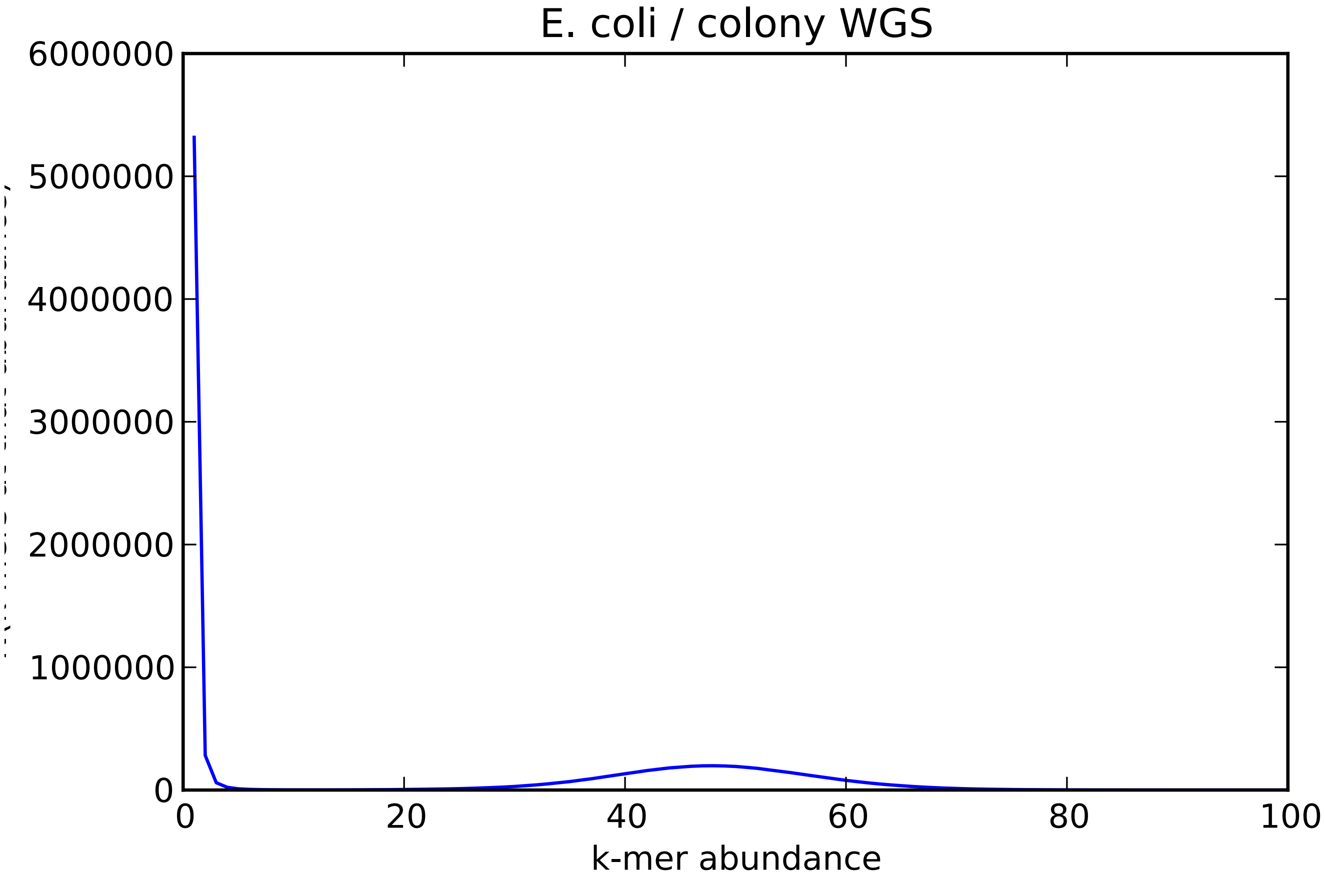
