De novo Assembly

Titus Brown

6/9/11

Assembly vs mapping

- No reference needed, for assembly!
  - De novo genomes, transcriptomes…
- But:
  - Scales poorly; need a much bigger computer.
  - Biology gets in the way (repeats!)
  - Need higher coverage
- But but:
  - Often your reference isn't that great, so assembly may actually be the best way to go.

Assembly

It was the best of times, it was the wor

, it was the worst of times, it was the

isdom, it was the age of foolishness

mes, it was the age of wisdom, it was th

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

…but for lots and lots of fragments!

Assemble based on word overlaps:

the quick brown fox jumped

jumped over the lazy dog

the quick brown fox jumped over the lazy dog

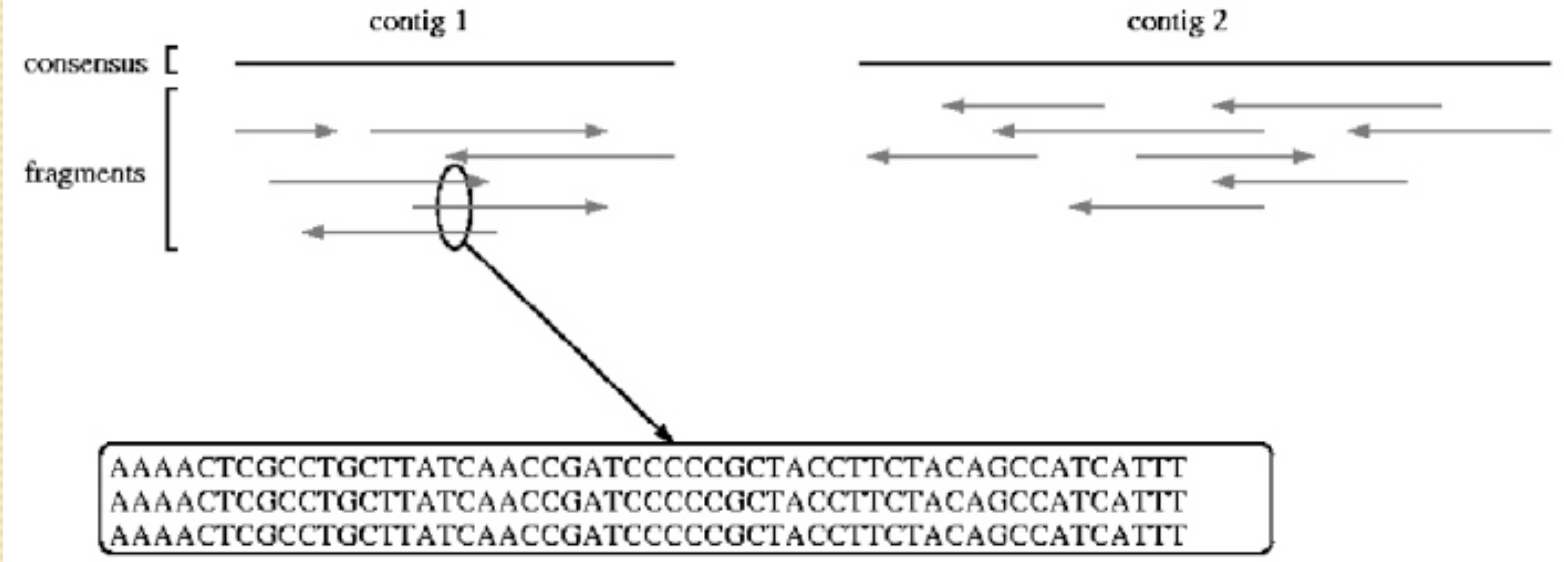
Repeats do cause problems:

my chemical romance: na na na

na na na, batman!

Shotgun sequencing & assembly

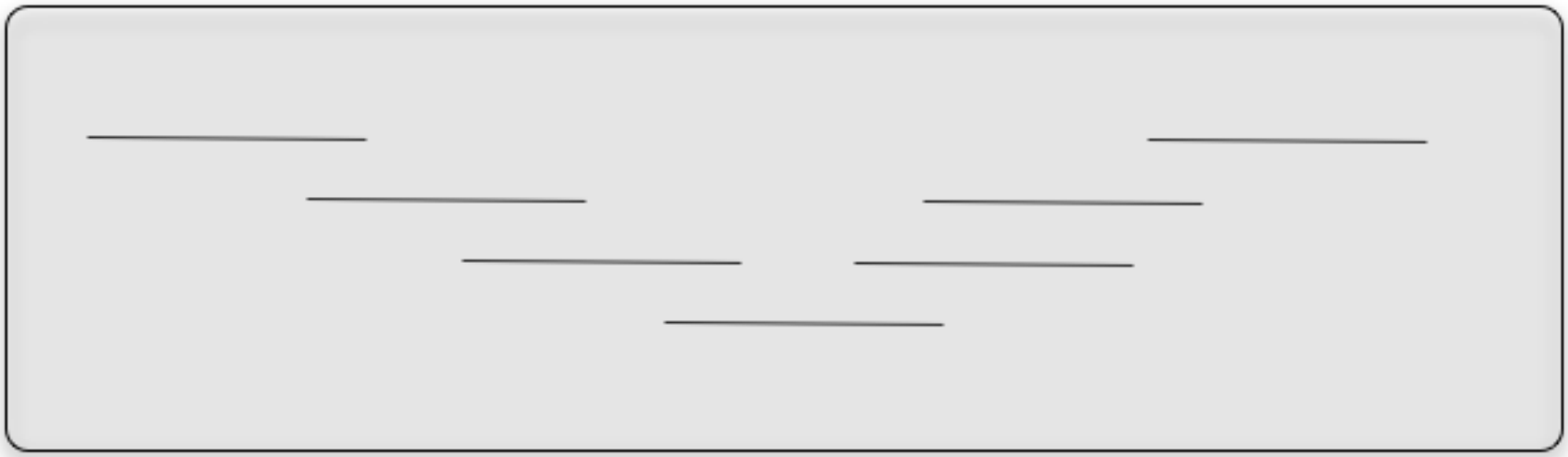
Randomly fragment & sequence from DNA; reassemble computationally.



UMD assembly primer (cbcb.umd.edu)

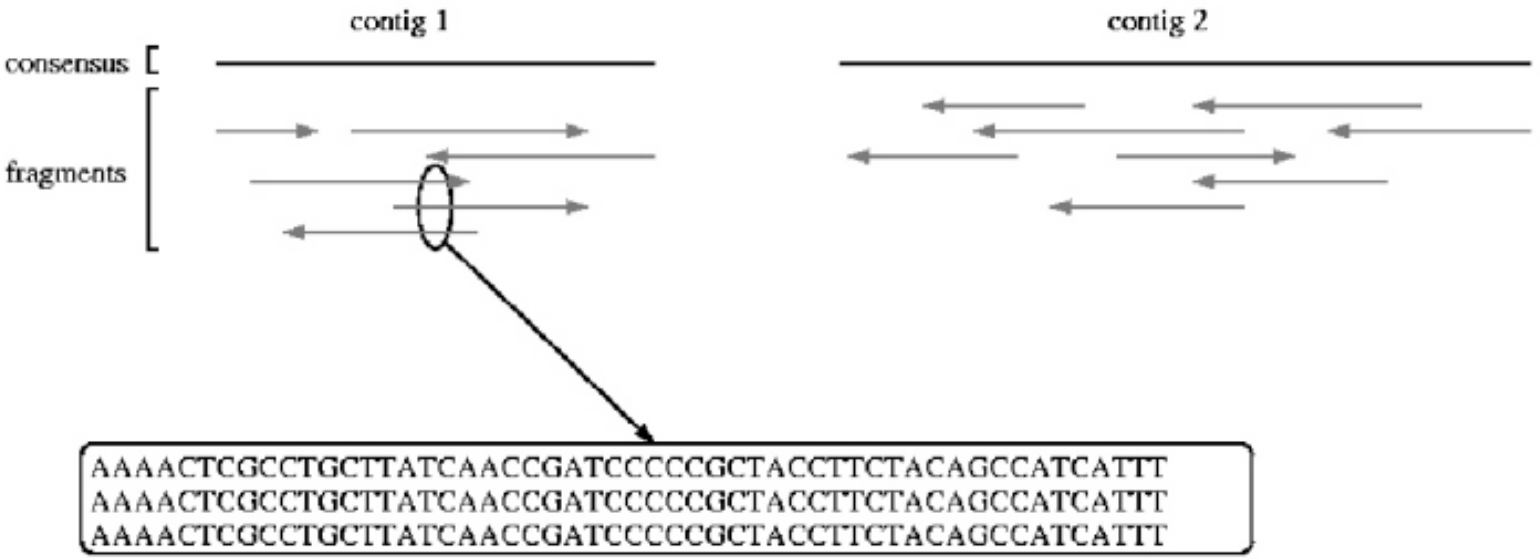
Assembly – no subdivision!

Assembly is inherently an *all by all* process. There is no good way to subdivide the reads without potentially missing a key connection



Short-read assembly

- Short-read assembly is problematic
- Relies on very deep coverage, ruthless read trimming, paired ends.



UMD assembly primer (cbcb.umd.edu)

Short read lengths are hard.

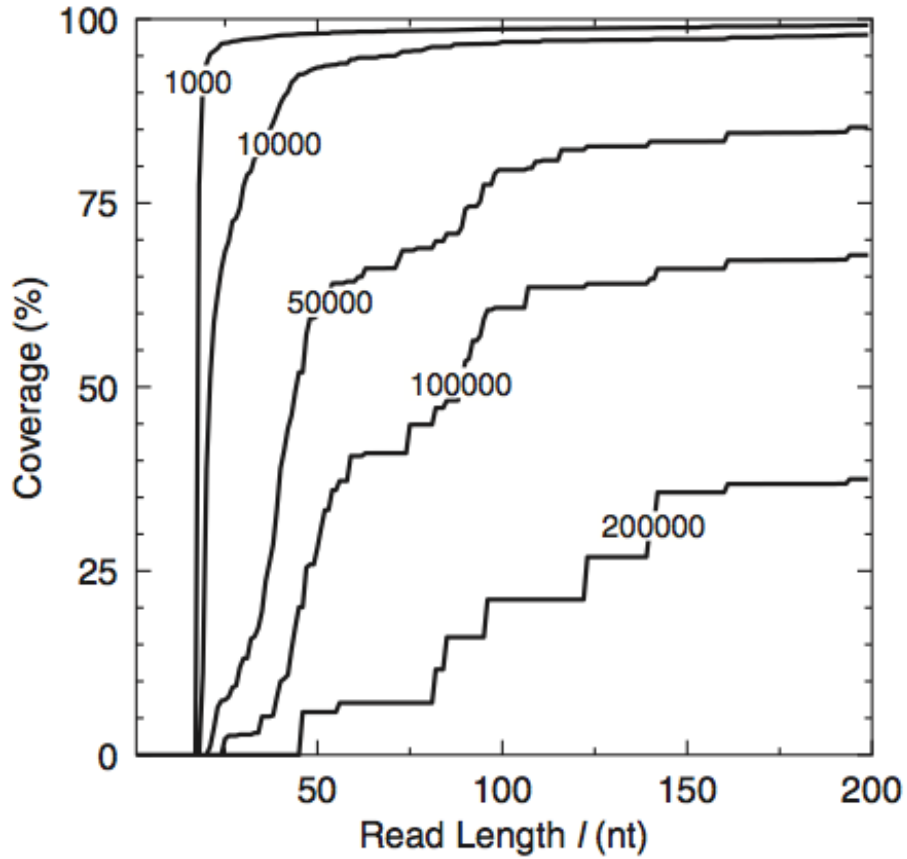


Figure 3. Percentage of the *E.coli* genome covered by contigs greater than a threshold length as a function of read length.

