

Graphs and Genomes

Michael Schatz

Bioinformatics Lecture 3
Quantitative Biology 2012



Dynamic Programming Matrix

Compute the optimal alignment of ABC...XY..N and DEF...UV...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 | I | 2 | 3 | | X | X+I | | N |
| D | I | | | | | | | | |
| E | 2 | | | | | | | | |
| F | 3 | | | | | | | | |
| ... | | | | | | | | | |
| U | U | | | | | | | | |
| V | U+I | | | | | | | | |
| ... | | | | | | | | | |
| 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

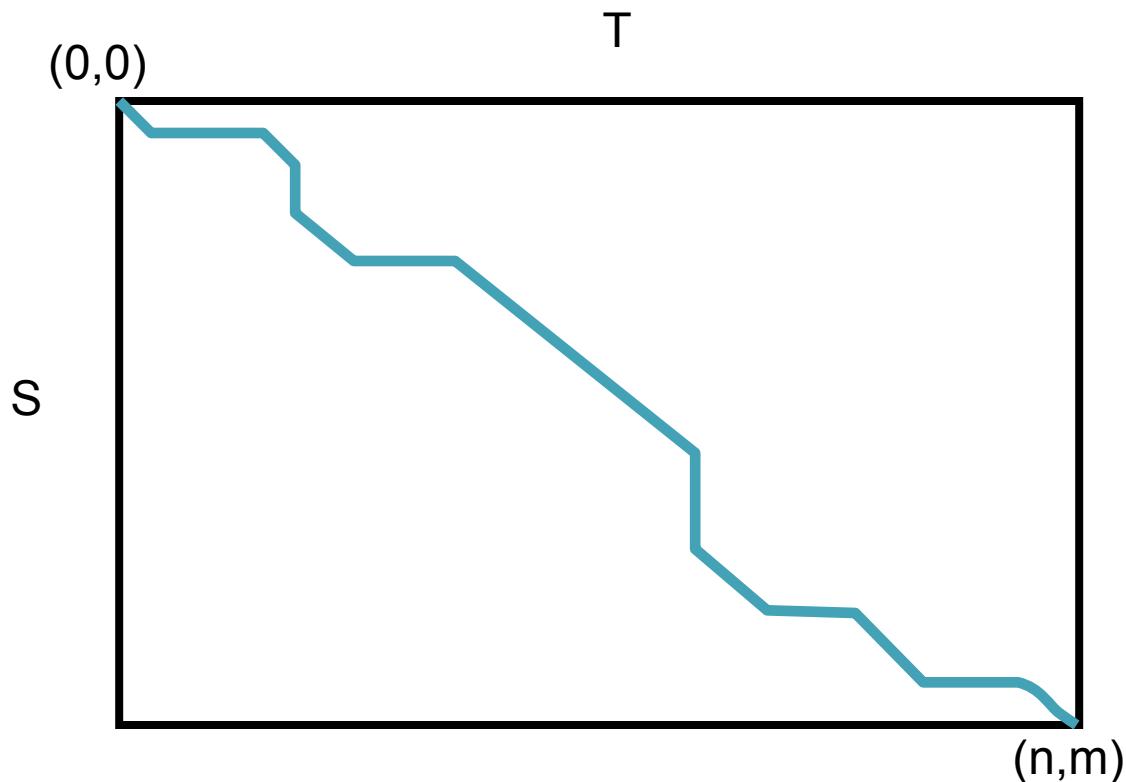
Compute the optimal alignment of “ABC...XY..N” and “DEF...UV...M”

| | 0 | A | B | C | ... | X | Y | ... | N |
|----------|----------|----------|----------|----------|-----|----------|----------|-----|----------|
| 0 | 0 | I | 2 | 3 | | X | X+I | | N |
| D | I | | | | | | | | |
| E | 2 | | | | | | | | |
| F | 3 | | | | | | | | |
| ... | | | | | | | | | |
| U | U | | | | | γ | α | | |
| V | U+I | | | | | β | Ω | | |
| ... | | | | | | | | | |
| M | M | | | | | | | | |

$$\Omega = \min \left\{ \begin{array}{lll} \text{“Up”} + I & \alpha+1 & \text{Up} \\ \text{“Left+”} + I & \beta+1 & \text{ABC...XY-} \\ \text{“Diagonal”} + 0/I & \gamma+1 & \text{DEF....UV} \end{array} \right. \quad \begin{array}{l} \text{Left} \\ \text{ABC.....XY} \\ \text{DEF....UV-} \end{array} \quad \begin{array}{l} \text{Diagonal} \\ \text{ABC...XY} \\ \text{DEF...UV} \end{array}$$

α β γ

Global Alignment Schematic



Evaluate all $N \times M$ cells in $O(N \times M)$ time.

Value in cell $D[n,m]$ is the edit distance.

$$D[AGCACACA, ACACACTA] = 2$$

AGCACAC-A

| * | | | | *

A-CACACTA

Biological Networks

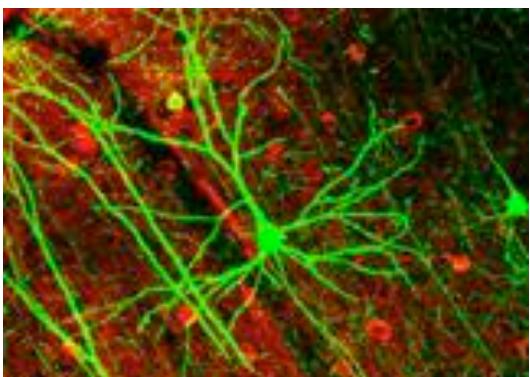
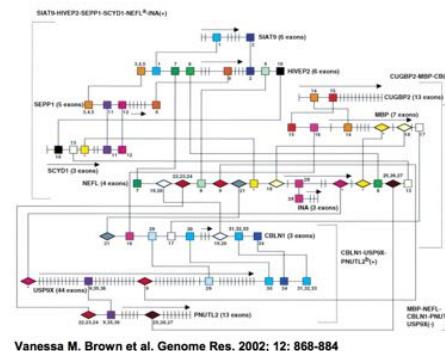
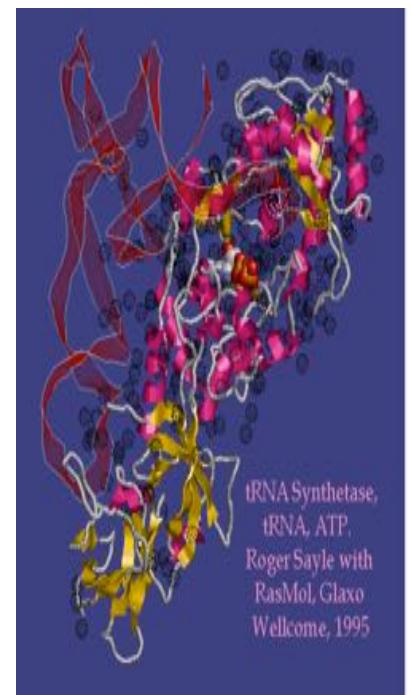
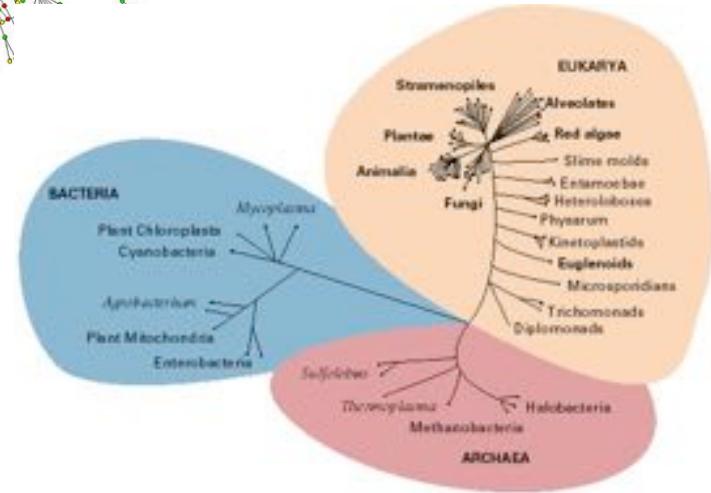
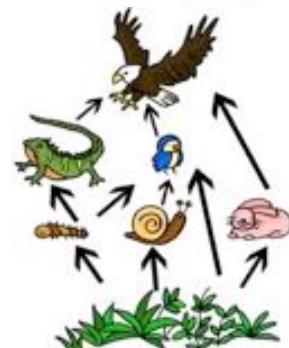
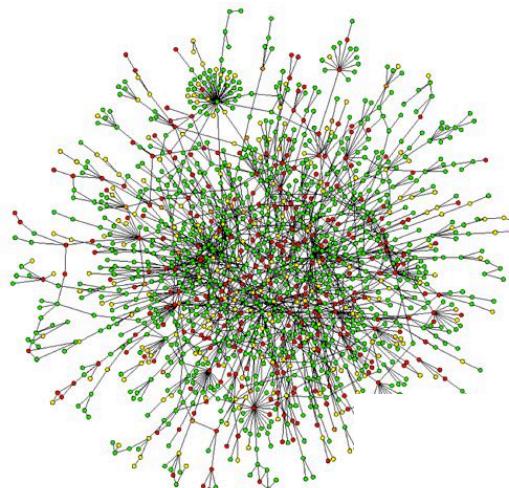
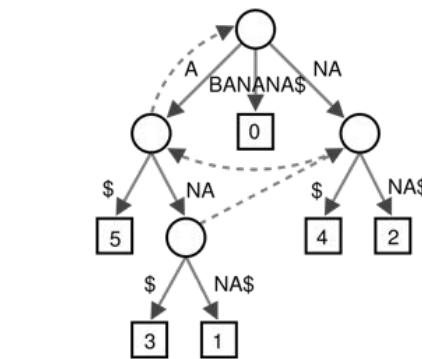
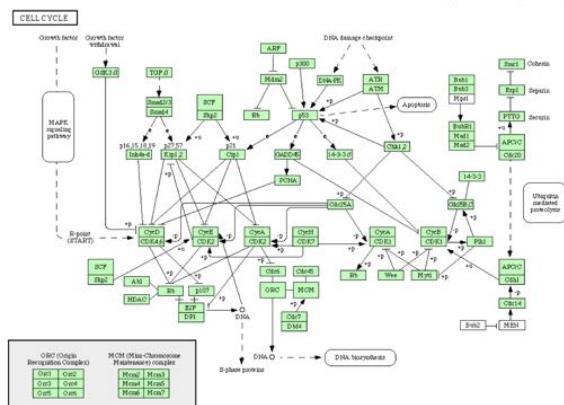


Figure 5 Putative regulatory elements shared between groups of correlated and anticorrelated genes

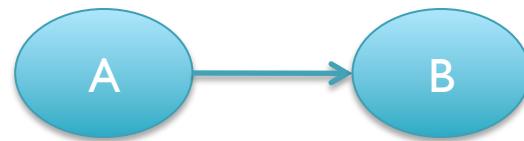


Vanessa M. Brown et al. Genome Res. 2002; 12: 868-884

Cold Spring Harbor Laboratory Press



Graphs

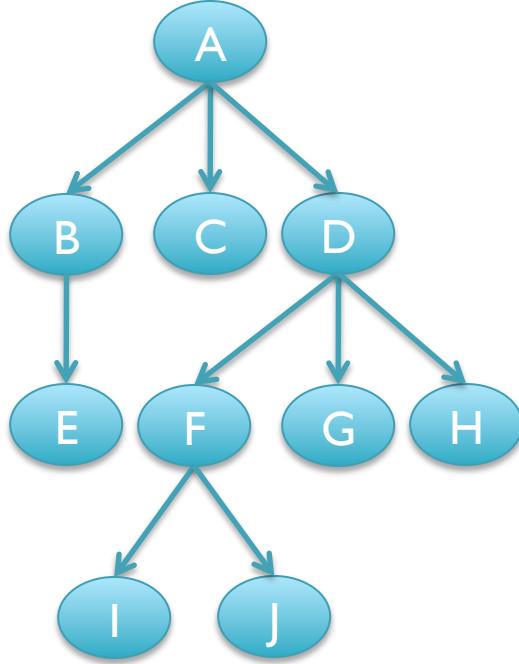


- **Nodes**
 - People, Proteins, Genes, Neurons, Sequences, Numbers, ...
- **Edges**
 - A is connected to B
 - A is related to B
 - A regulates B
 - A precedes B
 - A interacts with B
 - A activates B
 - ...

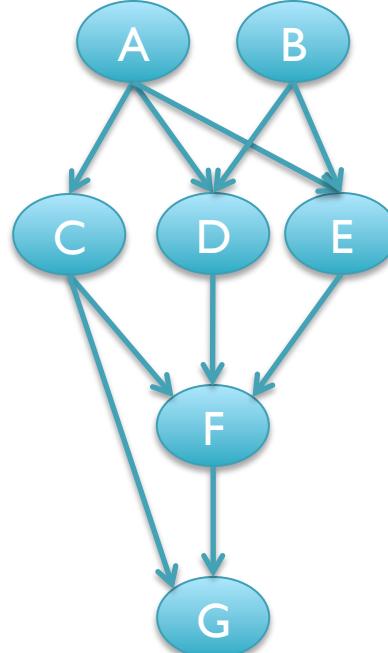
Graph Types



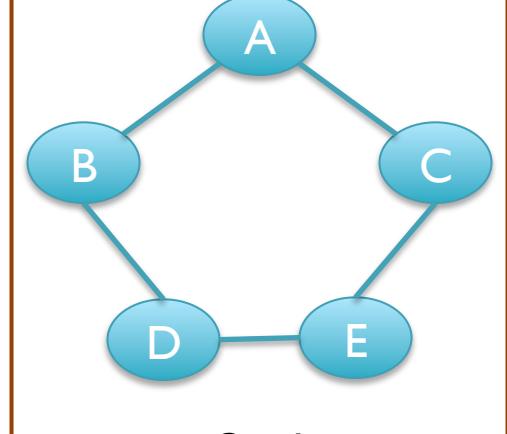
List



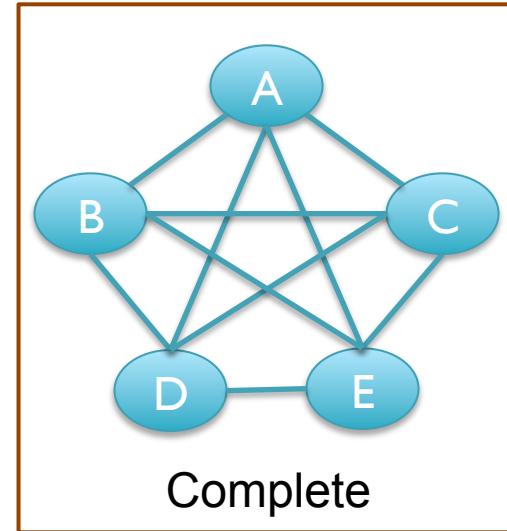
Tree



Directed
Acyclic
Graph

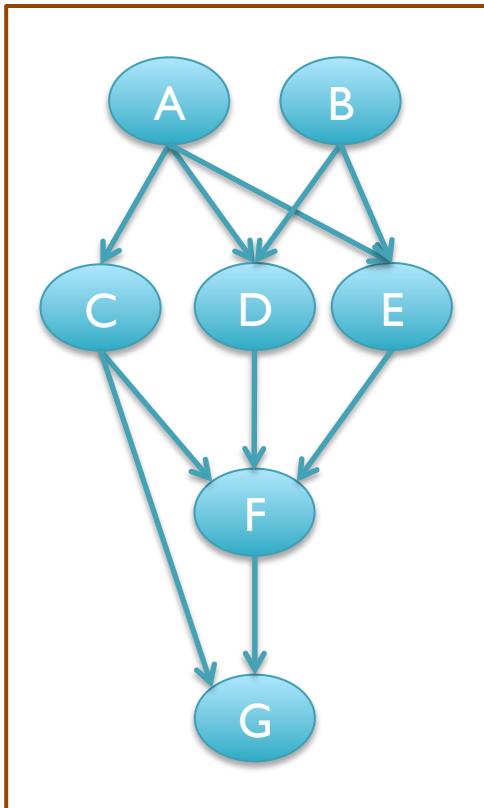


Cycle



Complete

Representing Graphs



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

| | A | B | C | D | E | F | G |
|---|---|---|---|---|---|---|---|
| A | | | | | | | |
| B | | | | | | | |
| C | | | | | | | |
| D | | | | | | | |
| E | | | | | | | |
| F | | | | | | | |
| G | | | | | | | |

Adjacency List
Good for sparse graphs
Compact storage: 4 bytes/edge

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

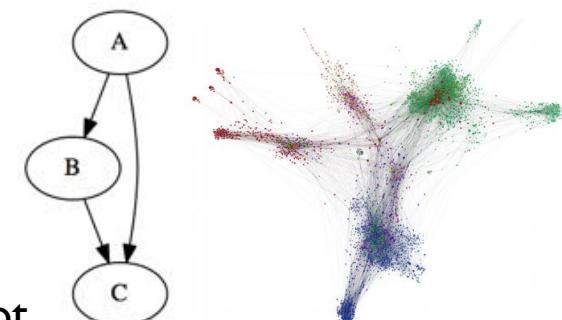
Matlab: <http://www.mathworks.com/>

Graphviz: <http://www.graphviz.org/>

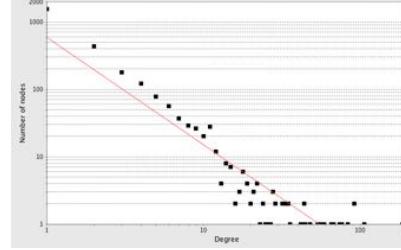
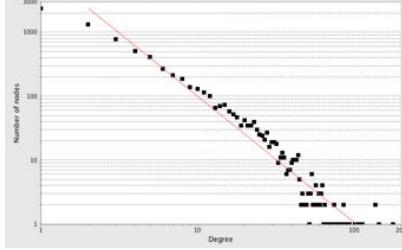
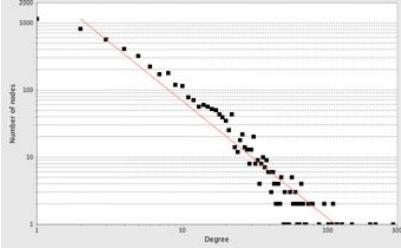
Gephi: <https://gephi.org/>

Cytoscape: <http://www.cytoscape.org/>

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



Network Characteristics

| | <i>C. elegans</i> | <i>D. melanogaster</i> | <i>S. cerevisiae</i> |
|----------------------|--|---|---|
| # 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