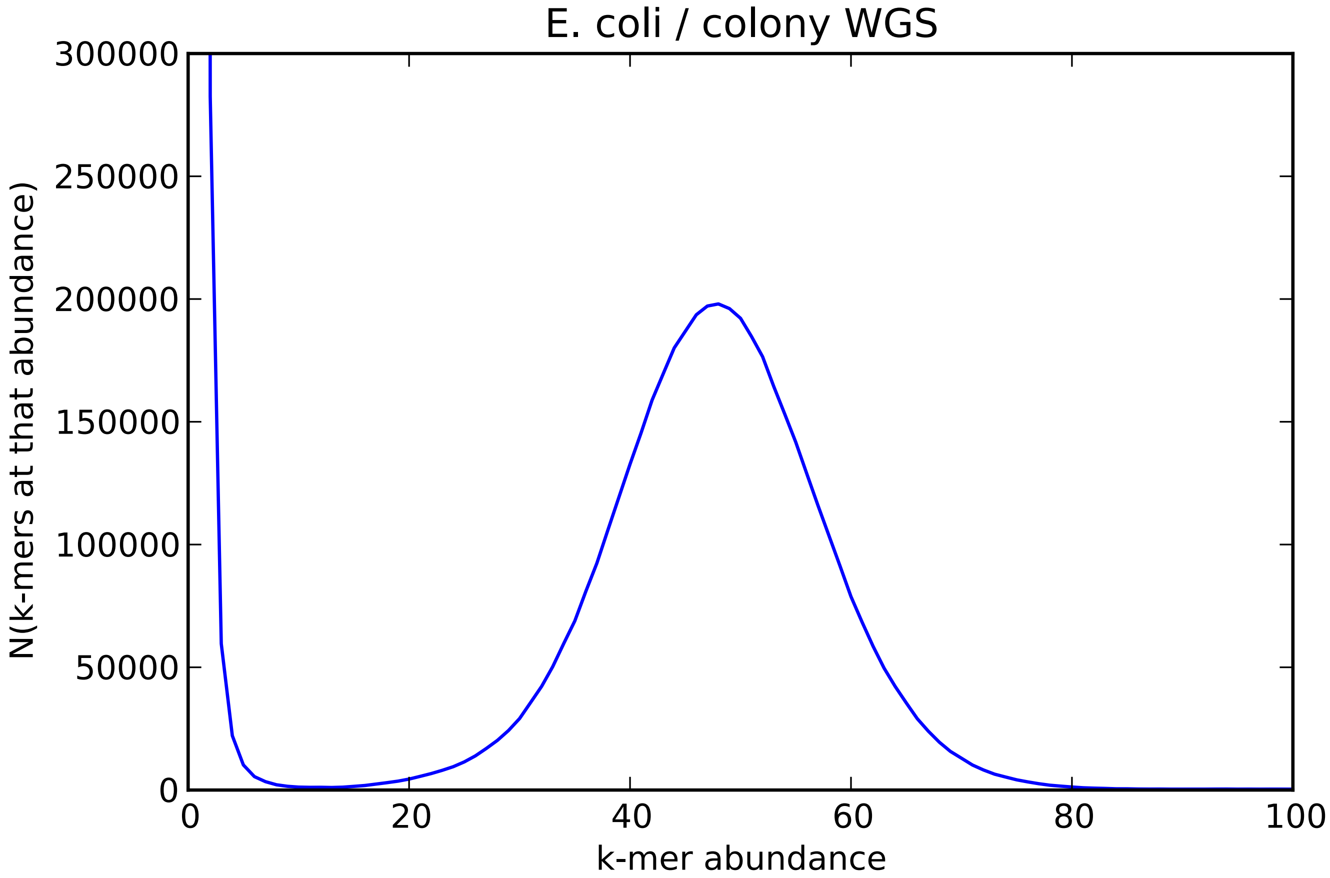
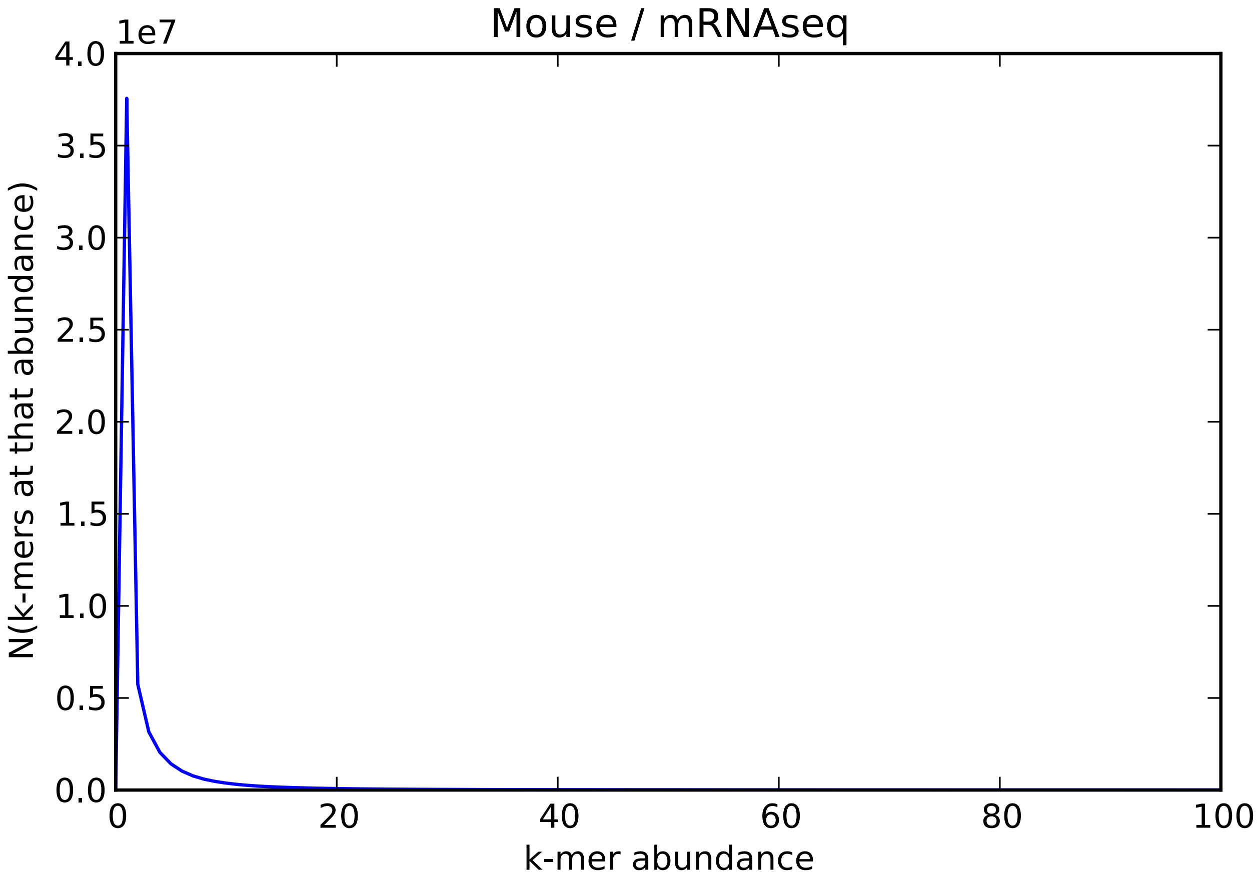