Whiteford et al., Nuc. Acid Res, 2005

Short read lengths are hard.

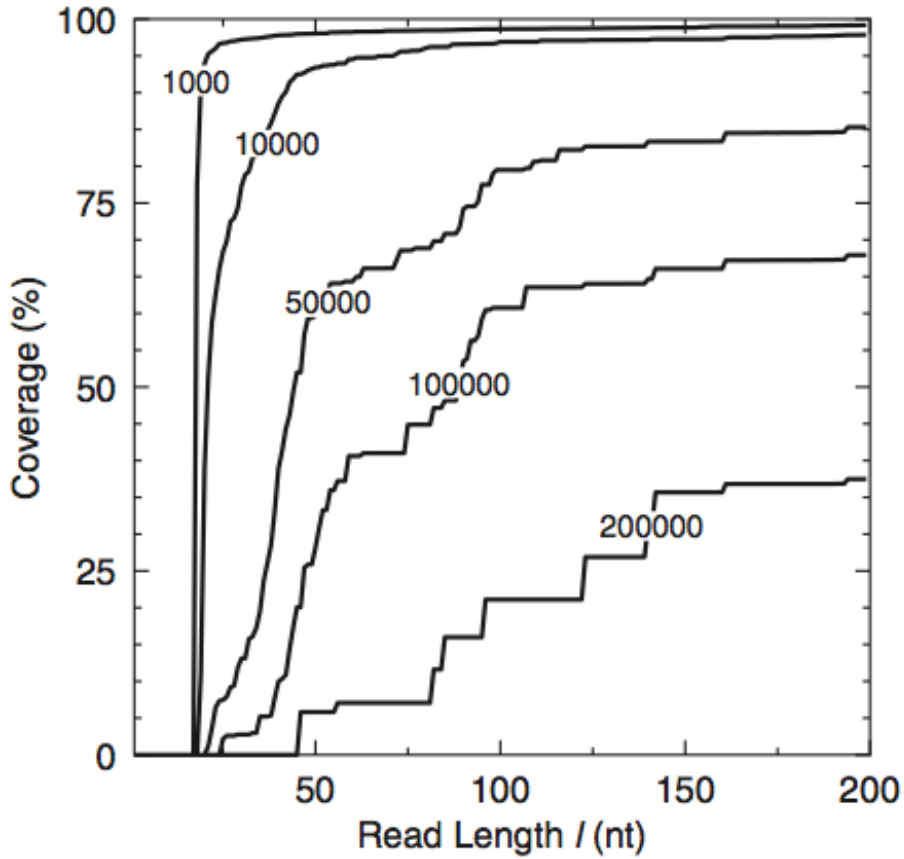


Figure 3. Percentage of the *E.coli* genome covered by contigs greater than a threshold length as a function of read length.

Conclusion: even with a read length of 200, the E. coli genome cannot be assembled completely.

Why?

Whiteford et al., Nuc. Acid Res, 2005

Short read lengths are hard.

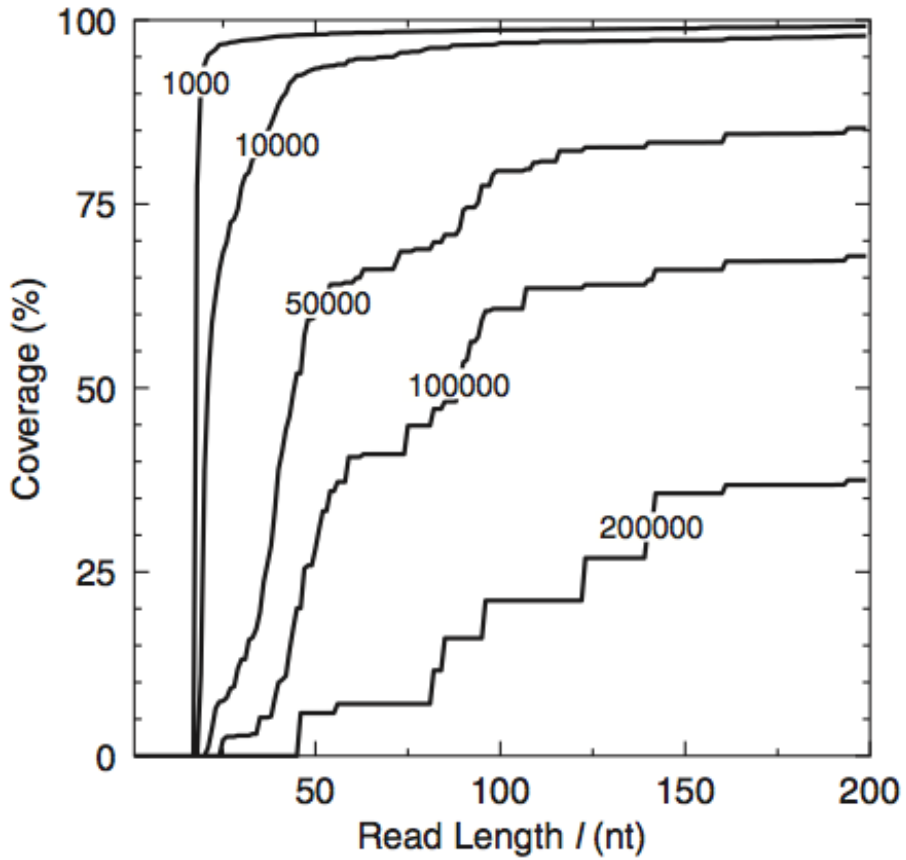


Figure 3. Percentage of the *E.coli* genome covered by contigs greater than a threshold length as a function of read length.

Conclusion: even with a read length of 200, the E. coli genome cannot be assembled completely.

Why? **REPEATS.**

This is why paired-end sequencing is so important for assembly.

Whiteford et al., Nuc. Acid Res, 2005

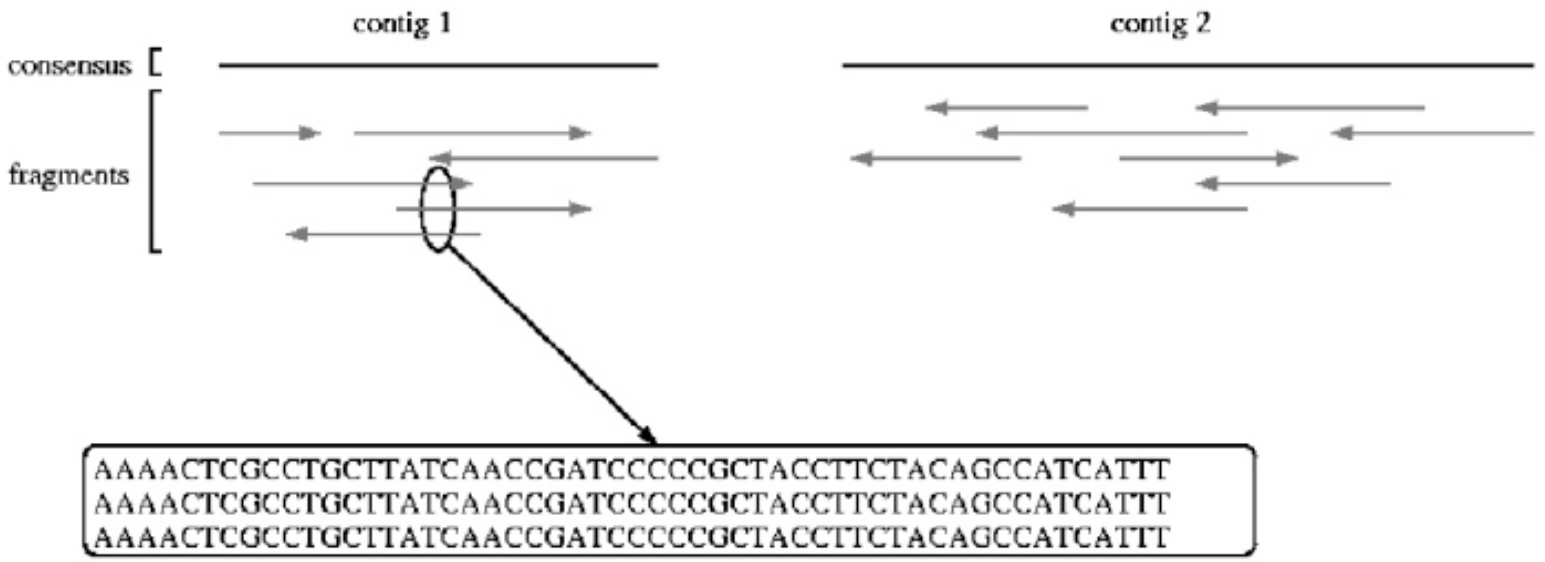
Two main challenges for *de novo* sequencing.

- Repeats.
- Low coverage.

Both introduce breaks in the construction of contigs.

Repeats

- Overlaps don't place sequences uniquely when there are repeats present.



UMD assembly primer (cbcb.umd.edu)

Coverage

Easy calculation:

