Assembly Bootcamp

Michael Schatz & Adam Phillippy

June 26, 2009

Institute for Genome Sciences

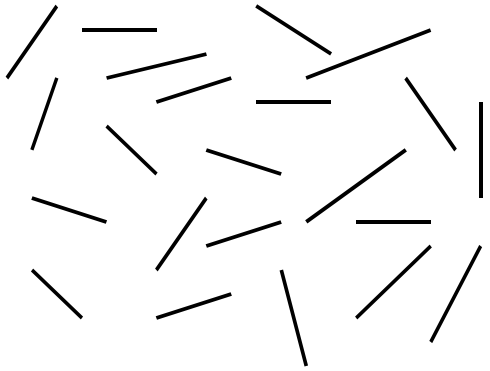
Outline

1. Assembly Theory and Practice
2. Assembly Validation & Comparison
3. Summary and Questions

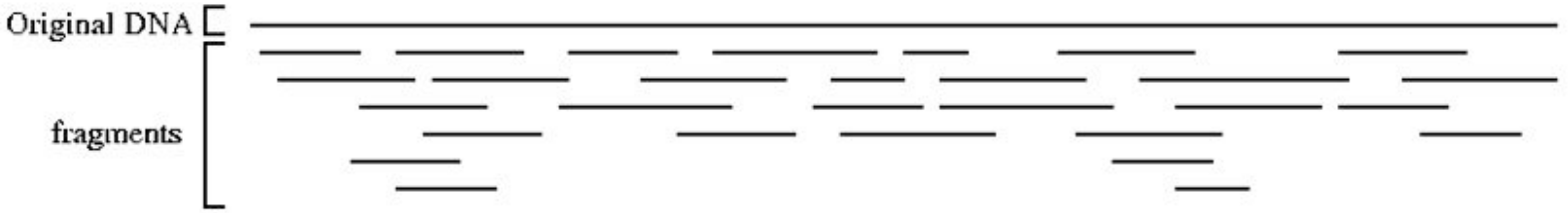
Sequencing a Genome

1. Collect DNA Sample

2. Break DNA into short random fragments & sequence



3. Assemble short sequences back into the genome

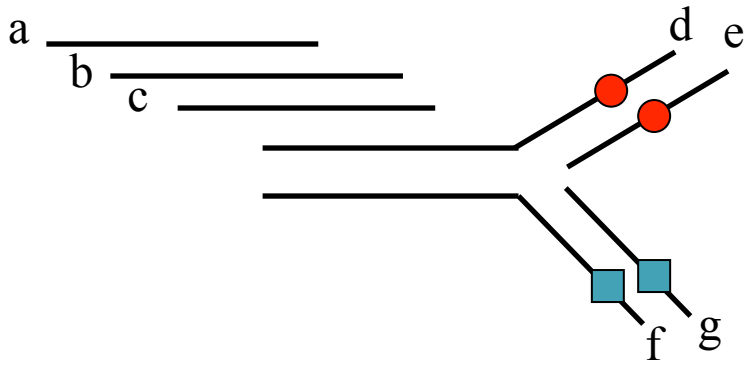


Assembly Overview

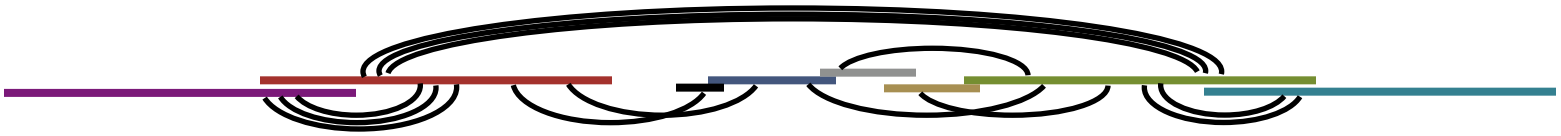
1. Compute overlaps between sequences

...AGCCTAGACCTACAGGATGCGCGGACACGT
GGATGCGCCGACACGTAGCTTATCCGGT...

