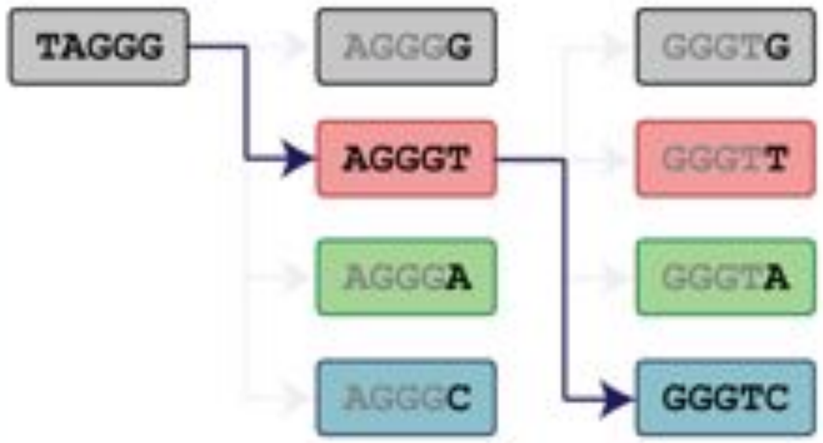


Nanopore Basecalling



```
ONT1 CCGACTCCGGTTACCCGCGTTGATTTGCTGGGGCAGGGCCG
     |||||||||||||||:|||||||||||||||||||||||||