(# reads x avg read length) / genome size

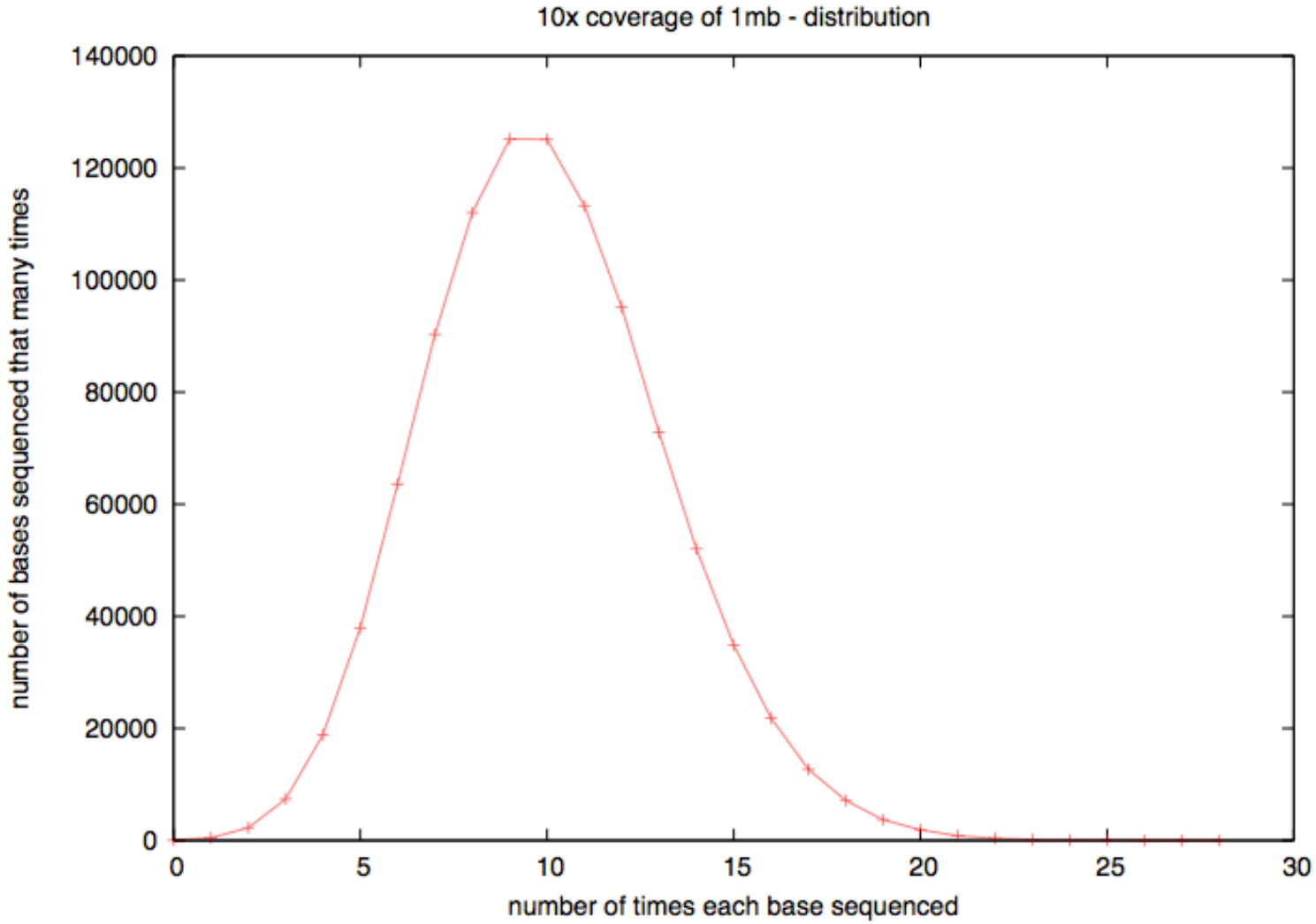
So, for haploid human genome:

30m reads x 100 bp = 3 bn

Coverage

- "1x" doesn't mean every DNA sequence is read once.
- It means that, if sampling were *systematic*, it would be.
- Sampling isn't systematic, it's random!

Actual coverage varies widely from the average, for low avg coverage



Two basic assembly approaches

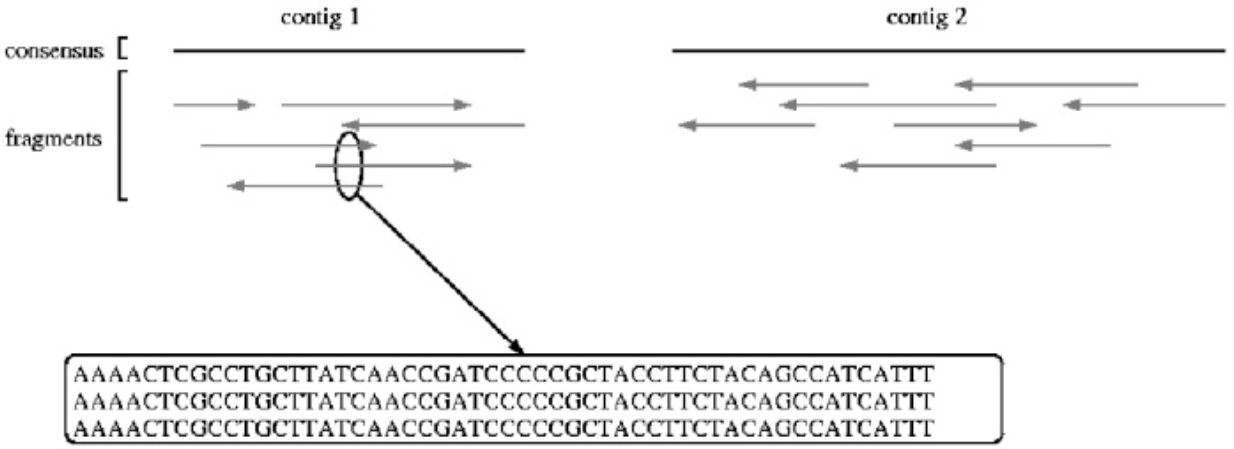
- Overlap/layout/consensus
- De Bruijn k-mer graphs

The former is used for long reads, esp all Sanger-based assemblies. The latter is used because of memory efficiency.

Overlap/layout/consensus

Essentially,

1. Calculate all overlaps
2. Cluster based on overlap.
3. Do a multiple sequence alignment



UMD assembly primer (cbcb.umd.edu)

K-mers

Essentially, break reads (of any length) down into multiple overlapping words of fixed length *k*.

```
ATGGACCAGATGACAC (k=12) =>

ATGGACCAGATG
 TGGACCAGATGA
  GGACCAGATGAC
   GACCAGATGACA
    ACCAGATGACAC
```

K-mers – what k to use?

Table 1A. Mean number of false placements of *K*-mers on the genome

K	*Escherichia coli*	*Saccharomyces cerevisiae*	*Arabidopsis thaliana*	*Homo sapiens*
200	0.063	0.26	0.053	0.18
160	0.068	0.31	0.064	0.49
120	0.074	0.39	0.086	1.7
80	0.082	0.49	0.15	7.2
60	0.088	0.58	0.27	18
50	0.091	0.63	0.39	32
40	0.095	0.69	0.65	78
30	0.11	0.77	1.5	330
20	0.15	1.0	5.7	2100
10	18	63.8	880	40,000

Butler et al., Genome Res, 2009

K-mers – what k to use?

Table 1B. Fraction of *K*-mers having a unique placement on the genome

K	*E. coli* (%)	*S. cerevisiae* (%)	*A. thaliana* (%)	*H. sapiens* (%)
200	98.5	95.9	97.4	97.6
160	98.3	95.6	97.1	97.2
120	98.2	95.2	96.6	96.6
80	98.0	94.7	95.4	95.2
60	97.8	94.4	94.4	93.1
50	97.7	94.2	93.4	91.2
40	97.6	93.9	92.2	88.3
30	97.4	93.5	90.4	83.4
20	97.0	92.9	86.5	71.8
10	0.0	0.0	0.0	0.0

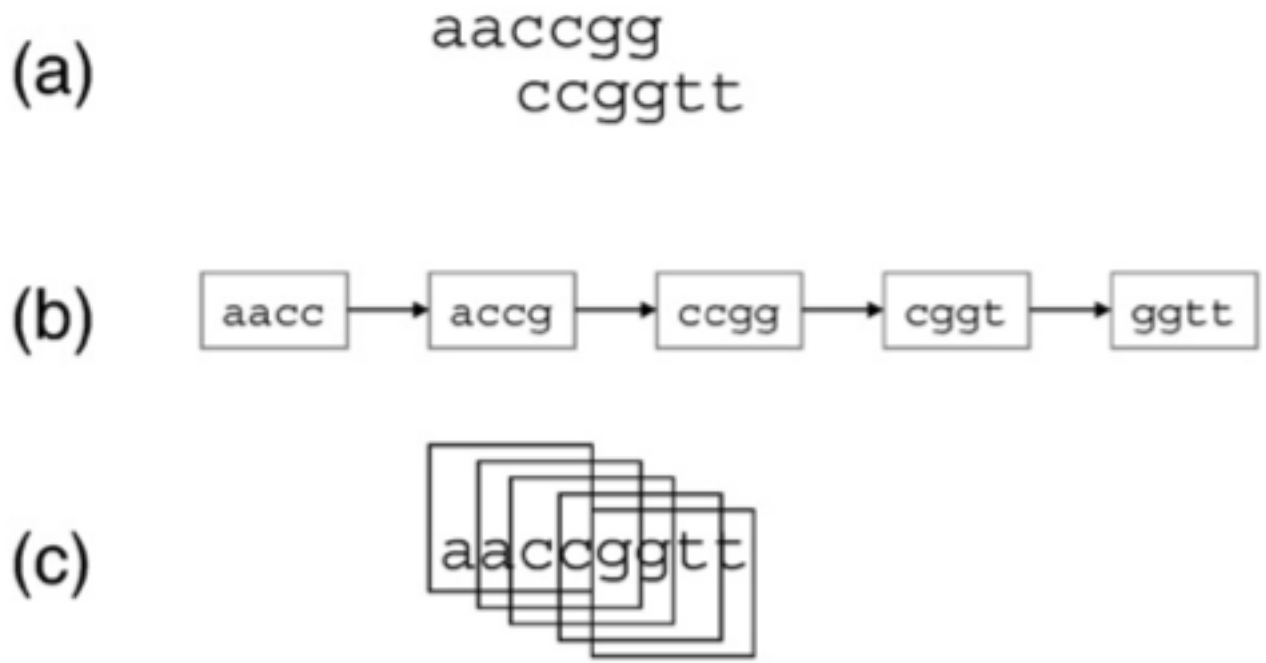
Butler et al., Genome Res, 2009

Big genomes are problematic

Species	Ploidy	Genome size (kb)	Reference N50 (kb)	Component N50 (kb)	Edge N50 (kb)	Ambiguities per megabase	Coverage (%)	Coverage by perfect edges ≥10 kb (%)
C. jejuni	1	1800	1800	1800	1800	0.0	100.0	100.0
E. coli	1	4600	4600	4600	4600	0.0	100.0	100.0
B. thailandensis	1	6700	3800	1800	890	2.7	99.8	99.5
E. gossypii	1	8700	1500	1500	890	2.6	100.0	99.9
S. cerevisiae	1	12,000	920	810	290	28.7	98.7	94.9
S. pombe	1	13,000	4500	1400	500	19.1	98.8	97.5
P. stipitis	1	15,000	1800	900	700	8.6	97.9	96.3
C. neoformans	1	19,000	1400	810	770	4.5	96.4	93.4
Y. lipolytica	1	21,000	3600	2200	290	6.2	99.1	98.6
Neurospora crassa	1	39,000	660	640	90	17.4	97.0	92.5
H. sapiens region	2	10,000	10,000	490	2	68.2	97.3	0.2