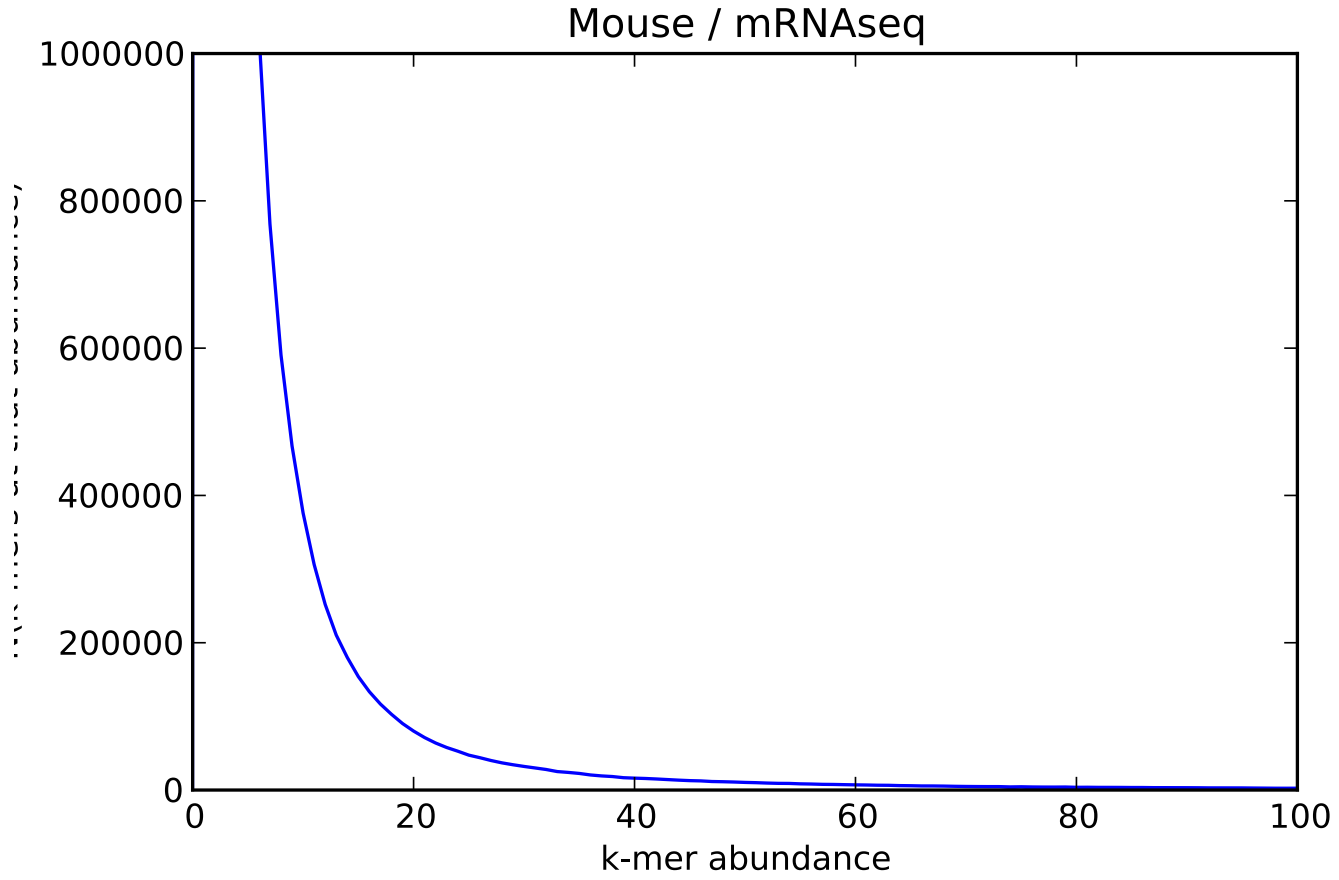
Reference-free analysis of genomes with k-mers