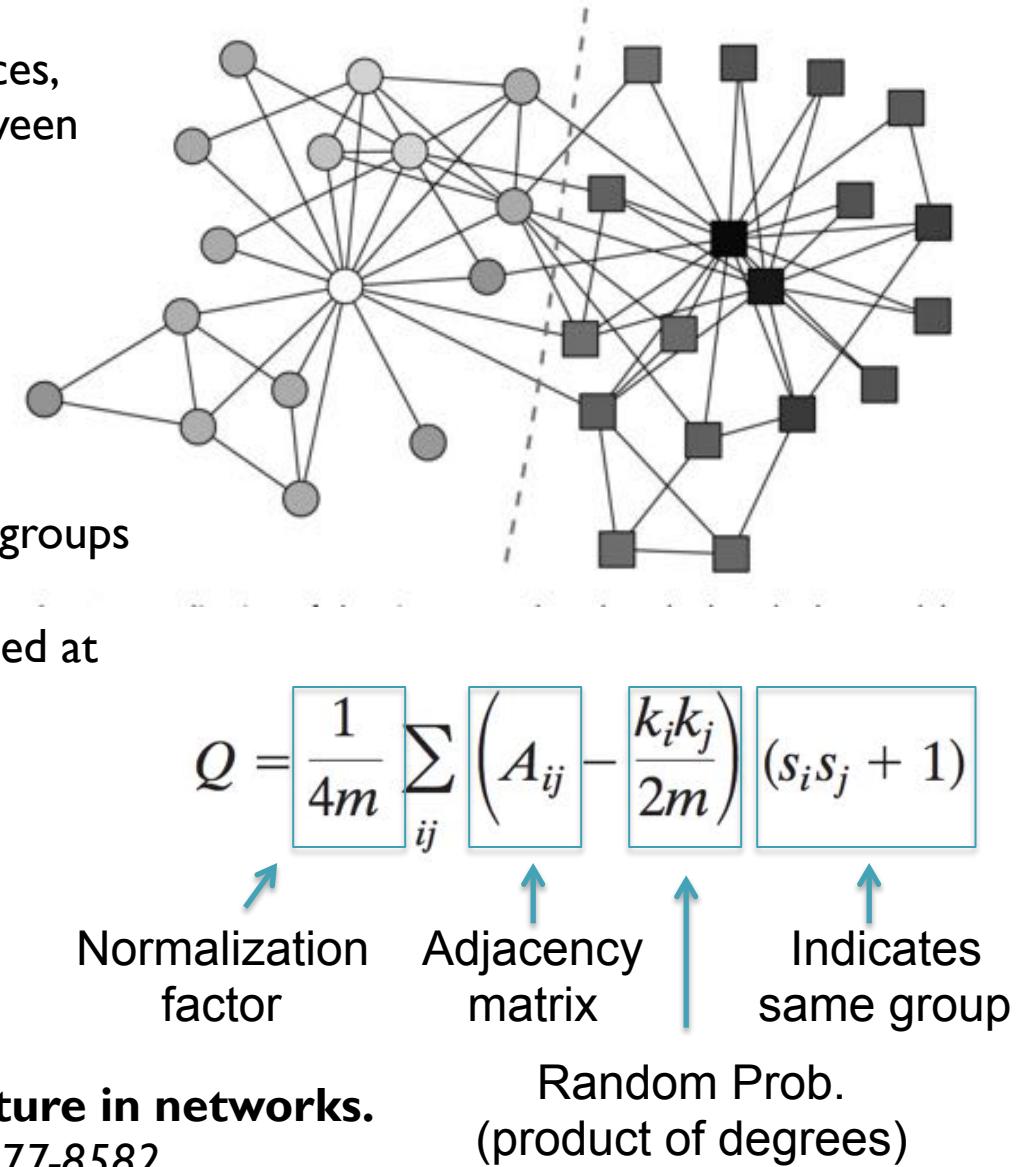
- Network Motif
 - Simple graph of connections
 - Exhaustively enumerate all possible 1, 2, 3, ... k node motifs
- Statistical Significance
 - Compare frequency of a particular network motif in a real network as compared to a randomized network
- Certain motifs are “characteristic features” of the network

| Network | Nodes | Edges | N_{real} | $N_{\text{rand}} \pm \text{SD}$ | Z score | N_{real} | $N_{\text{rand}} \pm \text{SD}$ | Z score | N_{real} | $N_{\text{rand}} \pm \text{SD}$ | Z score |
|---|---------|--------|-------------------|---------------------------------|---|-------------------|---------------------------------|--------------------------|-------------------|---------------------------------|---------|
| Gene regulation (transcription) | | | | X Y Y Z | Feed-forward loop | X Y Z W | | Bi-fan | | | |
| <i>E. coli</i> | 424 | 519 | 40 | 7 ± 3 | 10 | 203 | 47 ± 12 | 13 | | | |
| <i>S. cerevisiae*</i> | 685 | 1,052 | 70 | 11 ± 4 | 14 | 1812 | 300 ± 40 | 41 | | | |
| Neurons | | | | X Y Y Z | Feed-forward loop | X Y Z W | | Bi-fan | | | |
| <i>C. elegans†</i> | 252 | 509 | 125 | 90 ± 10 | 3.7 | 127 | 55 ± 13 | 5.3 | 227 | 35 ± 10 | 20 |
| Food webs | | | | X Y Y Z | Three chain | X Y Z W | | 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) | | | | X Y Y Z | Feed-forward loop | X Y Z W | | Bi-fan | | | |
| 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) | | | | X Y Y Z | Three-node feedback loop | X Y Z W | | Bi-fan | | | |
| 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 | | | | X Y Y Z | Feedback with two mutual dyads | X Y Z W | | Fully connected triad | | | |
| 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

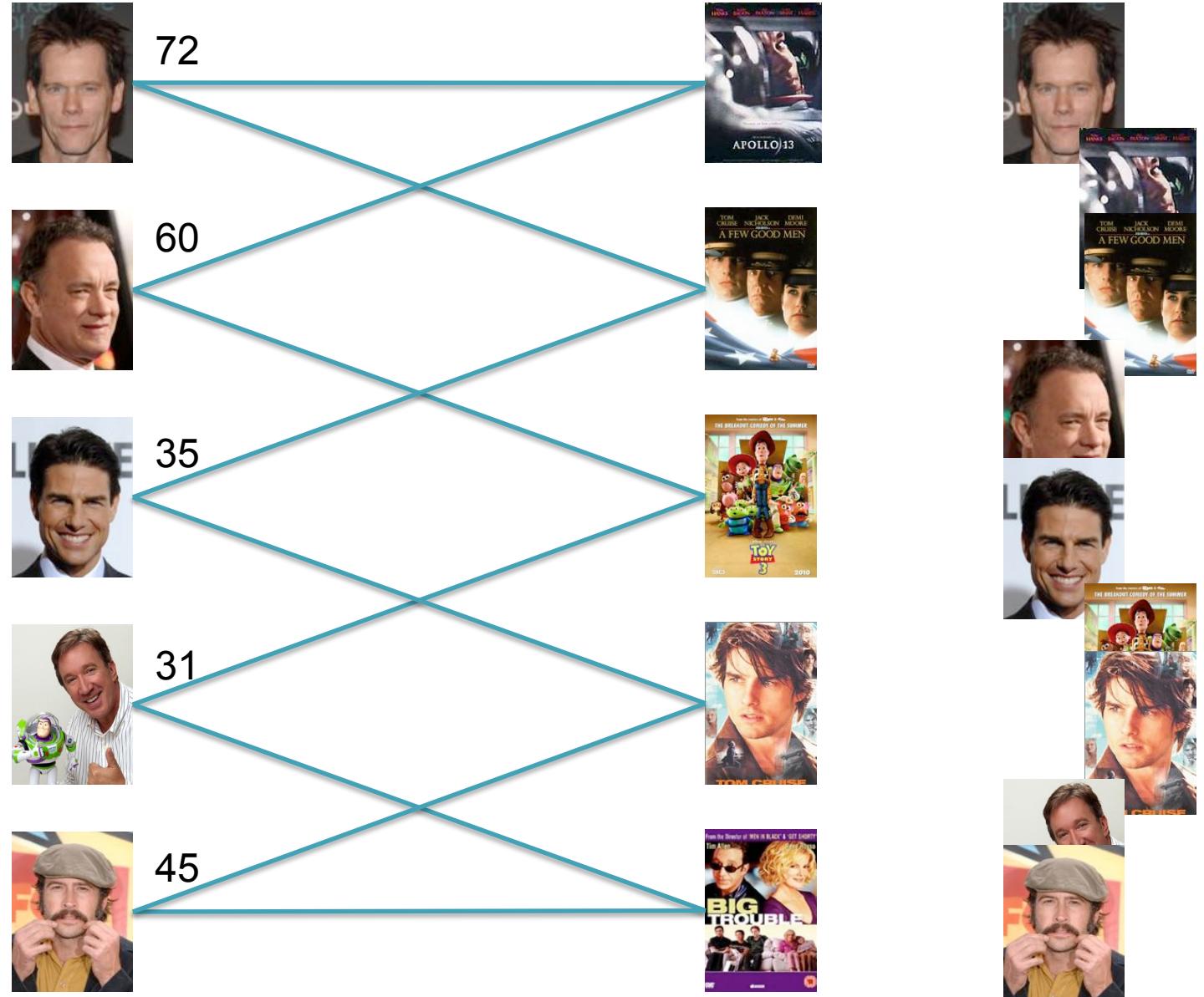
- Community structure
 - Densely connected groups of vertices, with only sparser connections between groups
 - Reveals the structure of large-scale network data sets
- Modularity
 - The number of edges falling within groups minus the expected number in an equivalent network with edges placed at random
 - Larger positive values => Stronger community structure
 - Optimal assignment determined by computing the eigenvector of the modularity matrix



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

Kevin Bacon and Bipartite Graphs

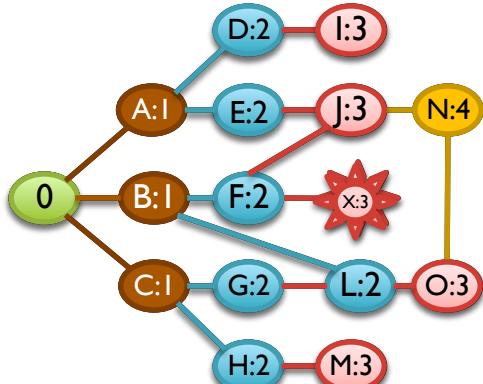
Find the **shortest** path from Kevin Bacon to Jason Lee



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)
```



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?]

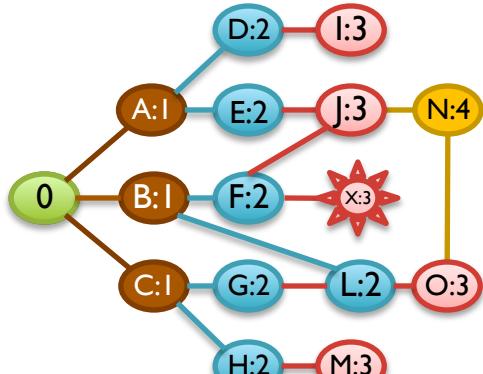
[What's the running time?]

[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)
```

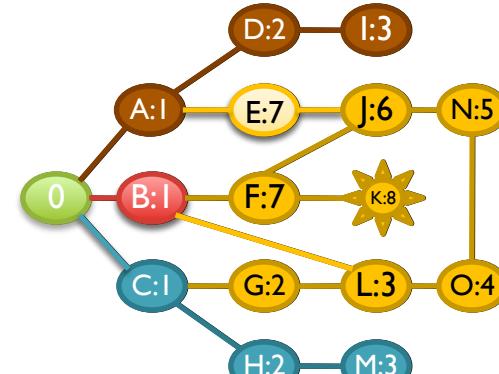
0

A,B,C

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

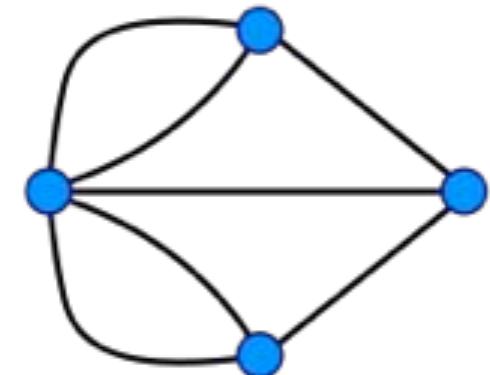
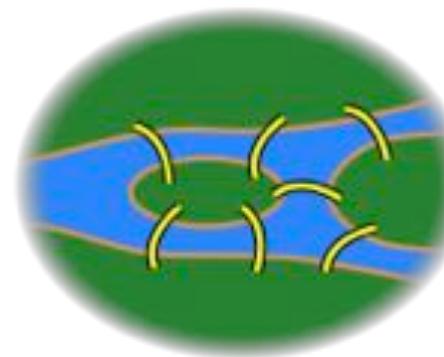
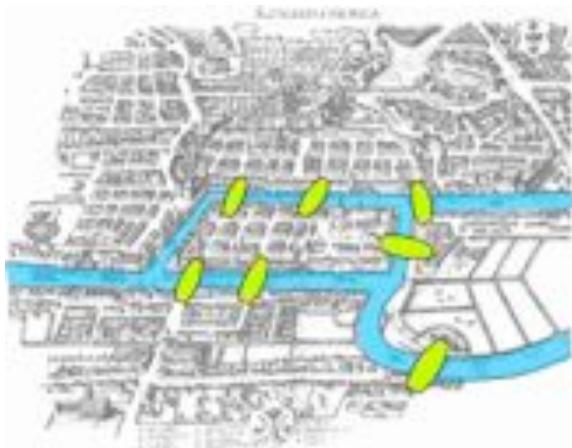
A,B,G
A,B,L
A,B,O
A,B,N
A,B,J
A,B,E,F
A,B,E,K
A,B,E

A,B
A
D
I



Eulerian Cycle Problem

- **Seven Bridges of Königsberg**
 - Find a cycle that visits every **edge** exactly once



[Can you find the cycle?]

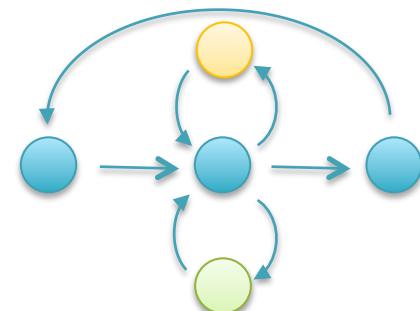
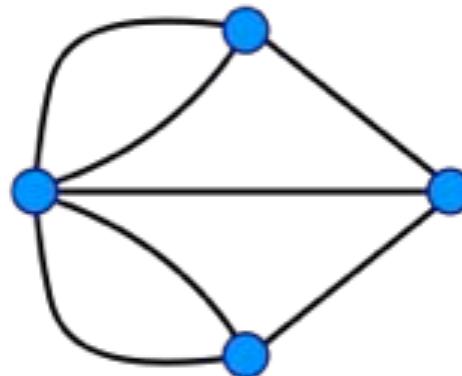
bioalgorithms.info

Euler Theorem

- A graph is ***balanced*** if for every vertex the number of incoming edges equals to the number of outgoing edges:

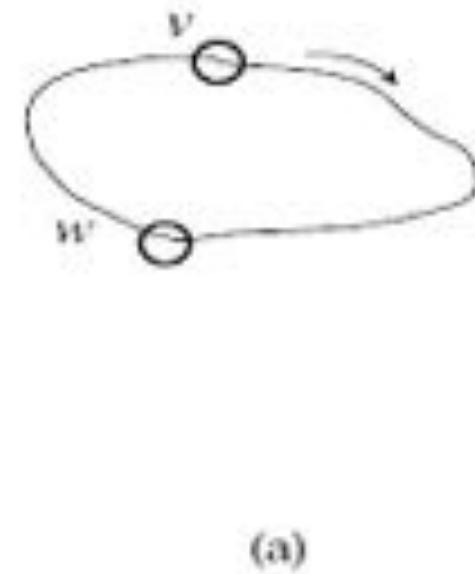
$$in(v) = out(v)$$

- **Theorem:** A connected graph is Eulerian if and only if each of its vertices is balanced.



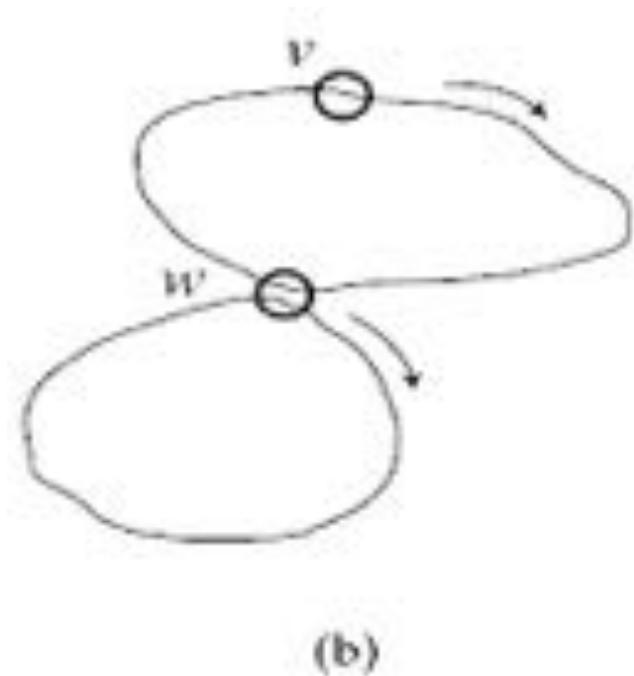
Algorithm for Constructing an Eulerian Cycle

- a. Start with an arbitrary vertex v and form an arbitrary cycle with unused edges until a dead end is reached. Since the graph is Eulerian this dead end is necessarily the starting point, i.e., vertex v .



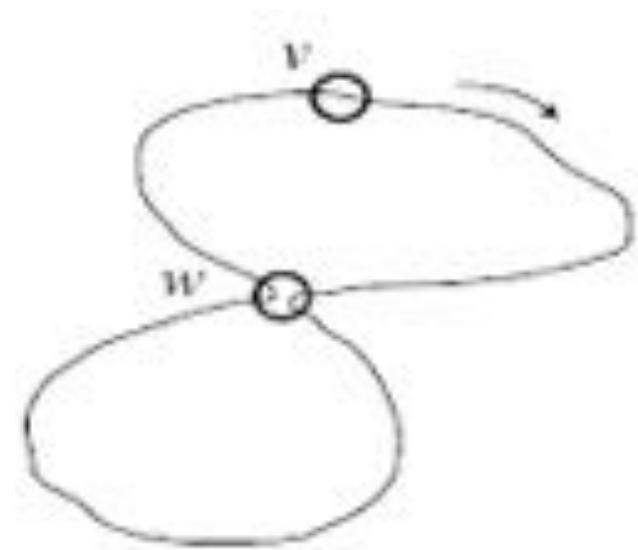
Algorithm for Constructing an Eulerian Cycle (cont' d)

- b. If cycle from (a) above is not an Eulerian cycle, it must contain a vertex w , which has untraversed edges. Perform step (a) again, using vertex w as the starting point. Once again, we will end up in the starting vertex w .



