Assembling metagenomes: a not so practical guide

C. Titus Brown
Assistant Professor
CSE, MMG, BEACON
Michigan State University
September 2013
ctb@msu.edu

Acknowledgements

Lab members involved

- **Adina Howe (w/Tiedje)**
- **Jason Pell**
- **Arend Hintze**
- **Rosangela Canino-Koning**
- **Qingpeng Zhang**
- Elijah Lowe
- Likit Preeyanon
- **Jiarong Guo**
- **Tim Brom**
- Kanchan Pavangadkar
- **Eric McDonald**

Collaborators

- Jim Tiedje, MSU; Janet Jansson, JGI; Susannah Tringe, JGI.

Funding

USDA NIFA; NSF IOS; BEACON, NIH.

Acknowledgements

Lab members involved

- ***Adina Howe*** **(w/Tiedje)**
- **Jason Pell**
- **Arend Hintze**
- **Rosangela Canino-Koning**
- **Qingpeng Zhang**
- Elijah Lowe
- Likit Preeyanon
- **Jiarong Guo**
- **Tim Brom**
- Kanchan Pavangadkar
- **Eric McDonald**

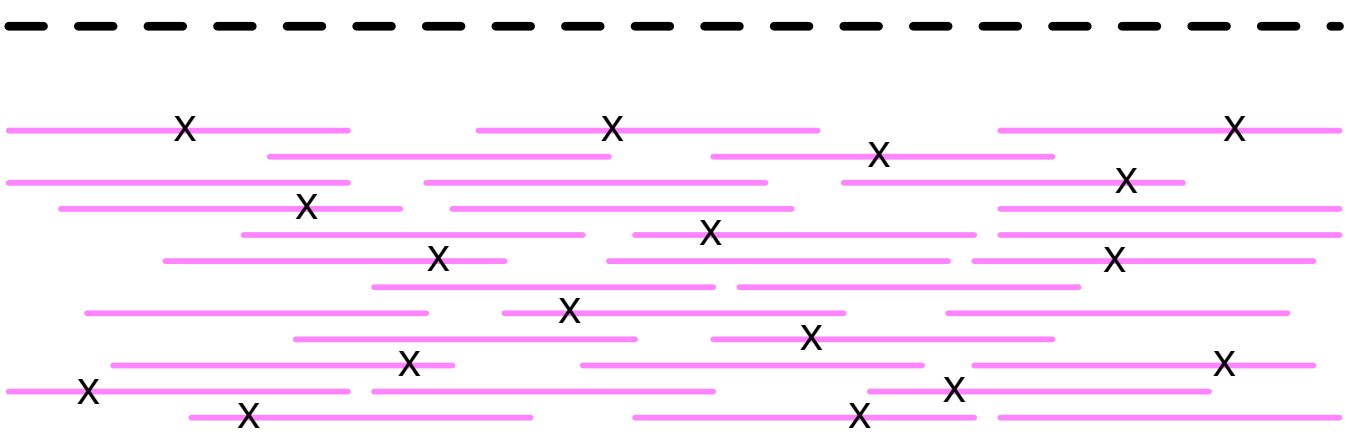
Collaborators

- Jim Tiedje, MSU; Janet Jansson, JGI; Susannah Tringe, JGI.

Funding

USDA NIFA; NSF IOS; BEACON, NIH.

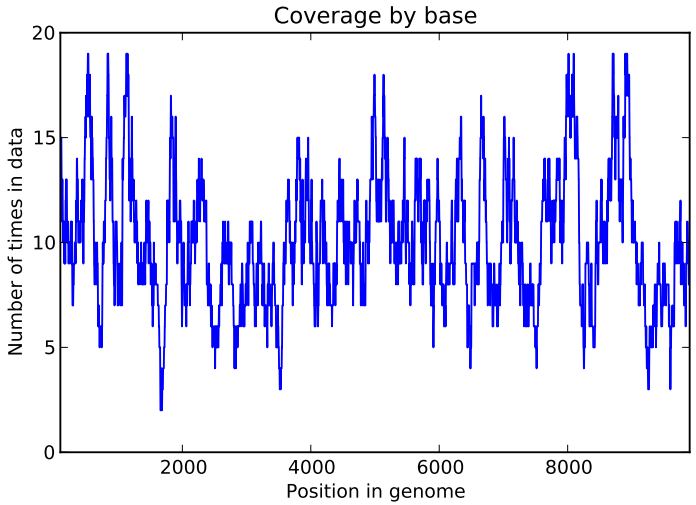
Shotgun sequencing and coverage

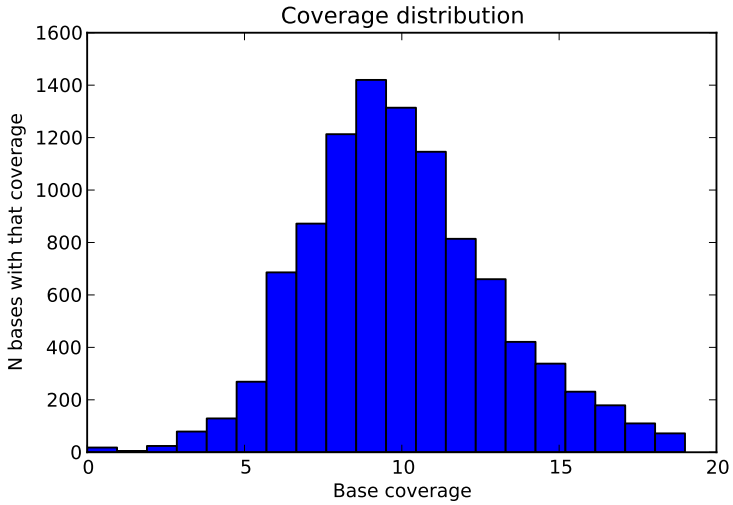


"Coverage" is simply the average number of reads that overlap each true base in genome.

Here, the coverage is ~10 – just draw a line straight down from the top through all of the reads.

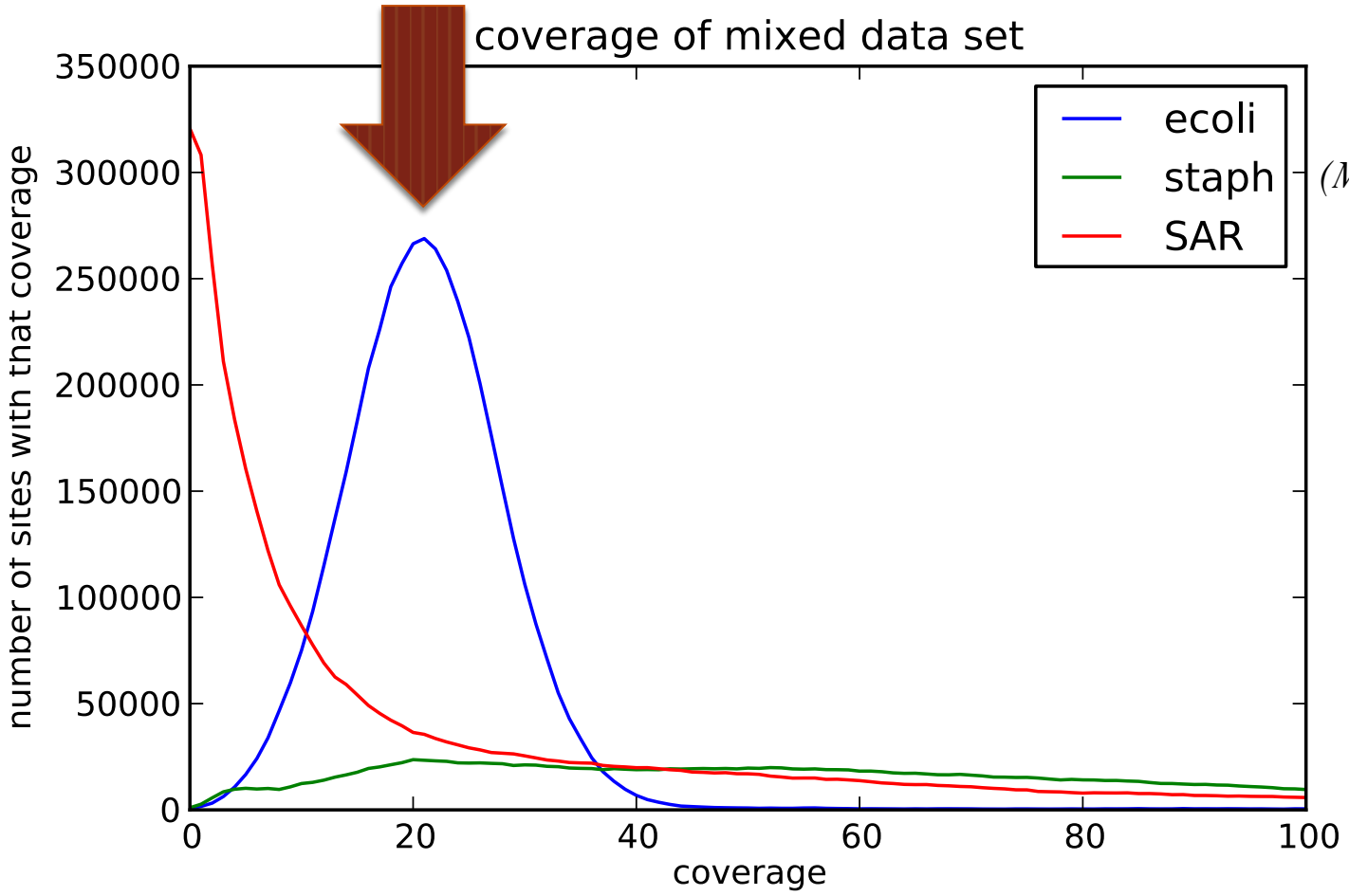
Random sampling => deep sampling needed





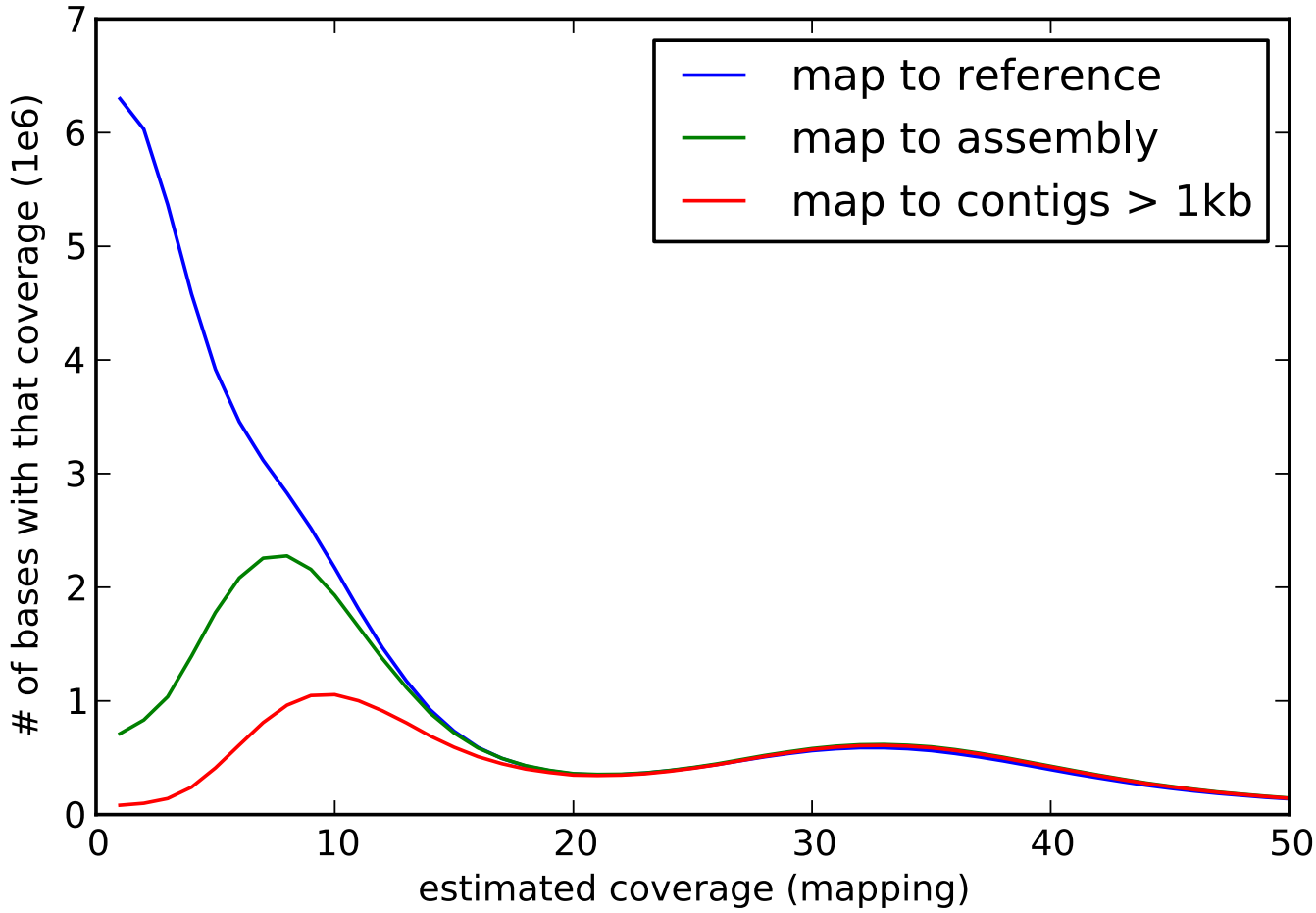
Typically 10-100x needed for robust recovery (300 Gbp for human)

Coverage distribution matters!