E. coli / colony WGS
6000000
5000000
4000000
3000000
2000000
1000000
0
0
20
40
60
80
100
k-mer abundance


E. coli / colony WGS
N(k-mers at that abundance)
300000
250000
200000
150000
100000
50000
0
0
20
40
60
80
100
k-mer abundance


Mouse / mRNAseq
1e7
4.0
3.5
3.0
2.5
2.0
1.5
1.0
0.5
0.0
N(k-mers at that abundance)
0
20
40
60
80
100
k-mer abundance


Mouse / mRNAseq
1000000
800000
600000
400000
200000
0
0
20
40
60
80
100
k-mer abundance

Preqc - repeats

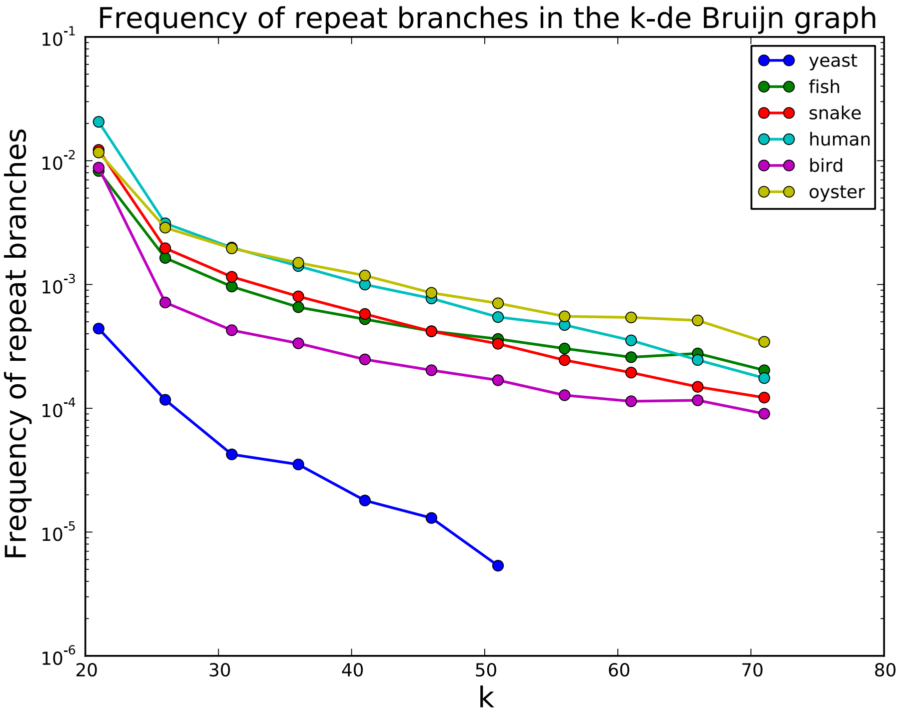
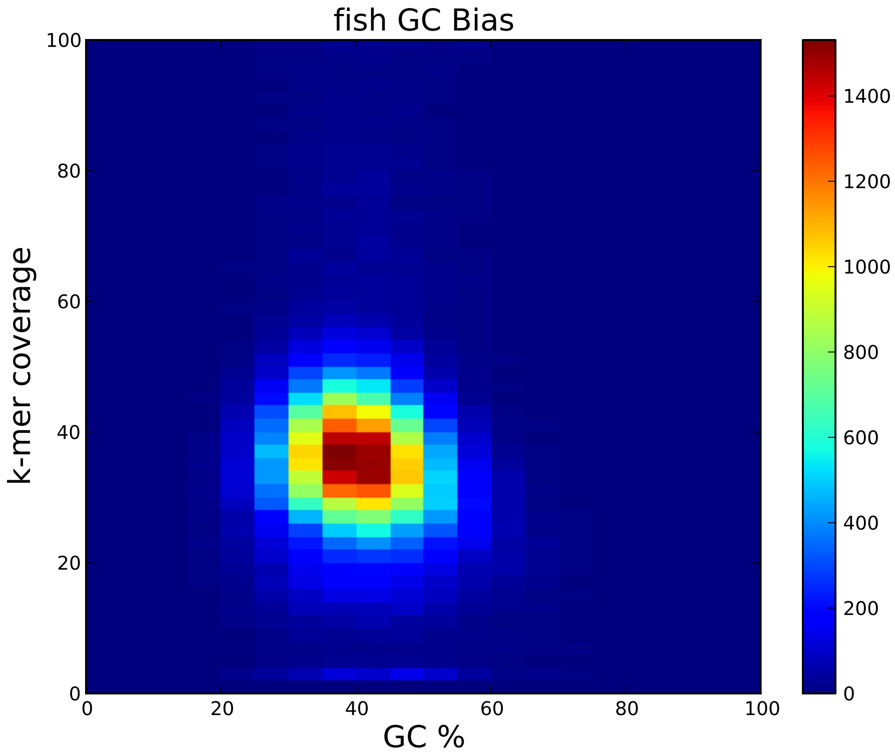


Figure 2: The estimated repeat branch rate for each genome as a function of k. The yeast data stops at k=51 as the number of repeat branches found falls below the minimum threshold for emitting an estimate.

https://github.com/jts/sga/wiki/Preqc

Preqc – GC bias / coverage



(a)