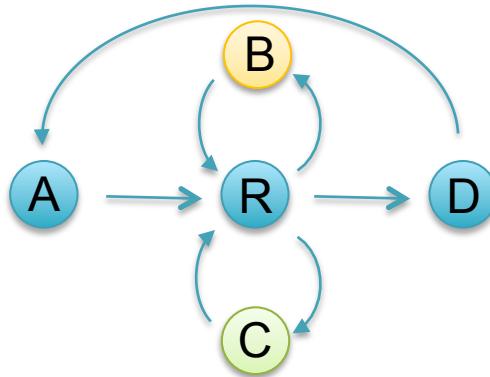
Algorithm for Constructing an Eulerian Cycle (cont' d)

- c. Combine the cycles from (a) and (b) into a single cycle and iterate step (b).



(c)

Counting Eulerian Cycles



ARBR_CRD
or
ARC_RB_BRD

Generally an exponential number of compatible sequences

- Value computed by application of the BEST theorem (Hutchinson, 1975)

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

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

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

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

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

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: $4+2+5+3 = 14$

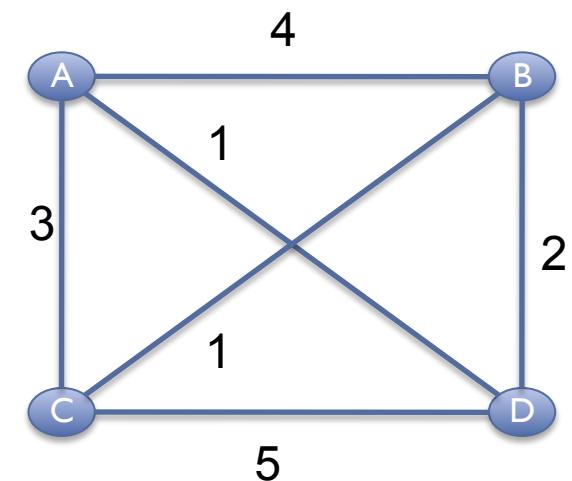
ACDBA: $3+5+2+4 = 14^*$

ABCDA: $4+1+5+1 = 11$

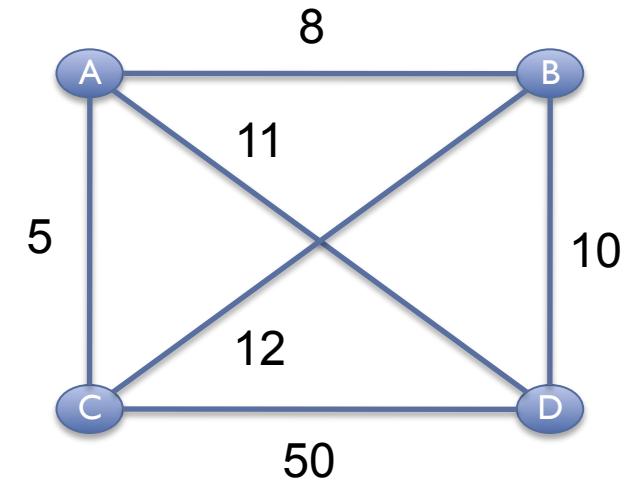
ADCBA: $1+5+1+4 = 11^*$

ACBDA: $3+1+2+1 = 7$

ADBKA: $1+2+1+3= 7 *$



Greedy Search



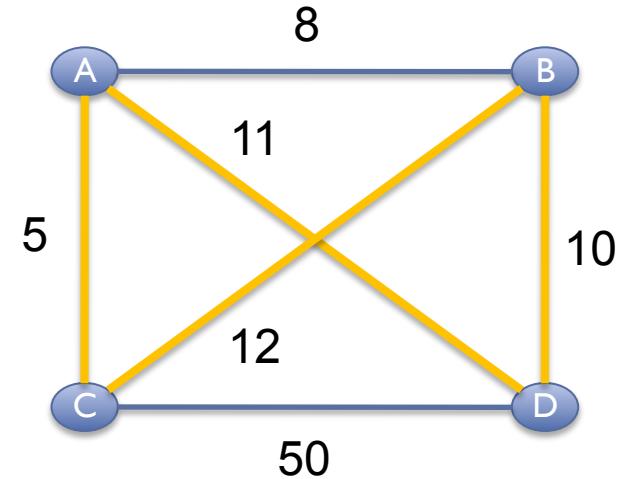
Greedy Search

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

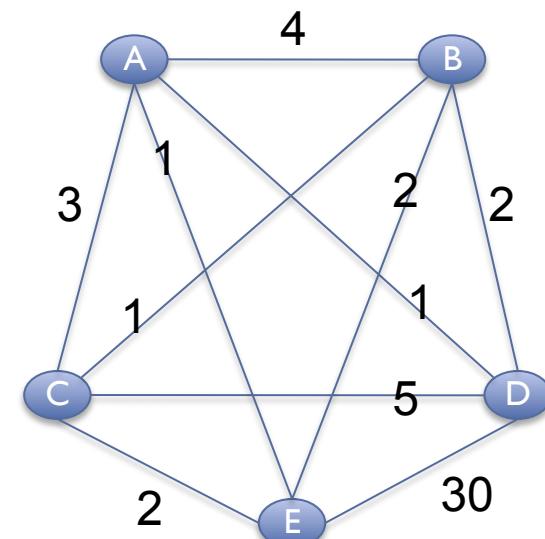
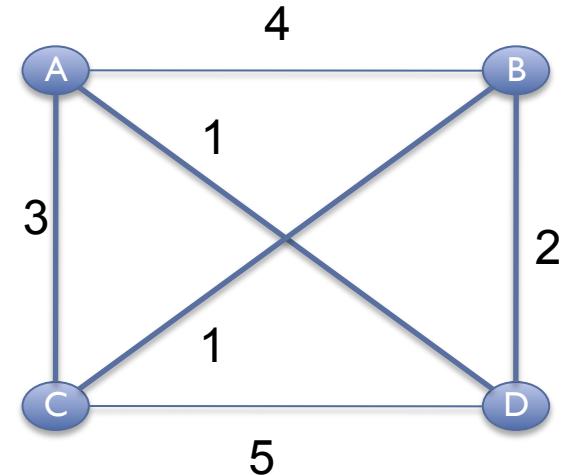
Optimal Greedy: Making change with the fewest number of coins

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?]

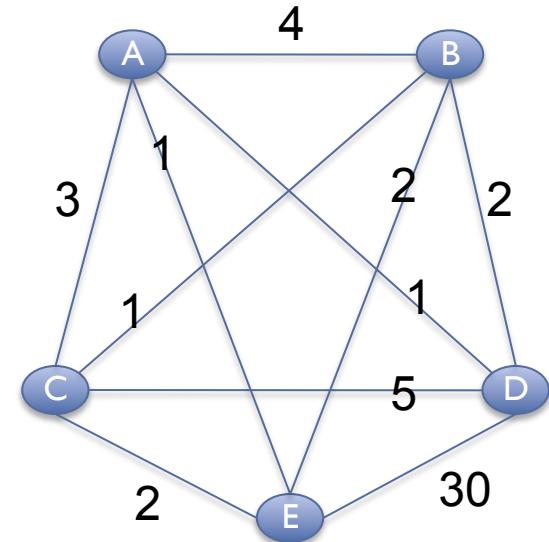
- Extensive searching is the only provably correct algorithm
 - Brute Force: $O(n!)$
 - ~20 cities max
 - $20! = 2.4 \times 10^{18}$



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

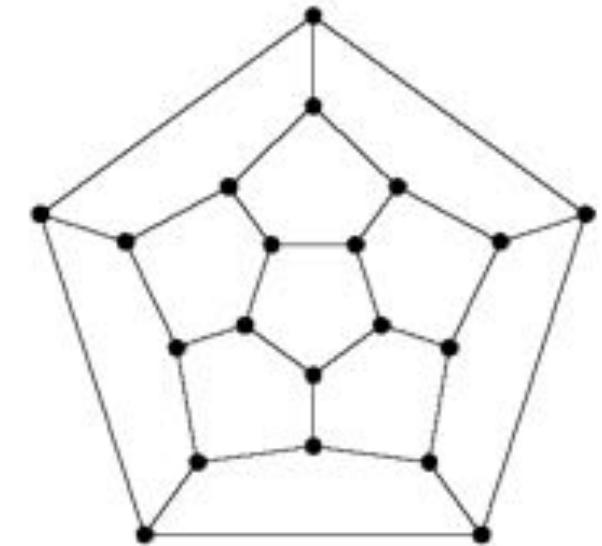
- Always gives the optimal answer
- Doesn't always help performance, but often does
- Current TSP record holder:

- 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



Shortest Common Superstring

Given: $S = \{s_1, \dots, s_n\}$

Problem: Find minimal length superstring of S

$s_1, s_2, s_3 = \text{CACCCGGGTGCACCC} \quad 15$

$s_1 \text{ CACCC} \quad s_1, s_3, s_2 = \text{CACCCACCGGGTGC} \quad 14$

$s_2 \text{ CCGGGTGC} \quad s_2, s_1, s_3 = \text{CCGGGTGACCCACC} \quad 15$

$s_3 \text{ CCACCC} \quad s_2, s_3, s_1 = \text{CCGGGTGCCACCC} \quad 13$

$s_3, s_1, s_2 = \text{CCACCCGGGTGC} \quad 12$

$s_3, s_2, s_1 = \text{CCA} \text{CCGGGTGCACCC} \quad 15$

NP-Complete by reduction from VERTEX-COVER and later DIRECTED-HAMILTONIAN-PATH

Break



Milestones in Genome Assembly

Nature Vol. 265 February 24 1977

487

articles

Nucleotide sequence of bacteriophage Φ X174 DNA

F. Sanger, G. M. Air¹, B. G. Barrell, N. L. Brown¹, A. R. Coulson, J. C. Fiddes,
C. A. Hutchison III¹, P. M. Slocombe² & M. Smith²

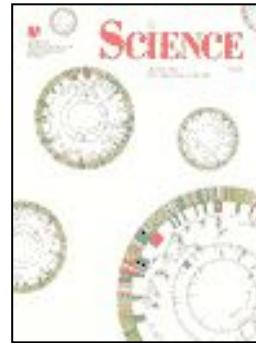
MRC Laboratory of Molecular Biology, Hills Road, Cambridge CB2 2QH, UK

A DNA sequence for the genome of bacteriophage Φ X174 of approximately 5,375 nucleotides has been determined using the rapid and simple "plus and minus" method. The sequence identifies many of the features responsible for the production of the various proteins known to be produced by the organism, including initiation and termination sites for the proteins and RNAs. Two pairs of genes are coded by the same region of DNA using different reading frames.

The genome of bacteriophage Φ X174 is a single-stranded, circular molecule of DNA. It contains 5,375 nucleotides and nine known proteins. The order of genes, as determined by genetic techniques¹⁻³, is A-B-C-D-E-F-F-G-H. Genes F, G and H code for structural proteins, while genes A, B, C and D code for enzymes involved in the metabolism of host bacteria.

Two sets of synthetic primers were used to prime the DNA with DNA polymerase were being developed⁴ and Scheraga⁵ found that one of these primers was complementary to part of the ribosome binding site. This was used to prime into the intervening regions between the T and G genes using DNA, ribonuclease and proteinase treatment. The ribonuclease technique⁶ facilitated the sequence determination of the intervening regions. The plus and minus method⁷ described was also used to develop the plus and minus method⁸. Suitable synthetic primers are, however, difficult to prepare and at

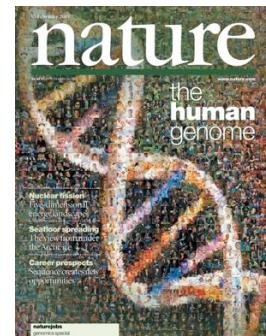
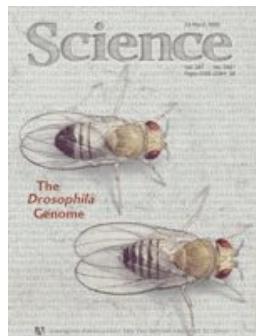
1977. Sanger et al.
1st Complete Organism
5375 bp



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

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

Assembly Applications

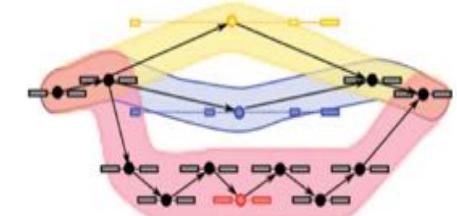
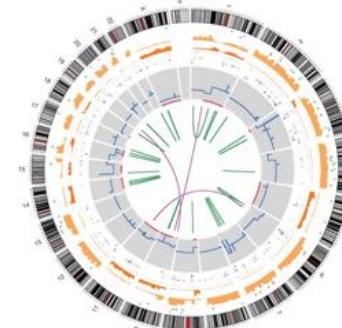
- Novel genomes



- Metagenomes

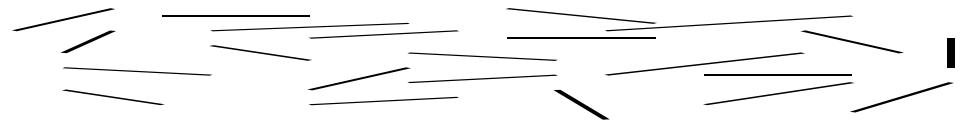


- Sequencing assays
 - Structural variations
 - Transcript assembly
 - ...



Assembling a Genome

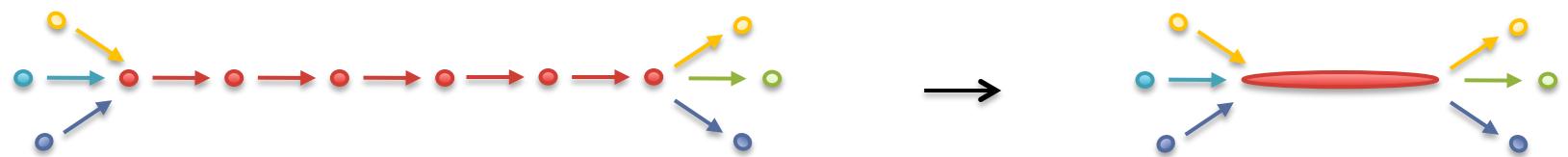
1. Shear & Sequence DNA



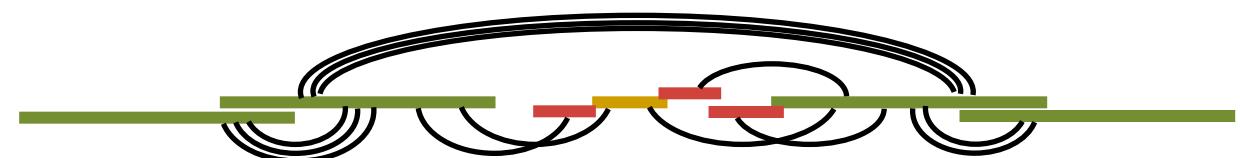
2. Construct assembly graph from overlapping reads

...AGCCTAGACCTACA**GGATGCGCGACACGT**
GGATGCGCGACACGTCGCATATCCGGT...

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:**

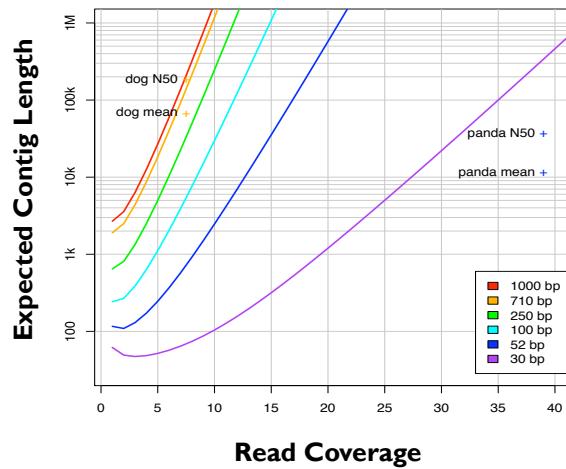
- (Very) Large genomes, complex structure

4. **Accuracy:**

- (Very) Hard to assess correctness

Ingredients for a good assembly

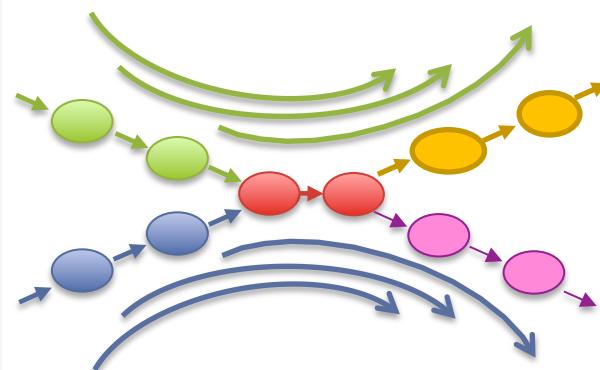
Coverage



High coverage is required

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

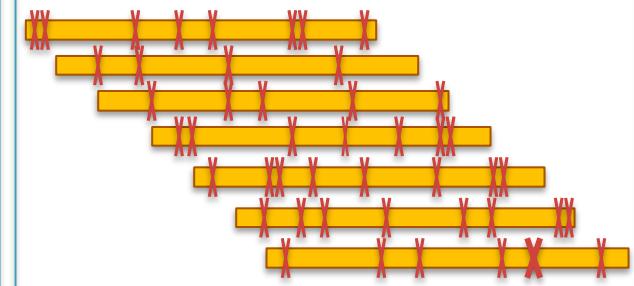
Read Length



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

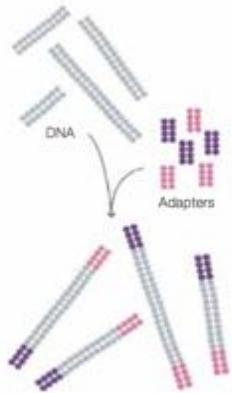


Errors obscure overlaps

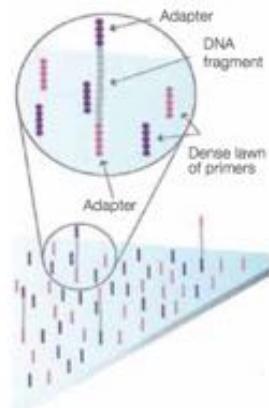
- Reads are assembled by finding kmers shared in pair of reads
- High error rate requires very short seeds, increasing complexity and forming assembly hairballs