REF  CCGACTCCGGTTACCAGCGTTGATTTGCTGGGGCAGGGCCG
```

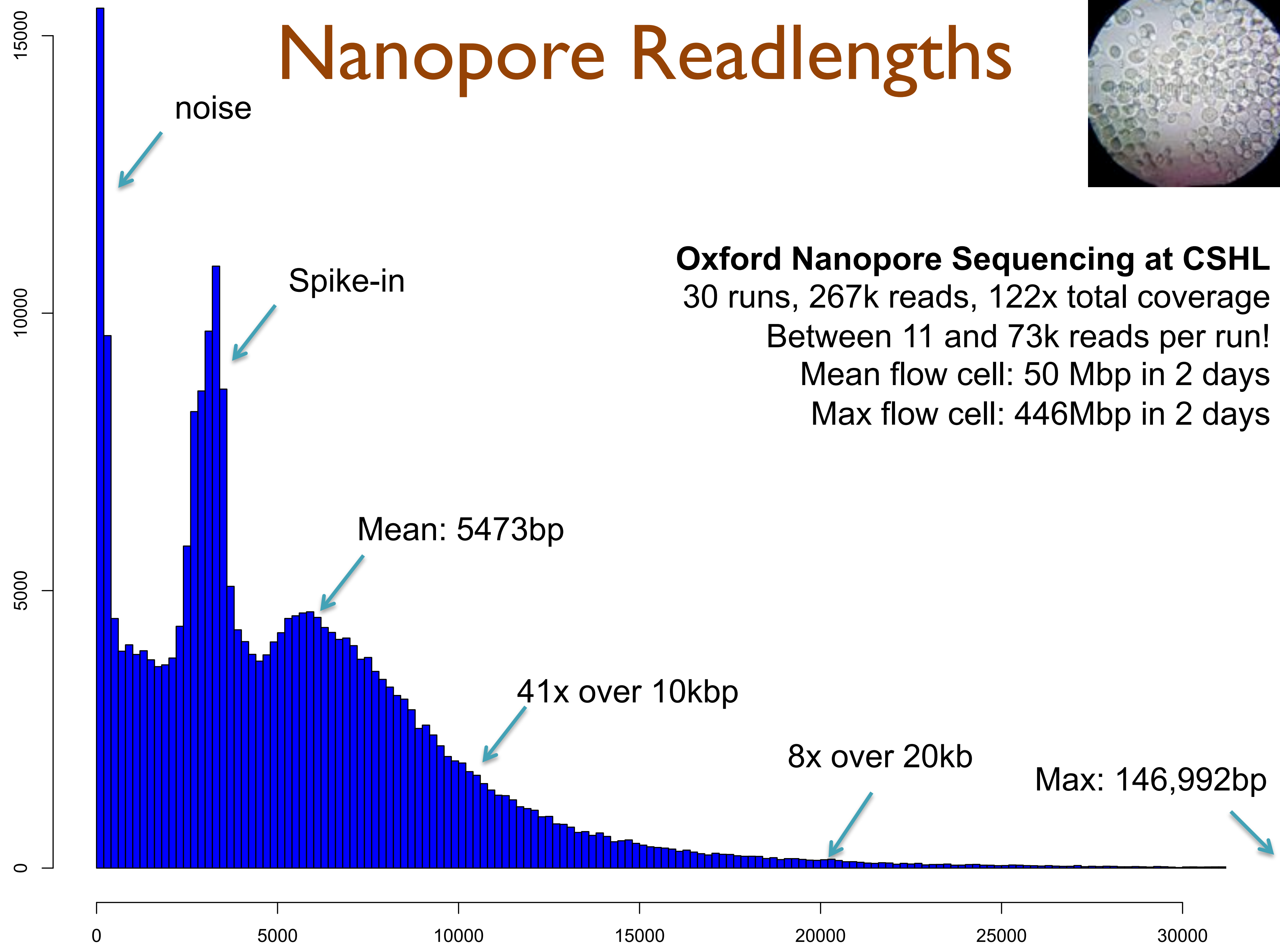
- Hidden Markov model
- Only four options per transition
- Pore type = distinct kmer length