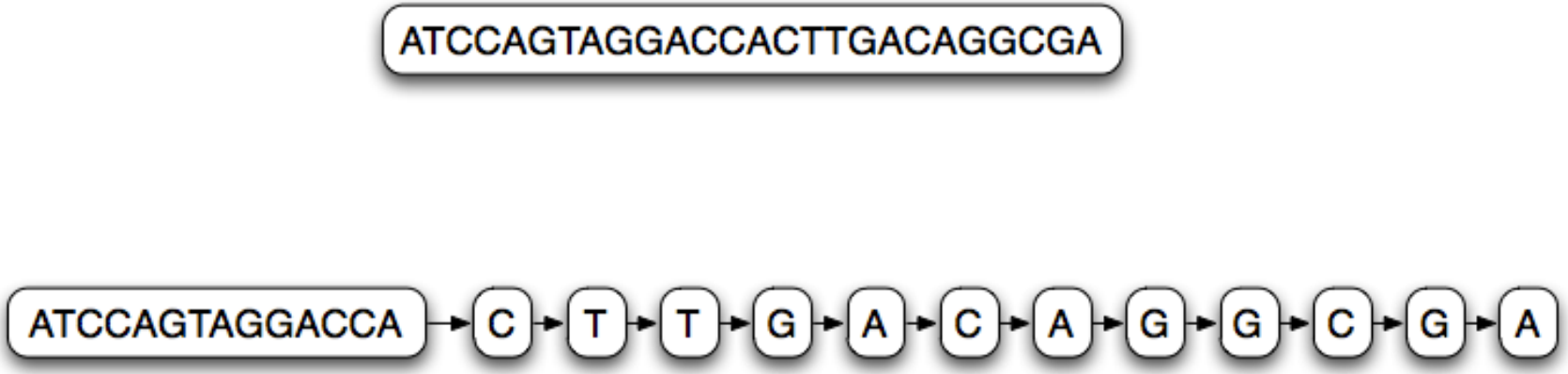
Butler et al., Genome Res, 2009

K-mer graphs - overlaps



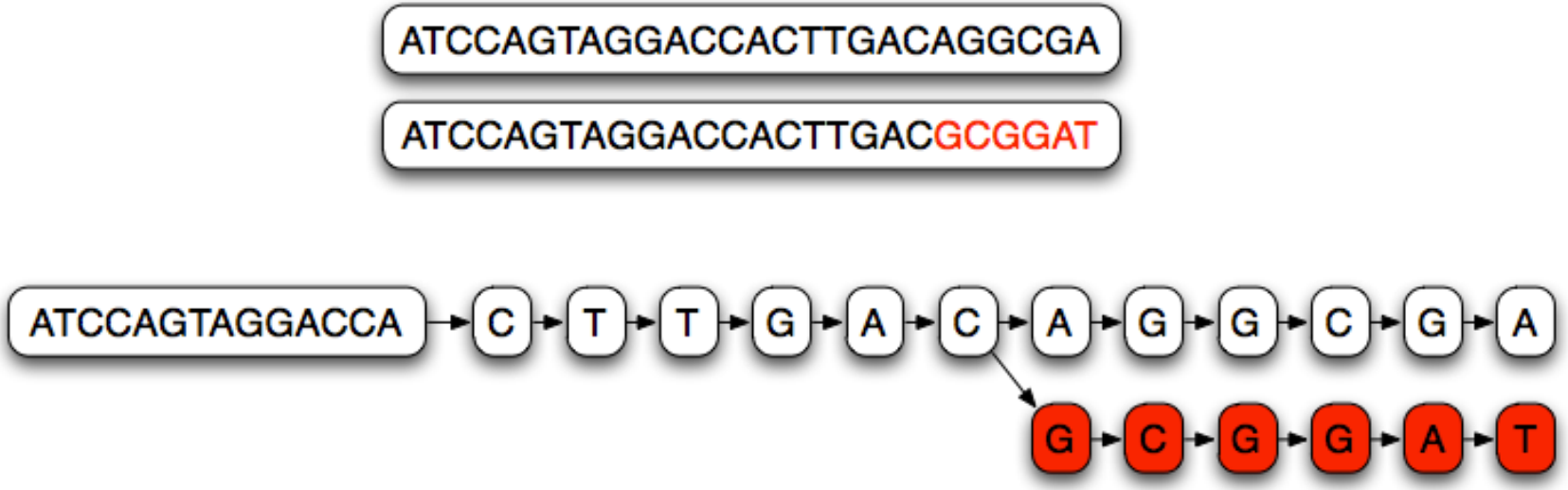
J.R. Miller et al. / Genomics (2010)

K-mer graph (k=14)



Each node represents a 14-mer;
Links between each node are 13-mer overlaps

K-mer graph (k=14)

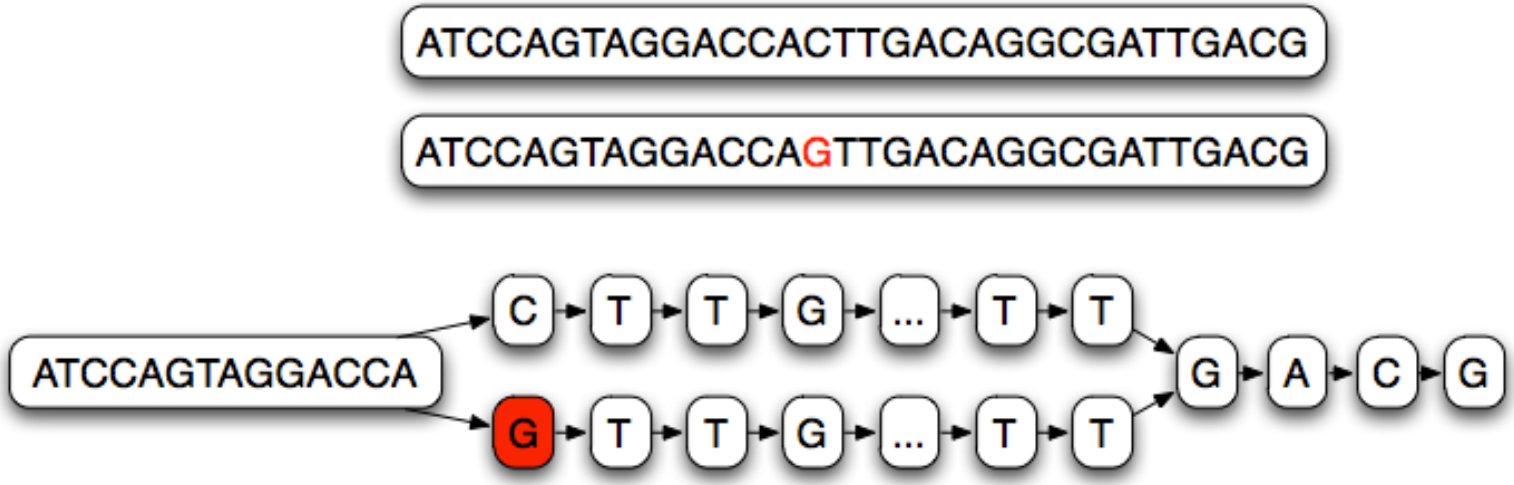


Branches in the graph represent partially overlapping sequences.

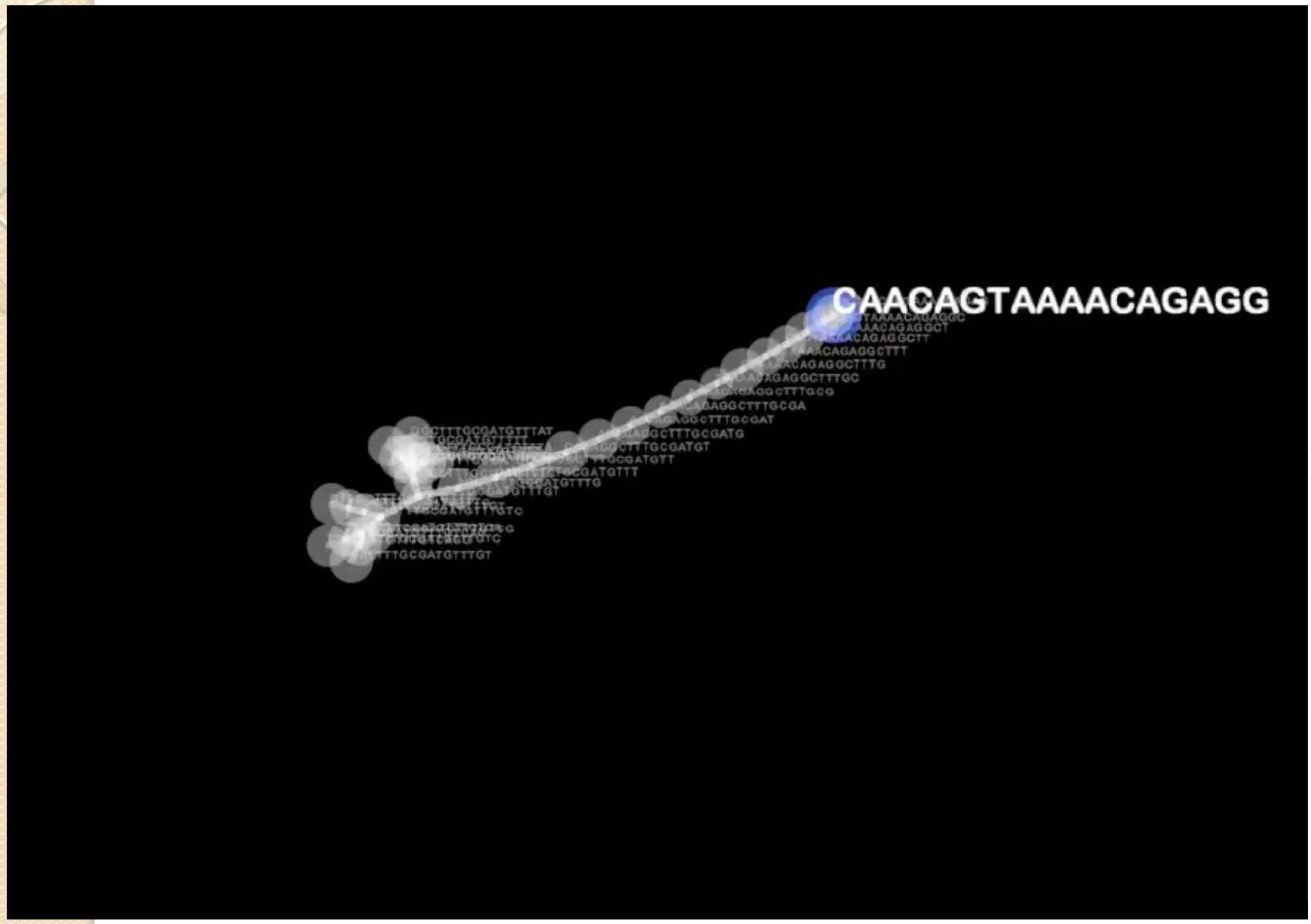
K-mer graph (k=14)

Single nucleotide variations cause long branches

K-mer graph (k=14)