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

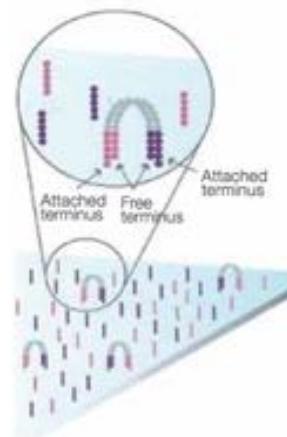
Illumina Sequencing by Synthesis



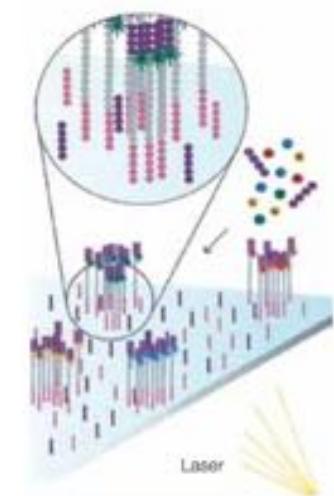
1. Prepare



2. Attach



3. Amplify



4. Image



5. Basecall

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

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

Paired-end and Mate-pairs

Paired-end sequencing

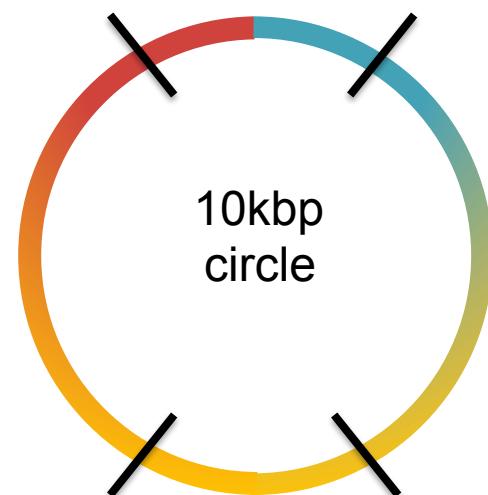
- Read one end of the molecule, flip, and read the other end
- Generate pair of reads separated by up to 500bp with inward orientation



Mate-pair sequencing

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

10kbp



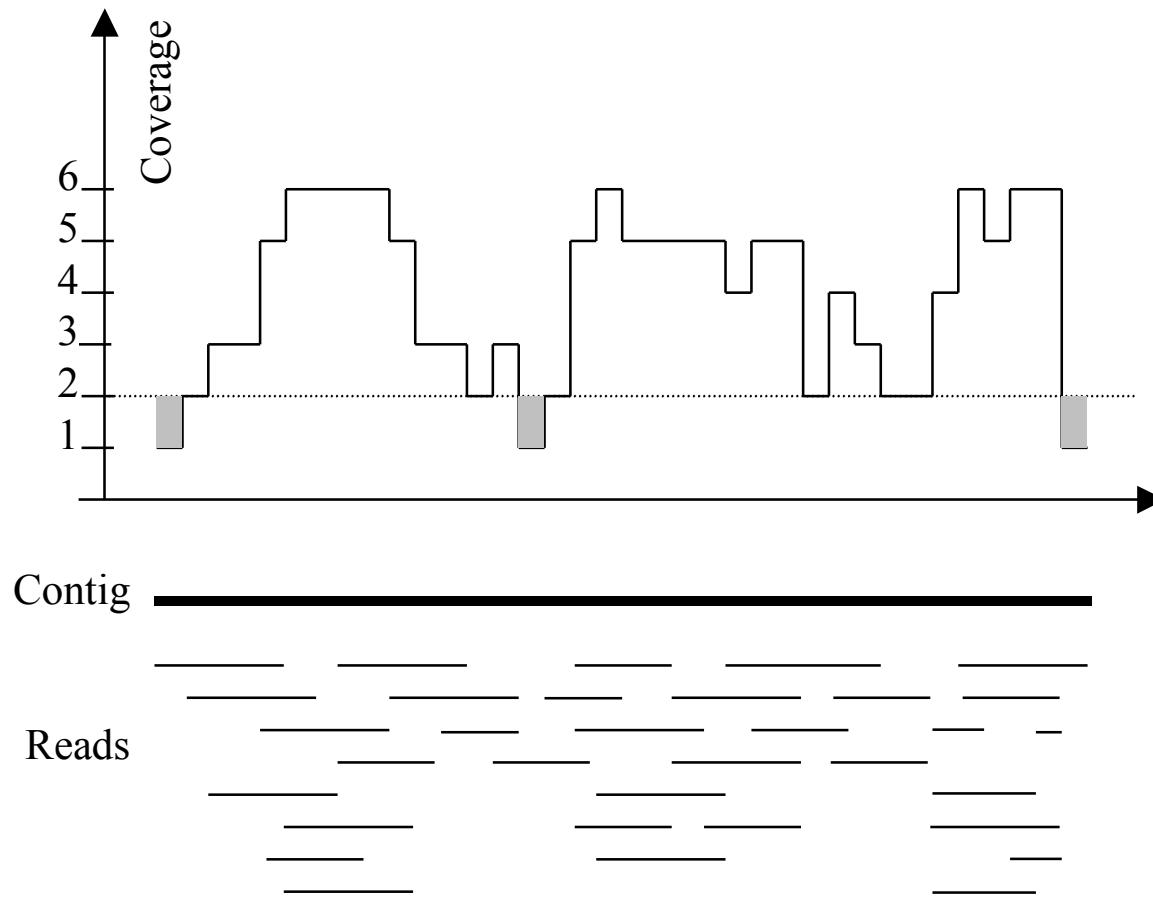
2x100 @ ~10kbp (outies)



2x100 @ 300bp (innies)

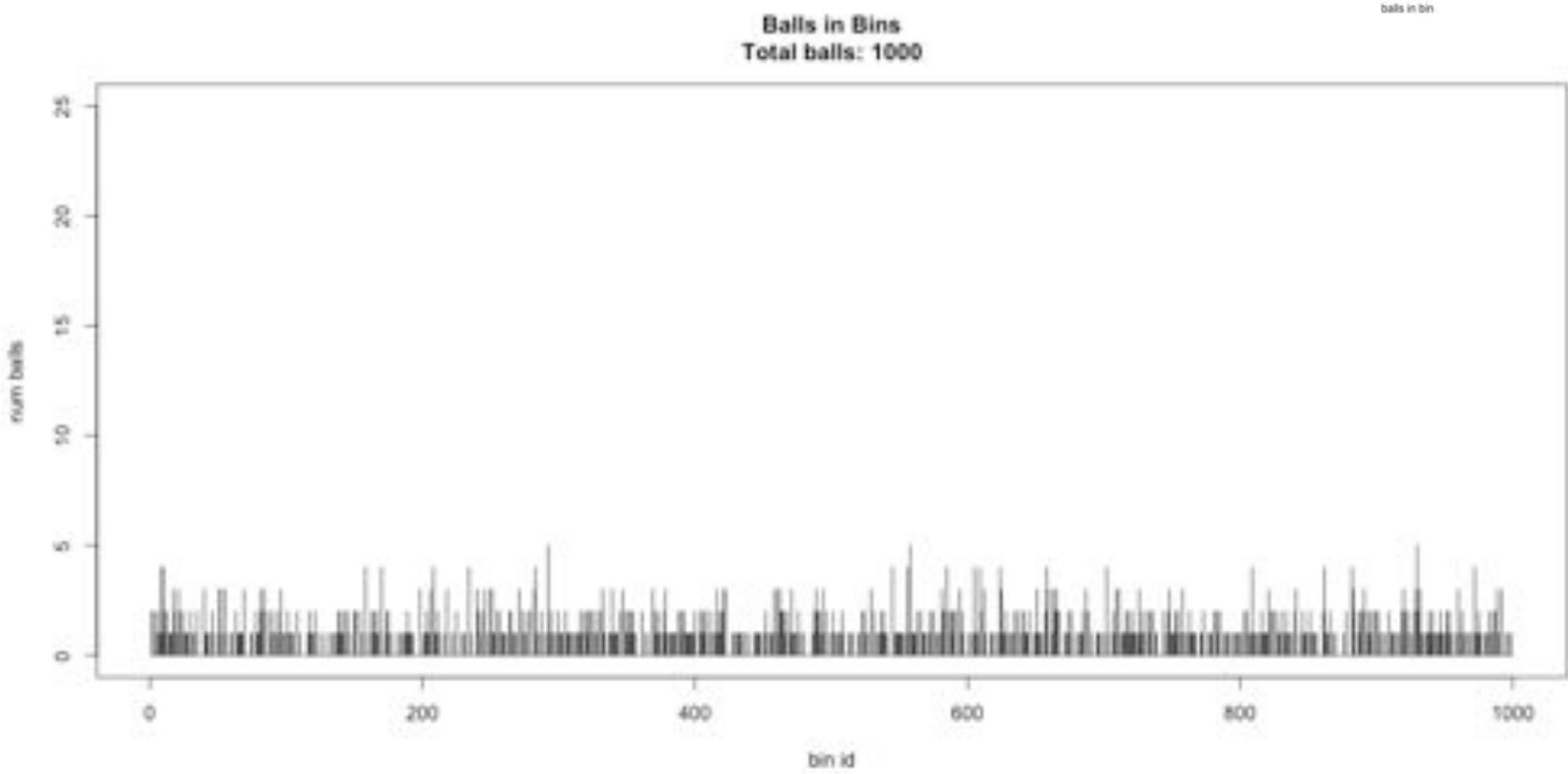


Typical contig coverage



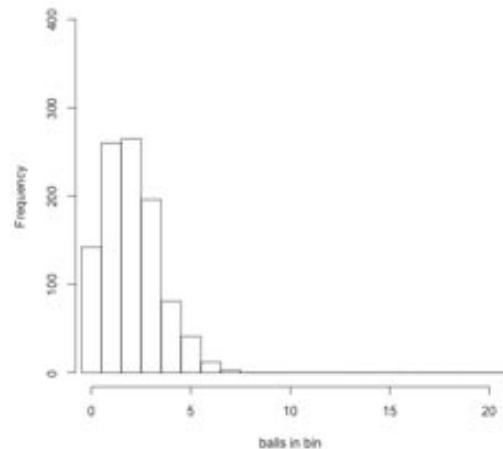
Imagine raindrops on a sidewalk

Balls in Bins IX

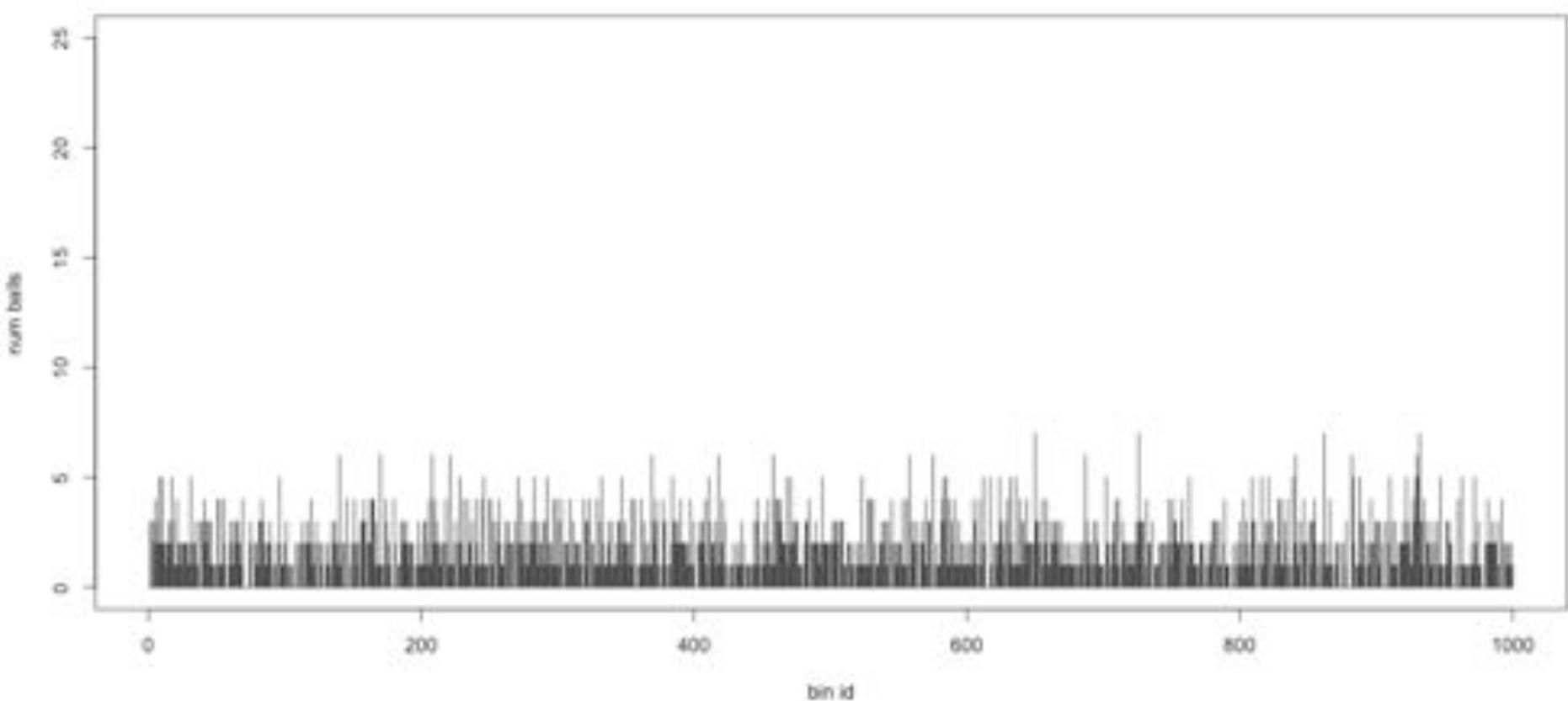


Balls in Bins 2x

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

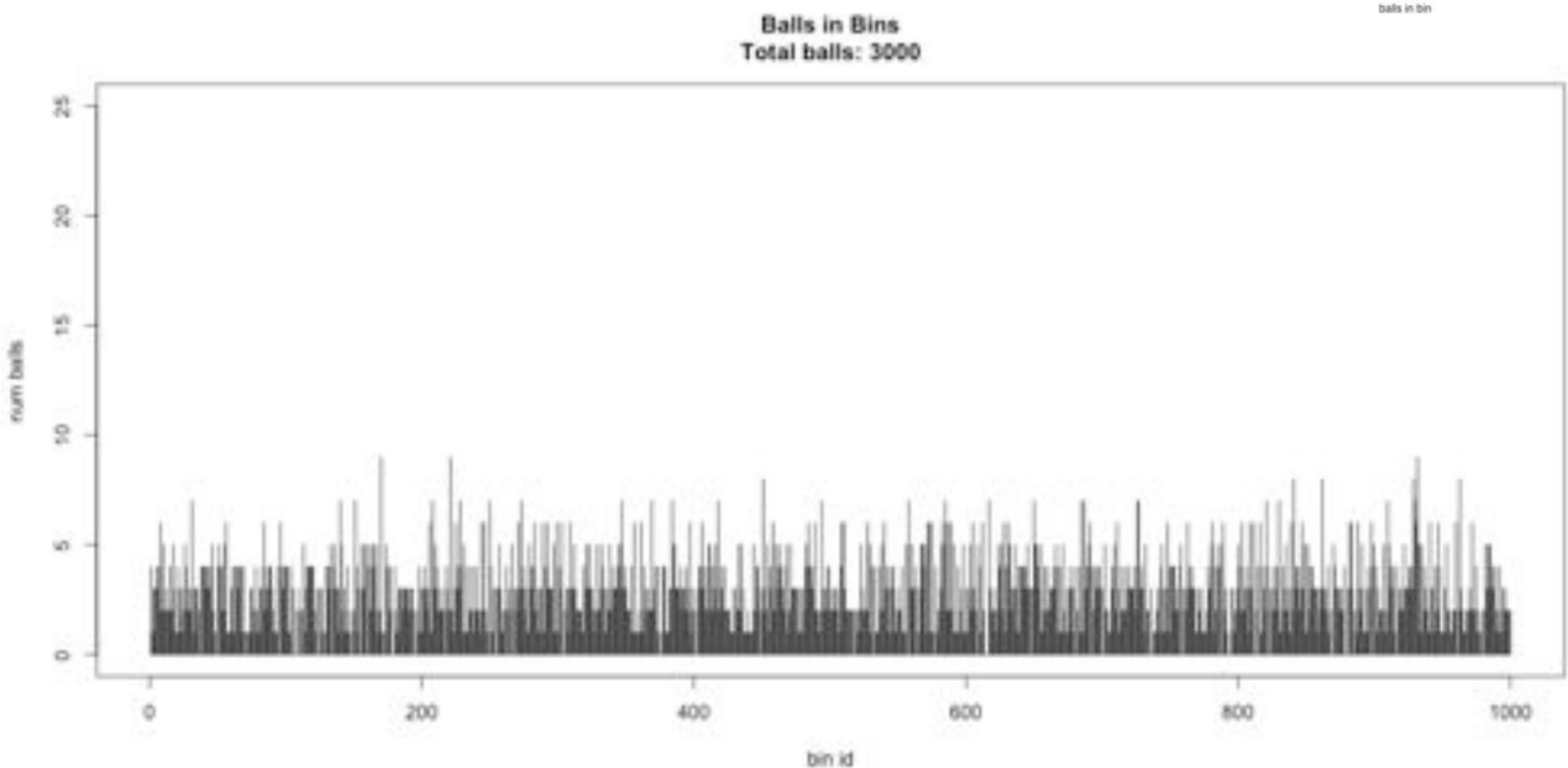
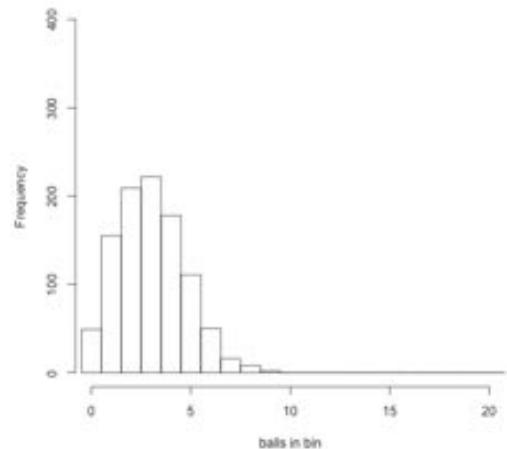