2. Compute unambiguous regions of genome



3. Disambiguate with mates, markers, and other information

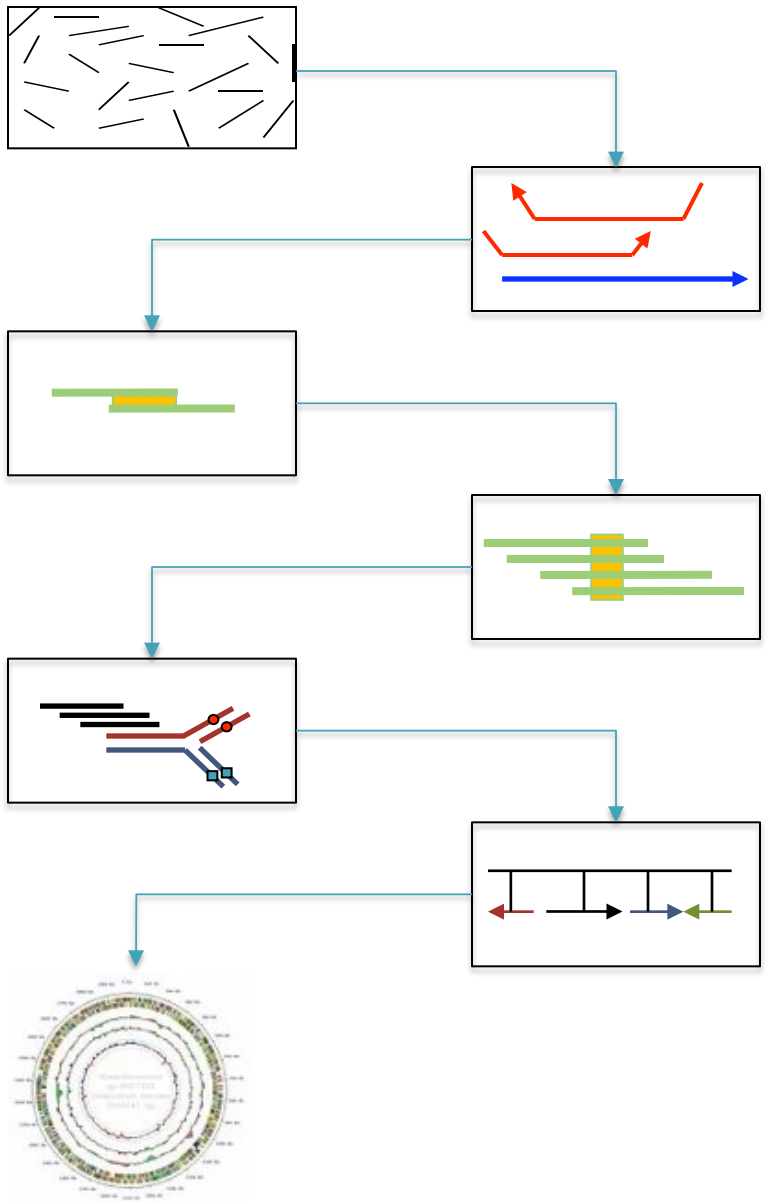


Running Celera Assembler

- Download release
  - http://wgs-assembler.sourceforge.net
- Prepare input frg files
  - 454 reads: sffToCA [–clear chop] [-linker name]
  - Sanger reads: tracedb-to-frg.pl
  - Solexa/ABI Solid: Unsupported* / Shred .ace files
- Launch Assembler
  - runCA –d <dir> -p <prefix> [options] reads.frg

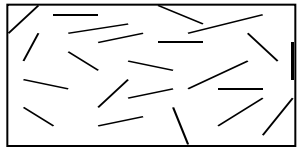
runCA Pipeline

1. Pre-overlap
   - gatekeeper
2. Trimming
   - Vector trimming & partial overlaps
3. Compute Overlaps
   - overlap, overlapStore
4. Error Correction
   - correct-frags, correct-olaps
5. Unitigging
   - unitigger, consensus –U
6. Scaffolding
   - cgw, consensus
7. Finalize Data
   - terminator, asm, fasta, & qc files

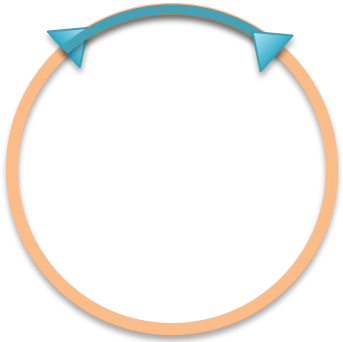


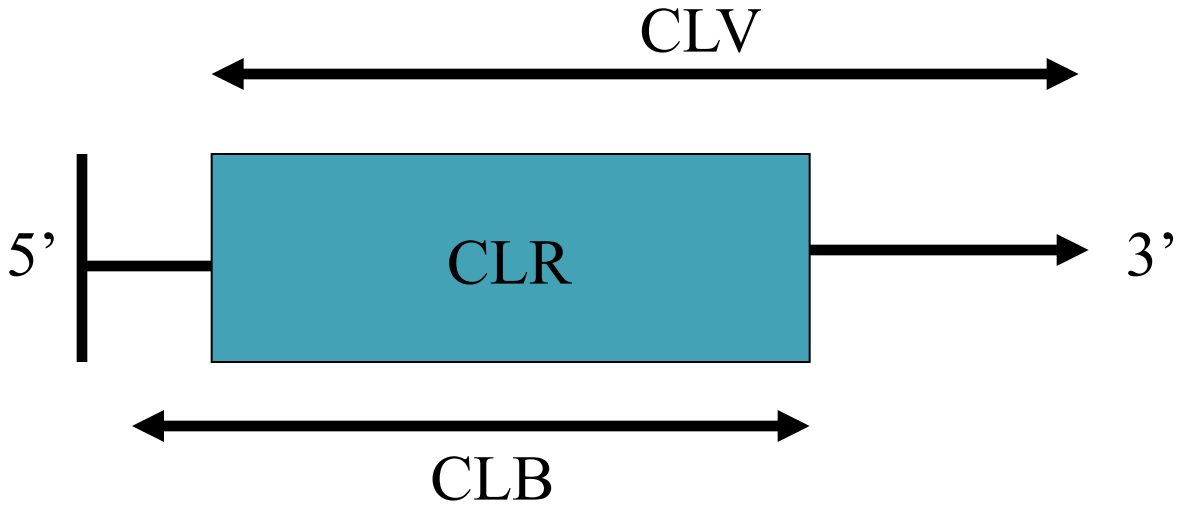
Gatekeeper

- Loads reads into gatekeeper store (gkpStore)
  - Sequences, qualities, clear ranges, libraries, ids
  - Binary database, not compatible across versions
- Initial sanity checks on data
  - Discards very short reads (< 64 bp)
- Dump reads, convert formats
  - gatekeeper -h



Trimming

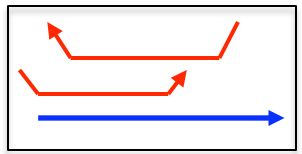




Identify the good region of each read (CLR)

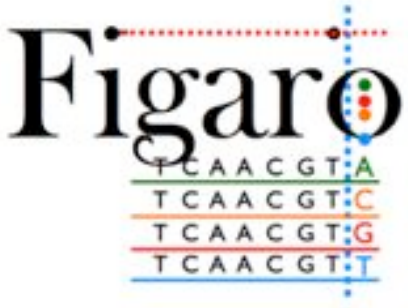
- Avoid vector sequence (CLV)
- Avoid low quality sequence (CLB)

*CLR is revised several times during assembly



Vector Trimming

- Figaro / CA vector screening
  - Identify kmers overrepresented at the beginning of reads



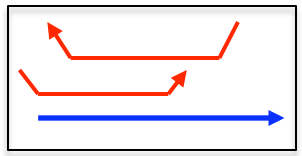
(White JR et al., 2008)