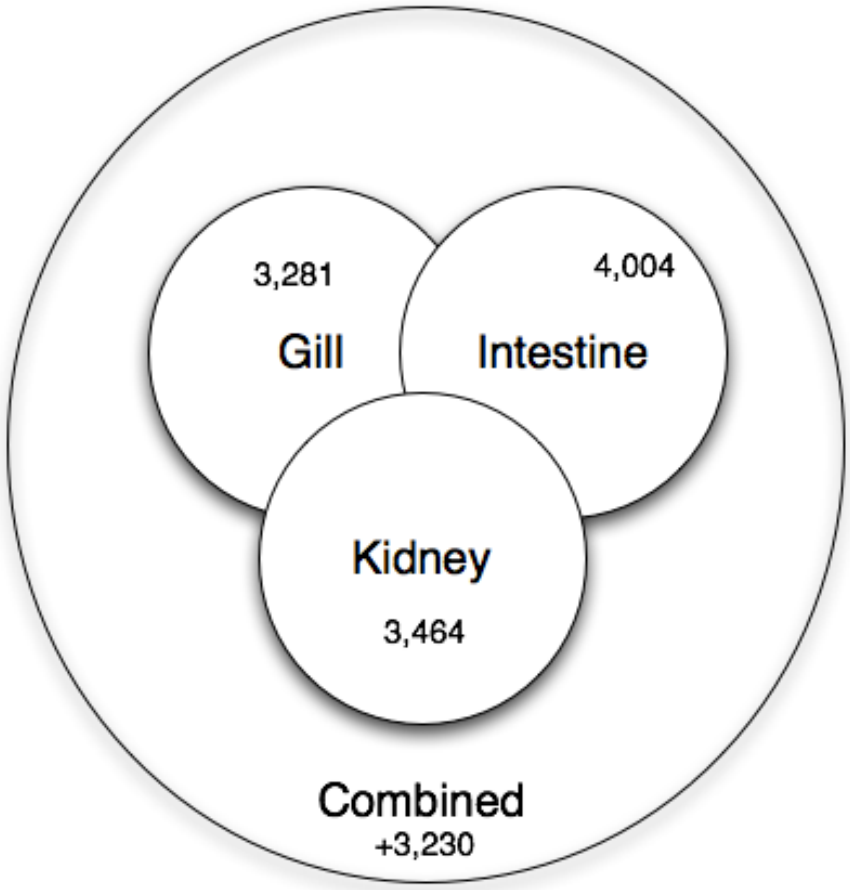
(MD amplified)

Assembly depends on high coverage



Co-assembly is important for sensitivity

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



K-mer based assemblers scale poorly

Why do big data sets require big machines??

Memory usage ~ "real" variation + number of errors

Number of errors ~ size of data set

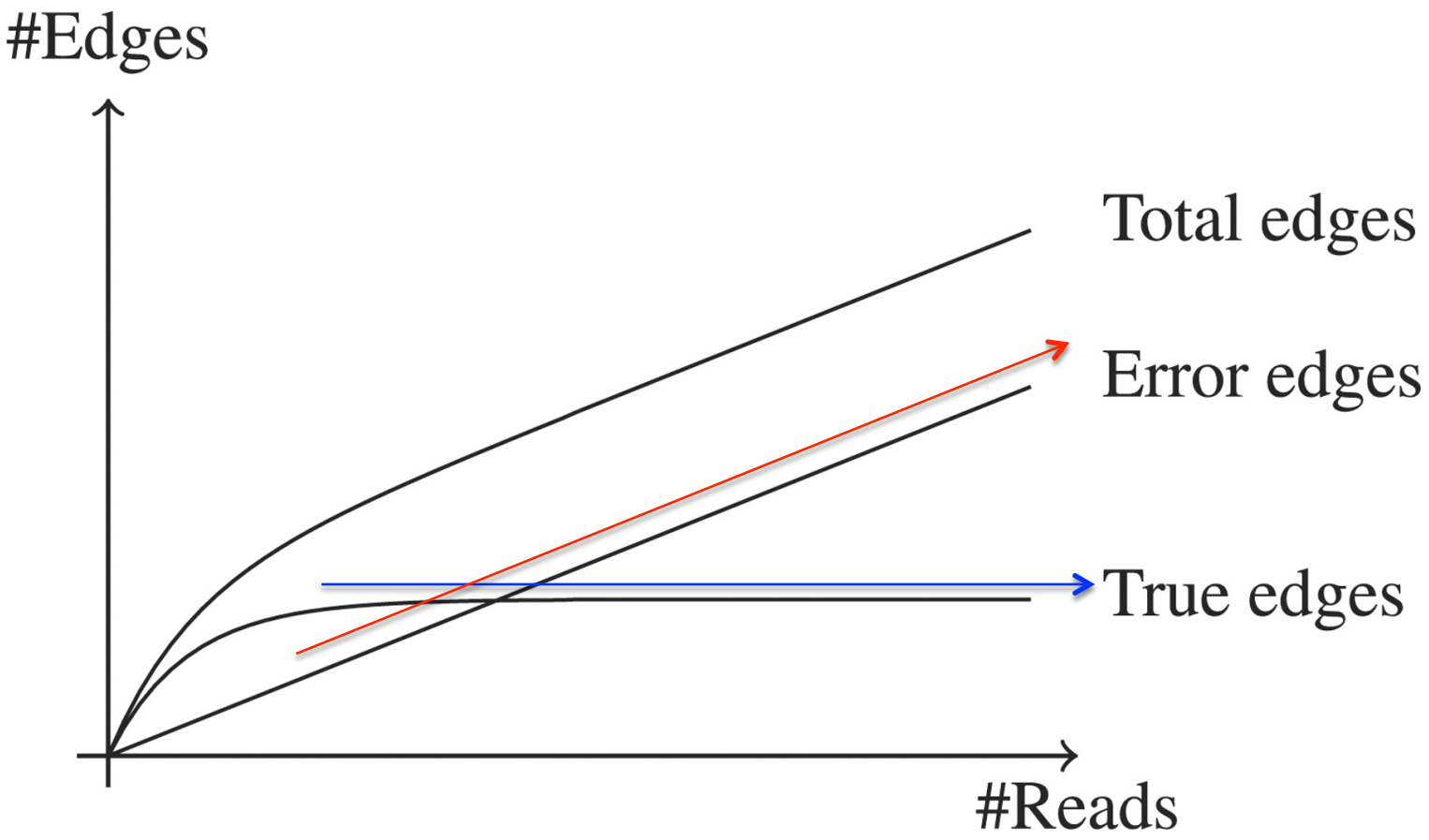
GCGTCAGGTAG**C**AGACCACCGCCATGGCGACGATG

GCGTCAGGTAGGAGACCACCG**T**CATGGCGACGATG

GCGT**T**AGGTAGGAGACCACCGCCATGGCGACGATG

GCGTCAGGTAGGAGACC**G**CCGCCATGGCGACGATG

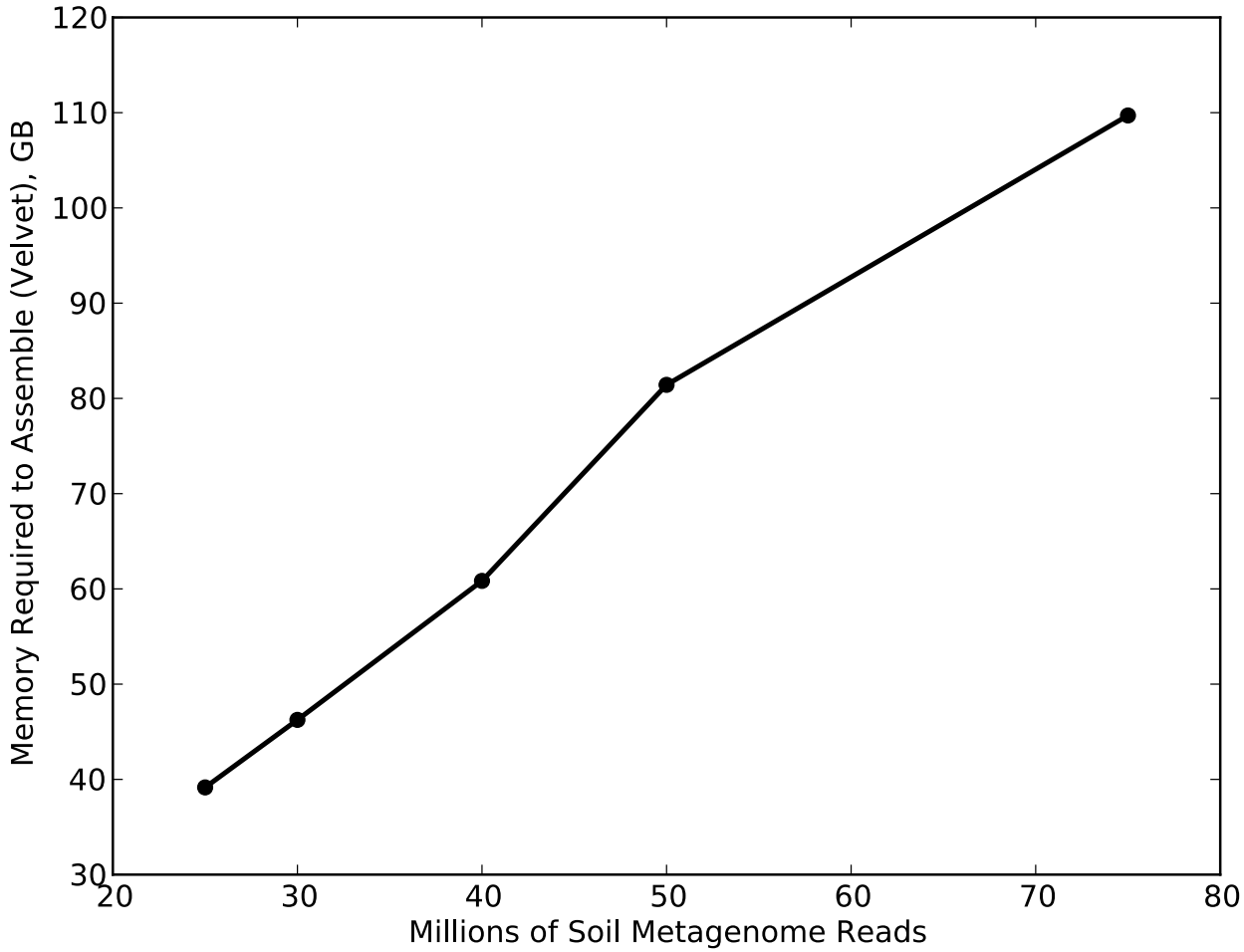
De Bruijn graphs scale poorly with data size



Conway T C , Bromage A J Bioinformatics 2011;27:479-486

Bioinformatics

Practical memory measurements



Velvet measurements (Adina Howe)

How much data do we need? (I)

("More" is rather vague…)

Assembly results for Iowa corn and prairie

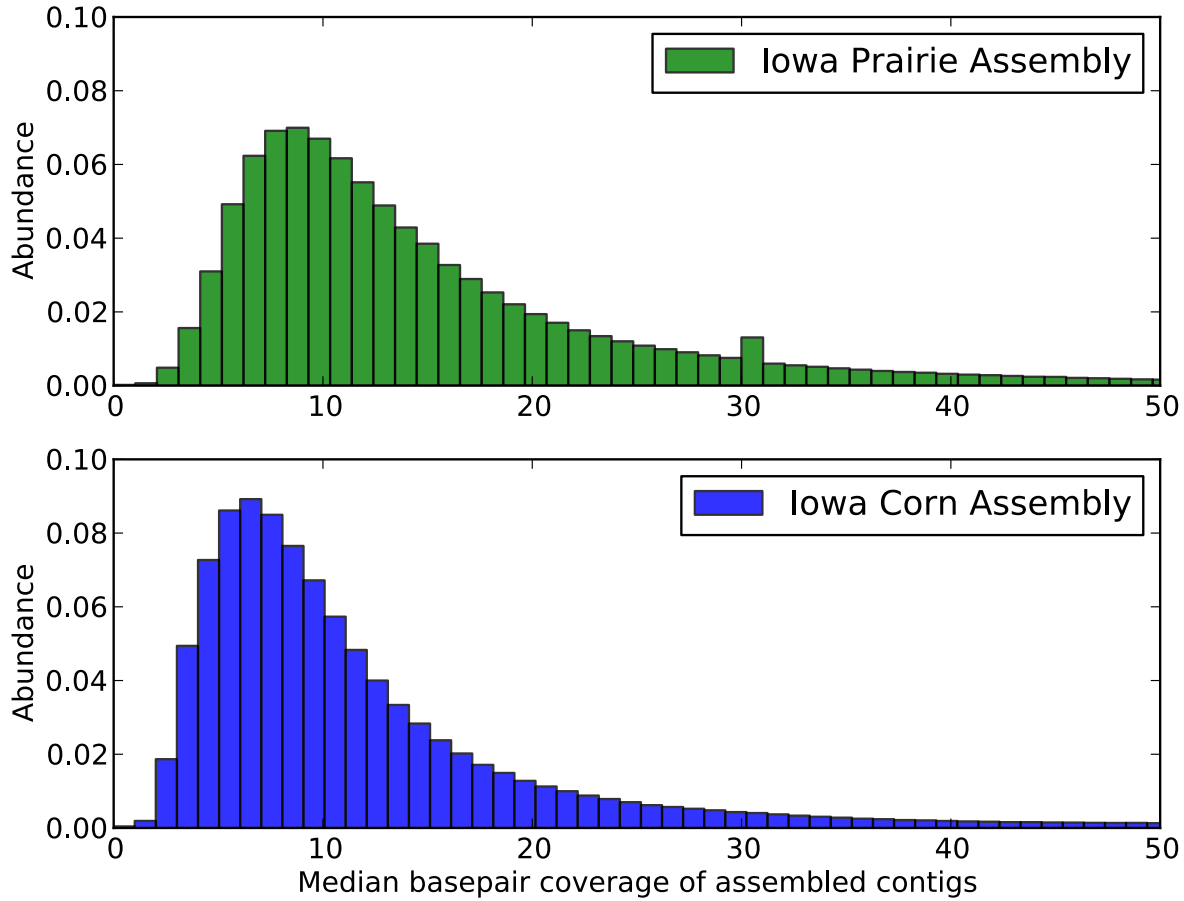
(2x ~300 Gbp soil metagenomes)

Total Assembly	Total Contigs (> 300 bp)	% Reads Assembled	Predicted protein coding
2.5 bill	4.5 mill	19%	5.3 mill
3.5 bill	5.9 mill	22%	6.8 mill

Putting it in perspective:
Total equivalent of ~1200 bacterial genomes
Human genome ~3 billion bp

Adina Howe

Resulting contigs are low coverage.