Balls in Bins
Total balls: 2000



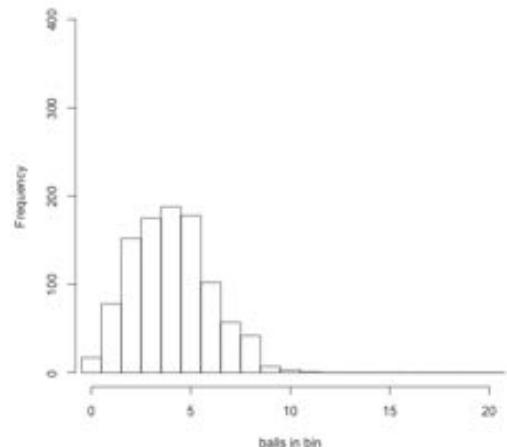
Balls in Bins 3x

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

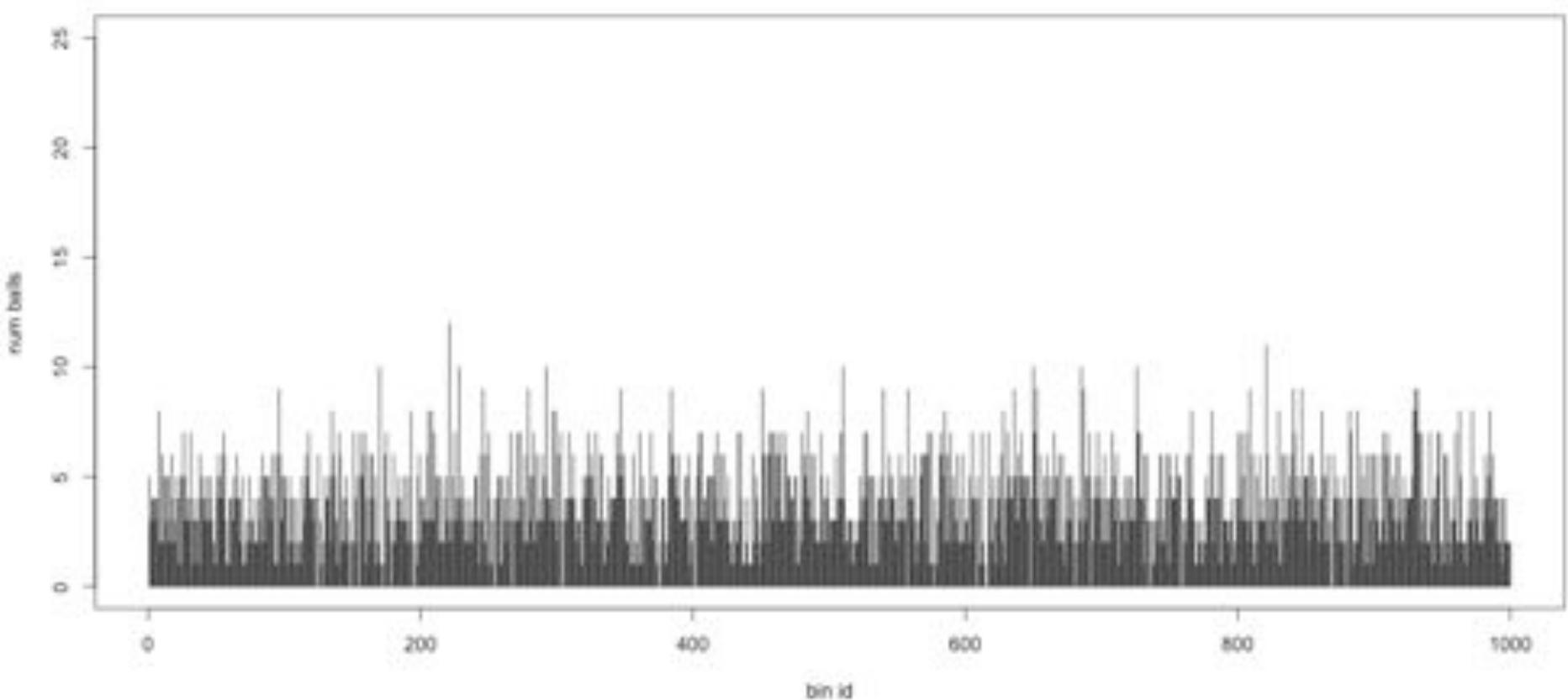


Balls in Bins 4x

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

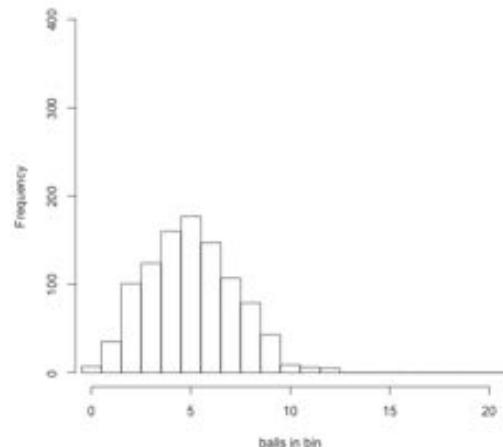


Balls in Bins
Total balls: 4000

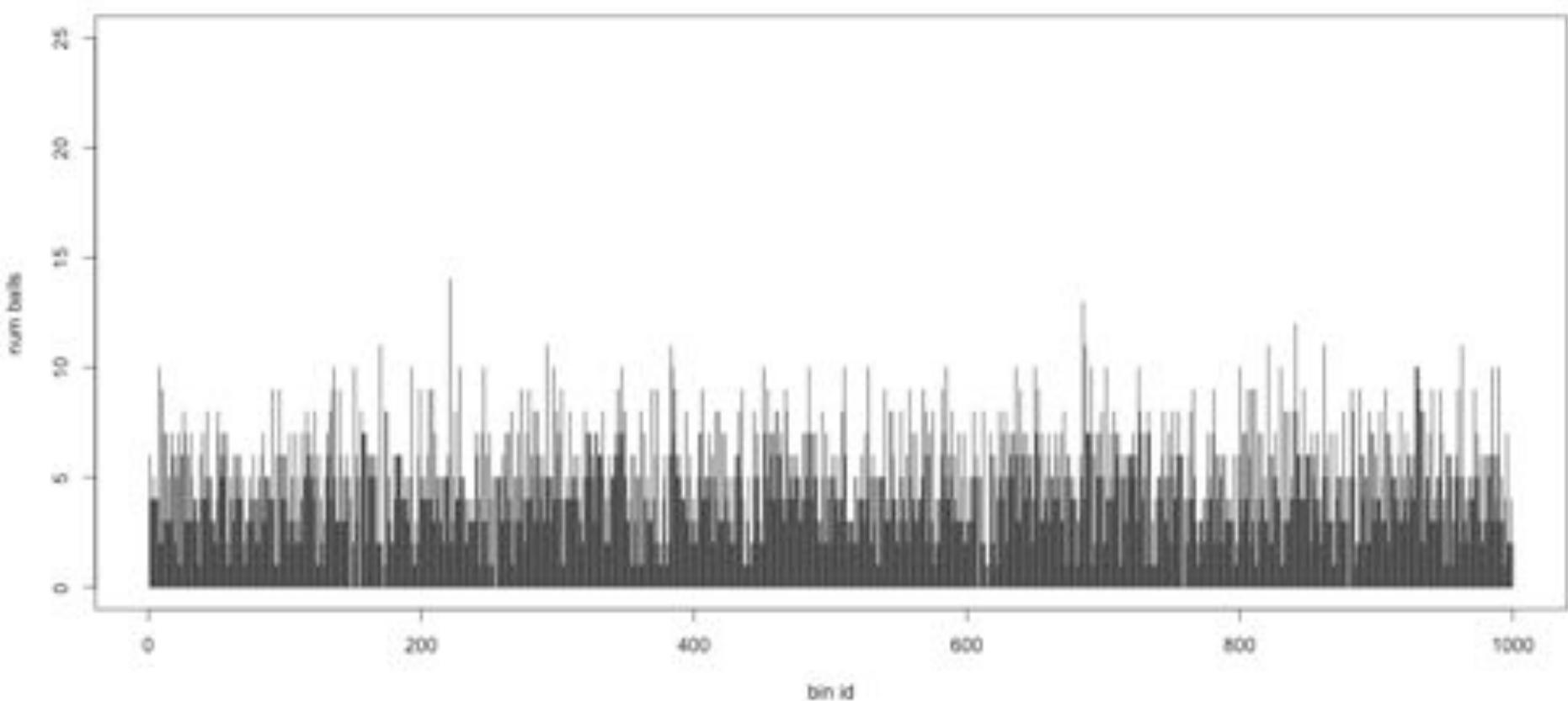


Balls in Bins 5x

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

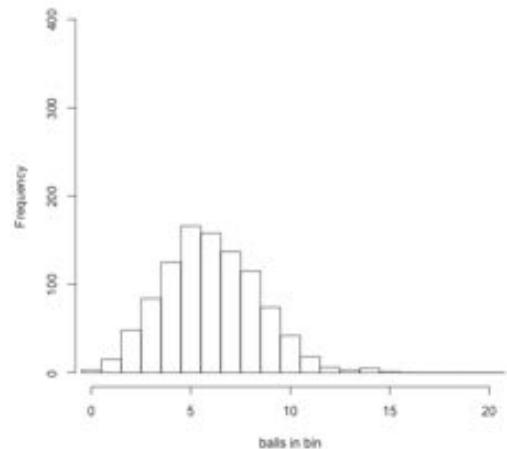


Balls in Bins
Total balls: 5000

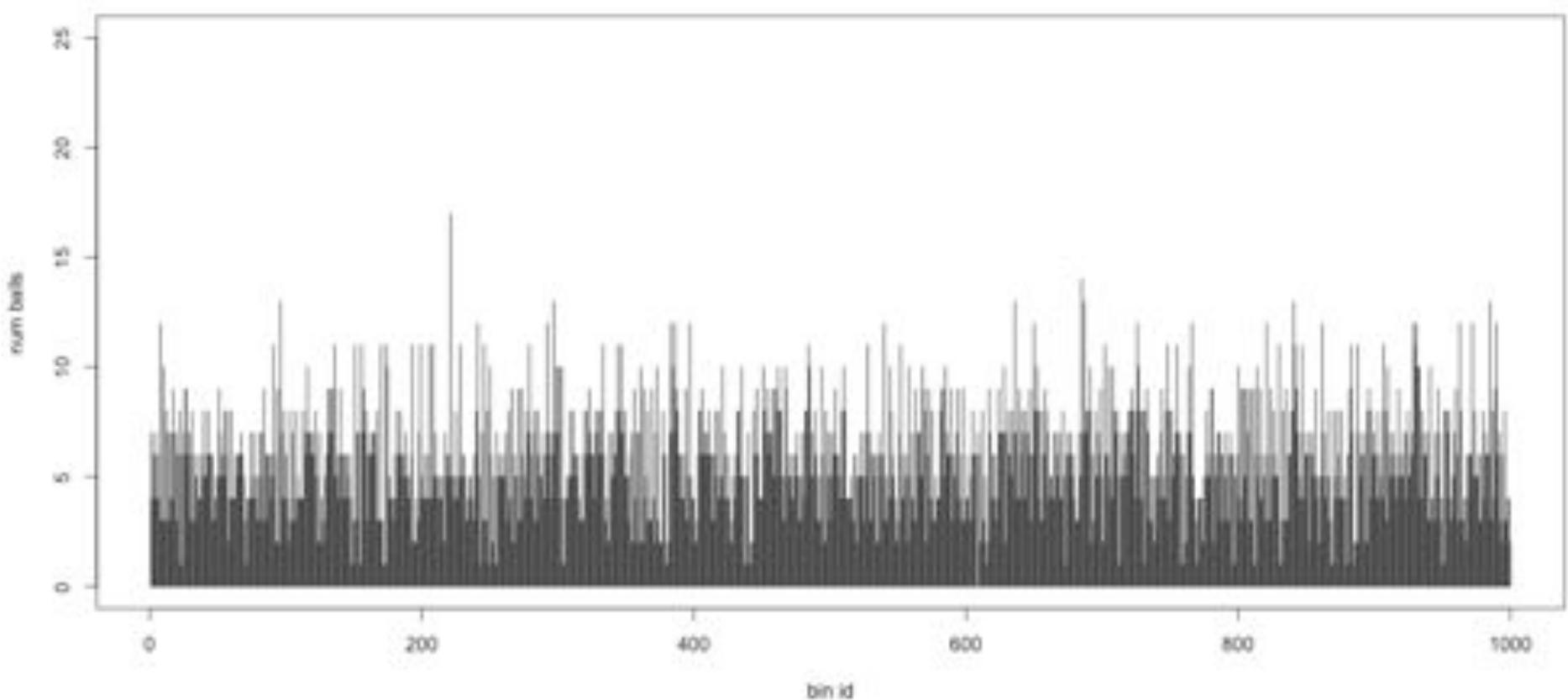


Balls in Bins 6x

Histogram of balls in each bin
Total balls: 6000 Empty bins: 3

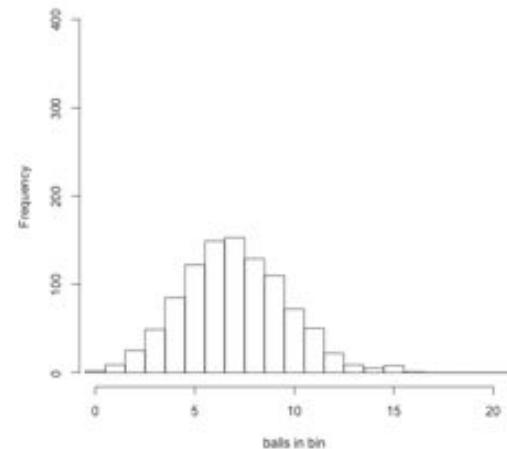


Balls in Bins
Total balls: 6000

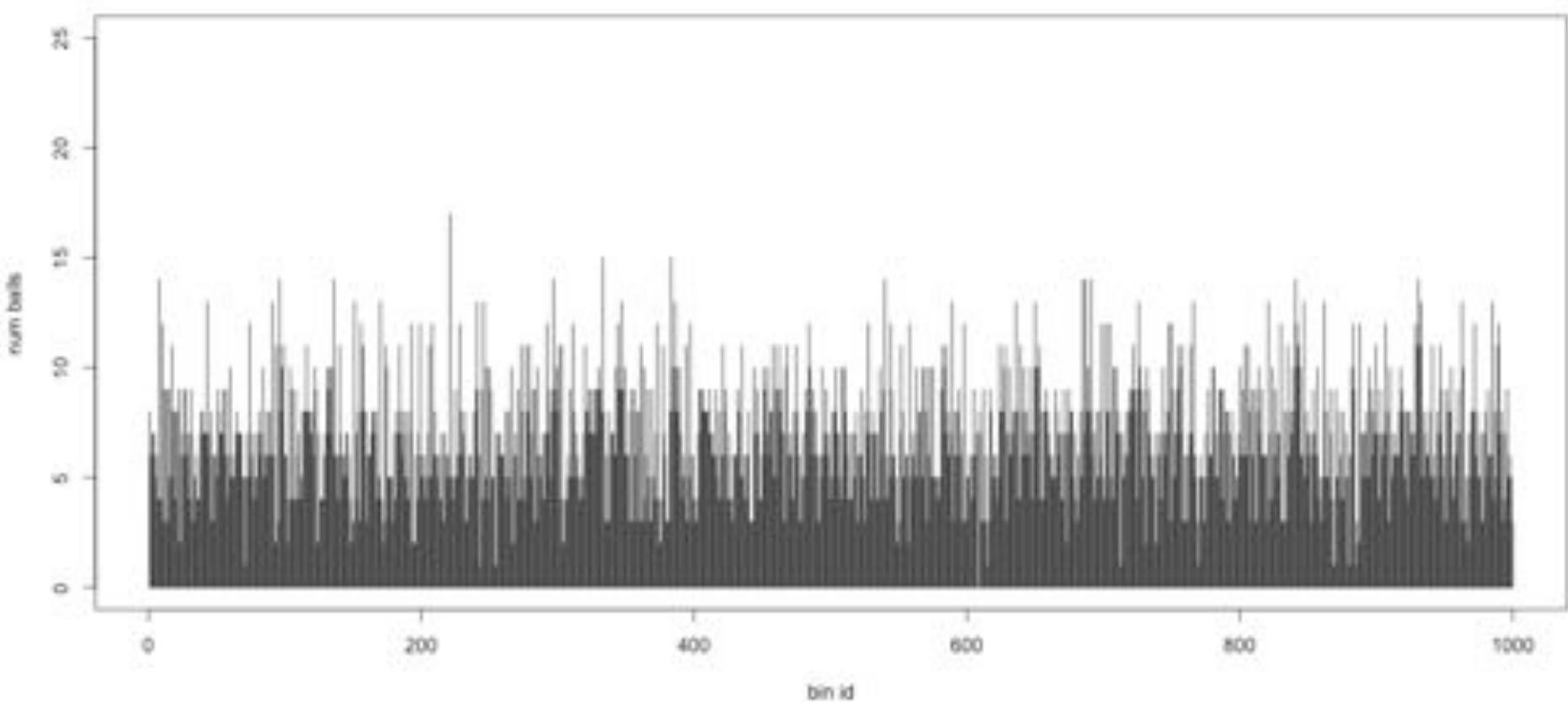


Balls in Bins 7x

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

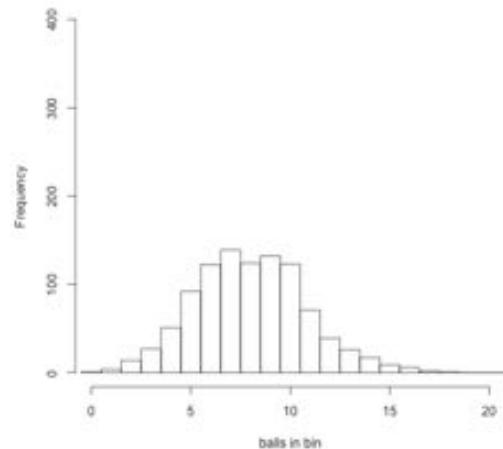


Balls in Bins
Total balls: 7000

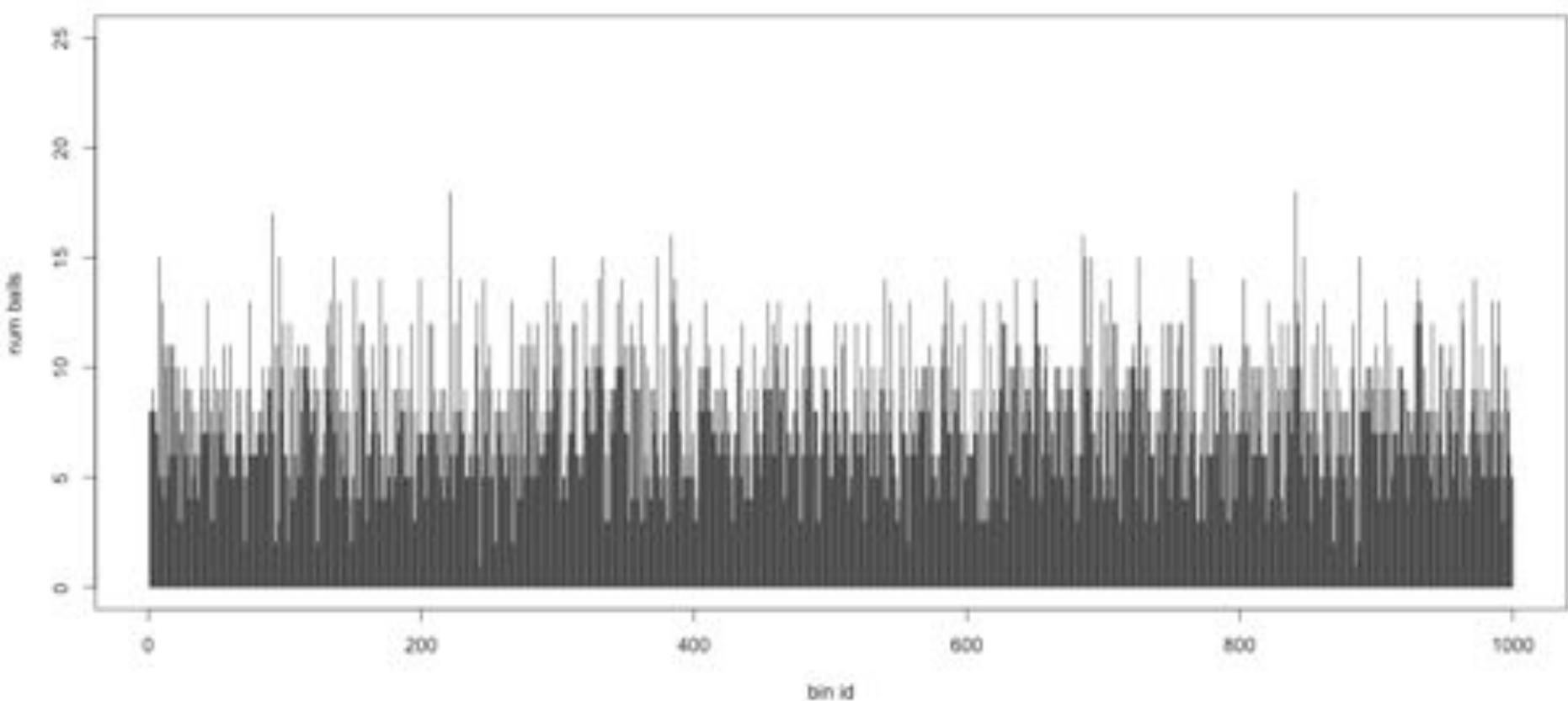


Balls in Bins 8x

Histogram of balls in each bin
Total balls: 8000 Empty bins: 1



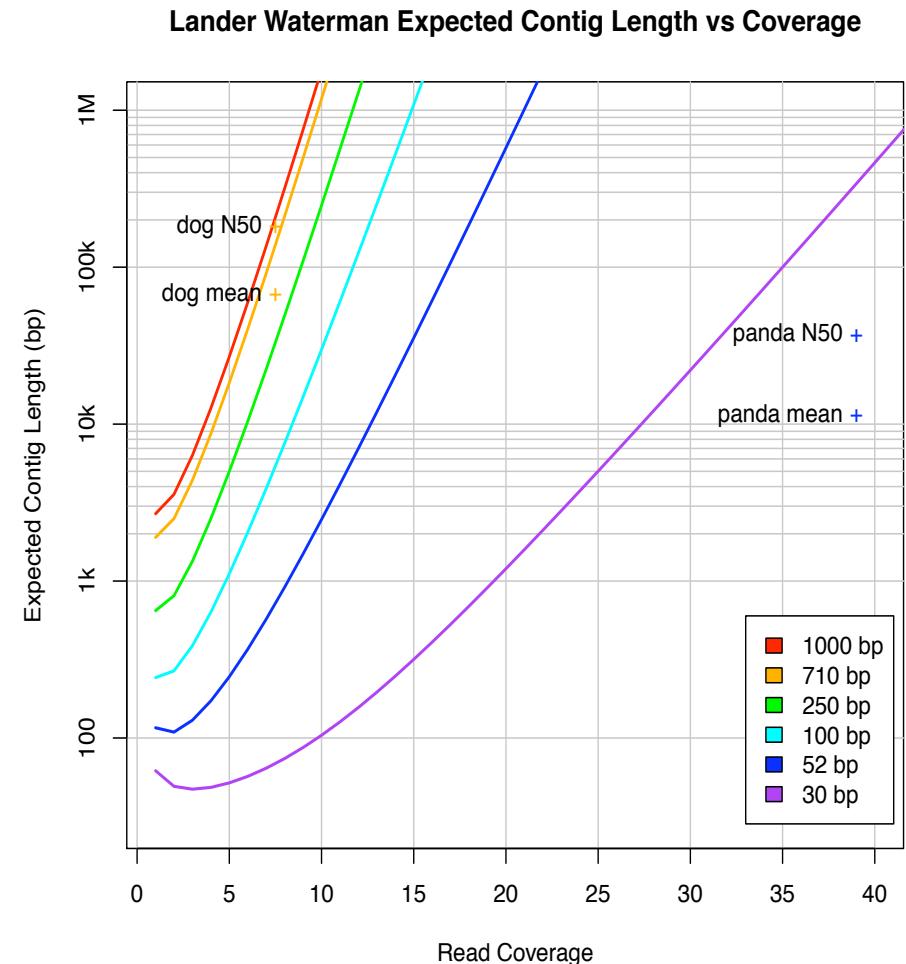
Balls in Bins
Total balls: 8000



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.