https://github.com/jts/sga/wiki/Preqc

Preqc – predicted contig lengths

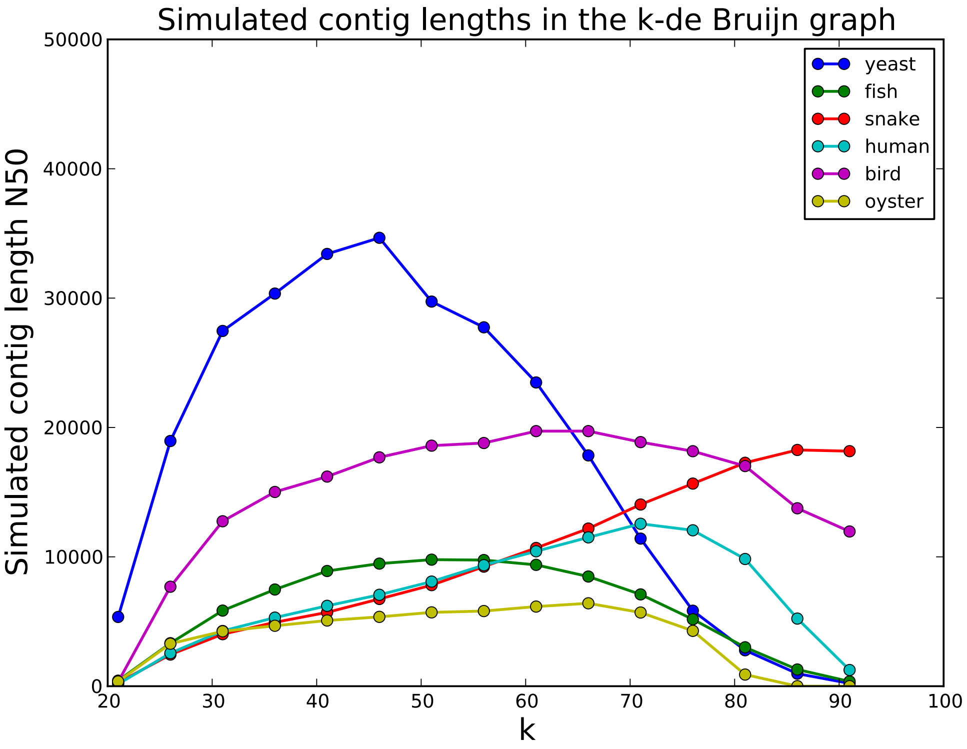
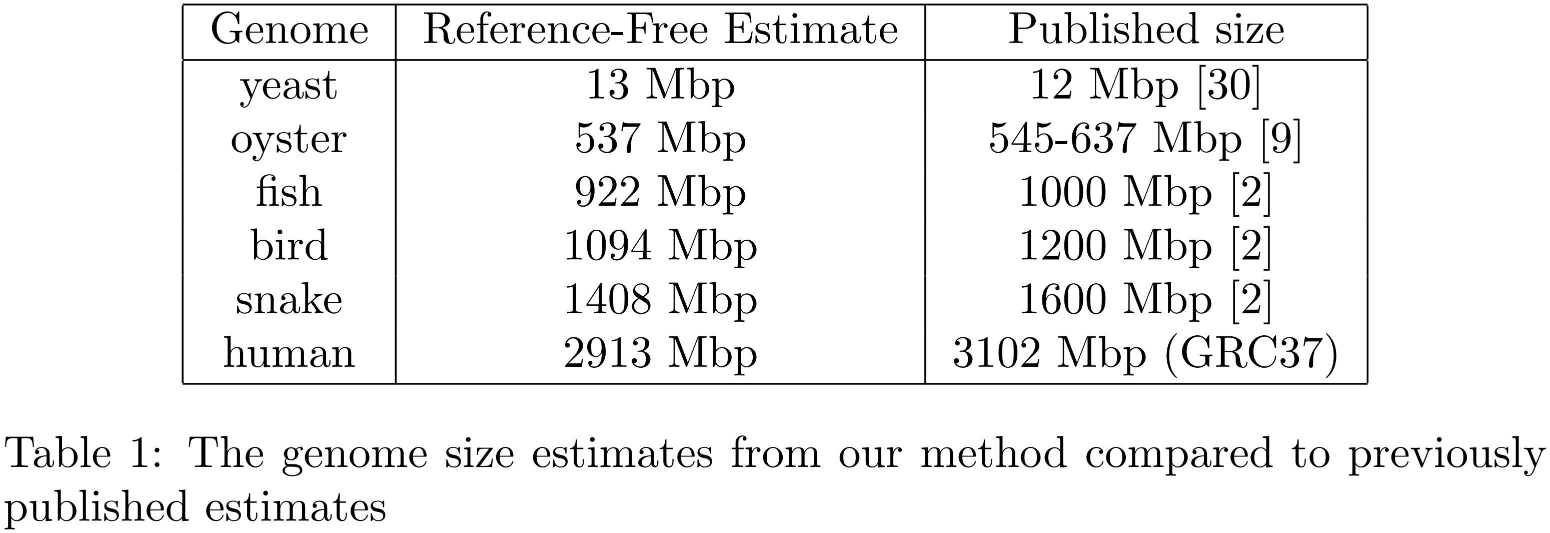


Figure 5: The N50 length of simulated contigs for k from 21 to 91, in increments of 5

https://github.com/jts/sga/wiki/Preqc

Preqc – estimated genome size



Genome	Reference-Free Estimate	Published size
yeast	13 Mbp	12 Mbp [30]
oyster	537 Mbp	545-637 Mbp [9]
fish	922 Mbp	1000 Mbp [2]
bird	1094 Mbp	1200 Mbp [2]
snake	1408 Mbp	1600 Mbp [2]
human	2913 Mbp	3102 Mbp (GRC37)