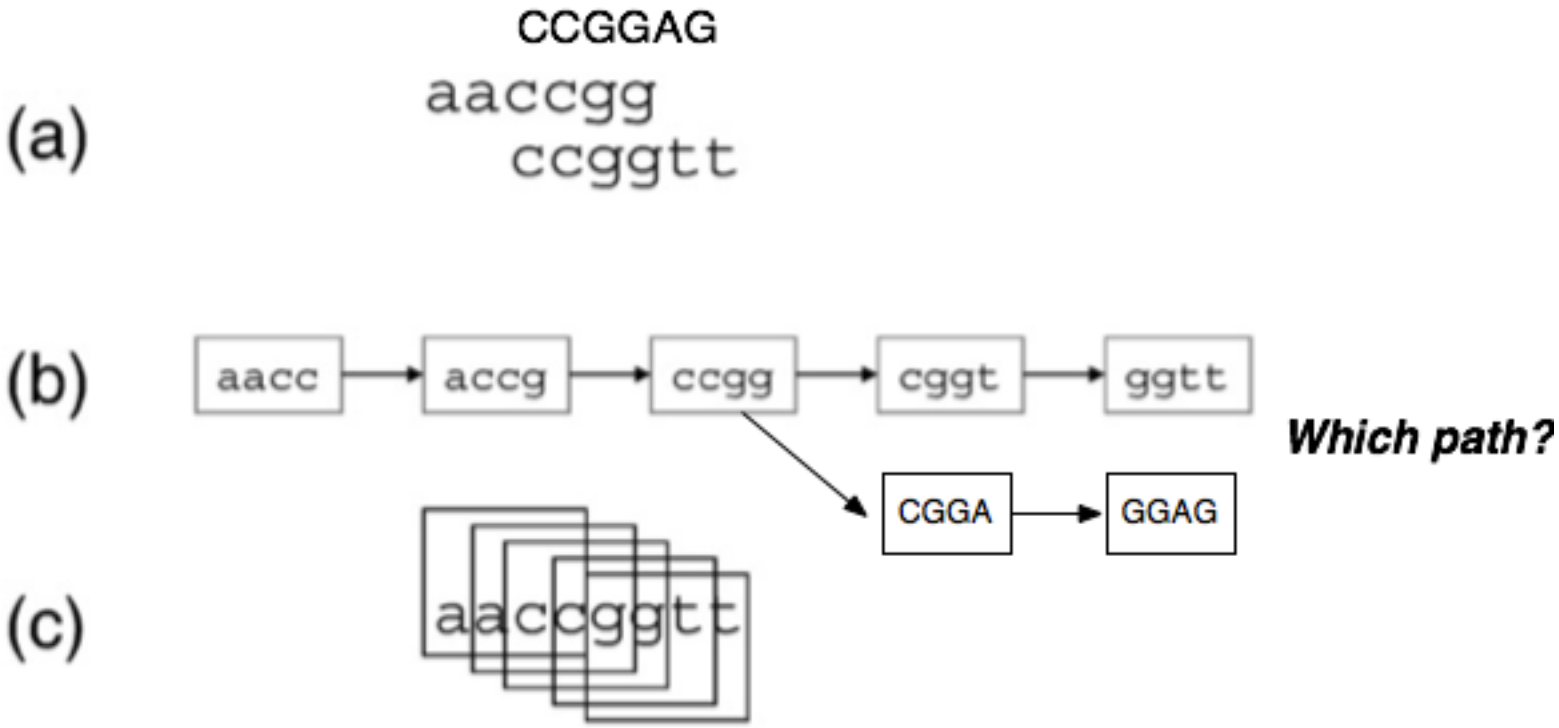


Single nucleotide variations cause long branches;
They don't rejoin quickly.



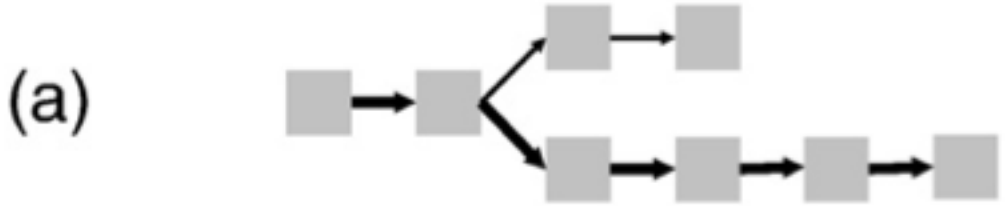
Groxel view of knot-like region / Arend Hintze

K-mer graphs - branching



For decisions about which paths etc, biology-based heuristics come into play as well.

K-mer graph complexity - spur



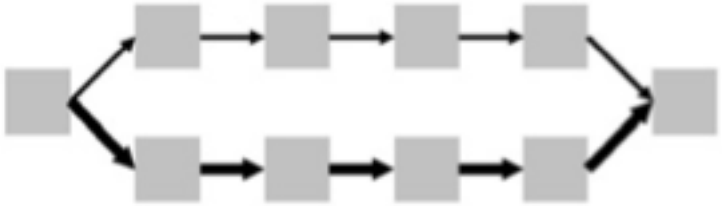
(Short) dead-end in graph.

Can be caused by error at the end of some overlapping reads, or low coverage

J.R. Miller et al. / Genomics (2010)

K-mer graph complexity - bubble

(b)



Multiple parallel paths that diverge and join.

Caused by sequencing error and true polymorphism / polyploidy in sample.

J.R. Miller et al. / Genomics (2010)

K-mer graph complexity – "frayed rope"

(c)

Converging, then diverging paths.

Caused by repetitive sequences.

J.R. Miller et al. / Genomics (2010)

Resolving graph complexity

- Primarily heuristic (approximate) approaches.
- Detecting complex graph structures can generally not be done efficiently.
- Much of the divergence in functionality of new assemblers comes from this.
- Three examples:

Read threading

Single read spans k-mer graph => extract the single-read path.

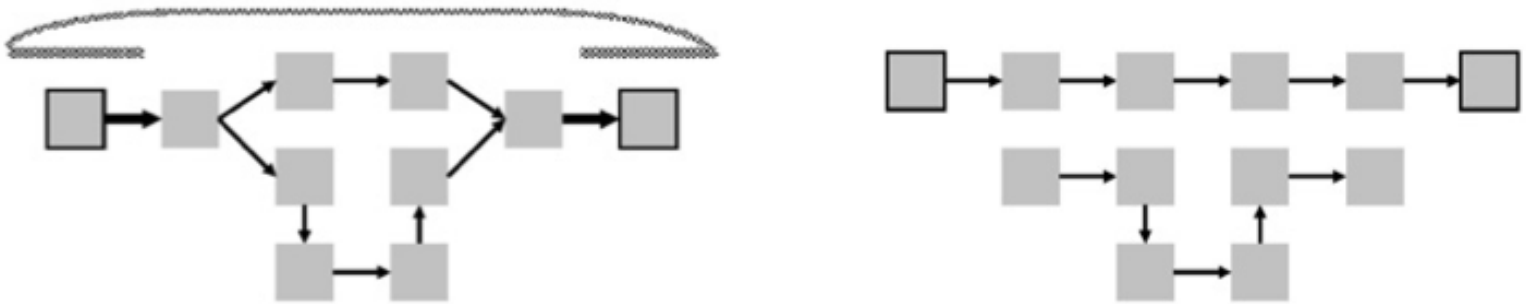
J.R. Miller et al. / Genomics (2010)

Mate threading

Resolve "frayed-rope" pattern caused by repeats, by separating paths based on mate-pair reads.

J.R. Miller et al. / Genomics (2010)

Path following

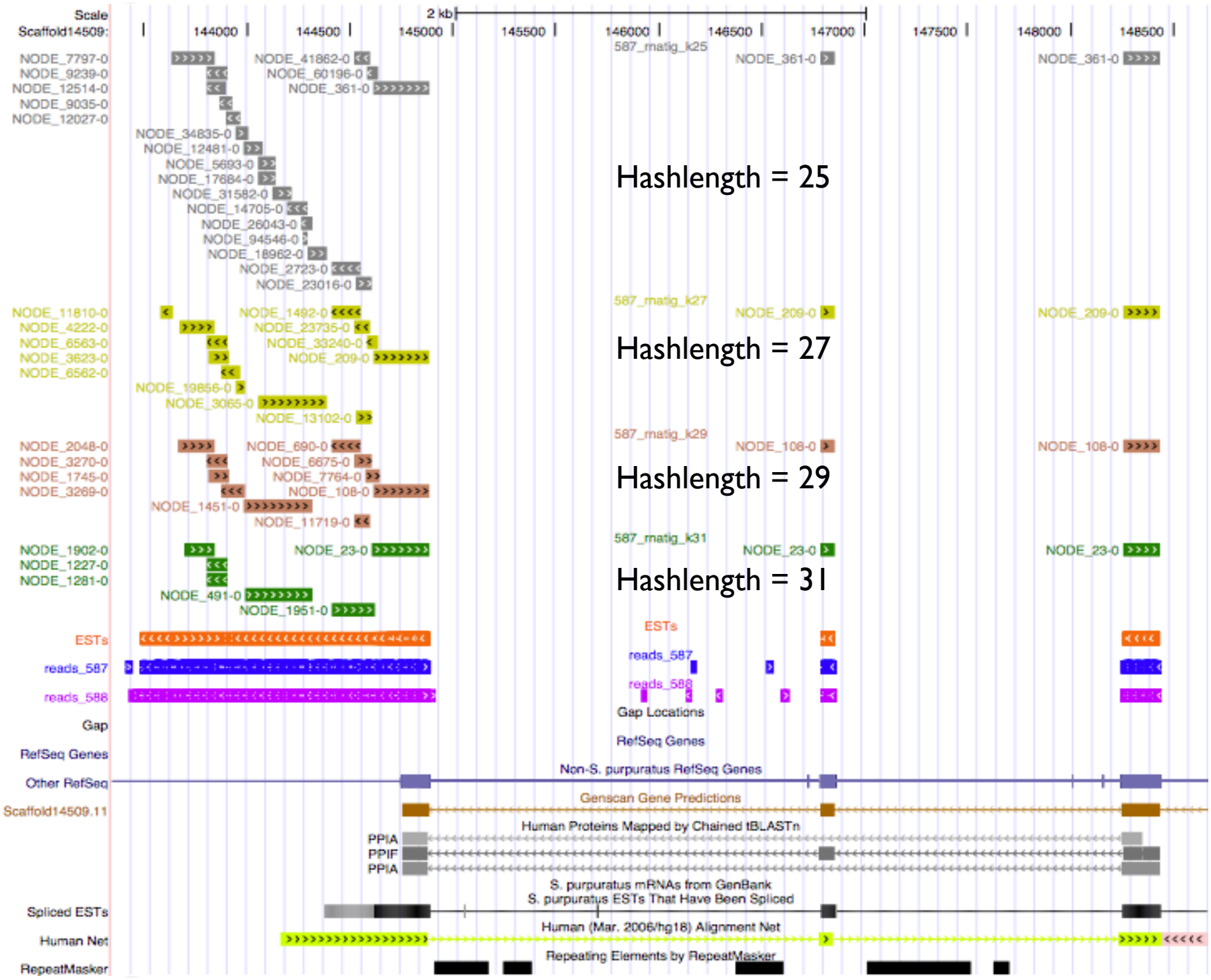


Reject inconsistent paths based on mate-pair reads and insert size.