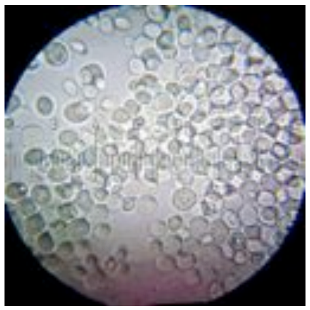


- Form probabilistic path through measured states currents and transitions
  - e.g. Viterbi algorithm

Basecalling currently performed at Amazon with frequent updates to algorithm

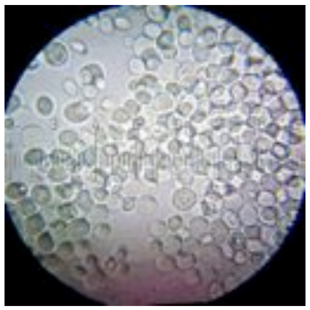
Nanopore Readlengths

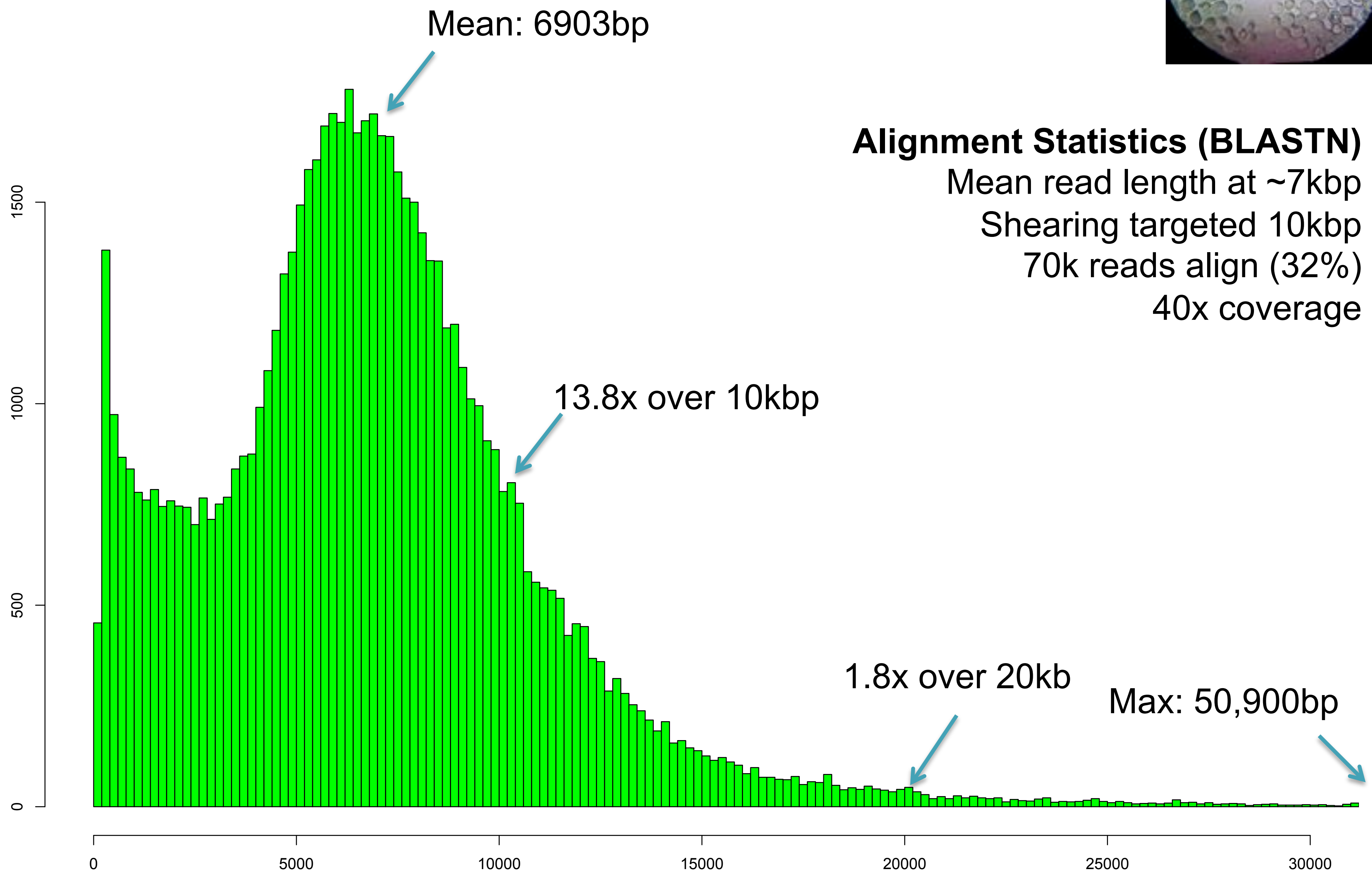




Oxford Nanopore Sequencing at CSHL
30 runs, 267k reads, 122x total coverage
Between 11 and 73k reads per run!
Mean flow cell: 50 Mbp in 2 days
Max flow cell: 446Mbp in 2 days