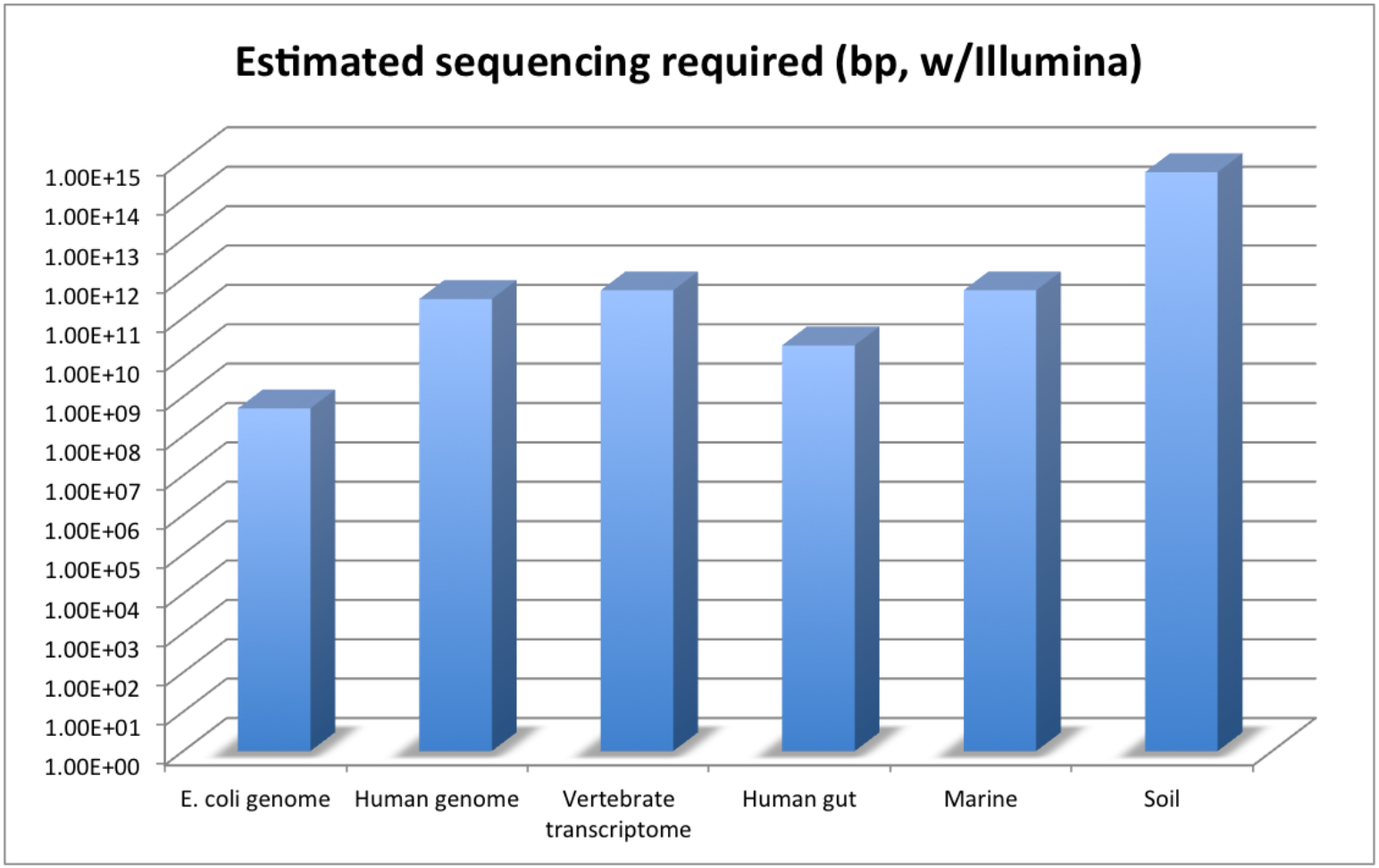
How much? (II)

- Suppose we need 10x coverage to assemble a microbial genome, and microbial genomes average 5e6 bp of DNA.
- Further suppose that we want to be able to assemble a microbial species that is "1 in a 100000", i.e. 1 in 1e5.
- Shotgun sequencing samples *randomly*, so must sample *deeply* to be sensitive.

10x coverage x 5e6 bp x 1e5 =~ 50e11, or 5 Tbp of sequence.

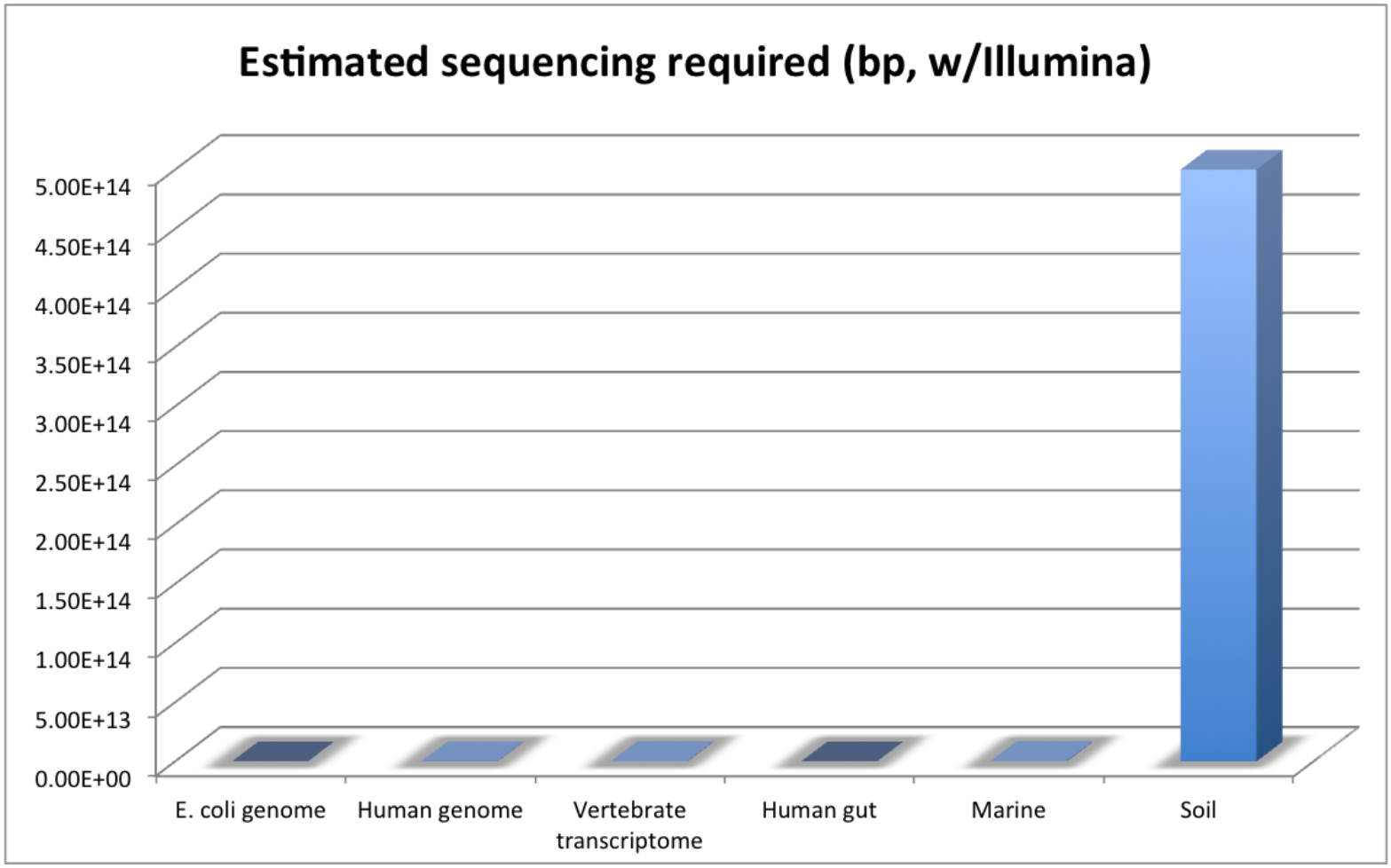
Currently this would cost approximately $100k, for 10 full Illumina runs, but in a year we will be able to do it for much less.

We can estimate sequencing req'd:



http://ivory.idyll.org/blog/how-much-sequencing-is-needed.html

“Whoa, that’s a lot of data…”



http://ivory.idyll.org/blog/how-much-sequencing-is-needed.html

Some approximate metagenome sizes

- Deep carbon mine data set: 60 Mbp (x 10x => 600 Mbp)
- Great Prairie soil: 12 Gbp (4x human genome)
- Amazon Rain Forest Microbial Observatory soil: 26 Gbp

How can we scale assembly!?

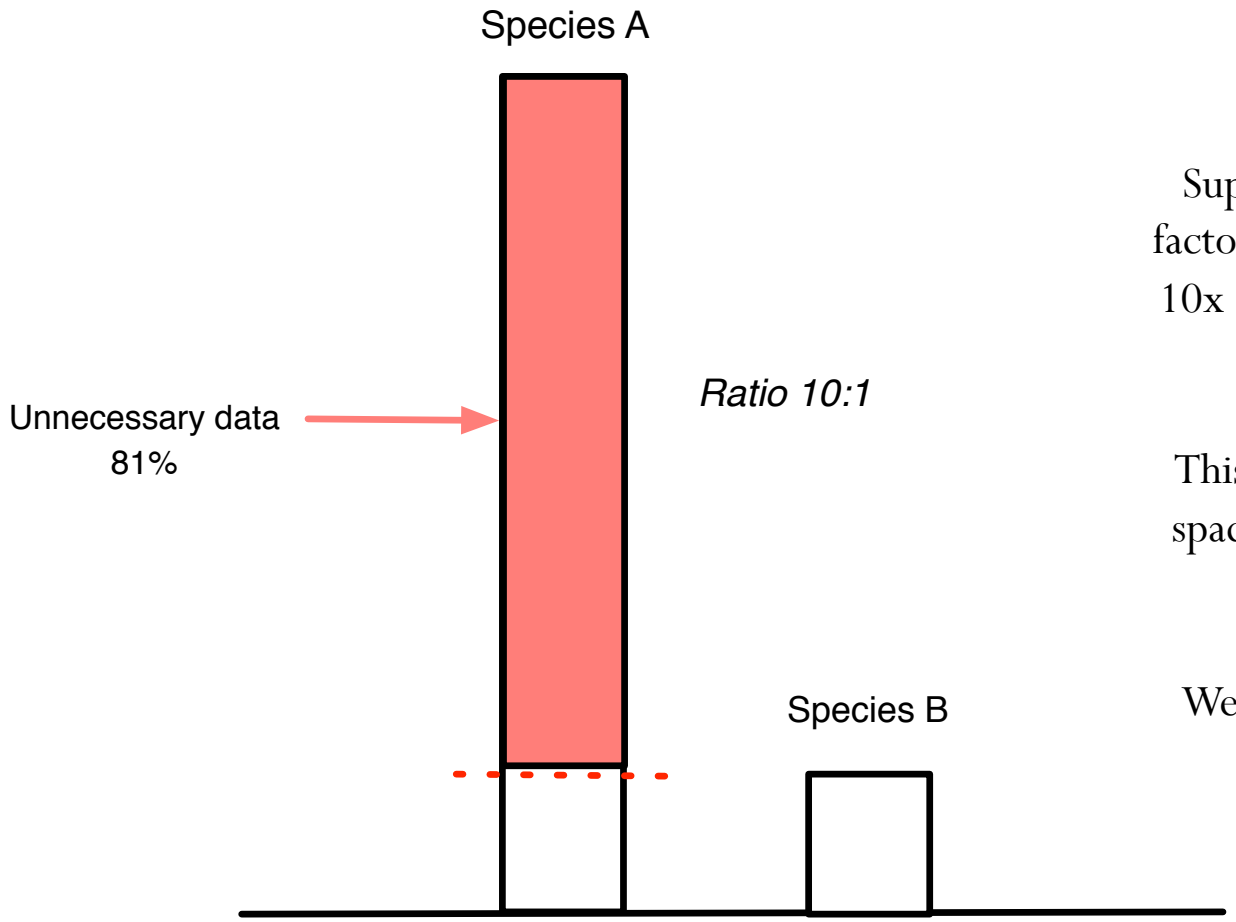
- We have developed two **prefiltering** approaches.
- Essentially, we preprocess your reads and (a) normalize their coverage and (b) subdivide them by graph partition.

"We take your reads and make them better! Satisfaction guaranteed or your money back!"*

* Terms may not apply to NSF, NIH, and USDA funding bodies.

Approach I: Digital normalization

(a computational version of library normalization)

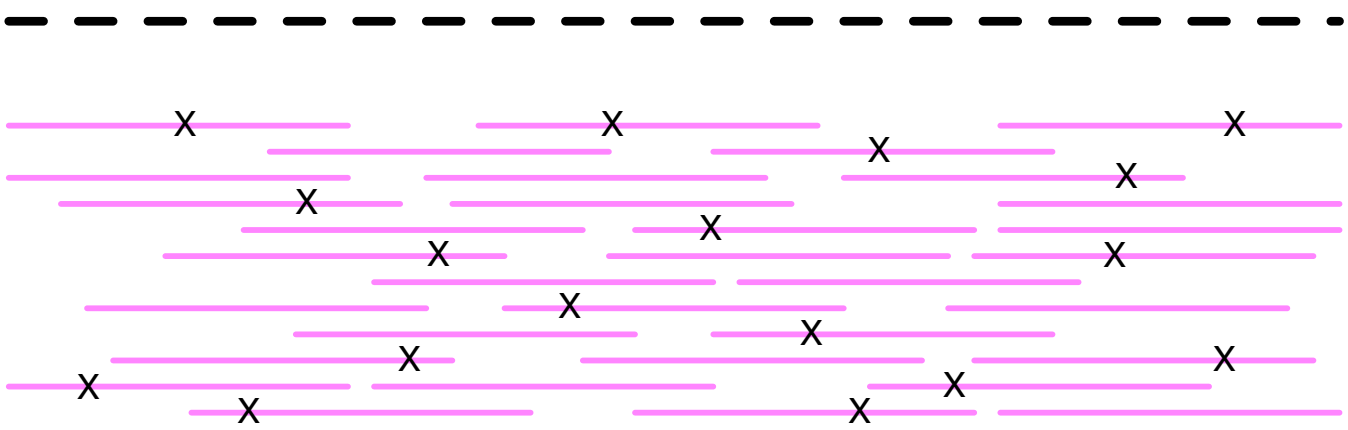


Suppose you have a dilution factor of A (10) to B(1). To get 10x of B you need to get 100x of A! Overkill!!