- Lucy
  - Compare each read to known vector sequence
  - Also scans for low quality values

(Li S and Chou H, 2004)

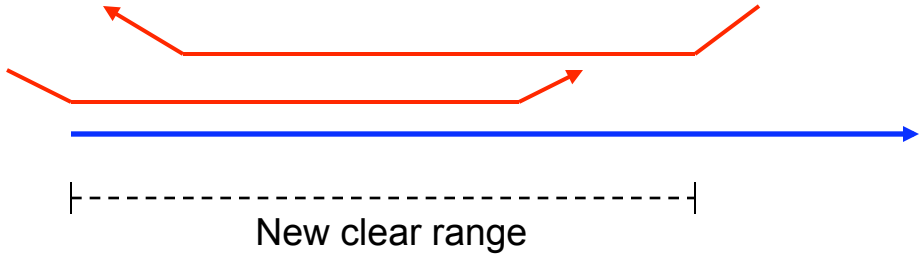
Note:

454 reads are generally vector free, but it is critical to identify the linker sequence in paired-end reads

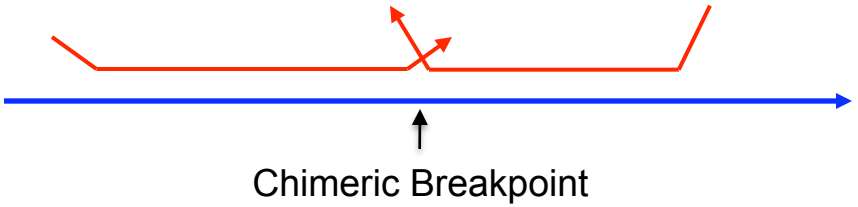


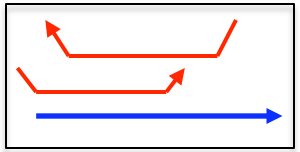
Overlap Based Trimming (OBT)

- Compute local alignments ("partial overlaps") between untrimmed reads.
  - Use confirmed alignment regions to set new clear range.



- Overlap forks indicate chimeric reads



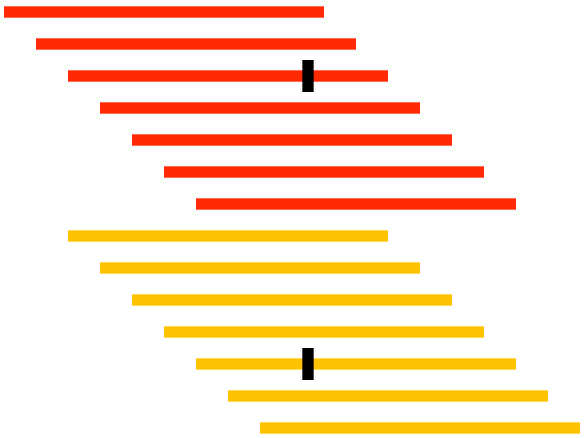


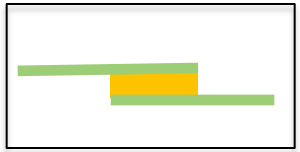
Overlapper

- Find all overlaps ≥ 40bp allowing 6% mismatch.
  - Use kmer (k=22) seed matches, skipping high frequency kmers
  - New support for masking homopolymer errors
    - runCA merCompression=X

- Warning:

  Sequencing error can accidentally cause low copy number seeds in high copy repeat regions, ovlMerThreshold=X





Error Correction

- For each read, construct a multiple alignment of the overlapping reads.
- If a discrepant base is not observed in the overlapping reads, consider it a sequencing error.
- Sequences are not actually changed, only overlaps are re-evaluated as single base pair errors are "corrected".

Before:

ACGTTCACGTATGGACAC

ACGTACACGTATGGACAC

ACGTACACGTATG–ACAC

ACGTACACGTATG–ACAC

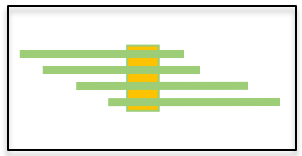
After:

ACGTACACGTATGGACAC

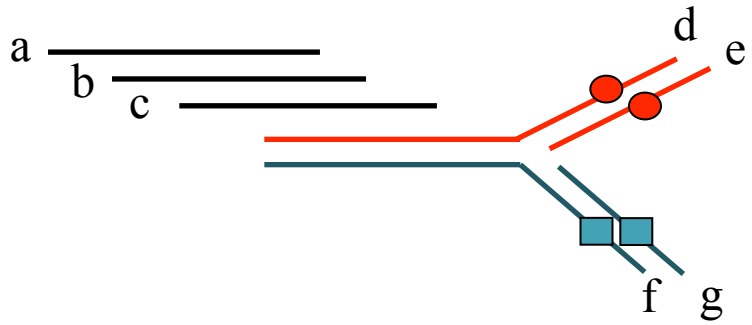
ACGTACACGTATGGACAC

ACGTACACGTATG–ACAC

ACGTACACGTATG–ACAC



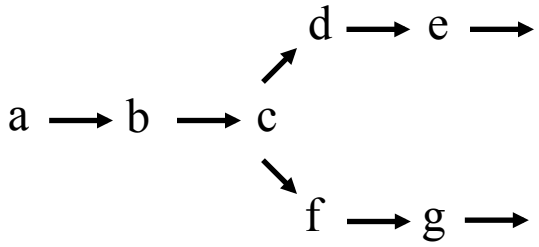
Unitigging



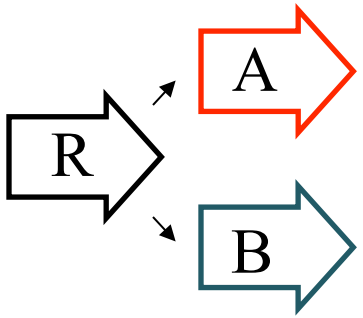
Overlaps

```
a b 50
a c 100
a d 150
a f 150
b c 50
b d 100
b f 100
c d 50
c f 50
d e 50
f g 50
```