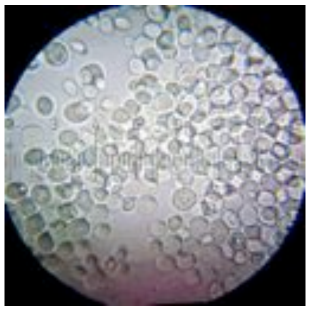
Nanopore Alignments



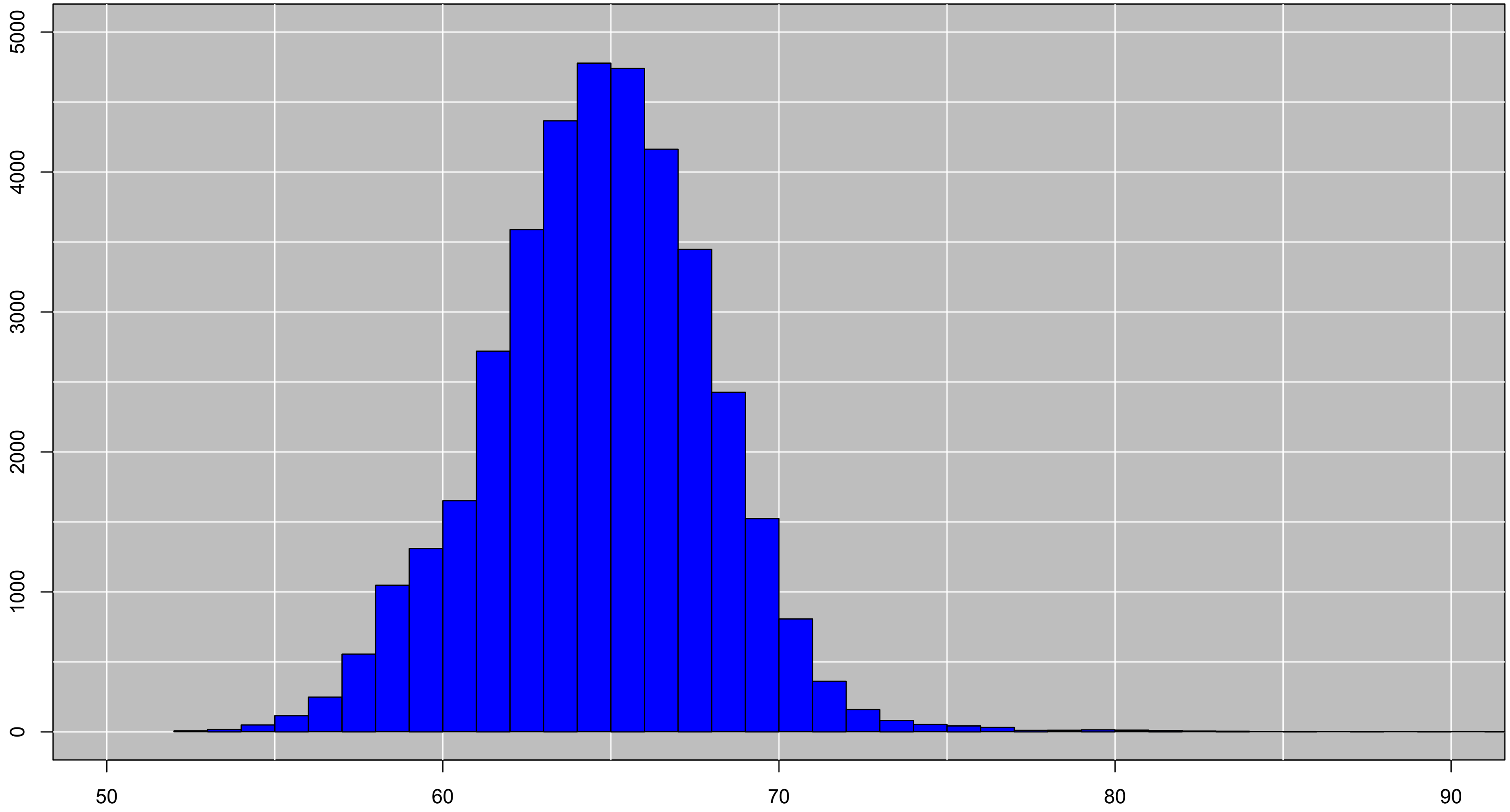


Alignment Statistics (BLASTN)
Mean read length at ~7kbp
Shearing targeted 10kbp
70k reads align (32%)
40x coverage