This 100x will consume disk space and, because of errors, **memory**.

We can discard it for you…

We only *need* ~5x at each point.



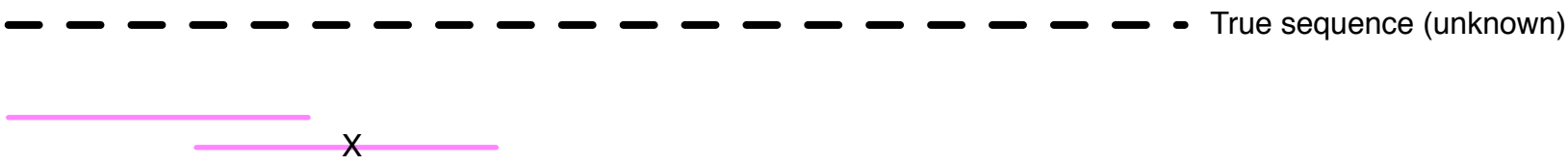
"Coverage" is simply the average number of reads that overlap each true base in genome.

Here, the coverage is ~10 – just draw a line straight down from the top through all of the reads.

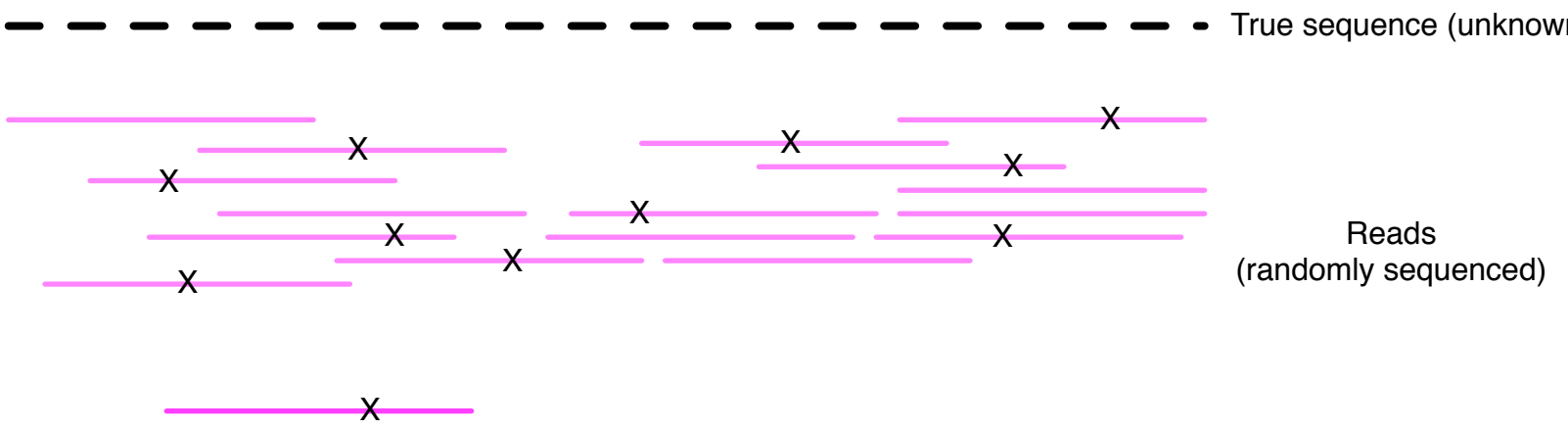
Digital normalization

Reads
(randomly sequenced)