Table 1: The genome size estimates from our method compared to previously published estimates

https://github.com/jts/sga/wiki/Preqc

Genome Biol. 2014; 15(12): 555.
Published online 2014 Dec 17. doi: 10.1186/s13059-014-0555-3

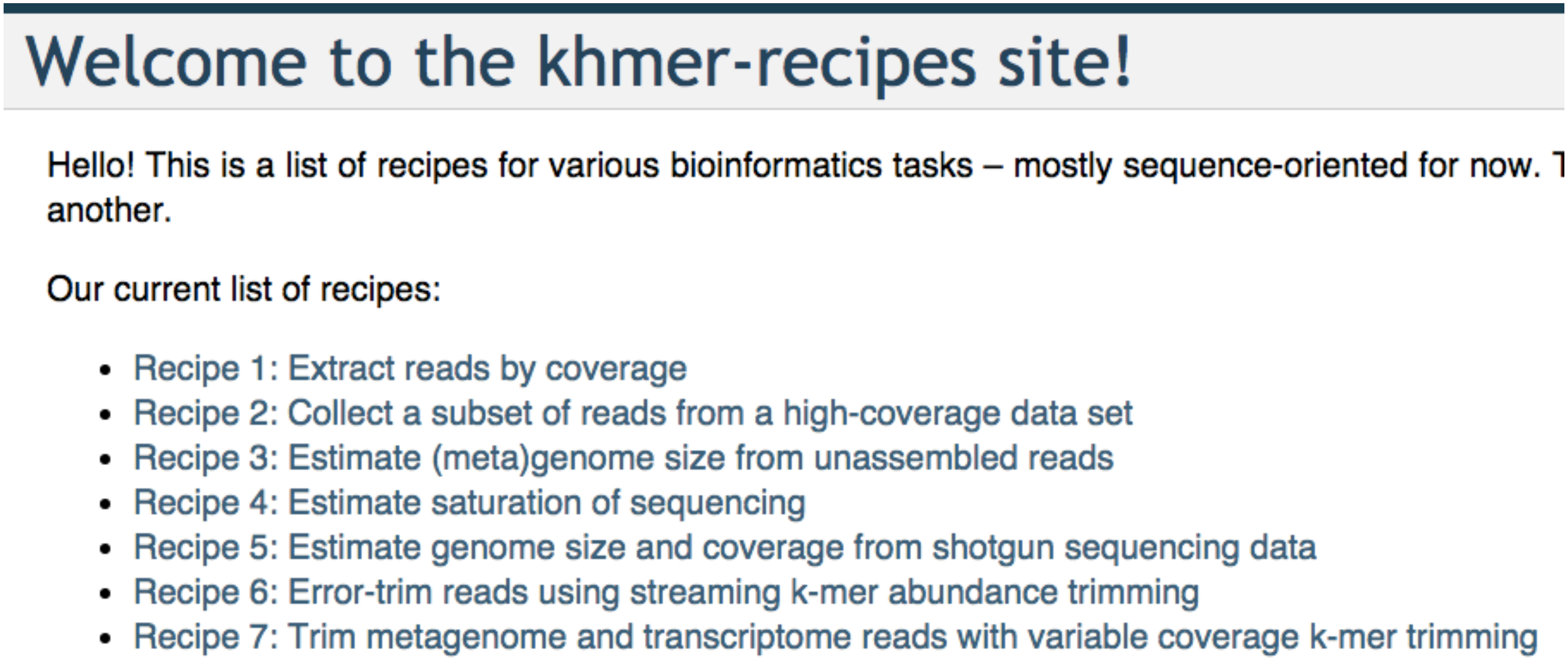
PMCID: PMC4298064

Determining the quality and complexity of next-generation sequencing data without a reference genome

Seyed Yahya Anvar,☒ Lusine Khachatryan, Martijn Vermaat, Michiel van Galen, Irina Pulyakhina, Yavuz Ariyurek, Ken Kraaijeveld, Johan T den Dunnen, Peter de Knijff, Peter AC 't Hoen, and Jeroen FJ Laros☒

Author information ▶ Article notes ▶ Copyright and License information ▶

Khmer-recipes



Welcome to the khmer-recipes site!