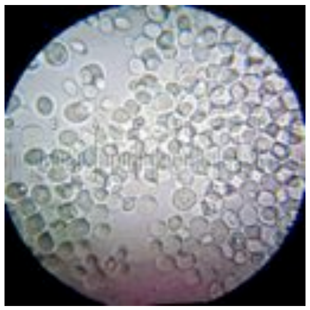
Nanopore Accuracy



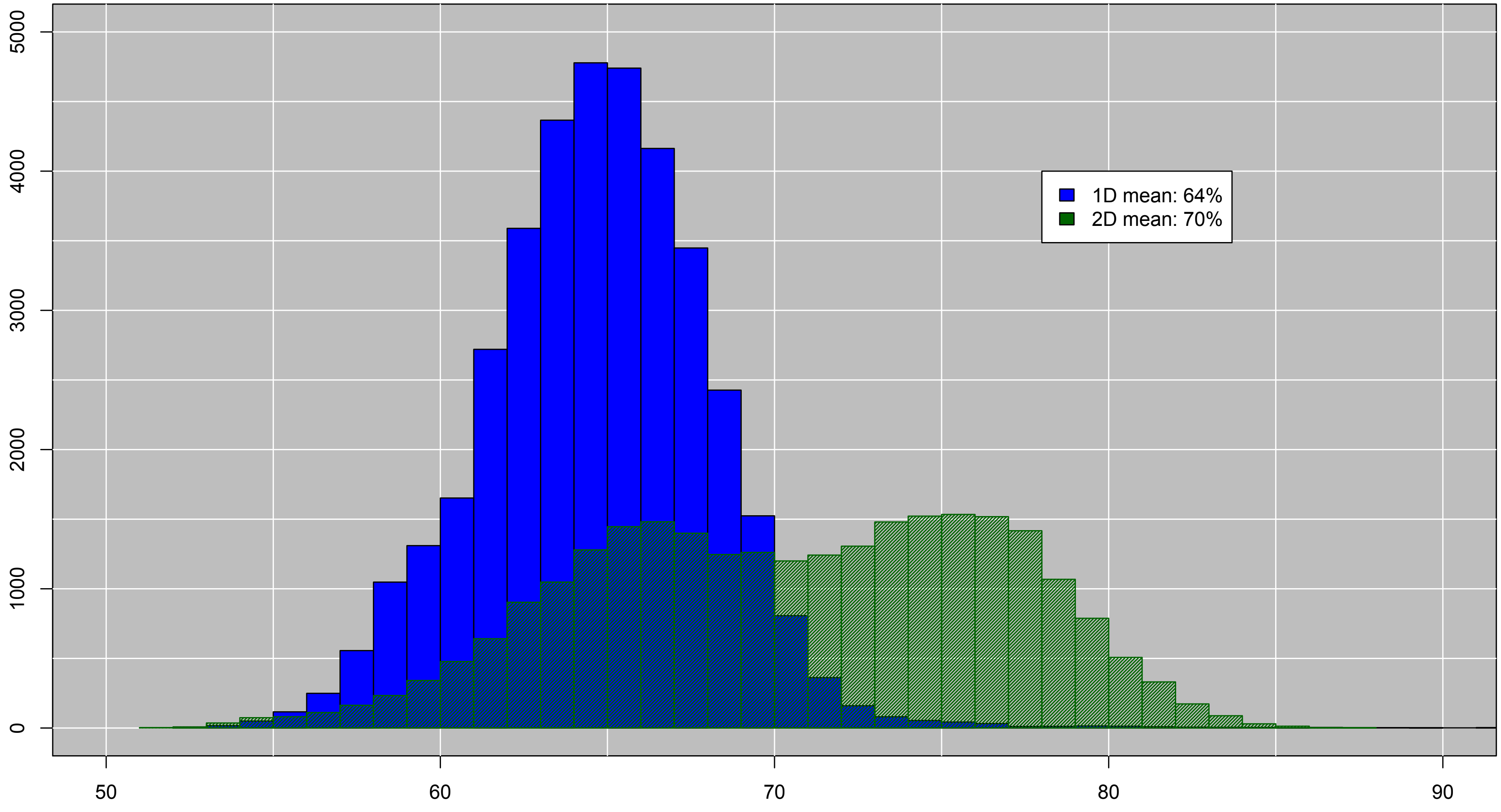
Alignment Quality (BLASTN)
Of reads that align, average ~64% identity