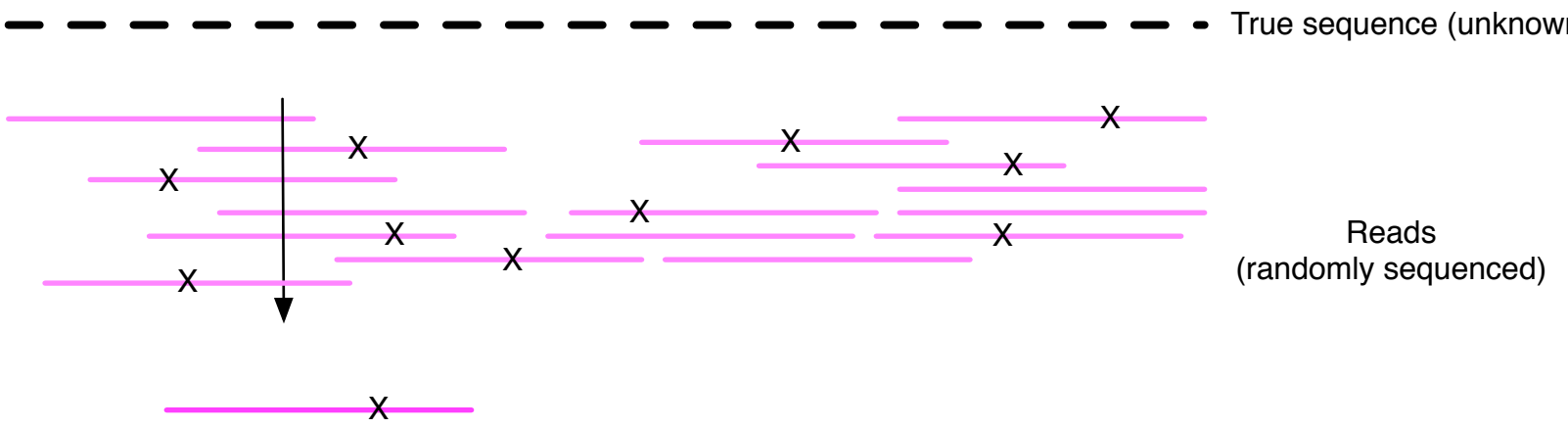
Digital normalization



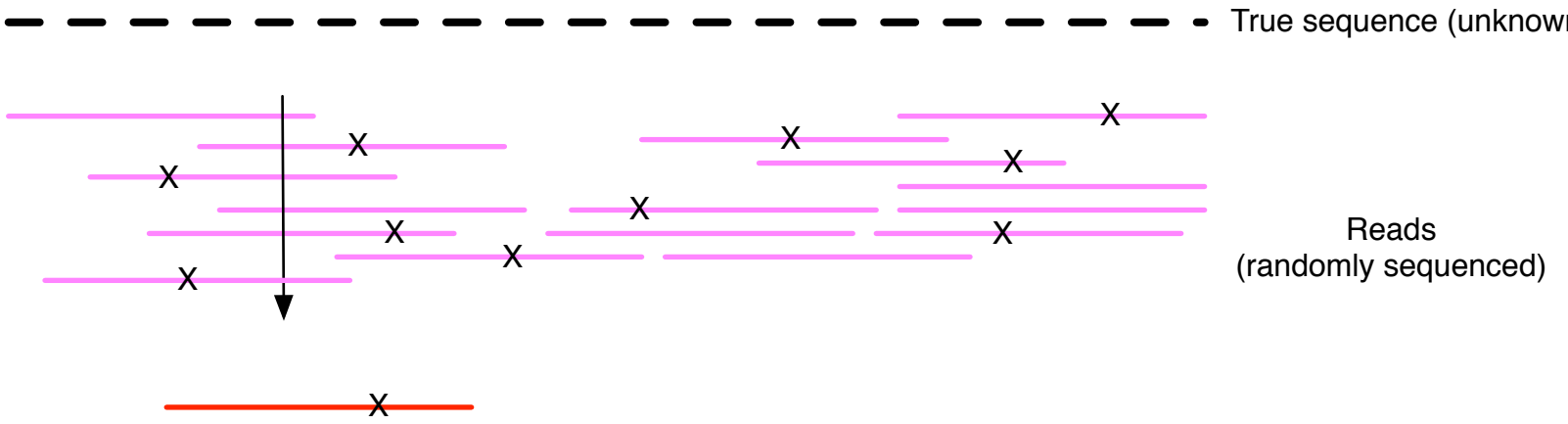
Digital normalization



Digital normalization

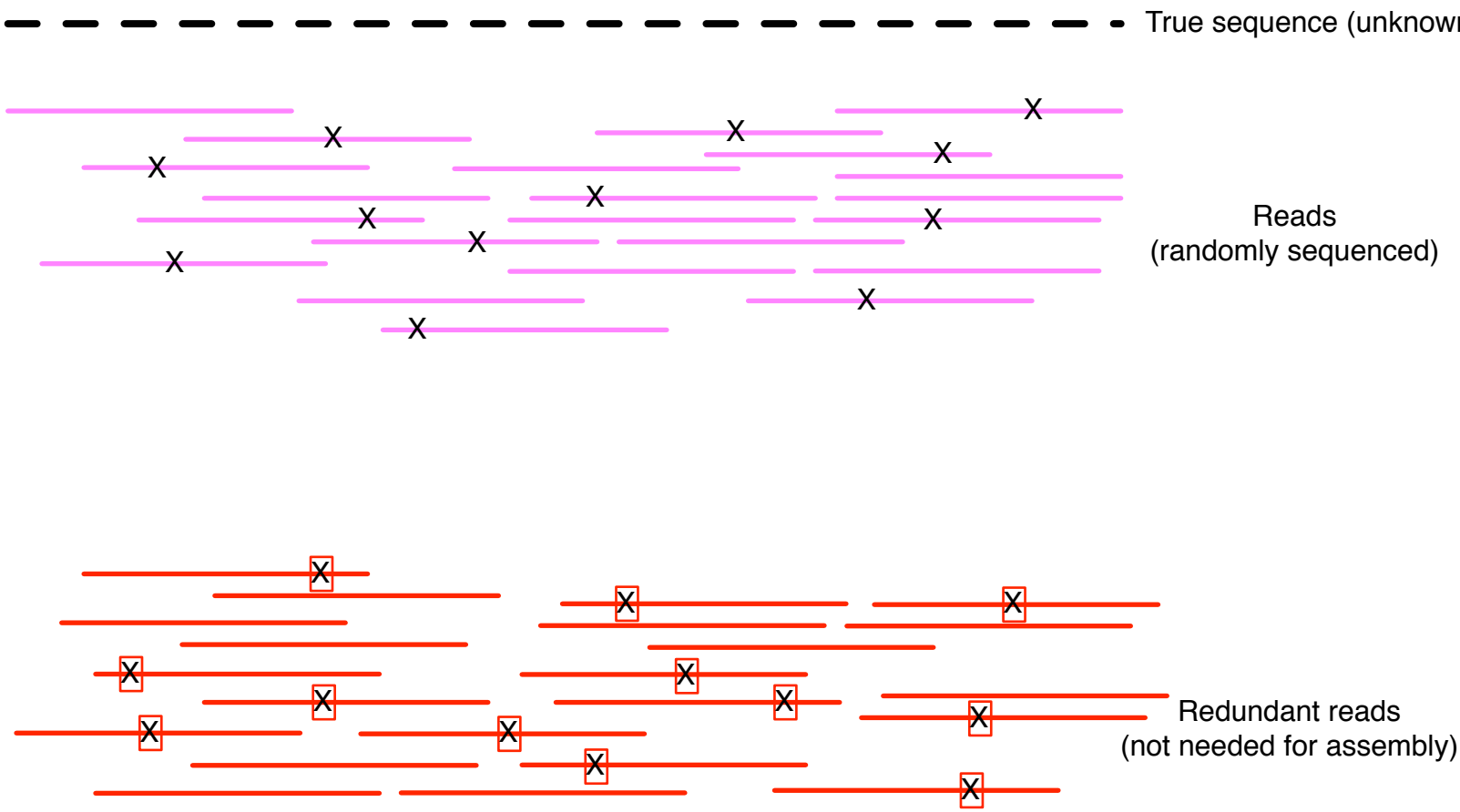


Digital normalization

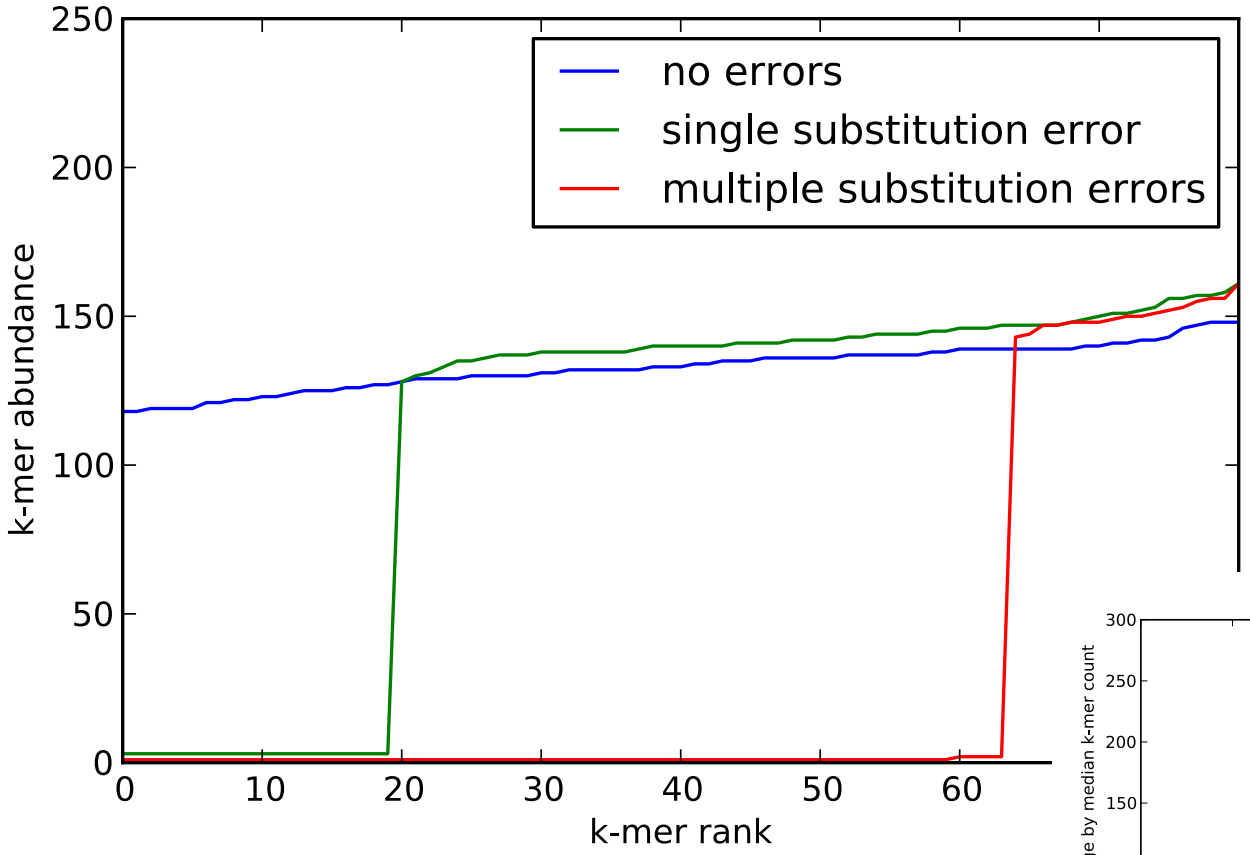


If next read is from a high coverage region - ***discard***

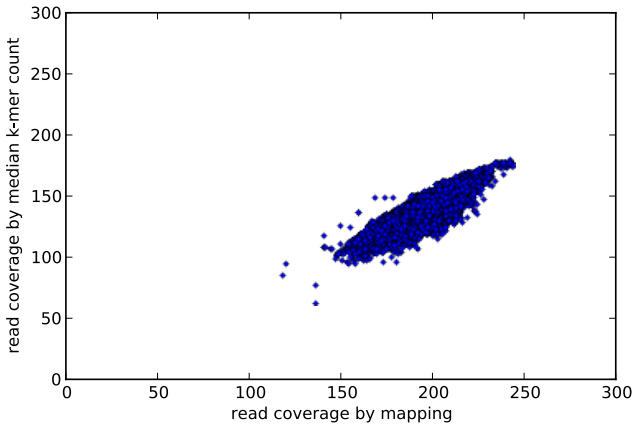
Digital normalization



A read's *median k-mer count* is a good estimator of "coverage".



This gives us a **reference-free** measure of coverage.

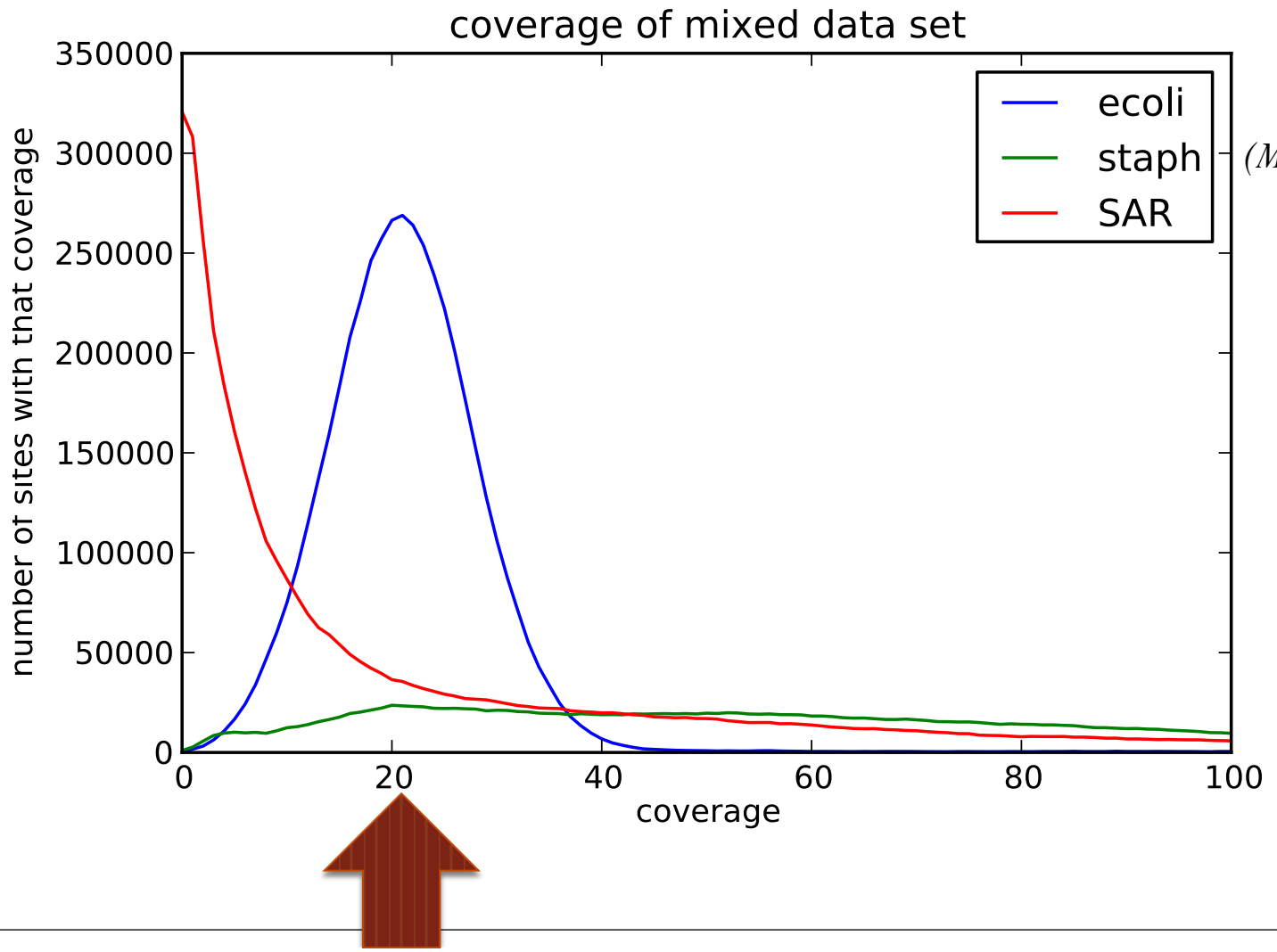


Digital normalization approach

A *digital* analog to cDNA library normalization, diginorm:

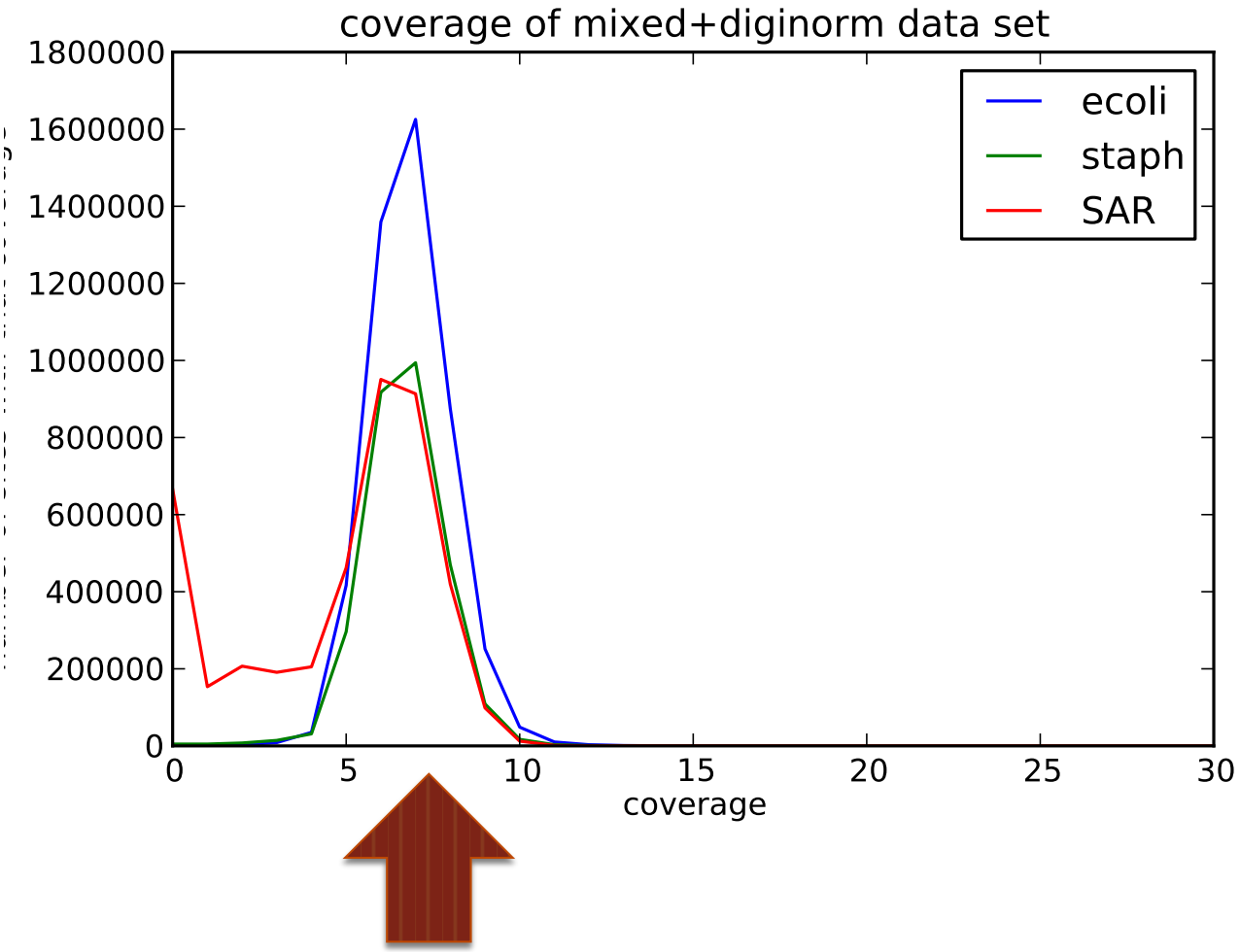
- Is single pass: looks at each read only once;
- Does not "collect" the majority of errors;
- Keeps all low-coverage reads;
- Smooths out coverage of regions.

Coverage before digital normalization:



(MD amplified)

Coverage after digital normalization:



Normalizes coverage

Discards redundancy

Eliminates majority of errors

Scales assembly dramatically.

Assembly is 98% identical.

Digital normalization approach

A *digital* analog to cDNA library normalization, diginorm is a read prefiltering approach that:

- Is single pass: looks at each read only once;
- Does not "collect" the majority of errors;
- Keeps all low-coverage reads;
- Smooths out coverage of regions.

Contig assembly is significantly more efficient and now scales with underlying genome size

Table 3. Three-pass digital normalization reduces computational requirements for contig assembly of genomic data.

Data set	N reads pre/post	Assembly time pre/post	Assembly memory pre/post
E. coli	31m / 0.6m	1040s / 63s (16.5x)	11.2gb / 0.5 gb (22.4x)
S. aureus single-cell	58m / 0.3m	5352s / 35s (153x)	54.4gb / 0.4gb (136x)
Deltaproteobacteria single-cell	67m / 0.4m	4749s / 26s (182.7x)	52.7gb / 0.4gb (131.8x)

- Transcriptomes, microbial genomes incl MDA, and most metagenomes can be assembled in under 50 GB of RAM, with identical or *improved* results.

Digital normalization retains information, while discarding data and errors

Table 1. Digital normalization to C=20 removes many erroneous k-mers from sequencing data sets. Numbers in parentheses indicate number of true k-mers lost at each step, based on reference.

Data set	True 20-mers	20-mers in reads	20-mers at C=20	% reads kept
Simulated genome	399,981	8,162,813	3,052,007 (-2)	19%
Simulated mRNAseq	48,100	2,466,638 (-88)	1,087,916 (-9)	4.1%
E. coli genome	4,542,150	175,627,381 (-152)	90,844,428 (-5)	11%
Yeast mRNAseq	10,631,882	224,847,659 (-683)	10,625,416 (-6,469)	9.3%
Mouse mRNAseq	43,830,642	709,662,624 (-23,196)	43,820,319 (-13,400)	26.4%

Table 2. Three-pass digital normalization removes most erroneous k-mers. Numbers in parentheses indicate number of true k-mers lost at each step, based on known reference.