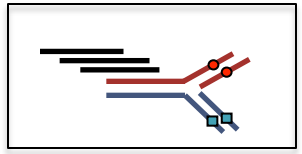
Best Buddy Graph



Unitig Graph

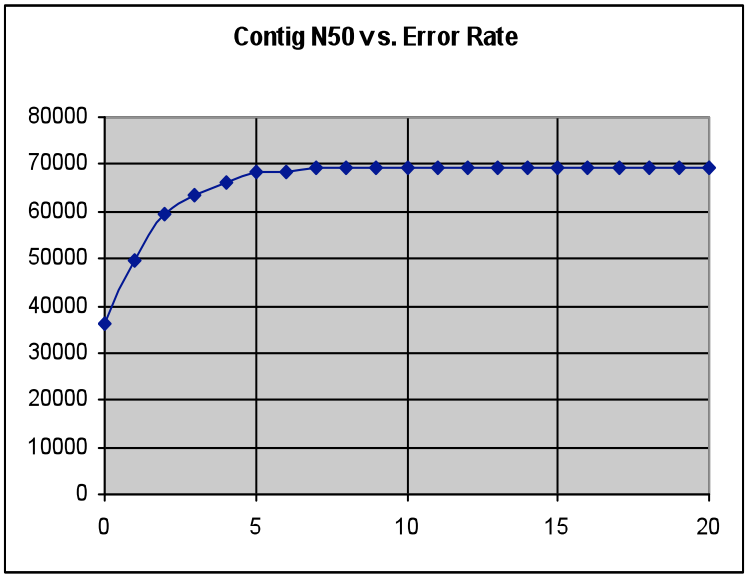


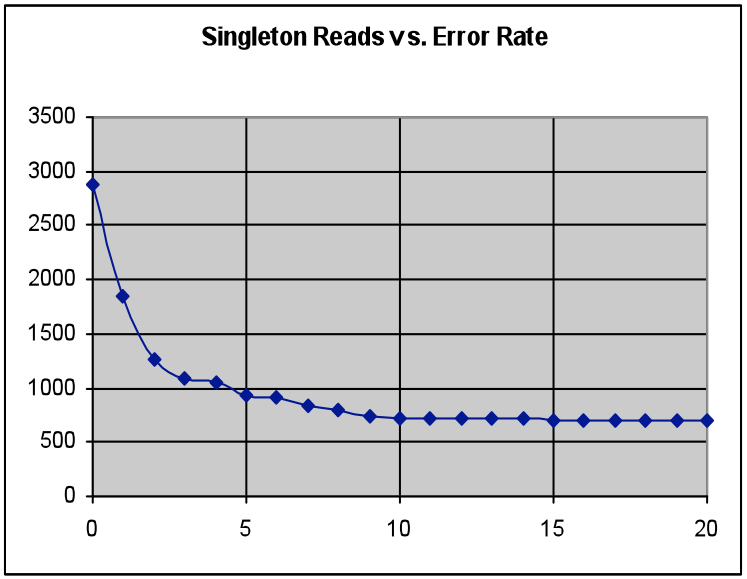
C overlaps D & F, but D & F don't overlap



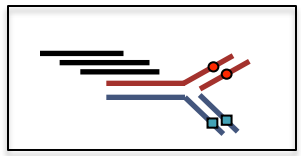
Sequencing Error Effect

runCA utgErrorRate=X

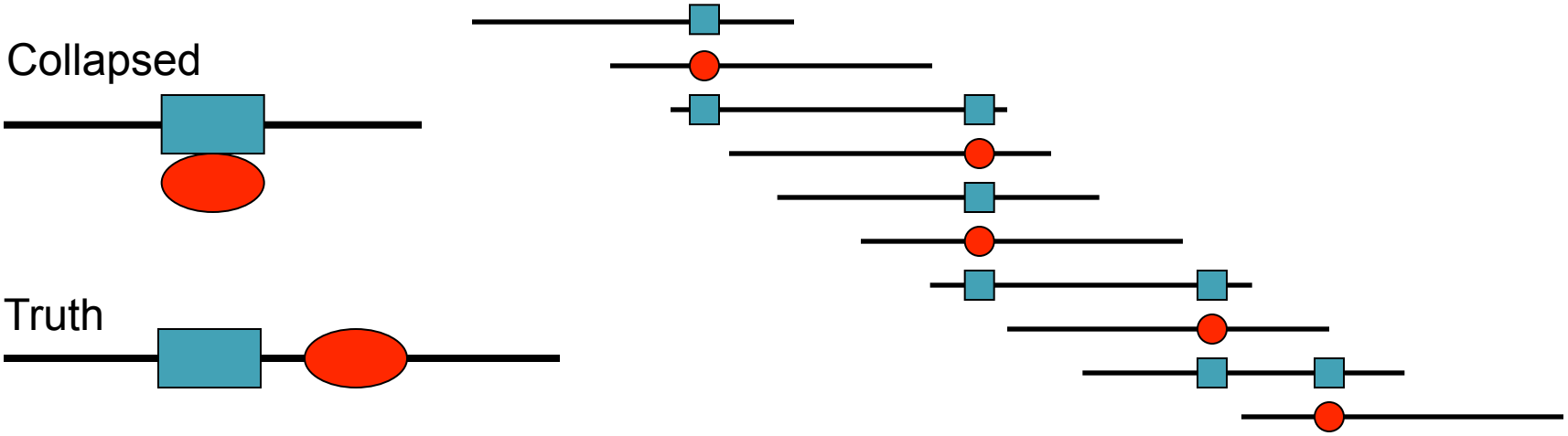




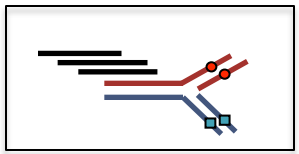
In general, contigs get larger and more reads are placed as the error rate threshold is increased.