Nanopore Accuracy



Alignment Quality (BLASTN)
Of reads that align, average ~64% identity
“2D base-calling” improves to ~70% identity



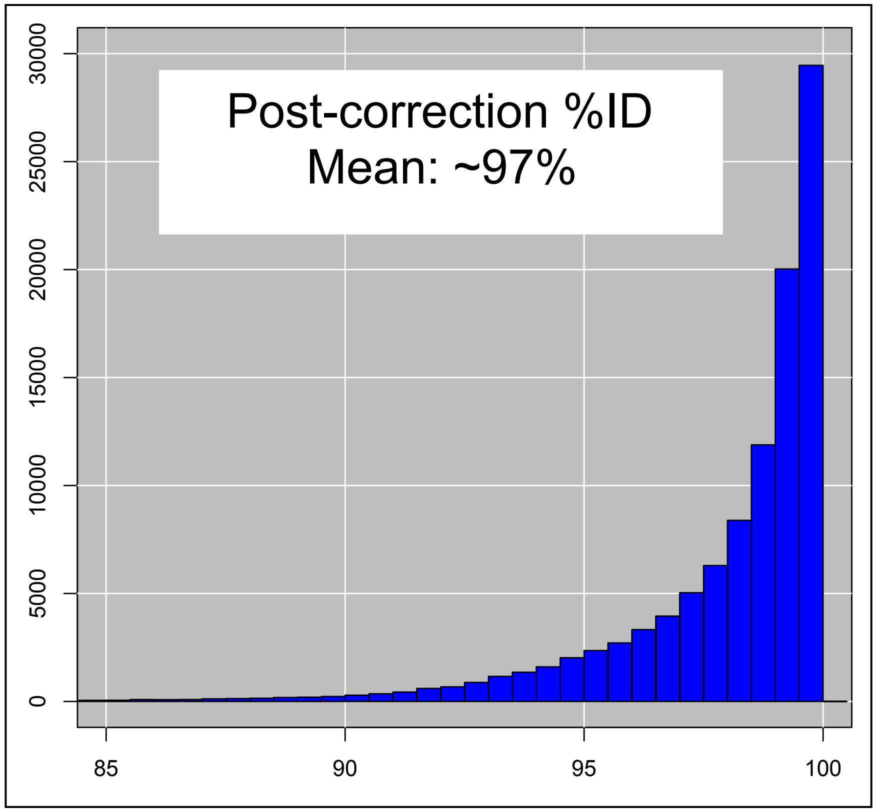
NanoCorr: Nanopore-Illumina Hybrid Error Correction

https://github.com/jgurtowski/nanocorr

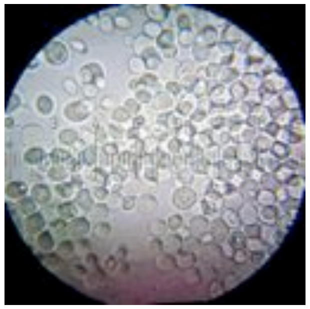
1. BLAST Miseq reads to all raw Oxford Nanopore reads

2. Select non-repetitive alignments
   - First pass scans to remove "contained" alignments
   - Second pass uses Dynamic Programming (LIS) to select set of high-identity alignments with minimal overlaps