Data set	True 20-mers	20-mers in reads	20-mers remaining	% reads kept
Simulated genome	399,981	8,162,813	453,588 (-4)	5%
Simulated mRNAseq	48,100	2,466,638 (-88)	182,855 (-351)	1.2%
E. coli genome	4,542,150	175,627,381 (-152)	7,638,175 (-23)	2.1%
Yeast mRNAseq	10,631,882	224,847,659 (-683)	10,532,451 (-99,436)	2.1%
Mouse mRNAseq	43,830,642	709,662,624 (-23,196)	42,350,127 (-1,488,380)	7.1%

Lossy compression

http://en.wikipedia.org/wiki/JPEG

Lossy compression

http://en.wikipedia.org/wiki/JPEG

Lossy compression

http://en.wikipedia.org/wiki/JPEG

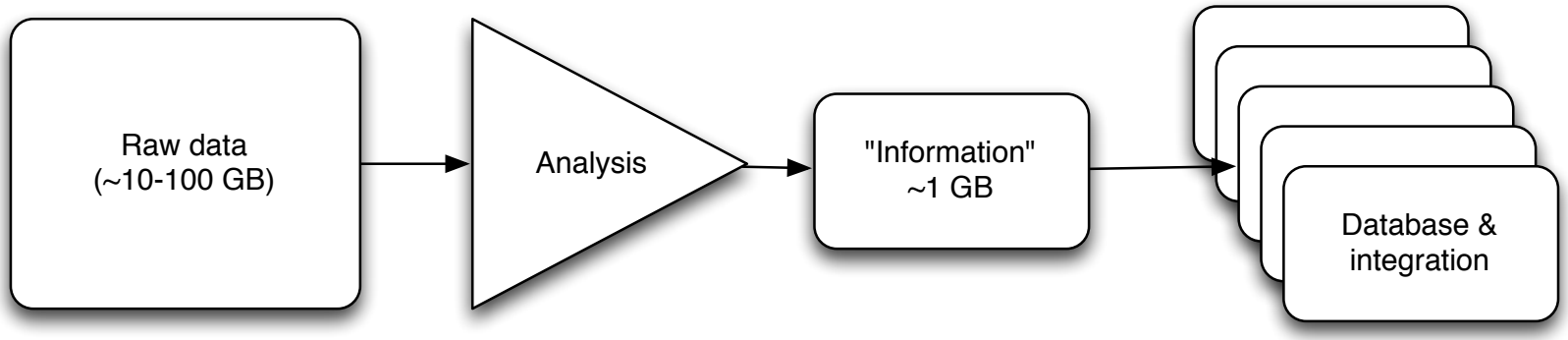
Lossy compression

http://en.wikipedia.org/wiki/JPEG

Lossy compression

http://en.wikipedia.org/wiki/JPEG

~2 GB – 2 TB of single-chassis RAM



We can use lossy compression approaches to make downstream analysis faster and better.

Metagenomes: Data partitioning

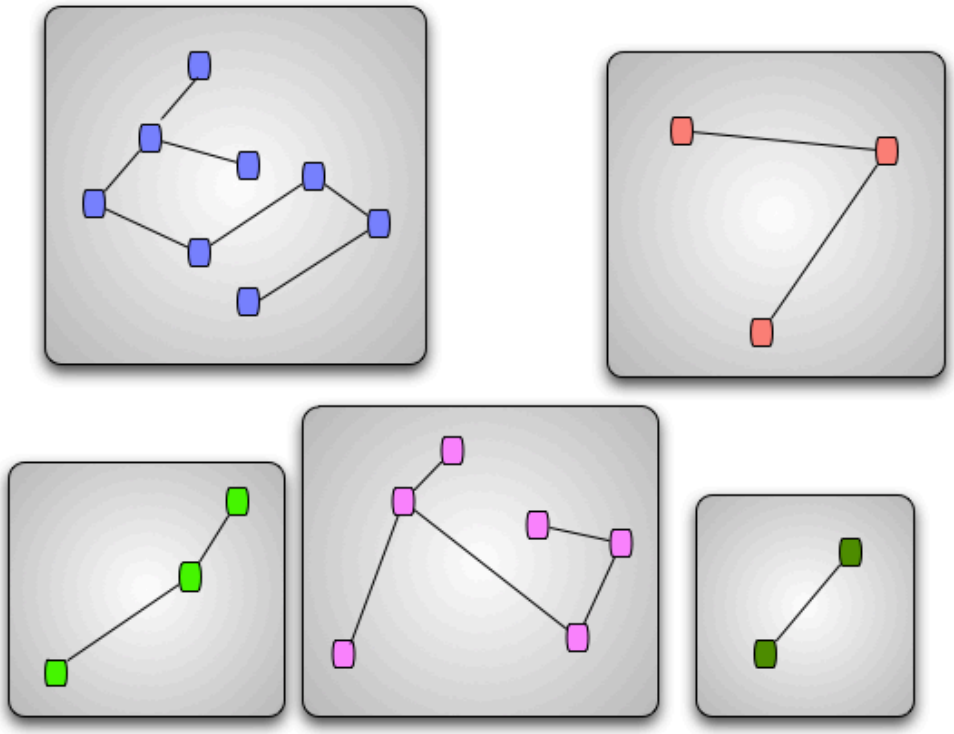
(a computational version of cell sorting)

Split reads into "bins" belonging to different source species.

Can do this based almost entirely on *connectivity* of sequences.

"Divide and conquer"

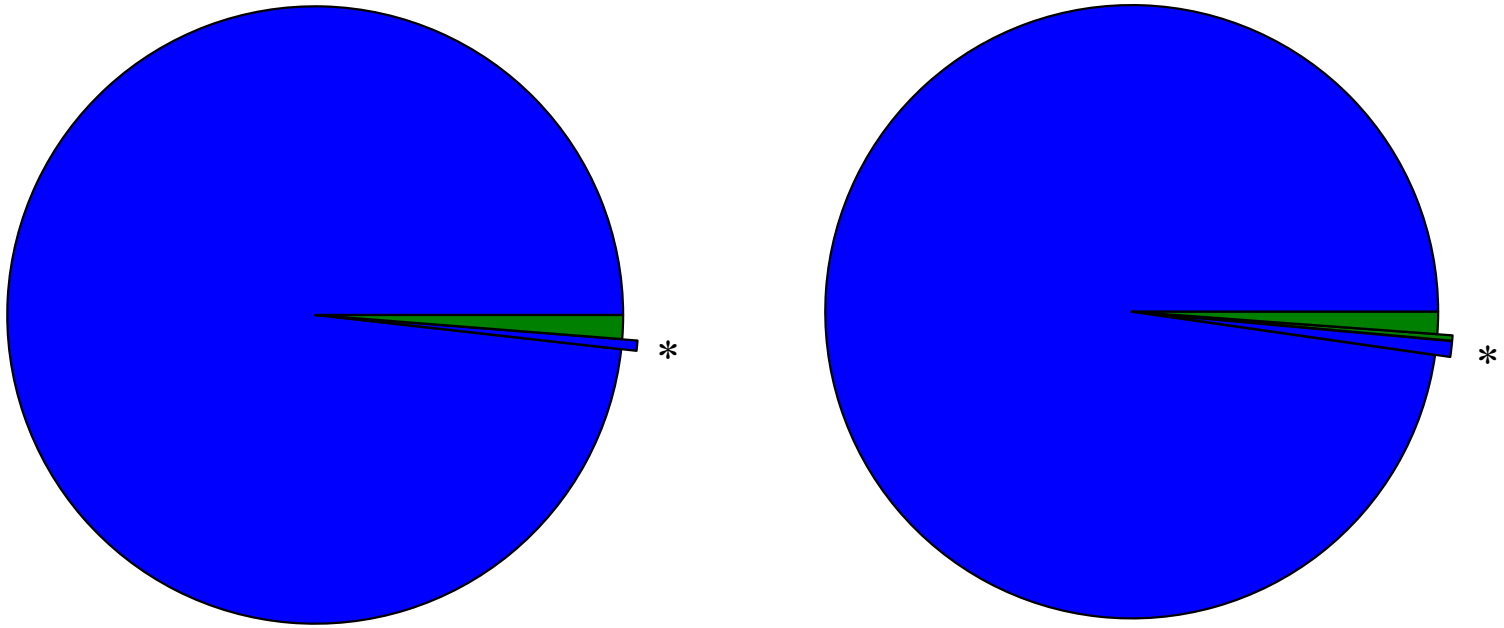
Memory-efficient implementation helps to scale assembly.



Pell et al., 2012, PNAS

Partitioning separates reads by genome.

When computationally spiking HMP mock data with one *E. coli* genome (left) or multiple *E. coli* strains (right), majority of partitions contain reads from only a single genome (blue) vs multi-genome partitions (green).



Partitions containing spiked data indicated with a *

Adina Howe

Partitioning: Technical challenges met (and defeated)

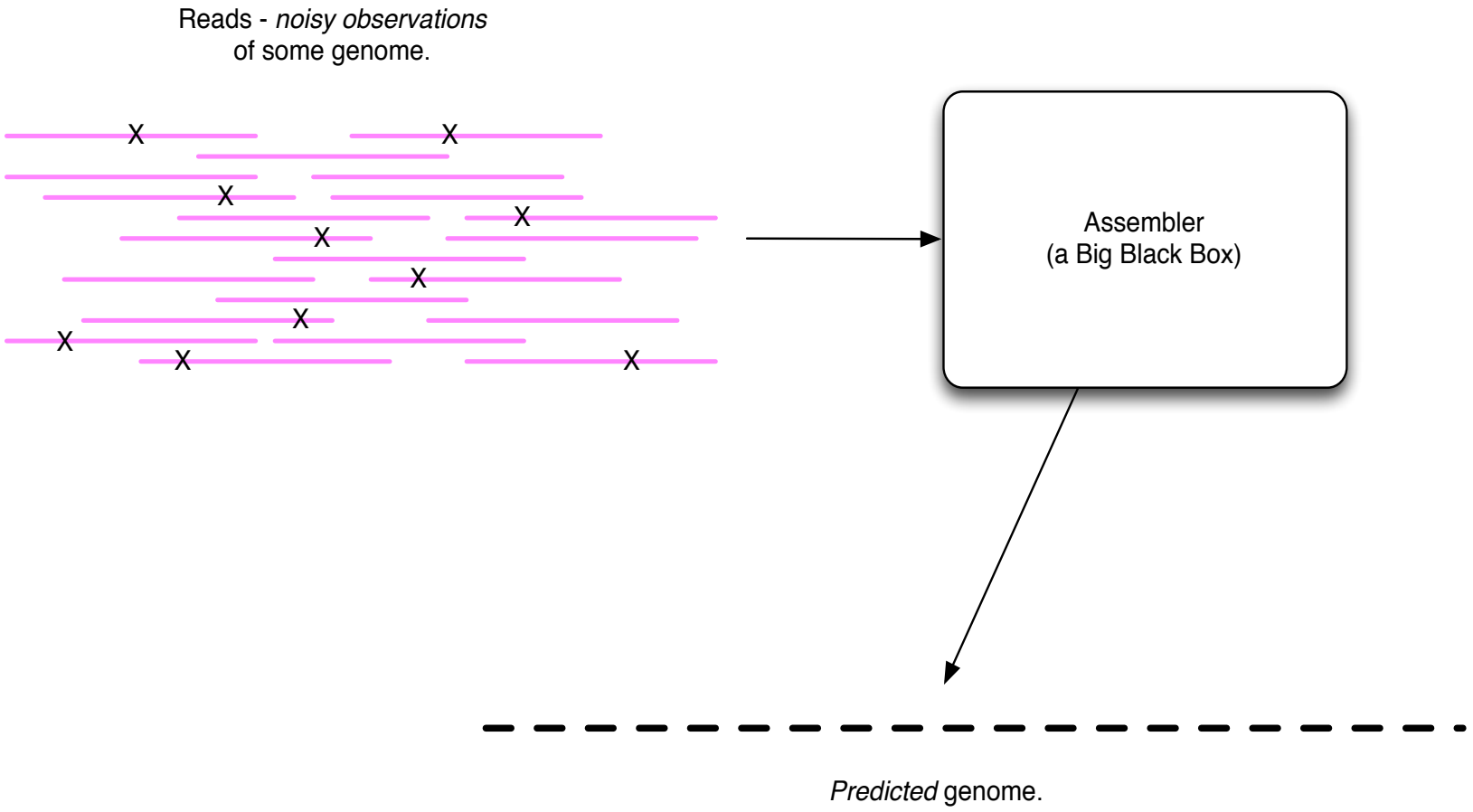
- Novel data structure properties elucidated via percolation theory analysis (Pell et al., 2012, PNAS).
- Exhaustive in-memory traversal of graphs containing 5-15 billion nodes.
- Sequencing technology introduces false sequences in graph (Howe et al., submitted.)
- Only 20x improvement in assembly scaling ☹.

Is your assembly good?

- **Truly reference-free assembly is hard to evaluate.**

- Traditional "genome" measures like N50 and NG50 simply do not apply to metagenomes, because very often you don't know what the genome "size" is.

Evaluating assembly



Evaluating correctness of metagenomes is still undiscovered country.

Evaluating assemblies

- Every assembly returns different results *for eukaryotic genomes.*

- For metagenomes, it's even worse.
  - No systematic exploration of precision, recall, etc.
  - Very little in the way of cross-comparable data sets
  - Often sequencing technology being evaluated is out of date
  - etc. etc.

Our experience

- Our metagenome assemblies compare well with others, but we have little in the way of ground truth with which to evaluate.

- Scaffold assembly is tricky; we believe in contig assembly for metagenomes, but not scaffolding.

- See arXiv paper, "Assembling large, complex metagenomes", for our suggested pipeline and statistics & references.

Metagenomic assemblies are highly variable

Table 2: Total number of contigs, assembly length, and maximum contig size was estimated for metagenomic datasets with multiple assemblers, as well as memory and time requirements of unfiltered read assembly (UF). Filtered reads (F) were processed in 24 GB of memory, and after filtering required less than 2 GB of memory to assemble. Velvet assemblies of the unfiltered human gut and large soil datasets (marked as *) could only be completed with K=33 due to computational limitations. The Meta-IDBA assembly of the large soil metagenome could not be completed in less than 100 GB.