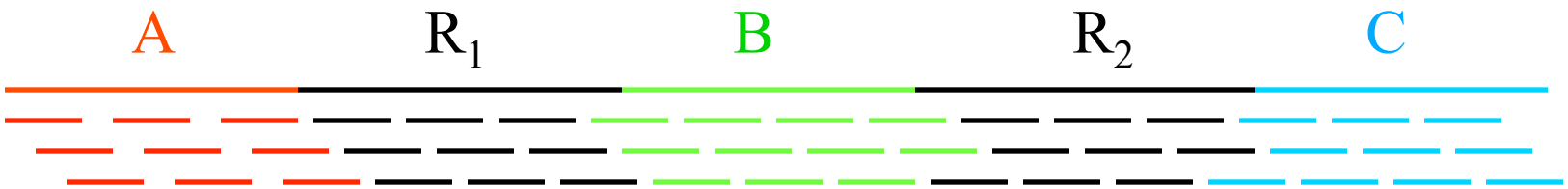
Collapsing Repeats

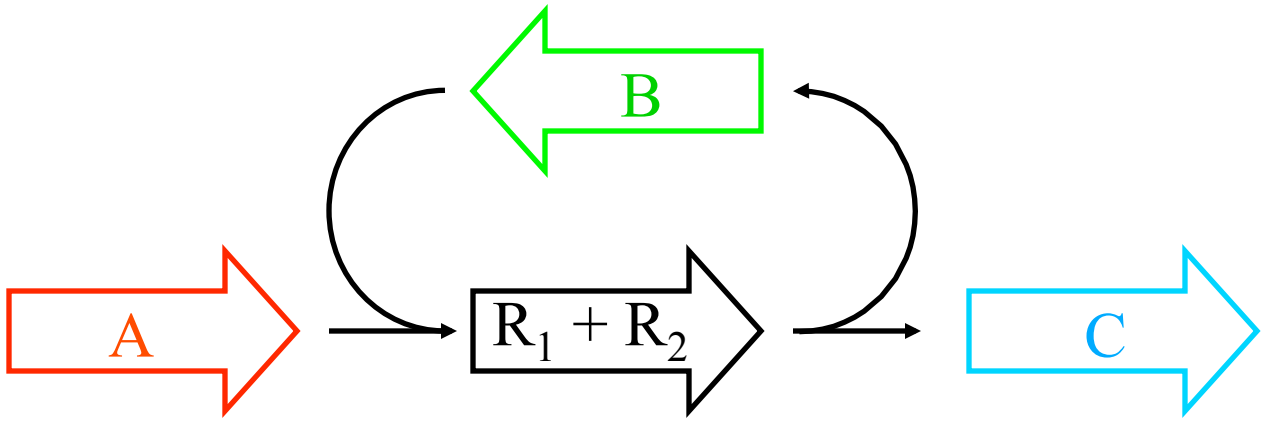


- Genome is mis-assembled as the unitigger becomes less sensitive to slight differences between repeats.
- Best practice is to try many error rates and compare assemblies