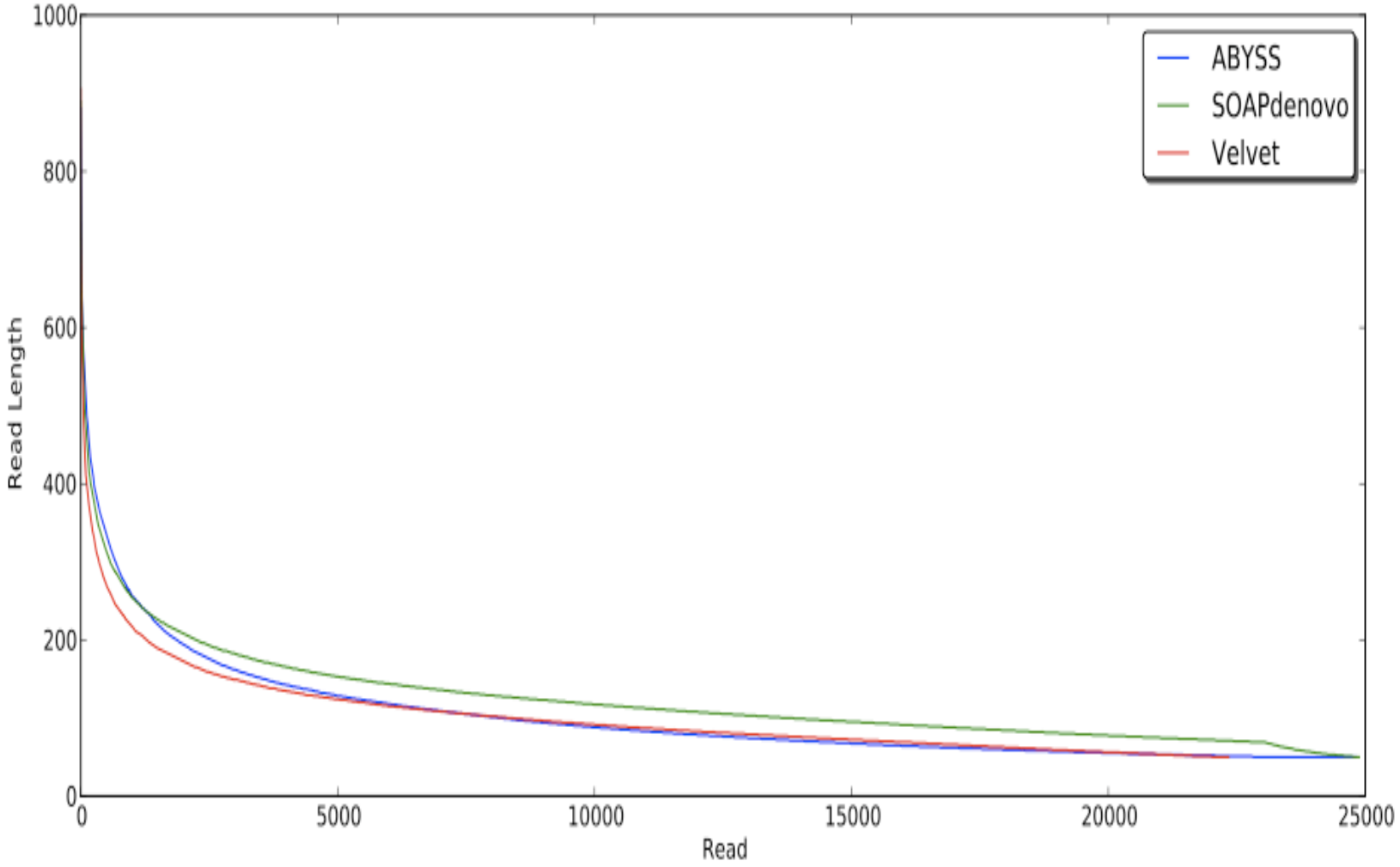
J.R. Miller et al. / Genomics (2010)

More assembly isses

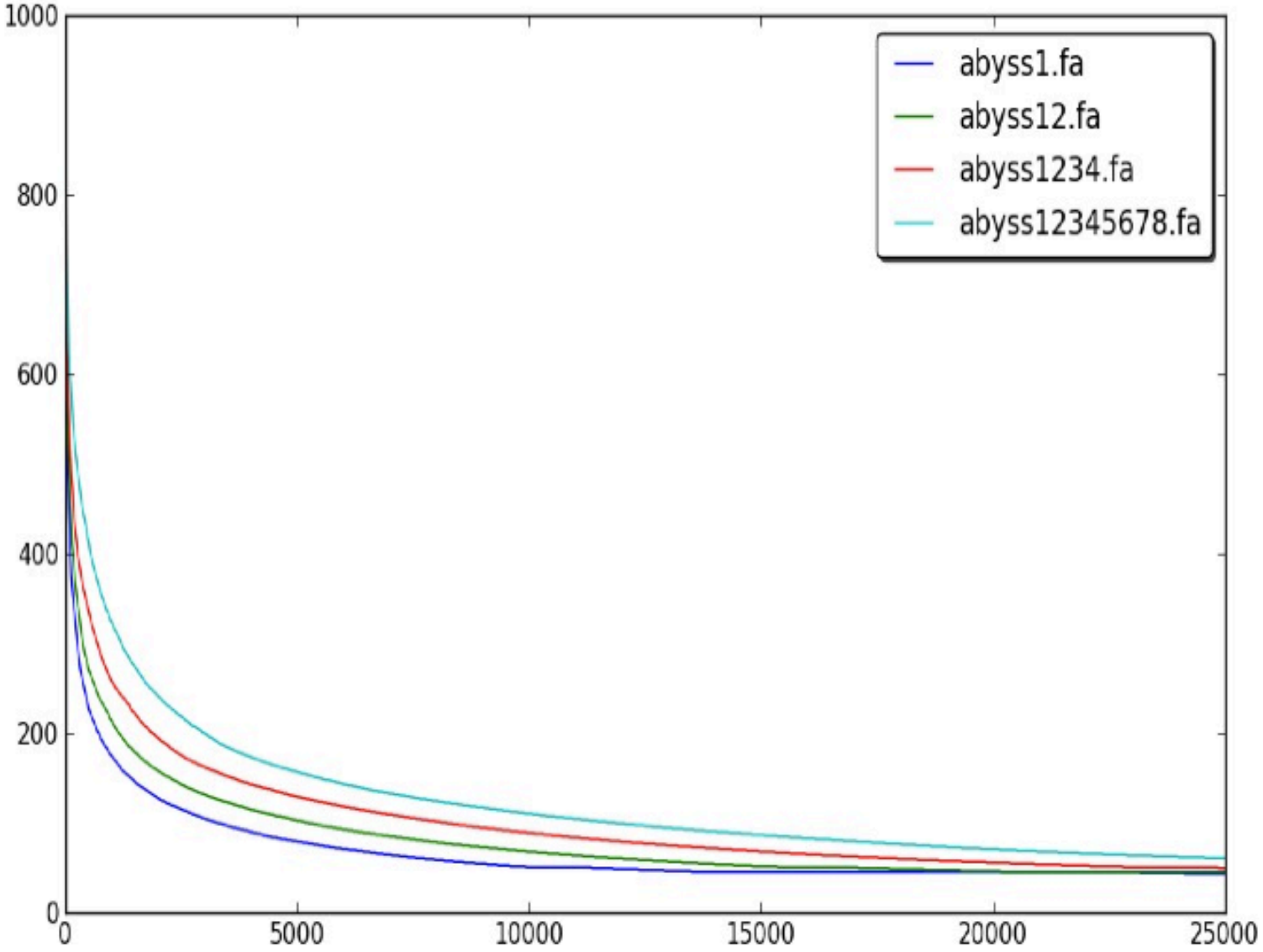
- Many parameters to optimize!
- RNAseq has variation in copy number; naïve assemblers can treat this as repetitive and eliminate it.
- Assembly requires gobs of memory (4 lanes, 60m reads => ~ 150gb RAM)
- How do we evaluate assemblies?
  - What's the best assembler?


Scale
Scaffold14509:
2 kb
144000
144500
145000
145500
146000
146500
147000
147500
148000
148500
587_rnatig_k25
NODE_7797-0
NODE_9239-0
NODE_12514-0
NODE_9035-0
NODE_12027-0
NODE_41862-0
NODE_60196-0
NODE_361-0
NODE_361-0
NODE_361-0
NODE_34835-0
NODE_12481-0
NODE_5693-0
NODE_17684-0
NODE_31582-0
NODE_14705-0
NODE_26043-0
NODE_94546-0
NODE_18962-0
NODE_2723-0
NODE_23016-0
Hashlength = 25
587_rnatig_k27
NODE_11810-0
NODE_4222-0
NODE_6563-0
NODE_3623-0
NODE_6562-0
NODE_1492-0
NODE_23735-0
NODE_33240-0
NODE_209-0
NODE_209-0
NODE_209-0
NODE_19856-0
NODE_3065-0
NODE_13102-0
Hashlength = 27
587_rnatig_k29
NODE_2048-0
NODE_3270-0
NODE_1745-0
NODE_3269-0
NODE_690-0
NODE_6675-0
NODE_7764-0
NODE_108-0
NODE_108-0
NODE_108-0
NODE_1451-0
NODE_11719-0
Hashlength = 29
587_rnatig_k31
NODE_1902-0
NODE_1227-0
NODE_1281-0
NODE_23-0
NODE_23-0
NODE_23-0
NODE_491-0
NODE_1951-0
Hashlength = 31
ESTs
reads_587
reads_588
Gap
RefSeq Genes
Other RefSeq
Scaffold14509.11
Spliced ESTs
Human Net
RepeatMasker
ESTs
reads_587
reads_588
Gap Locations
RefSeq Genes
Non-S. purpuratus RefSeq Genes
Genscan Gene Predictions
Human Proteins Mapped by Chained tBLASTn
PPIA
PPIF
PPIA
S. purpuratus mRNAs from GenBank
S. purpuratus ESTs That Have Been Spliced
Human (Mar. 2006/hg18) Alignment Net
Repeating Elements by RepeatMasker