3. Compute consensus of each Oxford Nanopore read
   - Currently using Pacbio's pbdagcon



Long Read Assembly

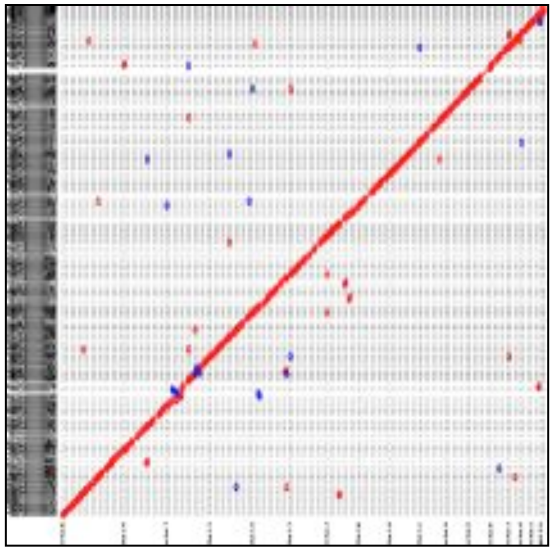


S288C Reference sequence

- 12.1Mbp; 16 chromo + mitochondria
- Chromosome N50: 924kbp

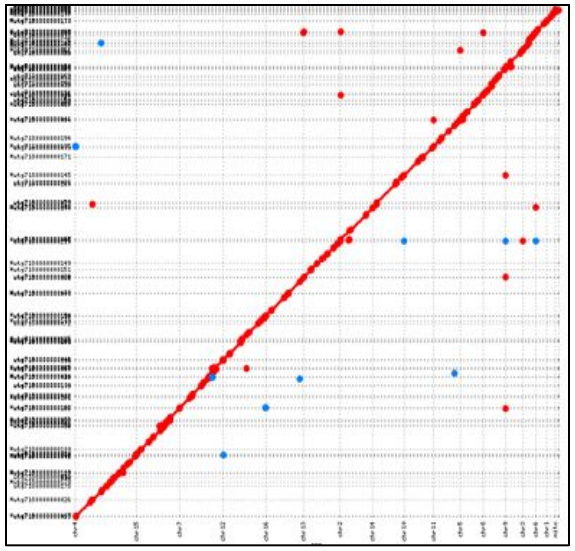
Illumina MiSeq
30x, 300bp PE (Flashed)

Celera Assembler
- 6953 non-redundant contigs
- N50: 59kb >99.9% id



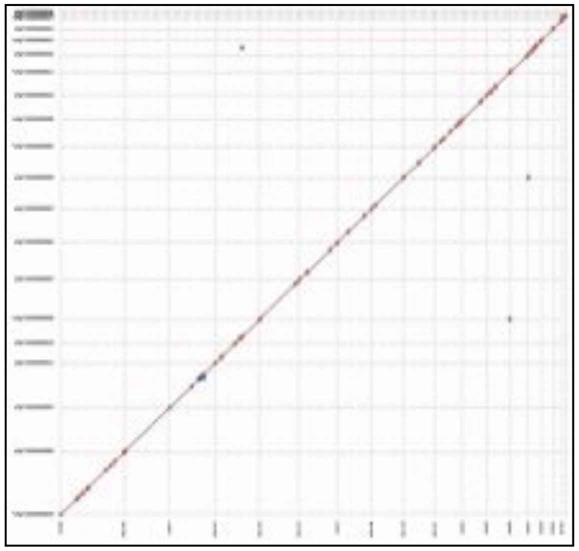
Oxford Nanopore
30x corrected reads > 6kb

NanoCorr + Celera Assembler
- 234 non-redundant contigs
- N50: 362kbp >99.78% id

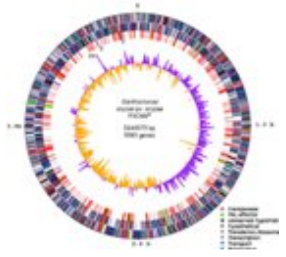


Pacific Biosciences
25x corrected reads > 10kb

HGAP + Celera Assembler
- 21 non-redundant contigs
- N50: 811kb >99.8% id



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

Thank You

http://schatzlab.cshl.edu/teaching/
@mike_schatz