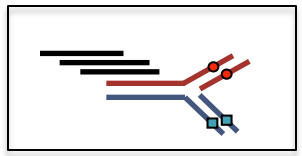


Unitig Scoring

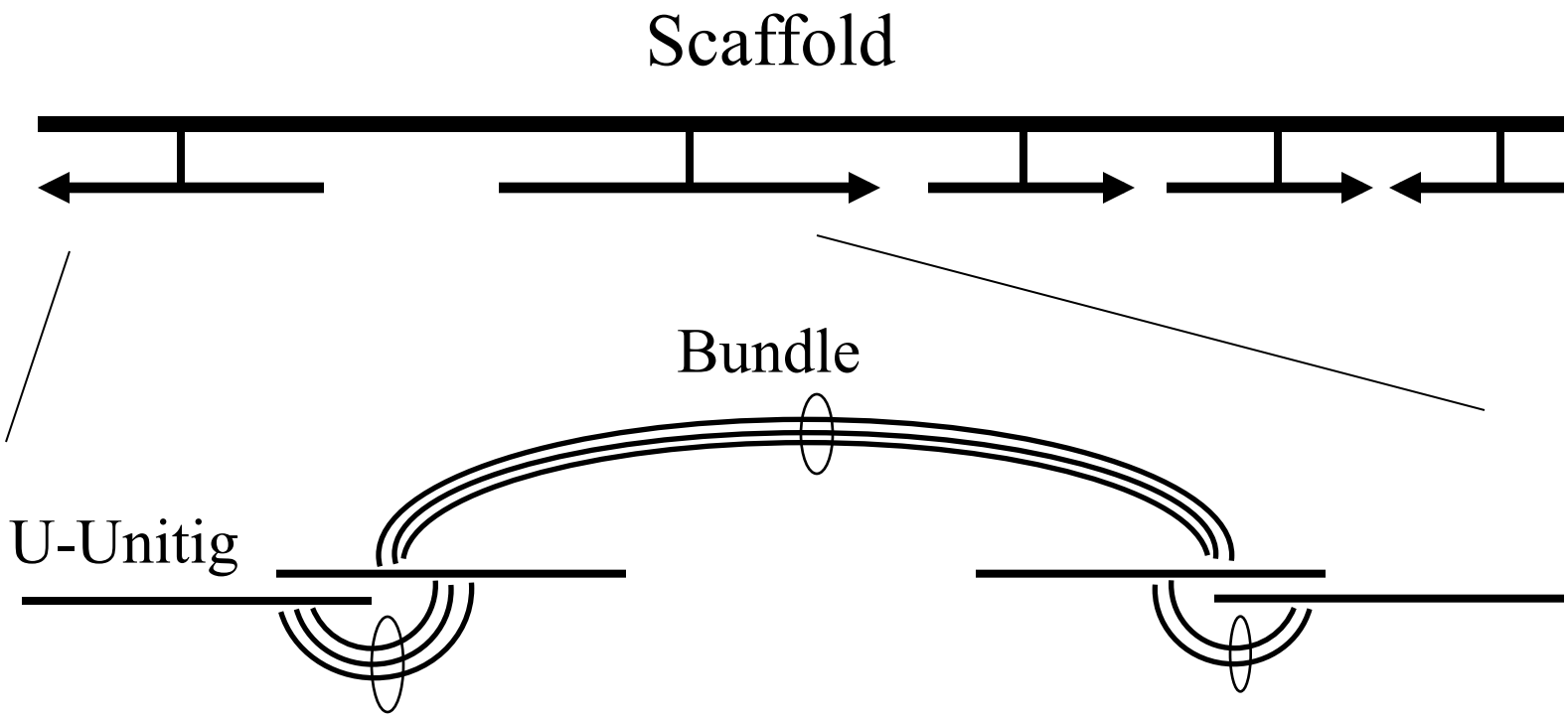




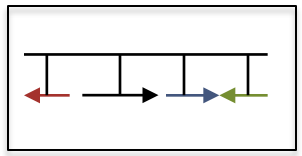
- The arrival rate statistic (A-stat) is the log-odds ratio of the probability the sequence is unique vs a 2 copy repeat.
  - Positive a-stat indicates unique unitigs (A,B,C)
  - Negative a-stat indicate repeats (R).



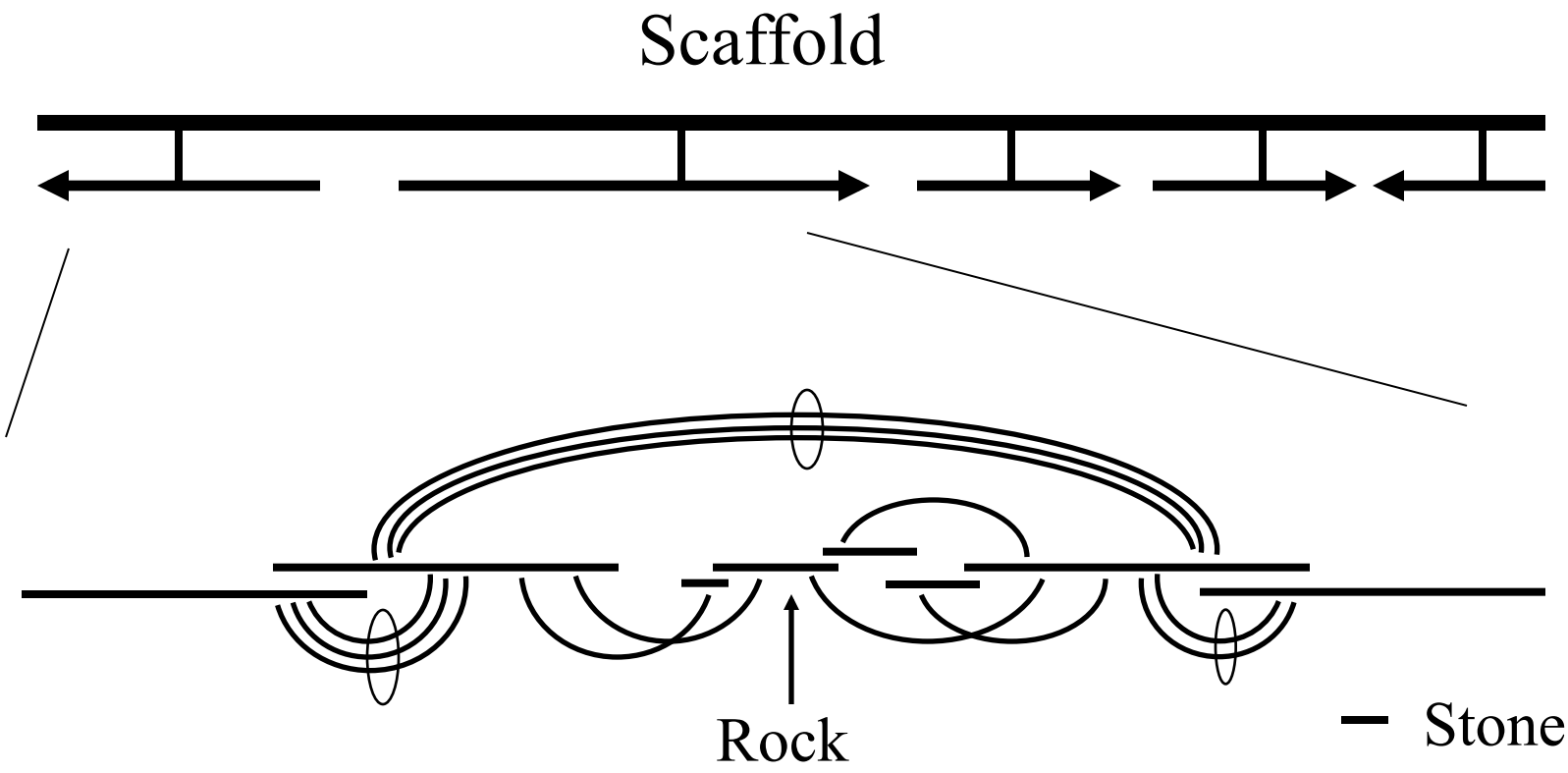
Initial Scaffolding



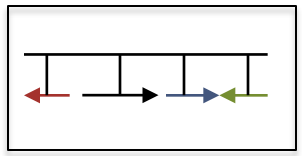
- Create initial scaffold of unique unitigs (U-Unitigs) with A-stat > 5.
- Also recruit borderline unitigs with A-stat is > 2 and have consistent mates with the U-Unitigs.