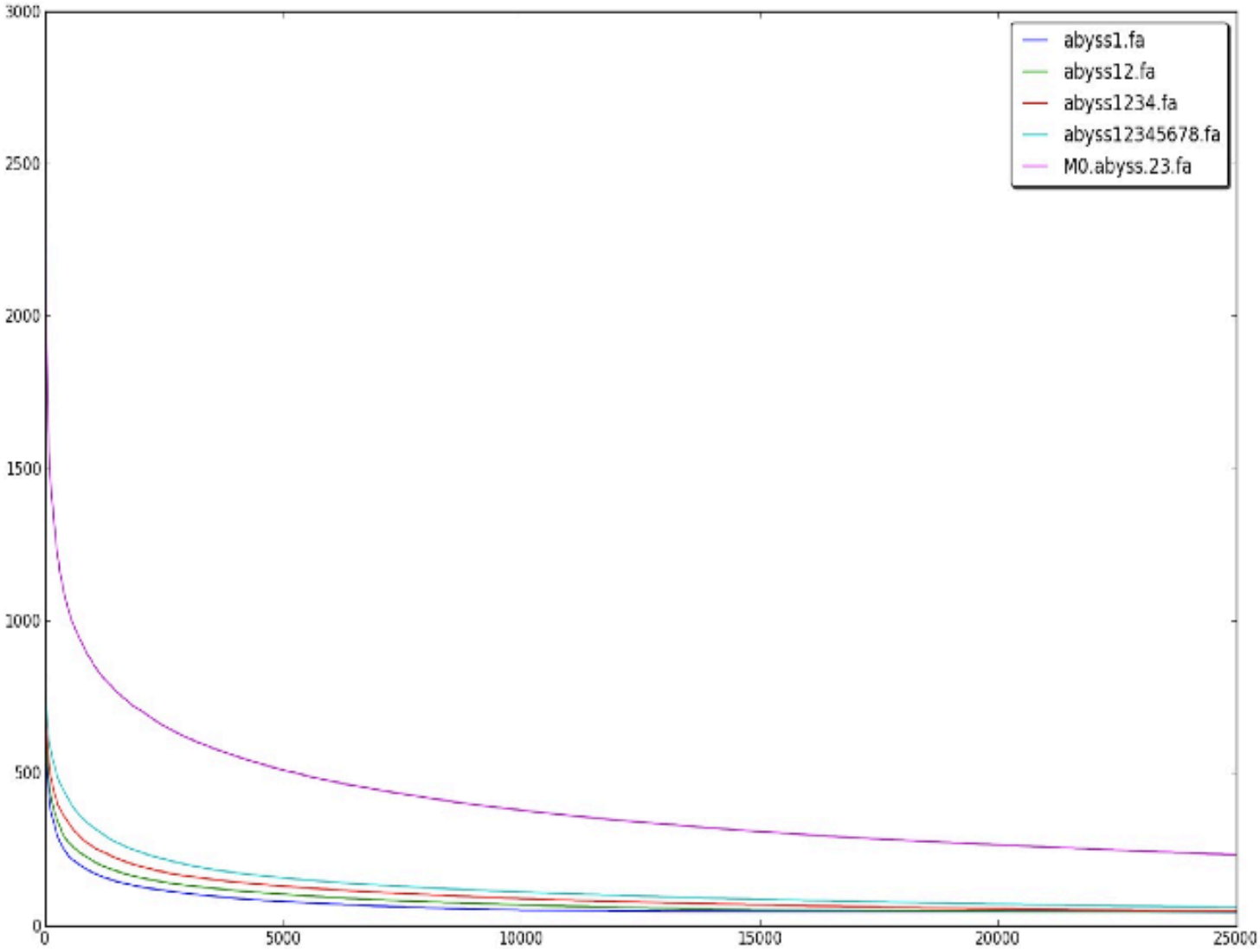
Trying out different assemblers



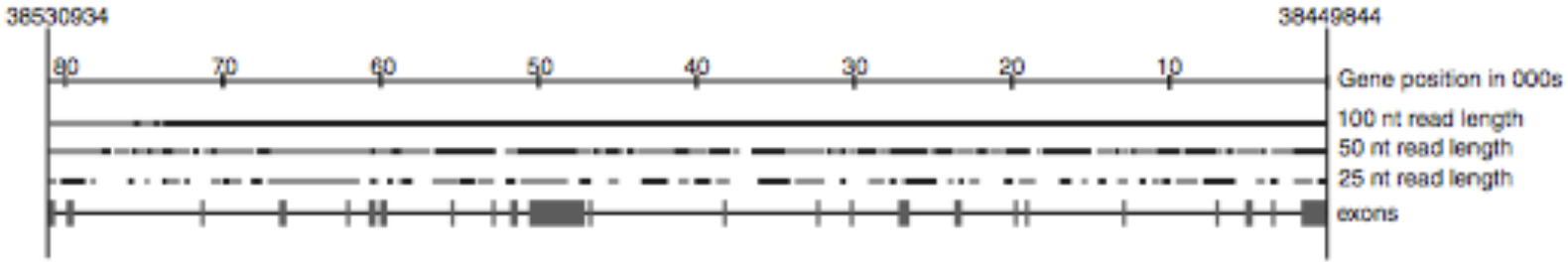
Adding more data starts to saturate



Longer reads help a lot… add 75bp

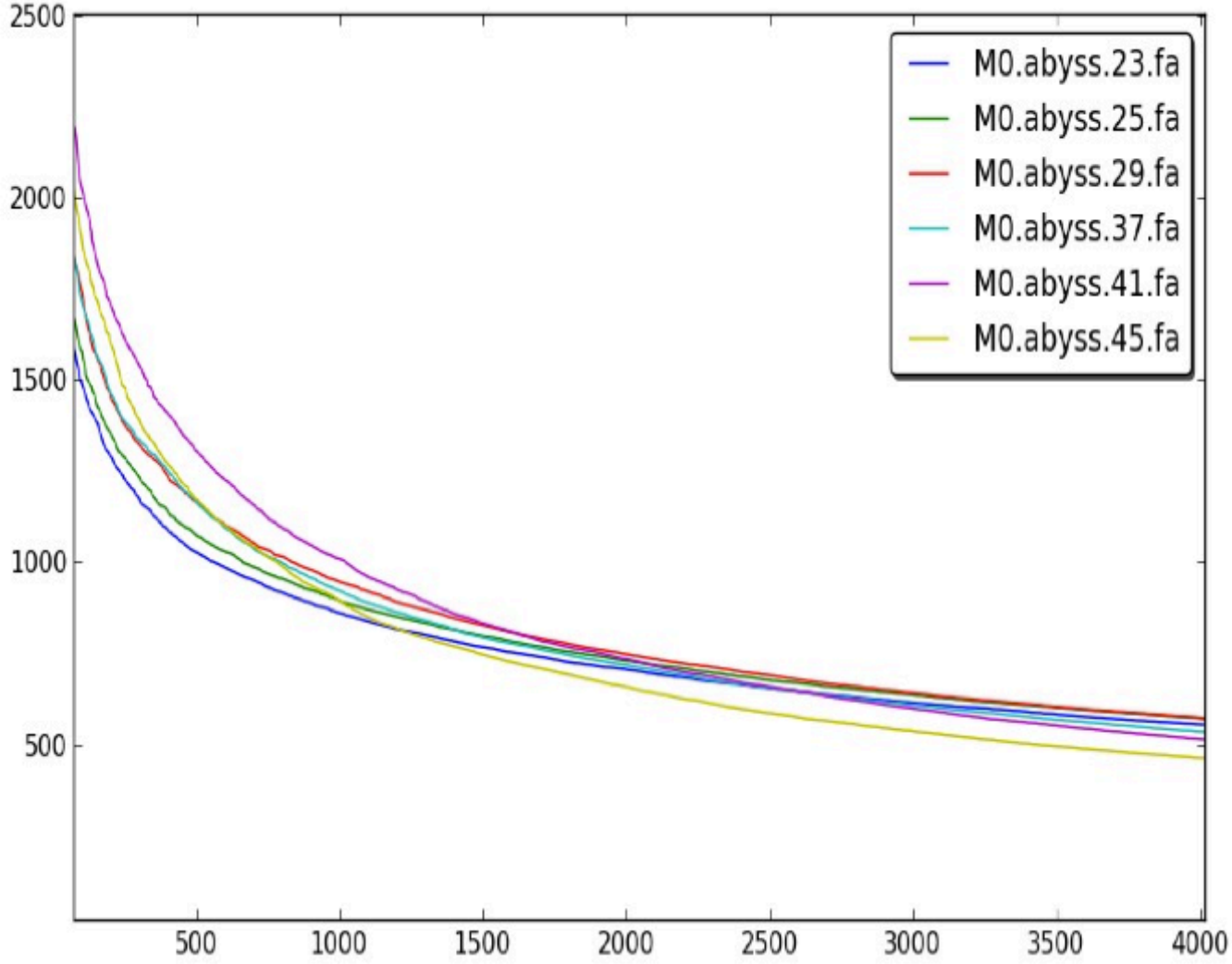


…this matches simulations

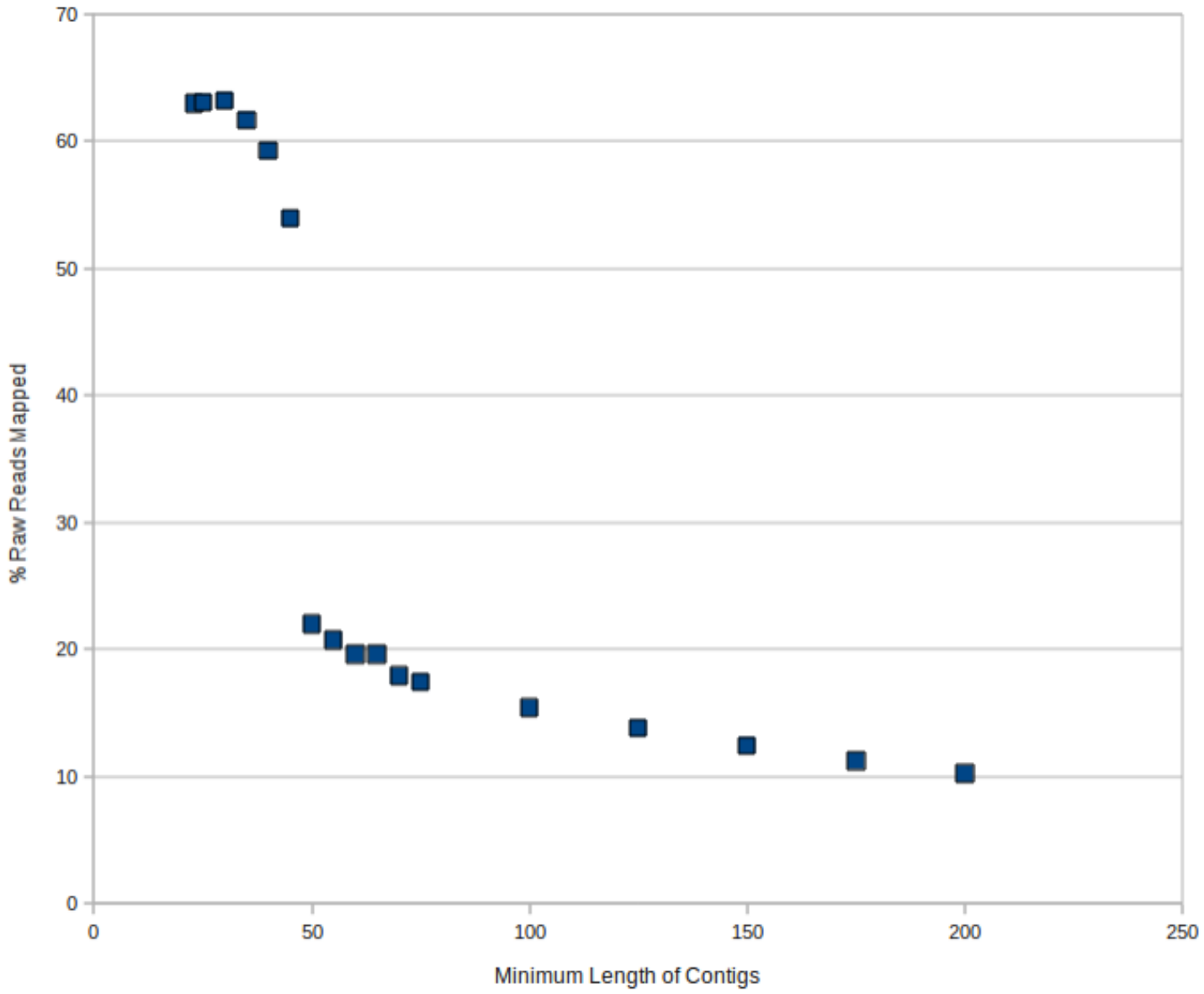


Whiteford et al., Nuc. Acid Res, 2005

Different k values matter a bit



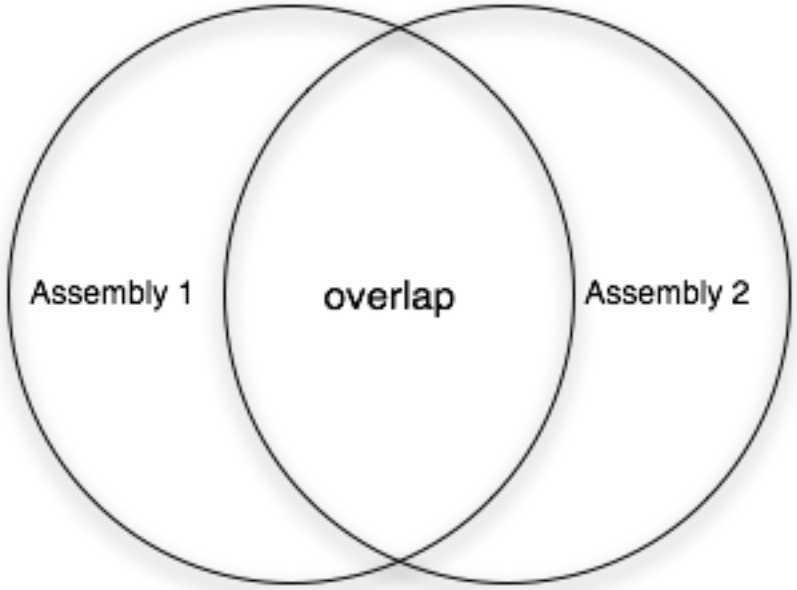
How useful is assembly, anyway?