de Bruijn Graph Construction

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

Original Fragment

It was the best of

Directed Edge

It was the best → was the best of

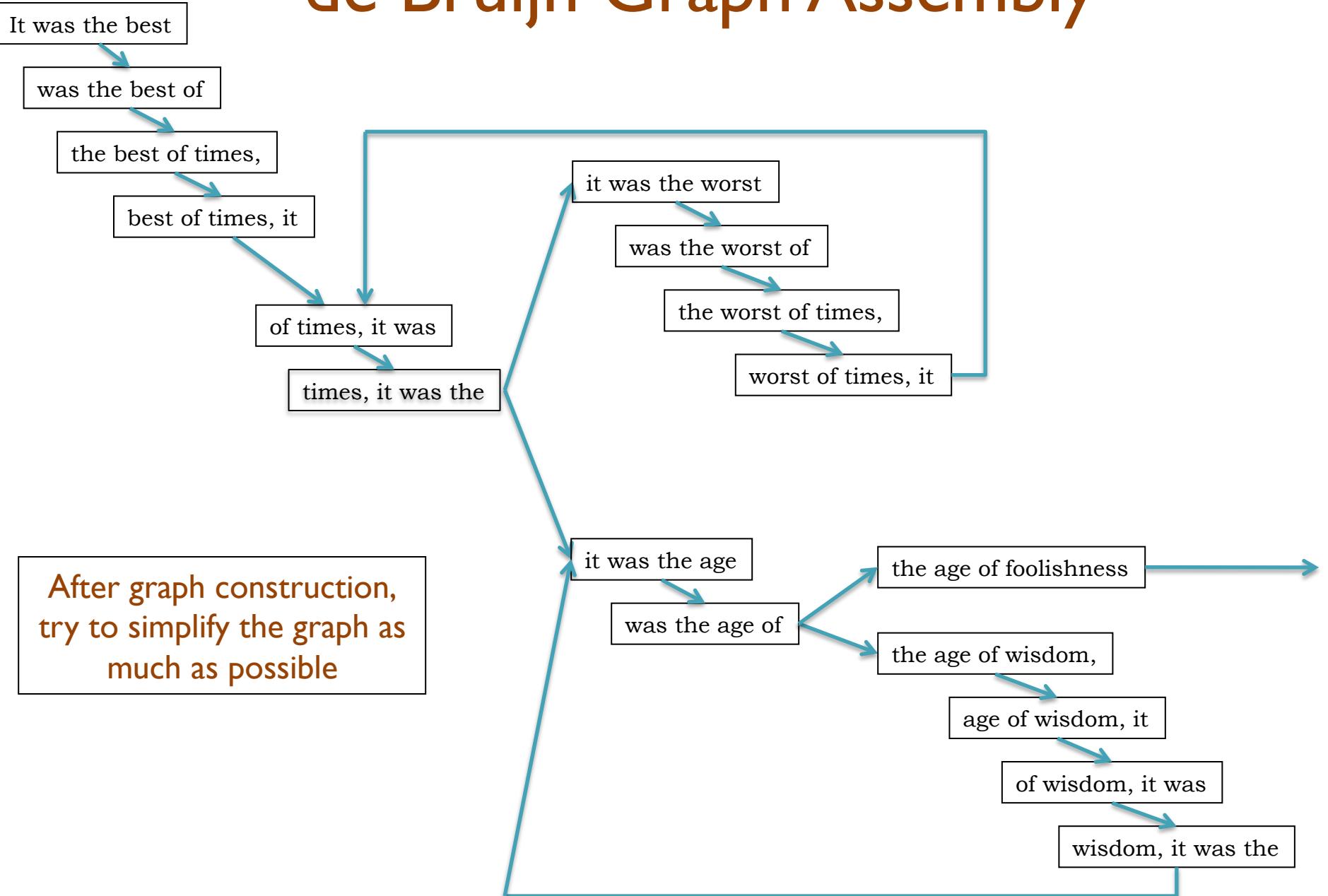
- Locally constructed graph reveals the global sequence structure
 - Overlaps between sequences implicitly computed

de Bruijn, 1946

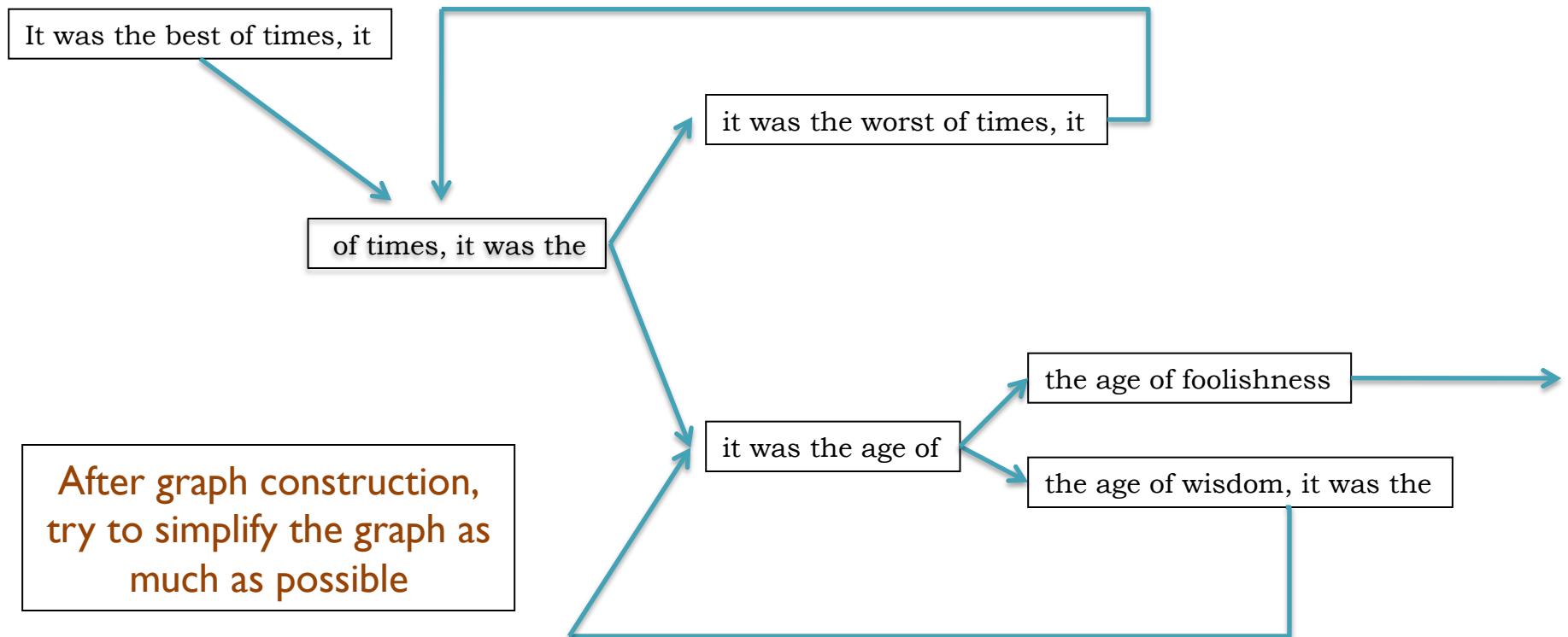
Idury and Waterman, 1995

Pevzner, Tang, Waterman, 2001

de Bruijn Graph Assembly

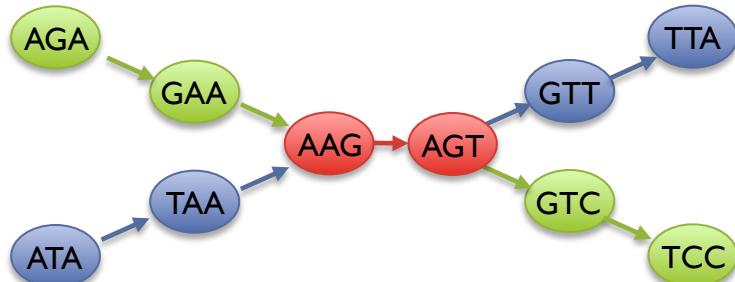


de Bruijn Graph Assembly



Two Paradigms for Assembly

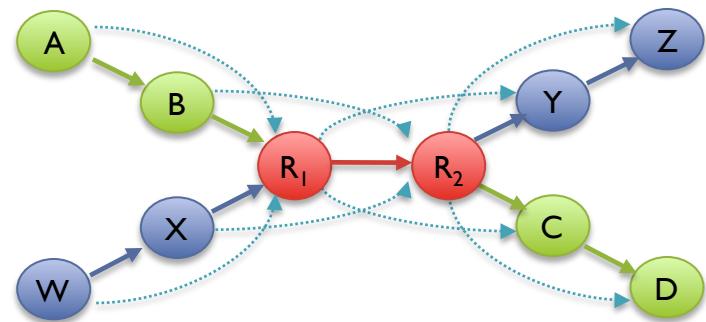
de Bruijn Graph



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.

Overlap between two sequences

overlap (19 bases) overhang (6 bases)

...AGCCTAGACCTACA**GGATGCGCGGACACGTAGCCAGGAC**
CAGTACTTGGATGCGCTGACACGTAGCTTATCCGGT...

overhang % identity = 18/19 % = 94.7%

overlap - region of similarity between regions

overhang - un-aligned ends of the sequences

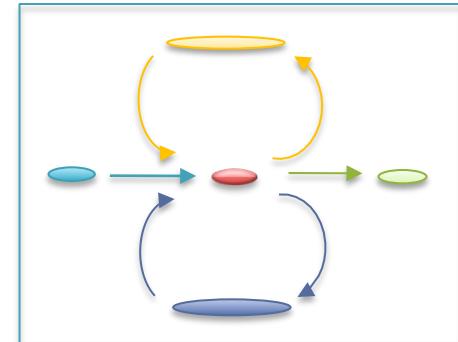
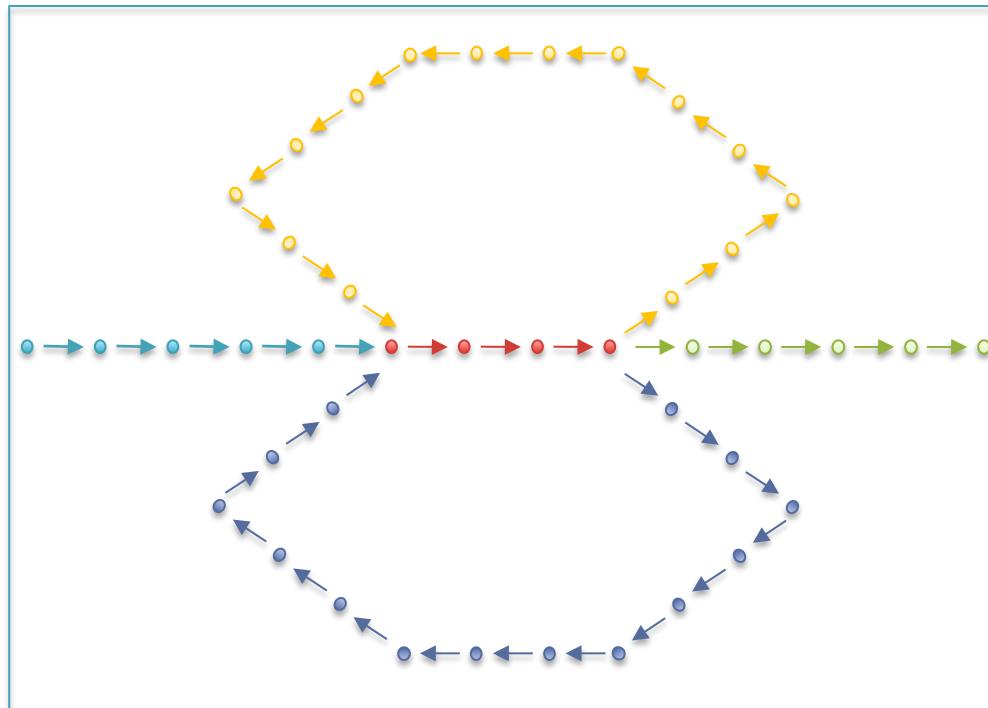
The assembler screens merges based on:

- length of overlap
- % identity in overlap region
- maximum overhang size.

[How do we compute the overlap?]