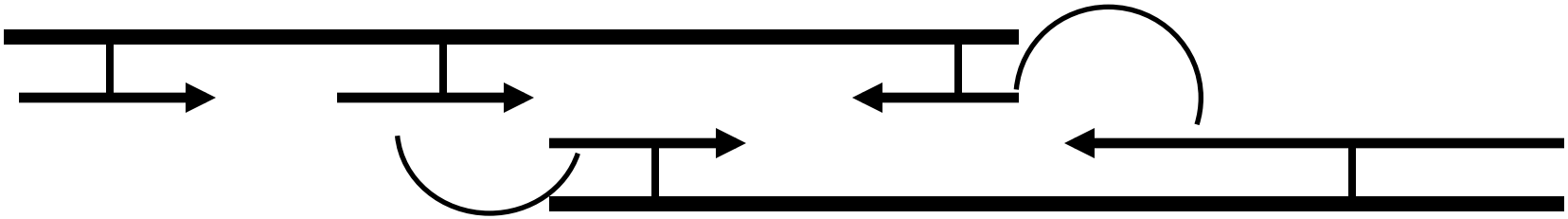
Repeat Resolution



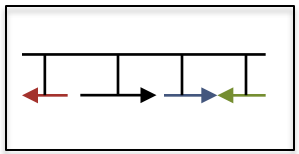
Use mates pairs to place rocks (A-stat > 0 with multiple consistent mates), and stones (single mate and overlap with placed objects) into the gaps.



Scaffold merging



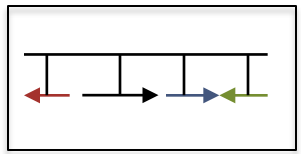
- After placing borderline unitigs and rocks, there may be new mates to merge scaffolds.
- This in turn may allow for new rocks and stones to be placed, so iterate these steps until the scaffolds stabilize.



Assembly Dregs

- Degenerate unitigs are unitigs with poor A-stat and not in any scaffold as a rock or stone.
  - In unpaired & metagenomic data, most of your contigs will be degenerate
  - Use runCA astat*Bound=X or runCA utgGenomeSize=X to promote borderline unitigs

- Non-unique surrogate unitigs are unitigs incorporated as stones in multiple places in the scaffold.
  - Consequently, their reads can be multiply placed.



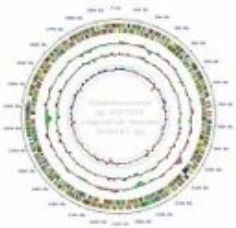
Assembler outputs

asmbl.asm - all the information in Celera message format
asmbl.qc - summary statistics

asmbl.ctg.fasta - all the contigs
asmbl.deg.fasta - all the degenerates
asmbl.singleton.fasta - all the singletons
asmbl.scf.fasta - all the scaffolds, 60 Ns replace the gaps

For unpaired reads, asmbl.utg.fasta: all unitigs

runCA –createAGP: creates agp file
runCA –createACE: creates ace file



Assembler Summary

- Overall assembly algorithm: bottom up assembly
  - Aggressively trim & overlap to build reliable unitigs
  - Use mates to resolve ambiguities and build scaffolds

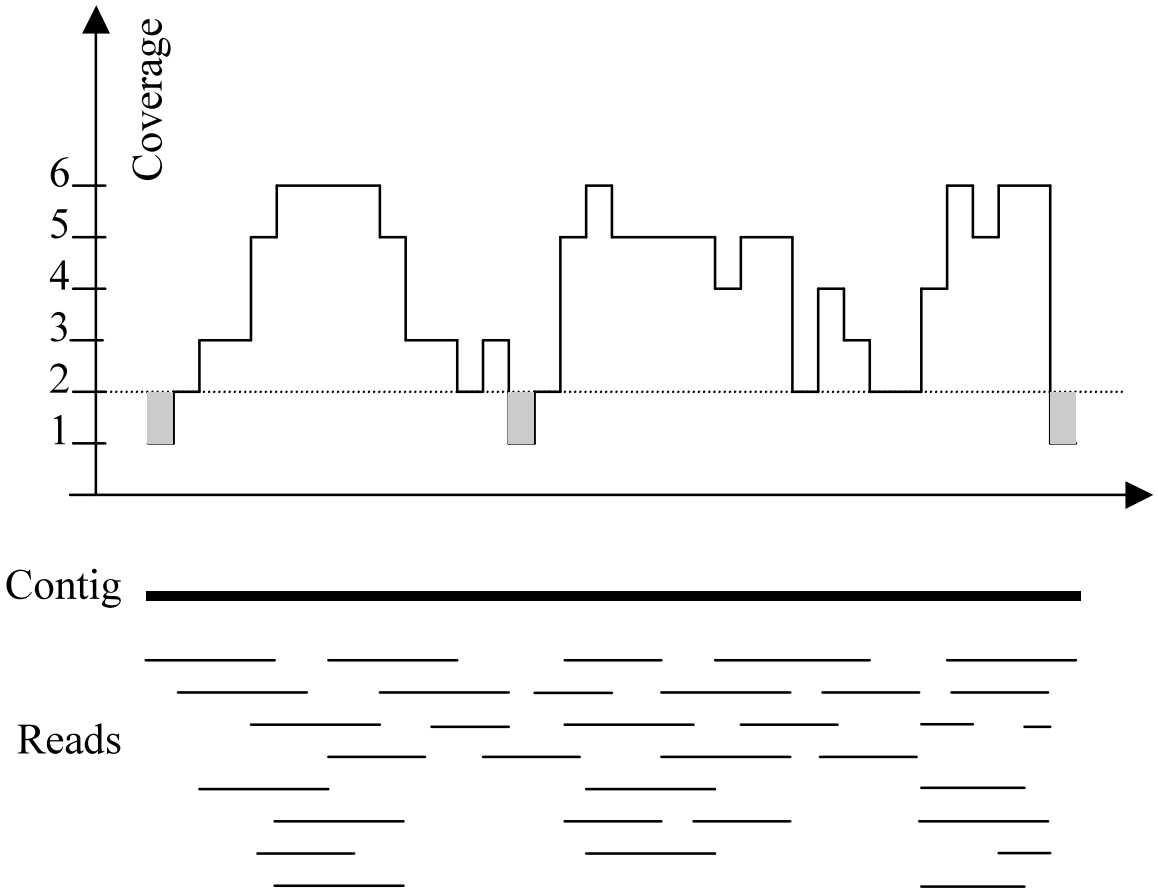
- The assembler is sensitive to data characteristics
  - Trimming, sequencing error, coverage levels
  - You'll get the best assembly if you tune the assembler
    - merCompression: allow homopolymer errors
    - ovlMerThreshold: repeat screening in overlaps
    - utgErrorRate: overlaps used for unitigging
    - utgGenomeSize, astat*Bound: repeat screening for scaffolding
    - *Concurrency: assembler performance