Is your assembly good?

- For genomes, N50 is an OK measure:
  - "50% or more of the genome is in contigs > this number"
- That assumes your contigs are correct…!
- What about mRNA and metagenomes??
- **Truly reference-free assembly is hard to evaluate.**

How do you compare assemblies?

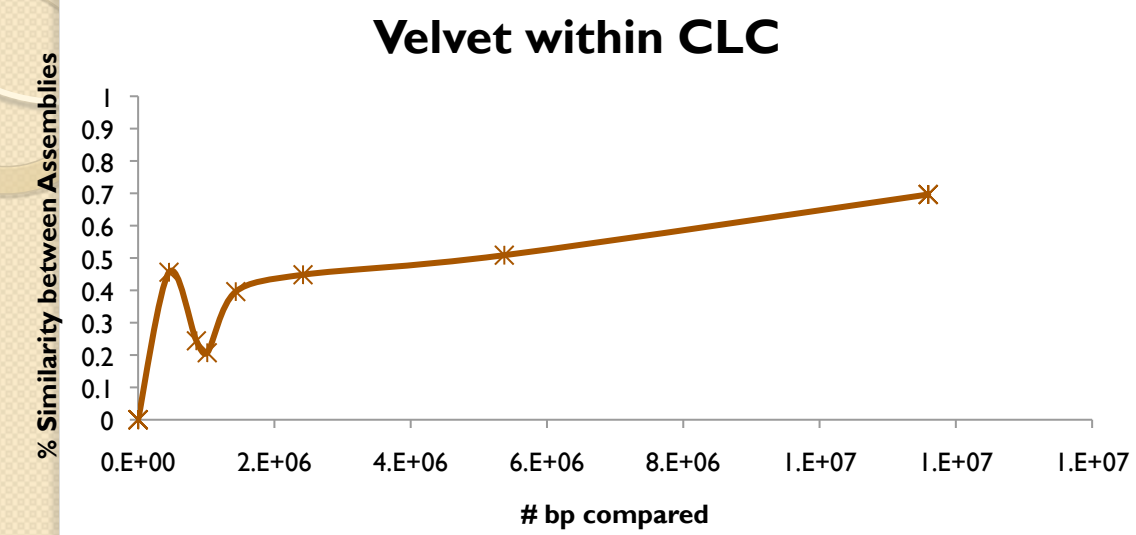


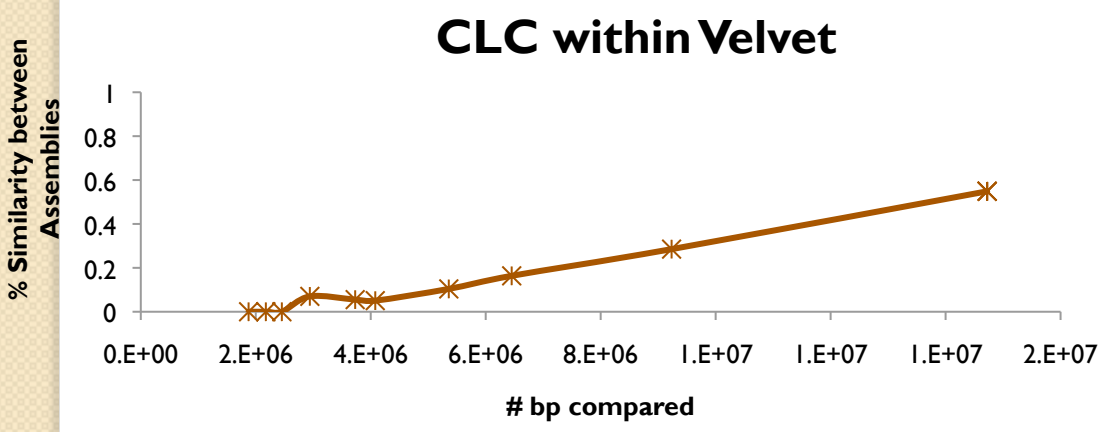
K-mer comparison technique

- Rank order contigs (shortest first)
- Compare all contigs from two assemblies that are >= cutoff
  - Decompose contigs into k-mers
  - Look for k-mer overlap

- This looks at *composition* of assemblies rather than structure; but if it reports differences, then those are *minimum* differences.

Comparison with small, clean data set



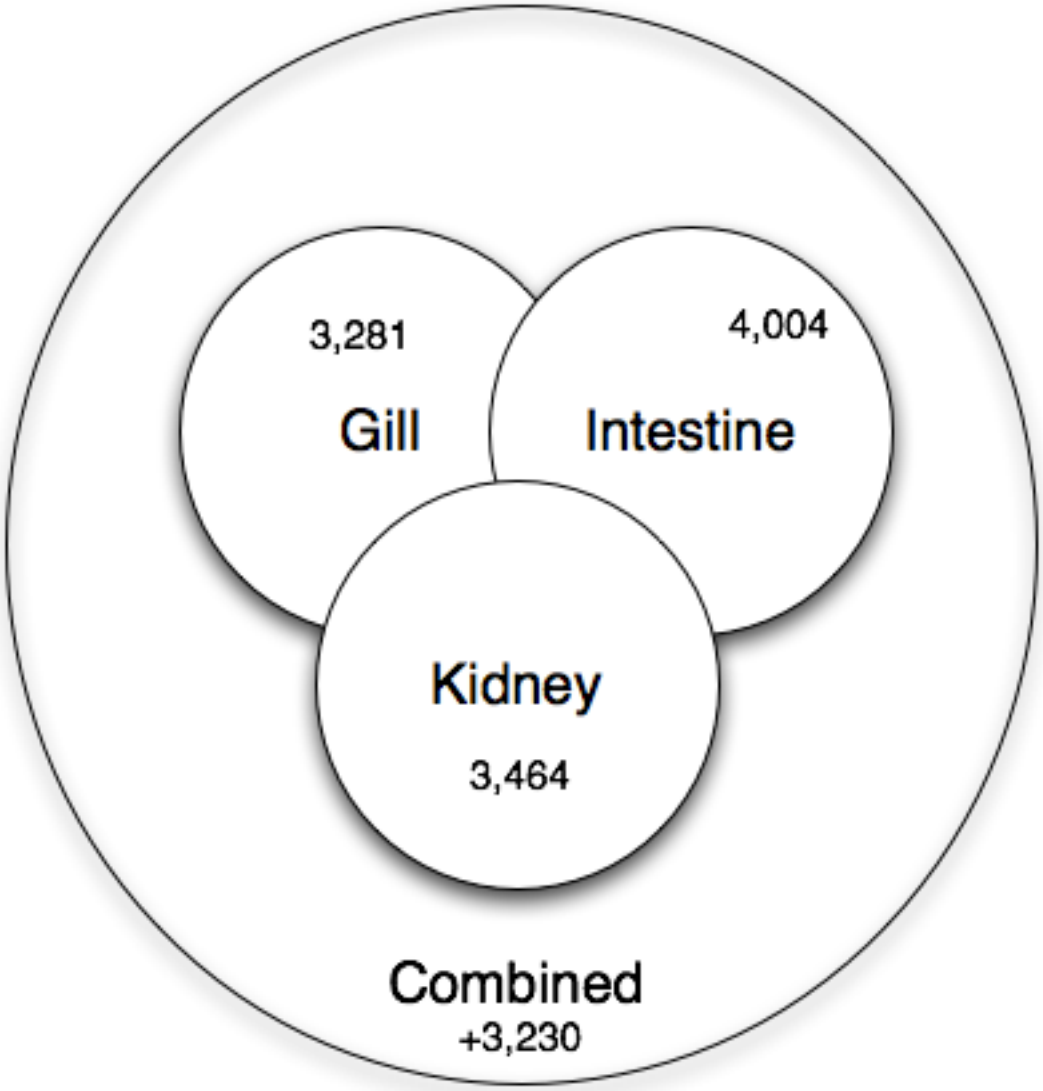


Conclusions:

CLC and Velvet are assembling quite a different set of contigs, even on a really simple, clean data set. !!

Co-assembly is important for sensitivity

Shared low-level transcripts may not reach the threshold for assembly.



Advertisement: next Thursday, Erich Schwarz on assembling a worm genome, de novo

H. contortus assembly statistics

	Word size (k)	Total (Mb)	Super-contigs	Max. sc. size	N50 (kb)	% non-N
Genomic	35	630	152 K	1.18 Mb	131	81.5
Top fraction	35	315	1.4 K	1.18 Mb	234	83.7
cDNA	43	67.6	132 K	22.4 kb	0.92	99.9

Without khmer filtering, assemblies with partial data took up to a week with ~150 Gb of RAM; assemblies failed completely for full data sets.

With khmer filtering, full assemblies took half a day with ~80 Gb of RAM.

Genome size: 315 ± 25 Mb; genomic coverage: ~87x.

Reads matching sheep DNA or MtDNA or known linkers, along with low-quality residues, were routinely removed before khmer filtering and/or assembly.

Best of the best (for synthetic genomes)

Team	Assembler	Affiliation
P	SOAPdenovo	BGI
Q	ALLPATHS	Broad Institute
D	SGA	Wellcome Trust Sanger Institute

These were the three assemblies that were consistently among the best by whatever metric was used. Note though that there were no assemblies that ranked first by the majority of metrics. It was far more common to see good assemblies consistently appear in the top 5 rankings of any particular metric. Having said that, even these assemblies would drop outside the top 10 based on certain metrics.

Practical issues

- Do you have enough memory?
- Trim vs use quality scores?
- When is your assembly as good as it gets?
- Paired-end vs longer reads?

- More data is not *necessarily* better, if it introduces more errors.

Go forth! And map/assemble!

- Assembly and mapping (and variations thereon) are the two basic approaches used to deal with next-gen sequencing data.
- After the next few tutorials, you will be a truly dangerous bioinformatician!
- Go forth & work with your own data, too!