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



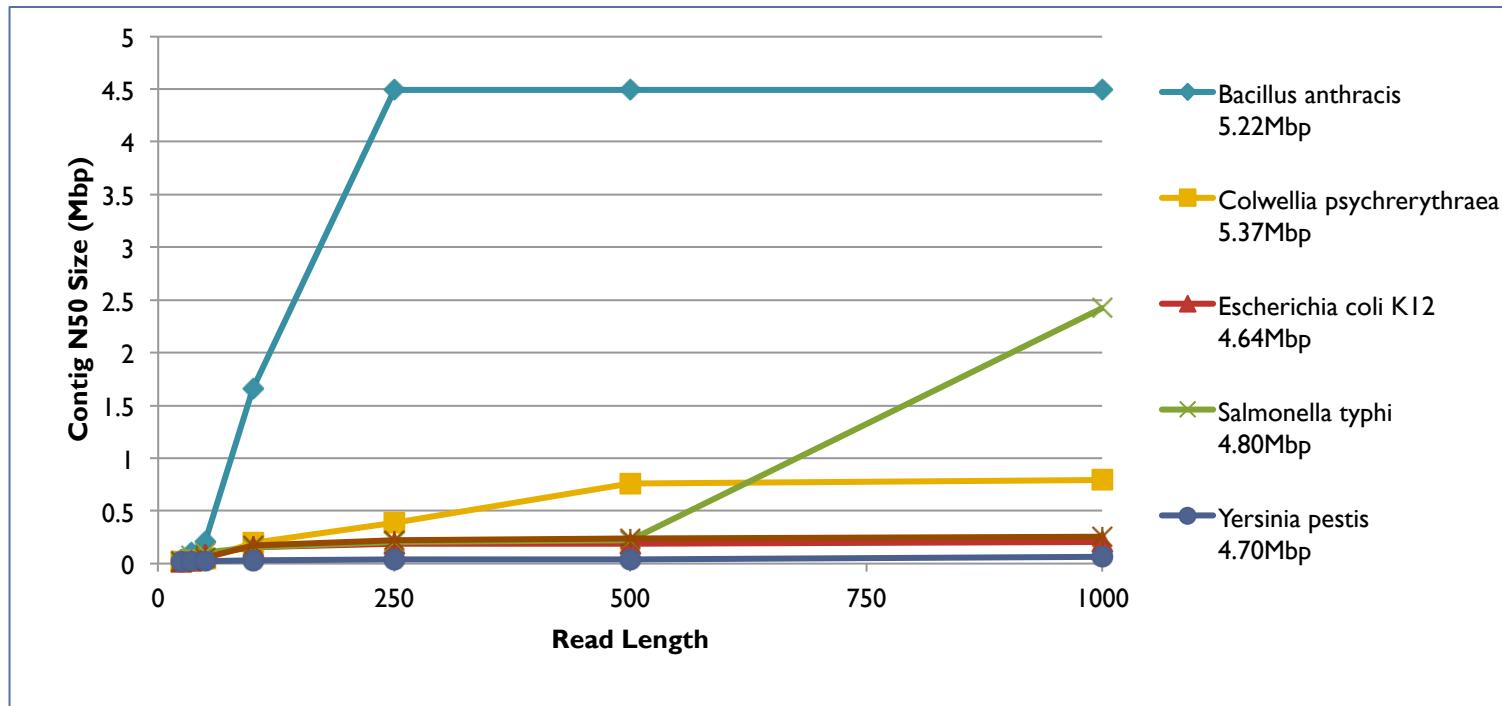
Errors in the graph



(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> |

Repeats and Read Length



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

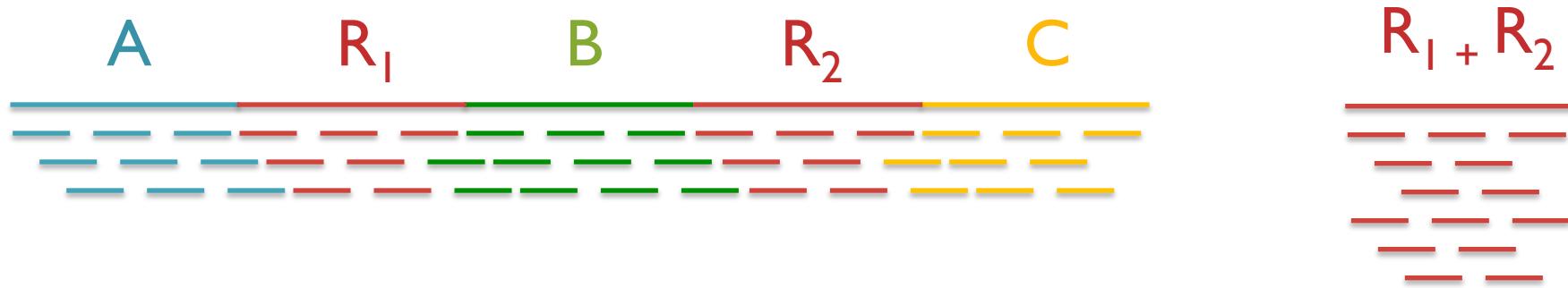
Assembly Complexity of Prokaryotic Genomes using Short Reads.
Kingsford C, Schatz MC, Pop M (2010) *BMC Bioinformatics*. 11:21.

Repetitive regions

| 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

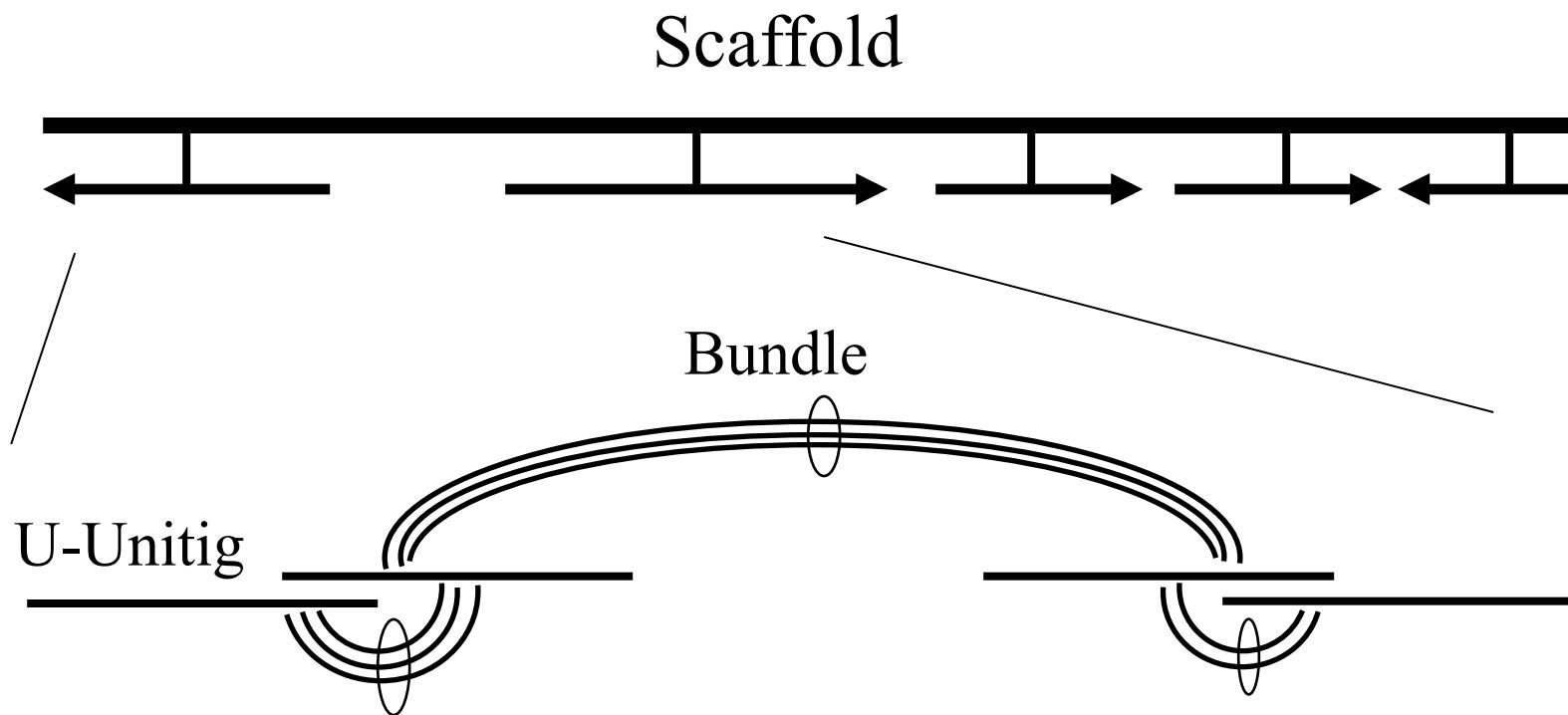


- If n reads are a uniform random sample of the genome of length G , we expect $k = n \Delta/G$ reads to start in a region of length Δ .
 - If we see many more reads than k (if the arrival rate is $> A$) , it is likely to be a collapsed repeat
 - Requires an accurate genome size estimate

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

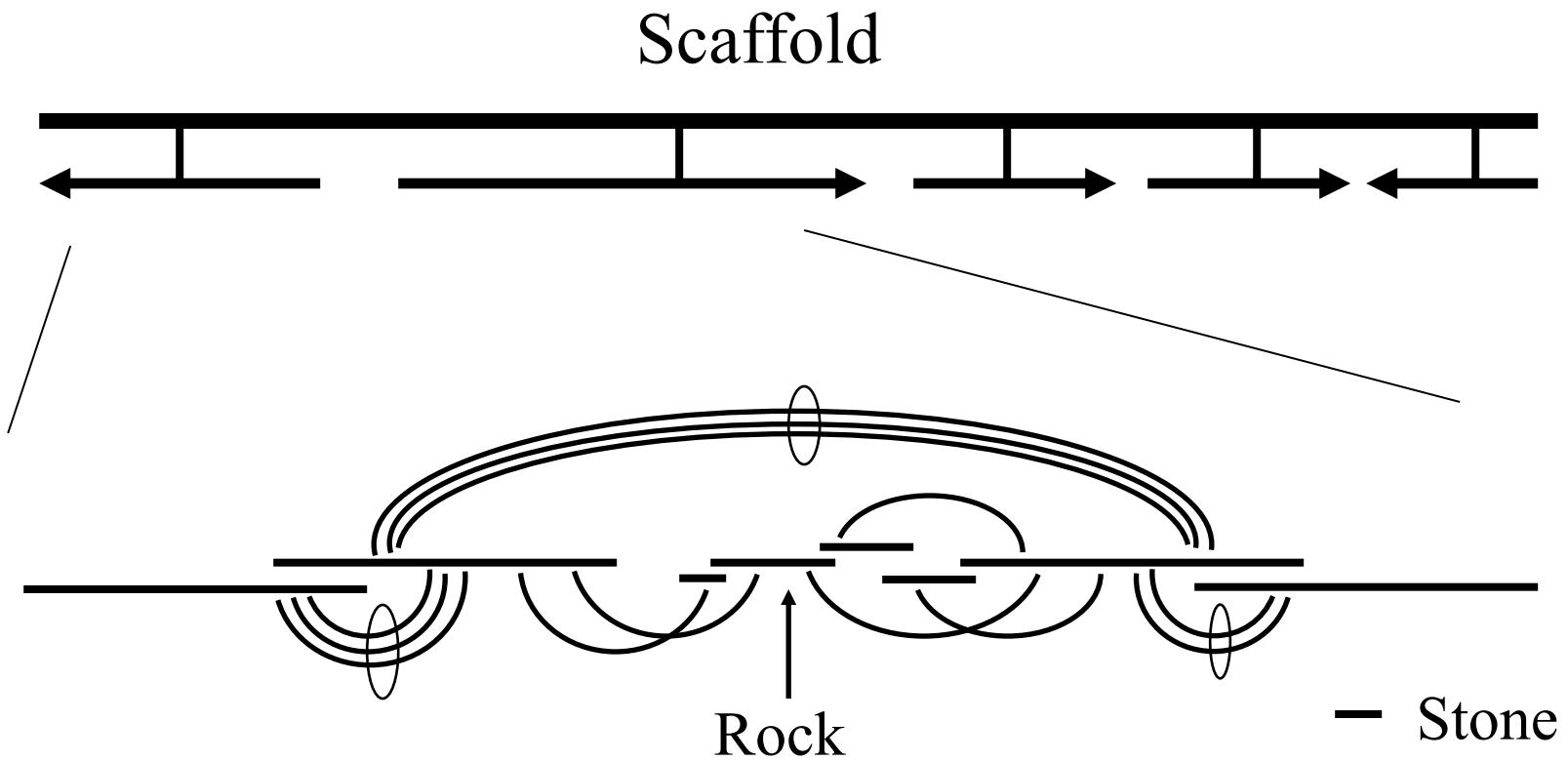
$$A(\Delta, k) = \ln \left(\frac{\Pr(1 - \text{copy})}{\Pr(2 - \text{copy})} \right) = \ln \left(\frac{\frac{(\Delta n / G)^k e^{-\Delta n}}{k!}}{\frac{(2\Delta n / G)^k e^{-2\Delta n}}{k!}} \right) = \frac{n\Delta}{G} - k \ln 2$$

Initial Scaffolding



Create a initial scaffold of unique unitigs (U-Unitigs) whose A-stat > 5. Also recruit borderline unitigs whose A-stat is > 2 and have consistent mates with the U-Unitigs.

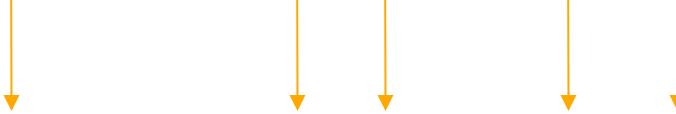
Repeat Resolution



Place rocks ($A\text{-stat} > 0$ with multiple consistent mates), and stones (single mate and overlap path with placed objects) into the gaps. Pebbles, unitigs lacking mates, are no longer incorporated regardless of overlap qualities.

Derive Consensus Sequence

TAGATTACACAGATTACTGA TTGATGGCGTAA CTA
TAGATTACACAGATTACTGACTTGATGGCGTAAACTA
TAG TTACACAGATTATGACTT**CAT**GGCGTAA CTA
TAGATTACACAGATTACTGACTTGATGGCGTAA CTA
TAGATTACACAGATTACTGACTTGATGGGTAA CTA



TAGATTACACAGATTACTGACTTGATGGCGTAA CTA

Derive **multiple alignment** from pairwise read alignments

Derive each consensus base by weighted voting

N50 size

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

Example: 1 Mbp genome



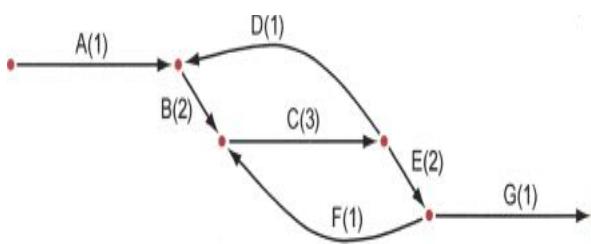
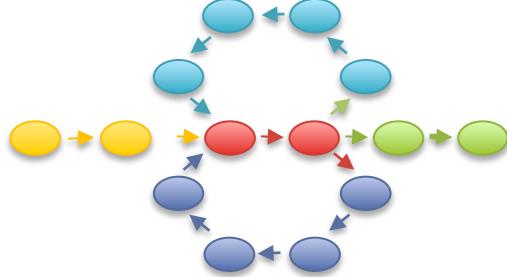
N50 size = 30 kbp

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

Note:

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

Assembly Algorithms

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

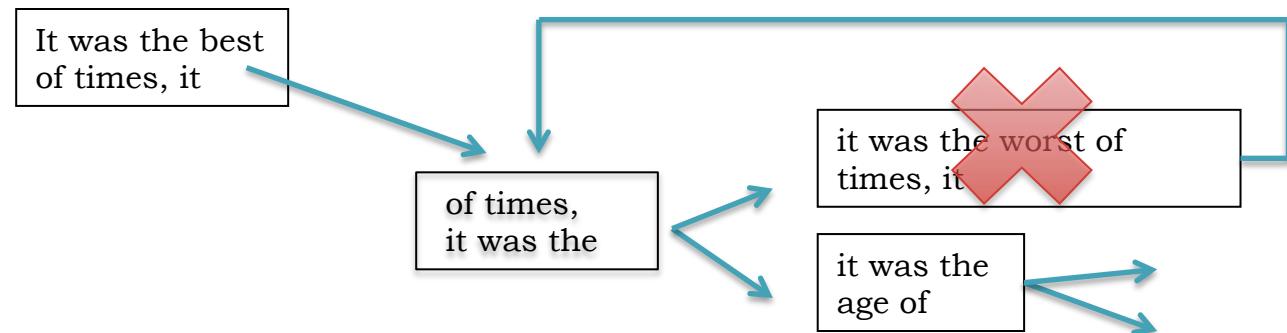
Assembly Validation



Automatically scan an assembly to locate misassembly signatures for further analysis and correction

Assembly-validation pipeline

1. Evaluate Mate Pairs & Libraries
2. Evaluate Read Alignments
3. Evaluate Read Breakpoints
4. Analyze Depth of Coverage

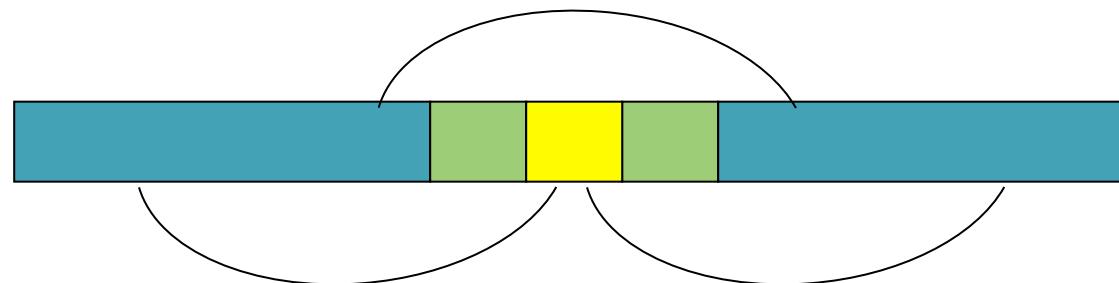


Genome Assembly forensics: finding the elusive mis-assembly.
Phillippy, AM, Schatz, MC, Pop, M. (2008) *Genome Biology* 9:R55.