Resources

- CA Wiki
  - http://wgs-assembler.sourceforge.net
- AMOS
  - http://amos.sourceforge.net
- Assembly Primer
  - http://www.cbcb.umd.edu/research/assembly.shtml

Thank You!

How much to sequence?



Imagine raindrops on a sidewalk

Lander Waterman Statistics

L = read length

T = minimum overlap

G = genome size

N = number of reads

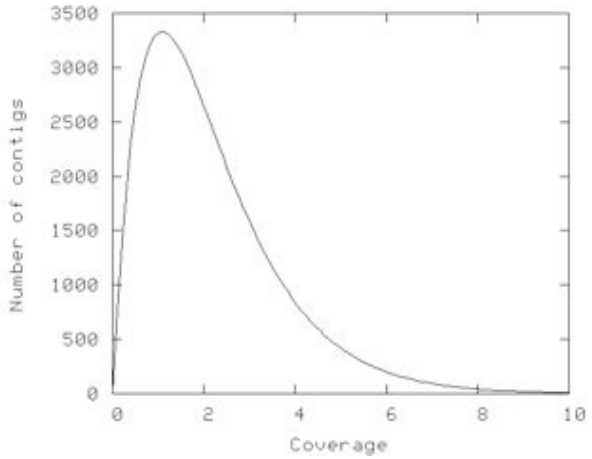
c = coverage (NL / G)

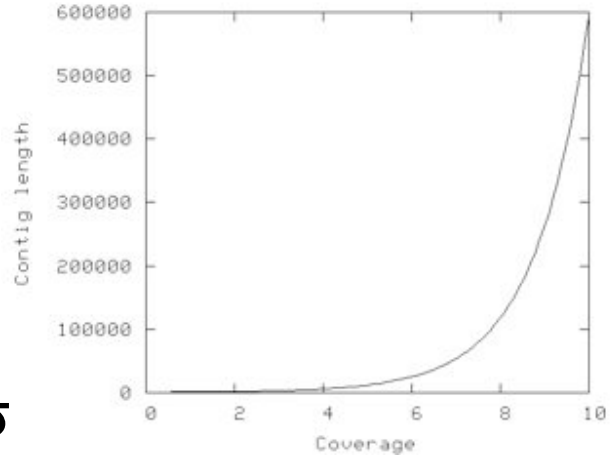
$\sigma = 1 - T/L$

$E(\#islands) = Ne^{-c\sigma}$

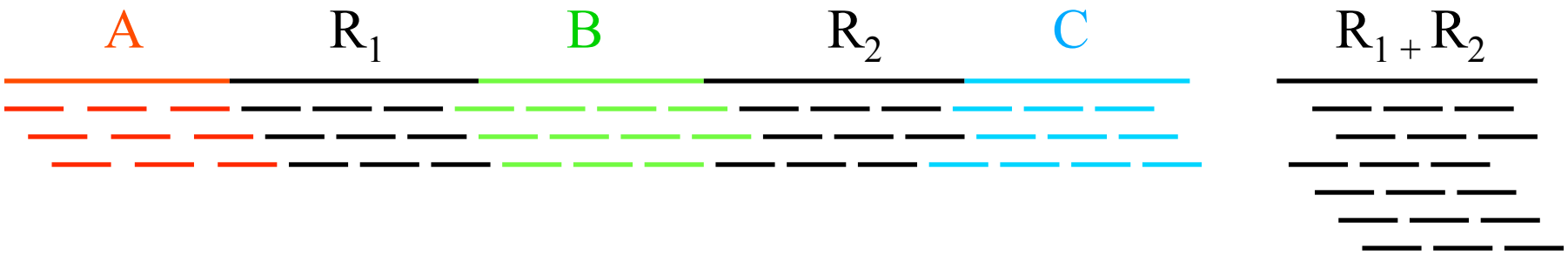
$E(island\ size) = L(e^{c\sigma} - 1) / c + 1 - \sigma$

contig = island with 2 or more reads





A-stat



- 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

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

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

The .qc file

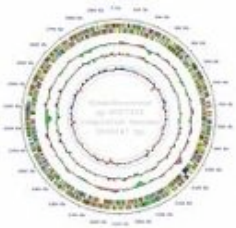
```
[Scaffolds]
TotalScaffolds=2
MeanContigsPerScaffold=23.50
MaxContigsPerScaffold=30

TotalBasesInScaffolds=3298141
MeanBasesInScaffolds=1649070.50
MaxBasesInScaffolds=2100614
N50ScaffoldBases=2100614

TotalSpanOfScaffolds=3310522
MeanSpanOfScaffolds=1655261.00
MaxScaffoldSpan=2104833
IntraScaffoldGaps=45
MeanSequenceGapSize=275.13

[Top_5_Scaffolds_contigs_size_span_avgContig_avgGap]
0=30 2100614 2104833 70020.47 145.48
```

http://www.cbcb.umd.edu/research/castats.shtml



N50 size

50% of genome is in contigs larger than N50

Example:

1 Mbp genome
Contigs: 300, 100, 50, 45, 30, 20, 15, 15, 10, ...
N50 size = 30 kbp
(300+100+50+45+30 = 525 >= 500kbp)

Note:

N50 is meaningful for comparison only when genome size is the same