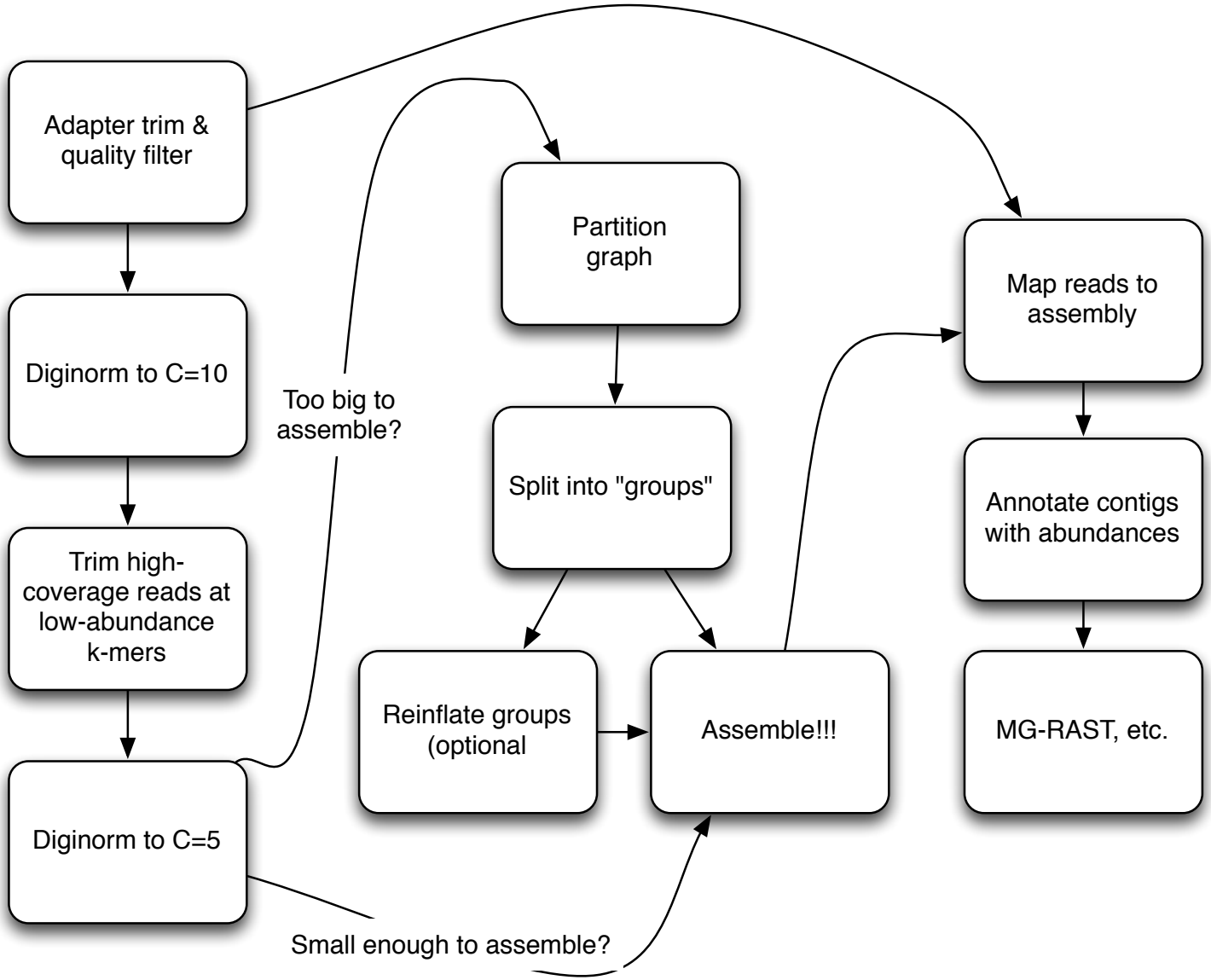
	UF Assembly (contigs / length / max size)	F Assembly (contigs / length / max size)	UF Requirements Memory (GB)/Time (h)
Velvet			
Small Soil	25,470 / 16,269,879 / 118,753	17,636 / 10,578,908 / 13,246	5 / 4
Medium Soil	113,613 / 81,660,678 / 57,856	79,654 / 54,424,264 / 23,663	18 / 21
Large Soil	554,825 / 306,899,884 / 41,217	290,018 / 159,960,062 / 41,423	33 / 12*
Rumen	92,044 / 74,813,072 / 182,003	72,705 / 49,518,627 / 34,683	11 / 14
Human Gut	543,331 / 234,686,983 / 85,596	203,299 / 181,934,800 / 145,740	76 / 8*
Simulated	11,204 / 6,506,248 / 5,151	9,859 / 5,463,067 / 6,605	<1 / <1
MetaIDBA			
Small Soil	15,739 / 9,133,564 / 37,738	12,513 / 7,012,036 / 17,048	<1 / <1
Medium Soil	76,269 / 45,844,975 / 37,738	52,978 / 30,040,031 / 18,882	2 / 2
Large Soil	395,122 / 228,857,098 / 37,738	N/A	>116 / incomplete
Rumen	60,330 / 47,984,619 / 54,407	48,940 / 33,276,502 / 22,083	12 / 3
Human Gut	173,432 / 211,067,996 / 106,503	132,614 / 142,139,101 / 85,539	58 / 15
Simulated	8,707 / 4,698,575 / 5,113	7,726 / 4,078,947 / 3,845	<1 / <1
SOAPdenovo			
Small Soil	14,275 / 7,100,052 / 37,720	12,801 / 6,343,110 / 13,246	3 / <1
Medium Soil	66,640 / 33,321,411 / 28,695	56,023 / 27,880,293 / 15,721	10 / <1
Large Soil	412,059 / 215,614,765 / 32,514	334,319 / 171,718,154 / 41,423	48 / 11
Rumen	62,896 / 40,792,029 / 22,875	55,975 / 34,540,861 / 19,044	5 / <1
Human Gut	190,963 / 171,502,574 / 57,803	161,795 / 139,686,630 / 56,034	35 / 5
Simulated	6,322 / 2,940,509 / 3,786	6,029 / 2,821,631 / 3,764	<1 / <1

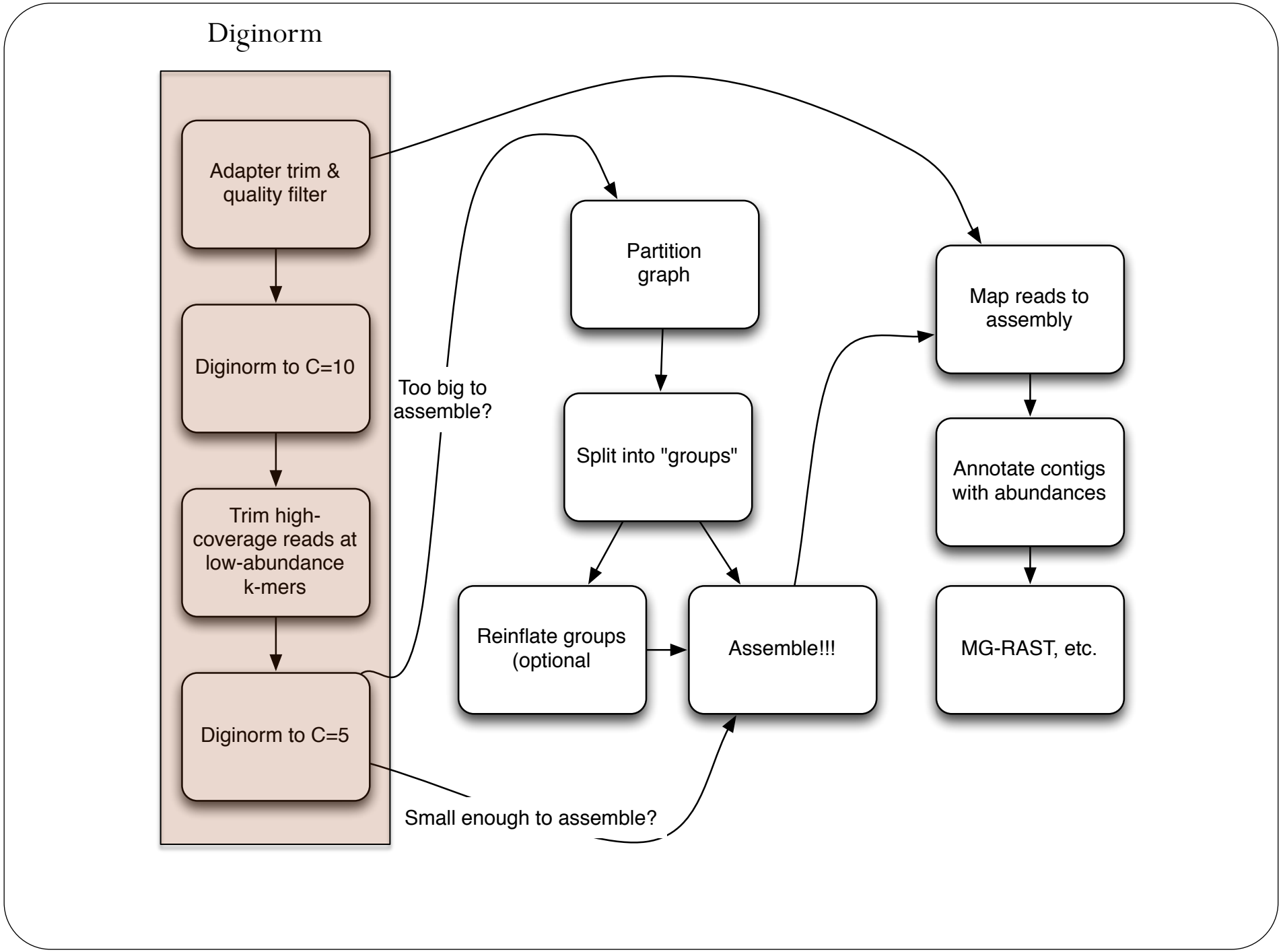
Adina Howe et al., arXiv 1212.0159

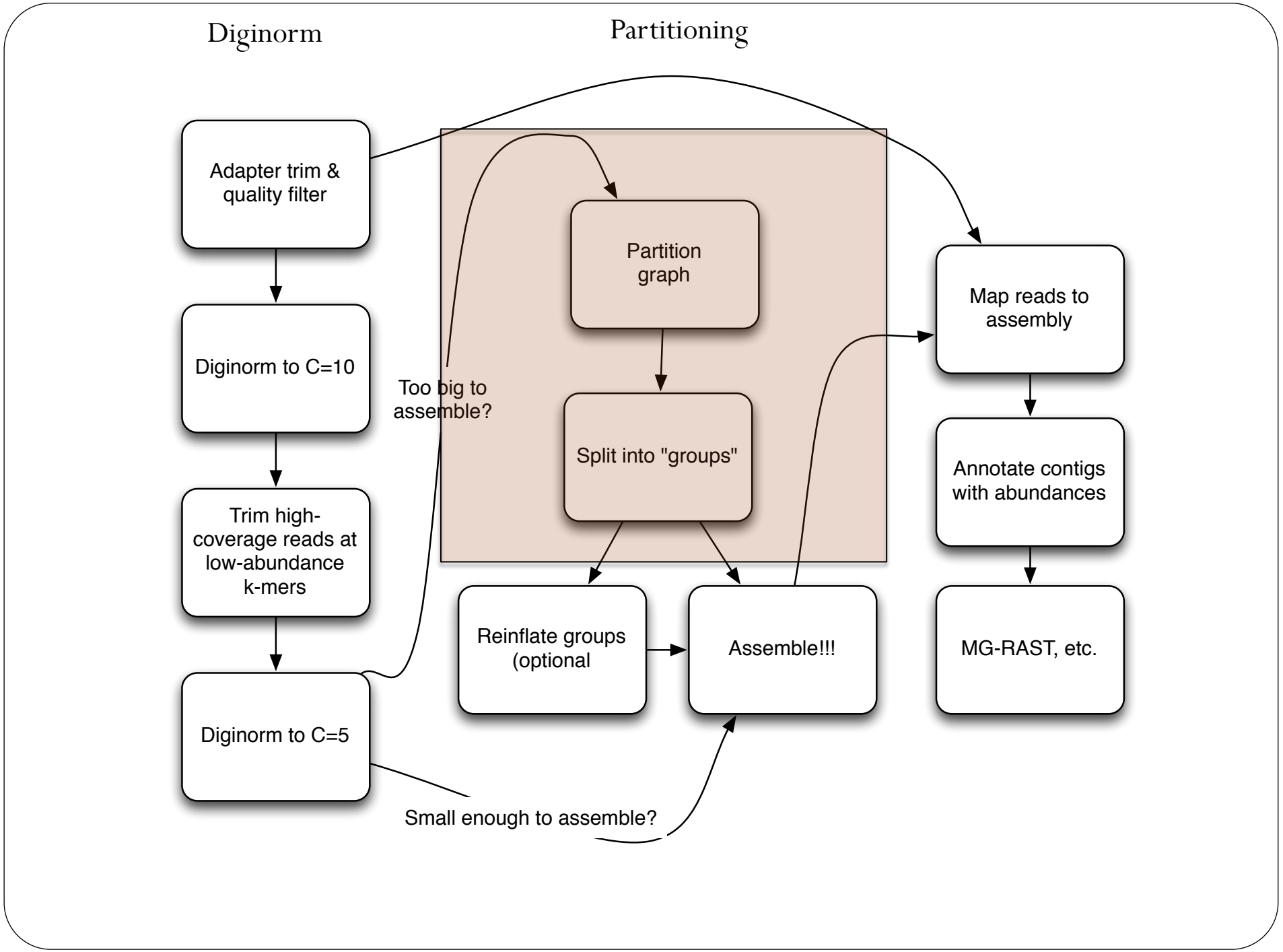
How to choose a metagenome assembler

- Try a few.
- Use what seems to perform best (most bp > some minimum)
- I've heard/read good things about
  - MetaVelvet
  - Ray Meta
  - IDBA-UD
- *Our pipeline doesn't specify an assembler.*

The Kalamazoo Metagenome Assembly Pipeline




Diginorm
Adapter trim & quality filter
Diginorm to C=10
Trim high-coverage reads at low-abundance k-mers
Diginorm to C=5
Too big to assemble?
Small enough to assemble?
Partition graph
Split into "groups"
Reinflate groups (optional
Assemble!!!
Map reads to assembly
Annotate contigs with abundances
MG-RAST, etc.


Diginorm
Partitioning
Adapter trim & quality filter
Diginorm to C=10
Trim high-coverage reads at low-abundance k-mers
Diginorm to C=5
Too big to assemble?
Partition graph
Split into "groups"
Reinflate groups (optional
Assemble!!!
Small enough to assemble?
Map reads to assembly
Annotate contigs with abundances
MG-RAST, etc.

Thoughts on our pipeline

- Should work with any metagenome; very generic approach.

- Diginorm can be decoupled from partitioning;
  - People report that diginorm "just works";
  - Partitioning is trickier and only needed for REALLY BIG data sets.