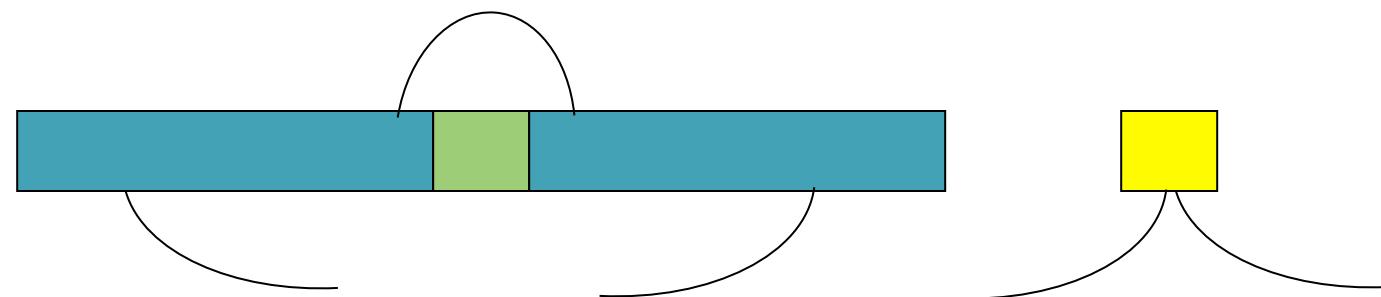
Mate-Happiness: asmQC

- Excision: Skip reads between flanking repeats

- Truth



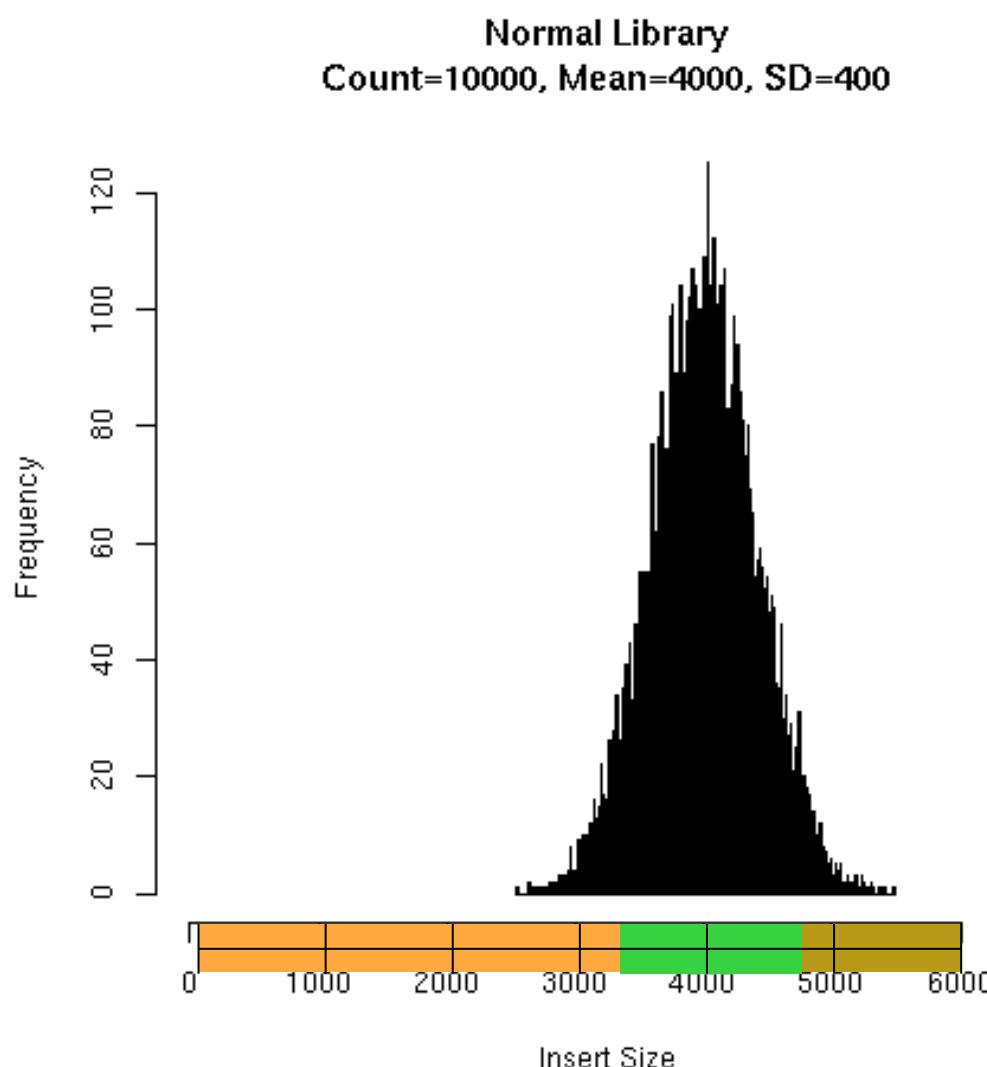
- Misassembly: Compressed Mates, Missing Mates



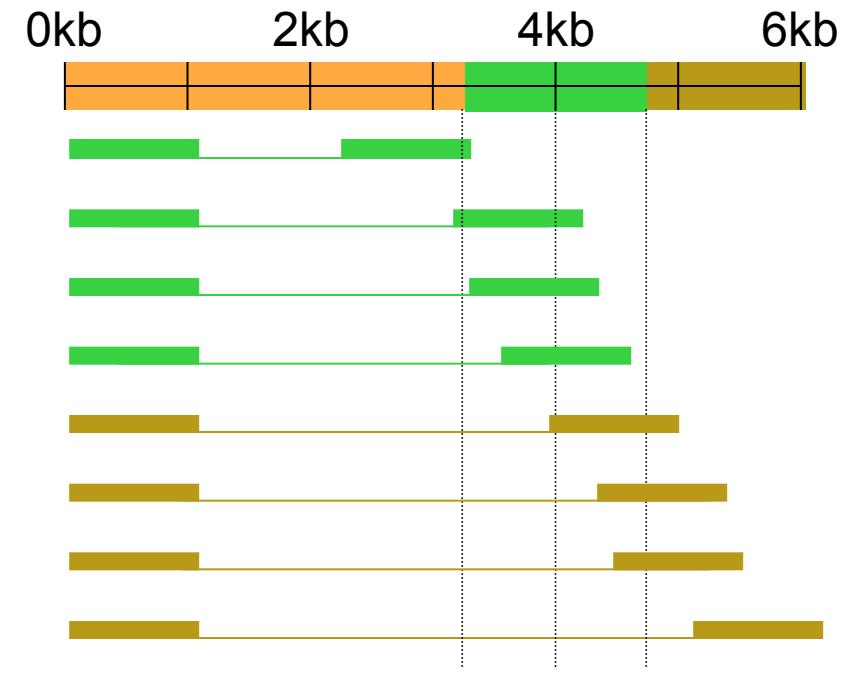
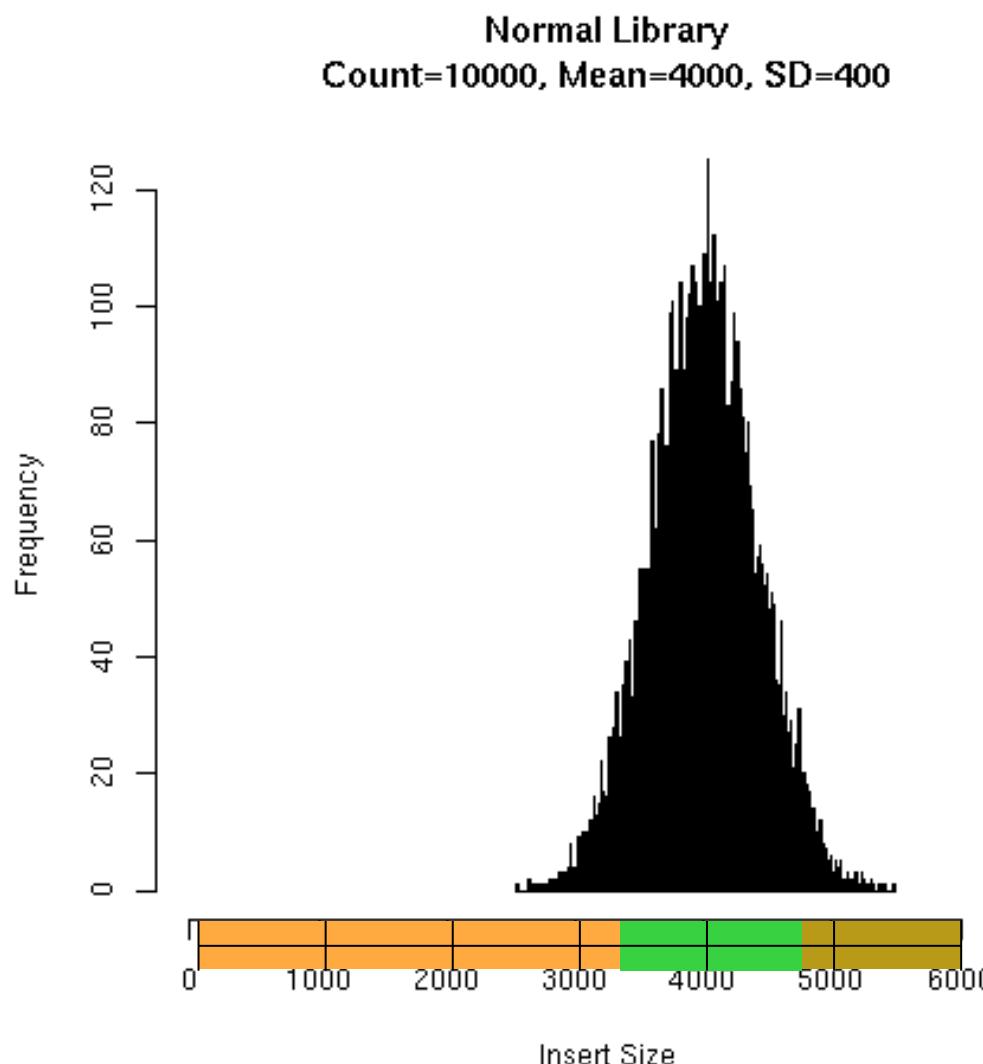
C/E Statistic

- The presence of individual compressed or expanded mates is rare but expected.
- Do the inserts spanning a given position differ from the rest of the library?
 - Flag large differences as potential misassemblies
 - Even if each individual mate is “happy”
- Compute the statistic at all positions
 - $(\text{Local Mean} - \text{Global Mean}) / \text{Scaling Factor}$
- Introduced by Jim Yorke’s group at UMD

Sampling the Genome



C/E-Statistic: Expansion



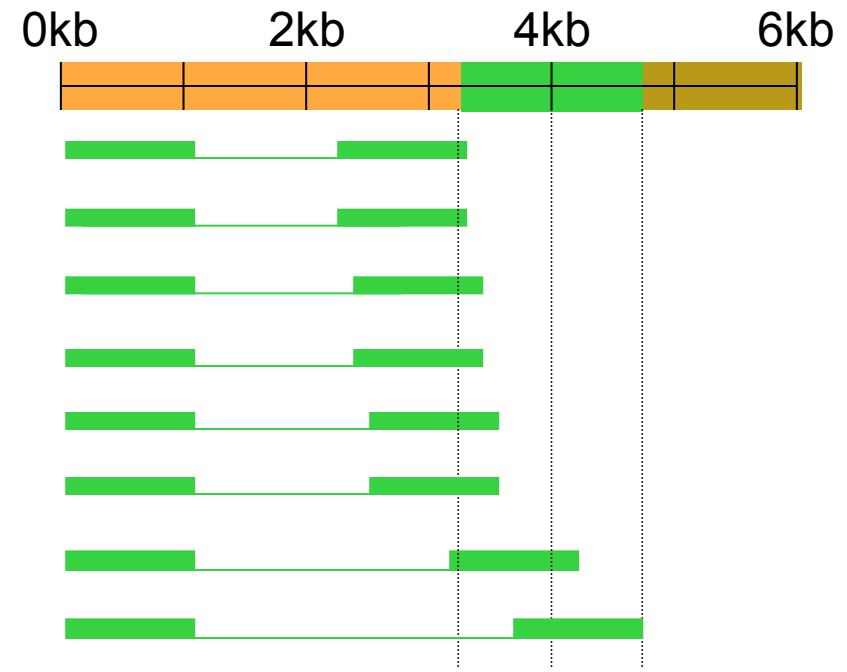
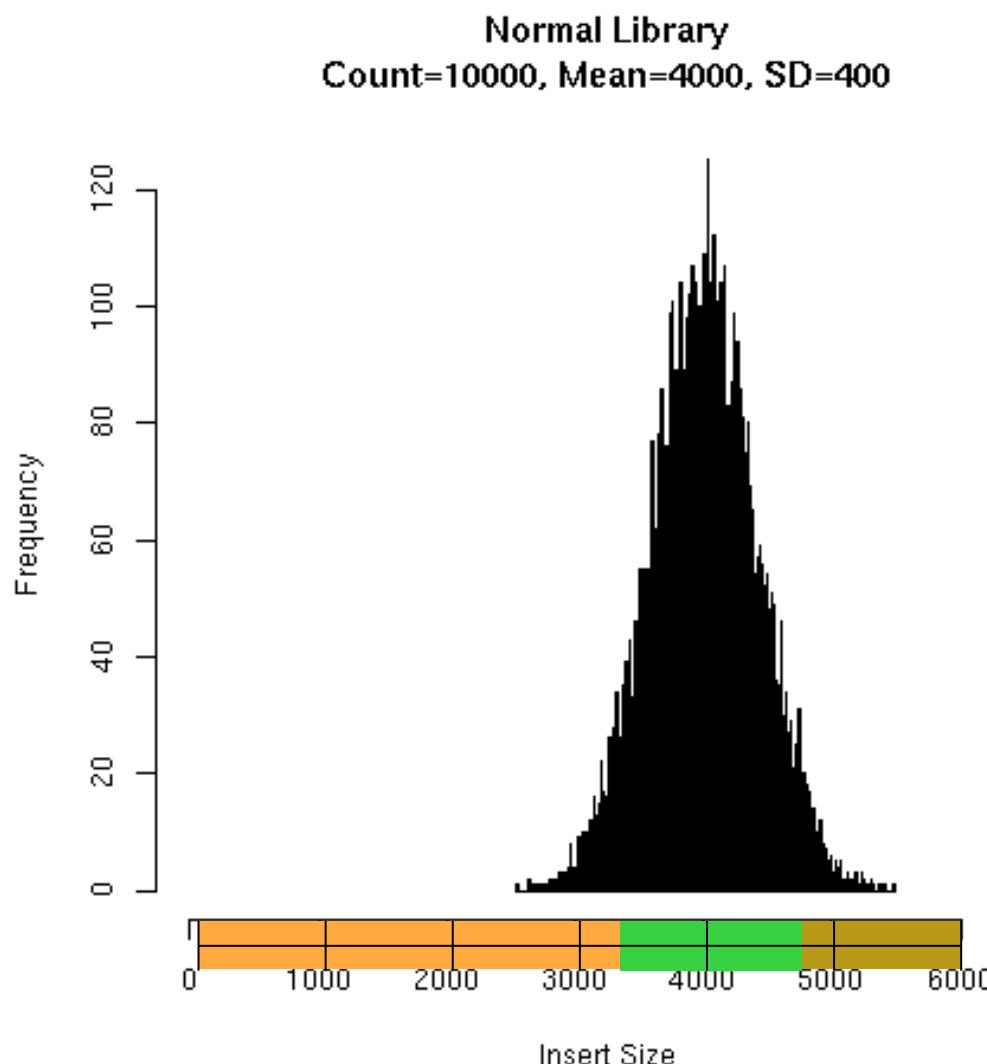
8 inserts: 3.2kb-6kb

Local Mean: 4461

$$\text{C/E Stat: } \frac{(4461 - 4000)}{(400 / \sqrt{8}}) = +3.26$$

C/E Stat ≥ 3.0 indicates Expansion

C/E-Statistic: Compression



8 inserts: 3.2 kb-4.8kb

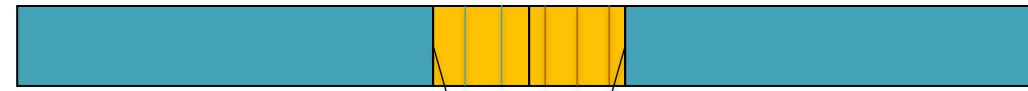
Local Mean: 3488

$$\text{C/E Stat: } \frac{(3488 - 4000)}{(400 / \sqrt{8}}) = -3.62$$

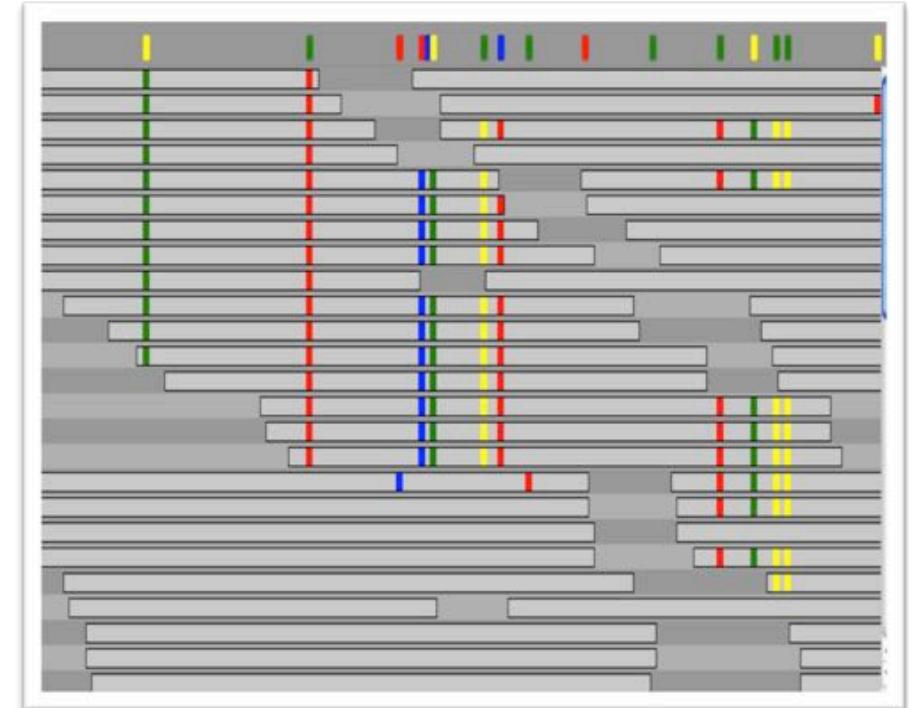
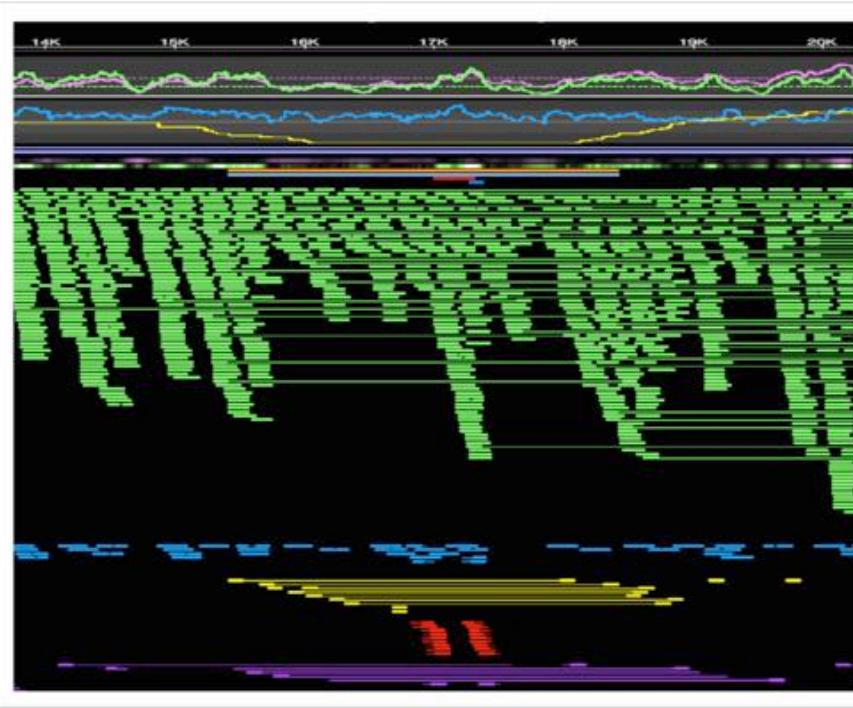
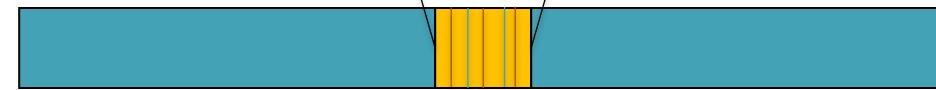
C/E Stat ≤ -3.0 indicates
Compression

Assembly Forensics

Truth:

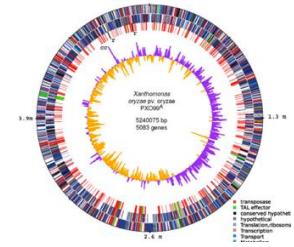


Mis-assembled:



Hawkeye & AMOS: Visualizing and assessing the quality of genome assemblies
Schatz, M.C. et al. (2011) *Briefings in Bioinformatics*. In Press.

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