- Diginorm does interact with some assemblers in a funny way, so suggest starting with Velvet and/or reinflating your partitions.

- This pipeline, esp diginorm part, is *faster* and *lower memory* than any other assembler out there (well, except maybe Minia).

Deep Carbon data set

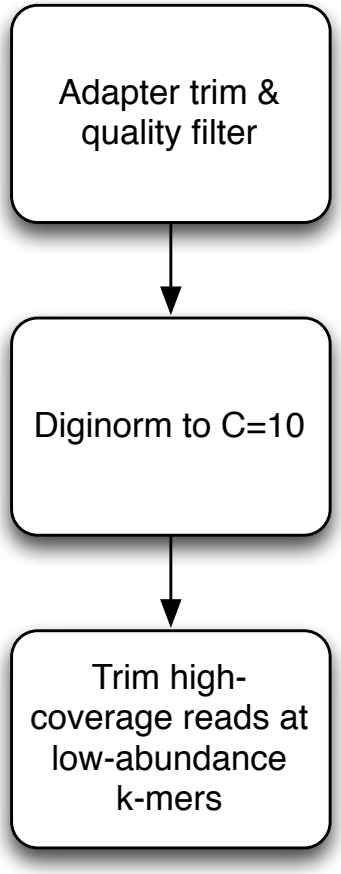
- Name: DCO_TCO_MM5

- Masimong Gold Mine; microbial cells filtered from fracture water from within a 1.9km borehole. (32,000 year old water?!)

- M.C.Y. Lau, C. Magnabosco, S. Grim, G. Lacrampe Couloume, K. Wilkie, B. Sherwood Lollar, D.N. Simkus, G.F. Slater, S. Hendrickson, M. Pullin, T.L. Kieft, O. Kuloyo, B. Linage, G. Borgonie, E. van Heerden, J. Ackerman, C. van Jaarsveld, and T.C. Onstott

DCO_TCO_MM5

20m reads / 2.1 Gbp



5.6m reads / 601.3 Mbp

"Could you take a look at this? MG-RAST is telling us we have a lot of artificially duplicated reads, i.e. the data is bad."

Entire process took ~4 hours of computation, or so.

(Minimum genome size est: 60.1 Mbp)

DCO_TCO_MM5

Assembly stats:

	All contigs		Contigs > 1kb		
k	*N contigs*	*Sum BP*	*N contigs*	*Sum BP*	*Max contig*
21	343263	63217837	6271	10537601	9987
23	302613	63025311	7183	13867885	21348
25	276261	62874727	7375	15303646	34272
27	258073	62500739	7424	16078145	48742
29	242552	62001315	7349	16426147	48746
31	228043	61445912	7307	16864293	48750
33	214559	60744478	7241	17133827	48768
35	203292	60039871	7129	17249351	45446
37	189948	58899828	7088	17527450	59437
39	180754	58146806	7027	17610071	54112
41	172209	57126650	6914	17551789	65207
43	165563	56440648	6925	17654067	73231

(Minimum genome size est: 60.1 Mbp)

DCO_TCO_MM5

Chose two:

- A: k=43 (“long contigs”)
- 165563 contigs
- 56.4 Mbp
- longest contig: **73231** bp

- B: k=43 (“high recall”)
- 343263 contigs
- **63.2** Mbp
- longest contig is 9987 bp

How to evaluate??

How many reads map back?

Mapped 3.8m paired-end reads (one subsample):

- high-recall: 41% of pairs map
- **longer-contigs: 70% of pairs map**

+ 150k single-end reads:

- high-recall: 49% of sequences map
- **longer-contigs: 79% of sequences map**

Annotation/exploration with MG-RAST

- You can upload genome assemblies to MG-RAST, and annotate them with coverage; tutorial to follow.
- What does MG-RAST do?

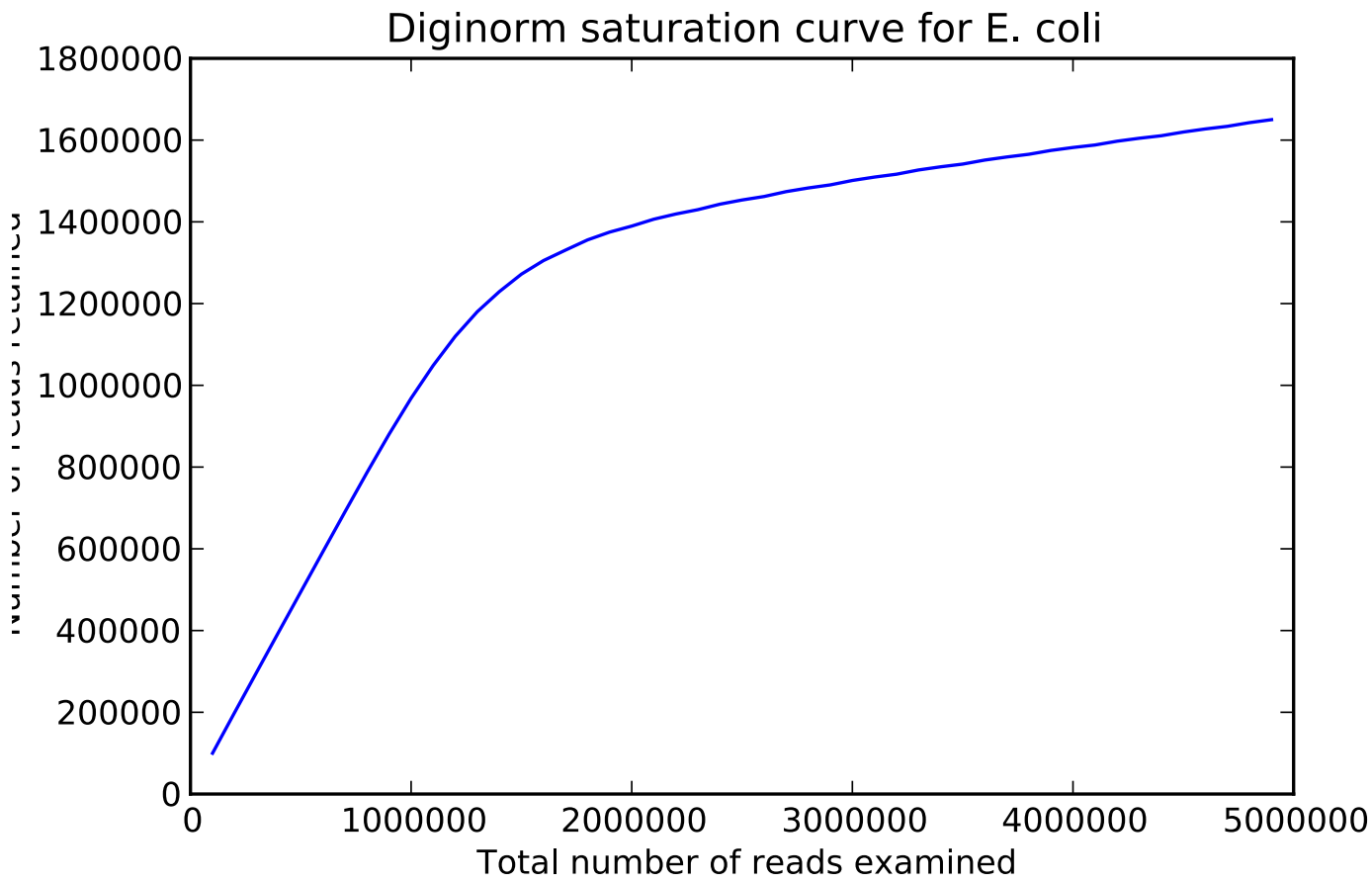
DCO_TCO_MM5

Conclusion

- **This is a pretty good metagenome assembly – > 80% of reads map!**
- Surprised that the larger dataset (6.32 Mbp, "high recall") accounts for a smaller percentage of the reads – 49% vs 79% for the 56.4 Mbp "long contigs" data set.
- I now suspect that different parameters are recovering different subsets of the sample…
- Don't trust MG-RAST ADR calls.

A few notes --

You can estimate metagenome size...



Estimates of metagenome size

Calculation: # reads * (avg read len) / (diginorm coverage)

Assumes: few entirely erroneous reads (upper bound); saturation (lower bound).

- *E. coli:* 384k * 86.1 / 5.0 => 6.6 Mbp est. (true: 4.5 Mbp)
- MM5 deep carbon: 60 Mbp
- Great Prairie soil: 12 Gbp
- Amazon Rain Forest Microbial Observatory: 26 Gbp

Diginorm changes your coverage.

Contigs > 1kb

		DN			*Reinflated*	
k	**N contigs**	**bp**	**longest**	**N contigs**	**bp**	**longest**
21	24	441844	80662	31	439074	79170
23	13	443330	86040	24	437988	80488
25	12	443565	84324	24	426949	84286
27	11	443256	89835	23	385473	89795
29	11	443665	89748	11	285725	89809
31	10	440919	102131	11	286508	89810
33	12	432320	85175	15	282373	85210
35	15	423541	85177	15	276158	85177
37	14	352233	121539	14	278537	85184
39	16	322968	121538	10	276068	85187
41	20	393501	121545	8	278483	85211
43	25	363656	121624	6	278380	121462

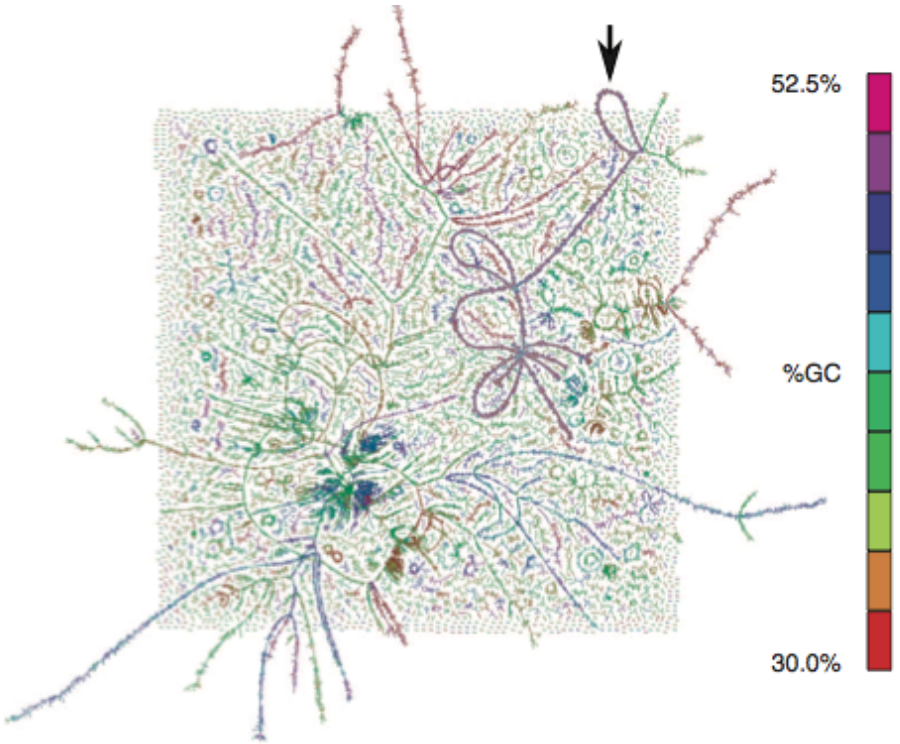
http://ivory.idyll.org/blog/the-k-parameter.html

Extracting whole genomes?

So far, we have only assembled *contigs*, but not whole genomes.

Can entire genomes be assembled from metagenomic data?

Iverson et al. (2012), from the Armbrust lab, contains a technique for scaffolding metagenome contigs into ~whole genomes. **YES.**



Concluding thoughts

- What works?
- What needs work?
- What will work?

What works?

Today,

- From deep metagenomic data, you can get the gene and operon content (including abundance of both) from communities.

- You can get microarray-like expression information from metatranscriptomics.

What needs work?

- Assembling ultra-deep samples is going to require more engineering, but is straightforward. ("Infinite assembly.")

- Building scaffolds and extracting whole genomes has been done, but I am not yet sure how feasible it is to do systematically with existing tools (c.f. Armbrust Lab).

What will work, someday?

- Sensitive analysis of strain variation.
  - Both assembly and mapping approaches do a poor job detecting many kinds of biological novelty.
  - The 1000 Genomes Project has developed some good tools that need to be evaluated on community samples.

- Ecological/evolutionary dynamics in vivo.
  - Most work done on 16s, not on genomes or functional content.
  - Here, sensitivity is really important!

The interpretation challenge

- For soil, we have generated approximately 1200 bacterial genomes worth of assembled genomic DNA from two soil samples.

- The vast majority of this genomic DNA contains unknown genes with largely unknown function.

- Most annotations of gene function & interaction are from a few phylogenetically limited model organisms
  - Est 98% of annotations are computationally inferred: transferred from model organisms to genomic sequence, using homology.
  - Can these annotations be transferred? (Probably not.)

This will be the biggest sequence analysis challenge of the next 50 years.

What are future needs?

- High-quality, medium+ throughput annotation of genomes?
  - Extrapolating from model organisms is both immensely important and yet lacking.
  - Strong phylogenetic sampling bias in existing annotations.

- Synthetic biology for investigating non-model organisms?

  (Cleverness in experimental biology doesn't scale ☹)

- Integration of microbiology, community ecology/evolution modeling, and data analysis.

Papers on our work.

- 2012 PNAS, Pell et al., pmid 22847406 (partitioning).
- Submitted, Brown et al., arXiv:1203.4802 (digital normalization).
- Submitted, Howe et al, arXiv: 1212.0159 (artifact removal from Illumina metagenomes).
- Submitted, Howe et al., arXiv: 1212.2832 – Assembling large, complex environmental metagenomes.
- In preparation, Zhang et al. – efficient k-mer counting.

Recommended reading

- "Comparative metagenomic and rRNA microbial diversity characterization using archaeal and bacterial synthetic communities." Shakya et al., pmid 23387867.
  - Good benchmark data set!

"*The results … indicate that a single gene marker such as rRNA is a poor determinant of the community structure in metagenomic sequence data from complex communities.*"