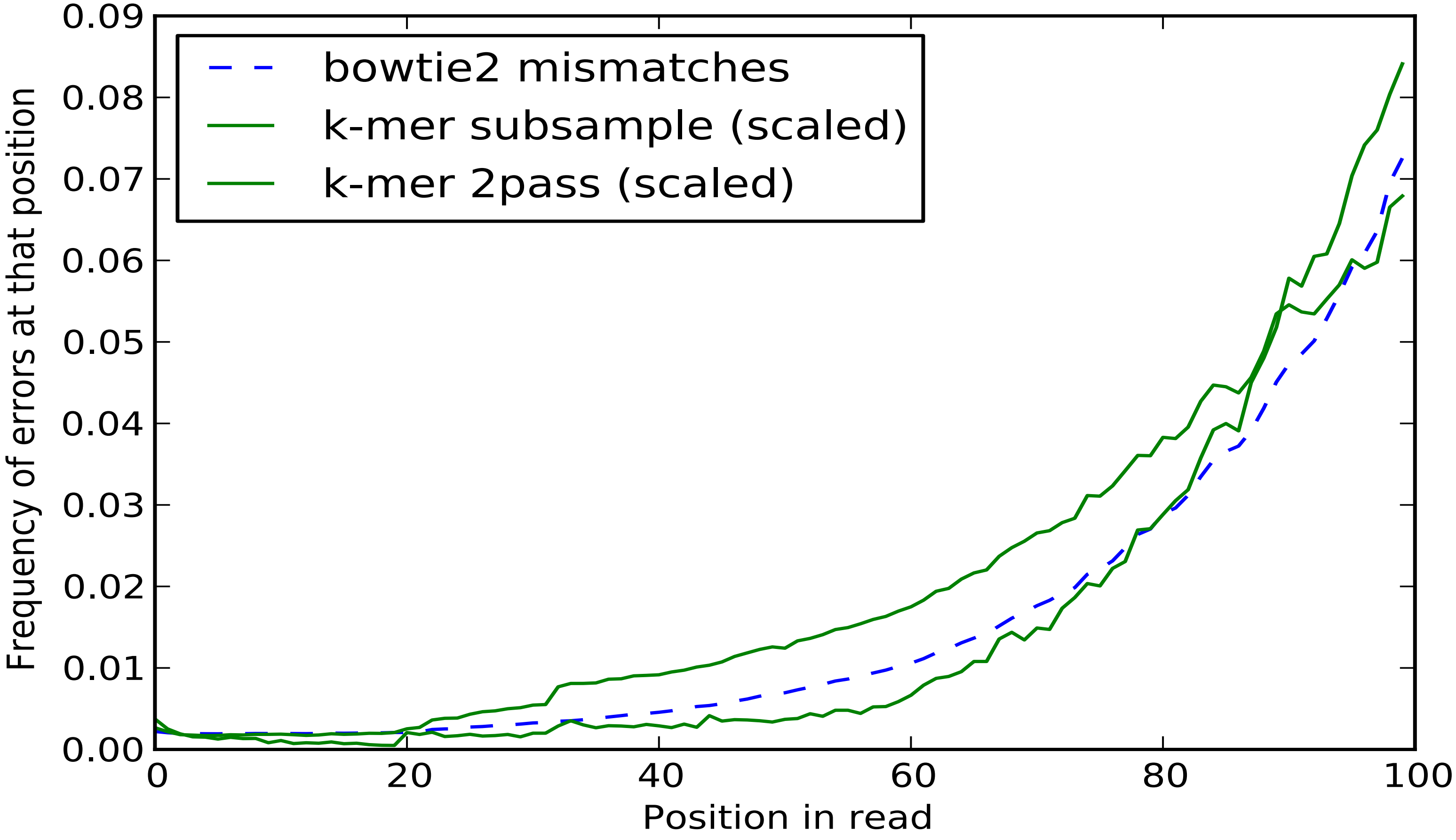
Hello! This is a list of recipes for various bioinformatics tasks – mostly sequence-oriented for now. T
another.

Our current list of recipes:

- Recipe 1: Extract reads by coverage
- Recipe 2: Collect a subset of reads from a high-coverage data set
- Recipe 3: Estimate (meta)genome size from unassembled reads
- Recipe 4: Estimate saturation of sequencing
- Recipe 5: Estimate genome size and coverage from shotgun sequencing data
- Recipe 6: Error-trim reads using streaming k-mer abundance trimming
- Recipe 7: Trim metagenome and transcriptome reads with variable coverage k-mer trimming

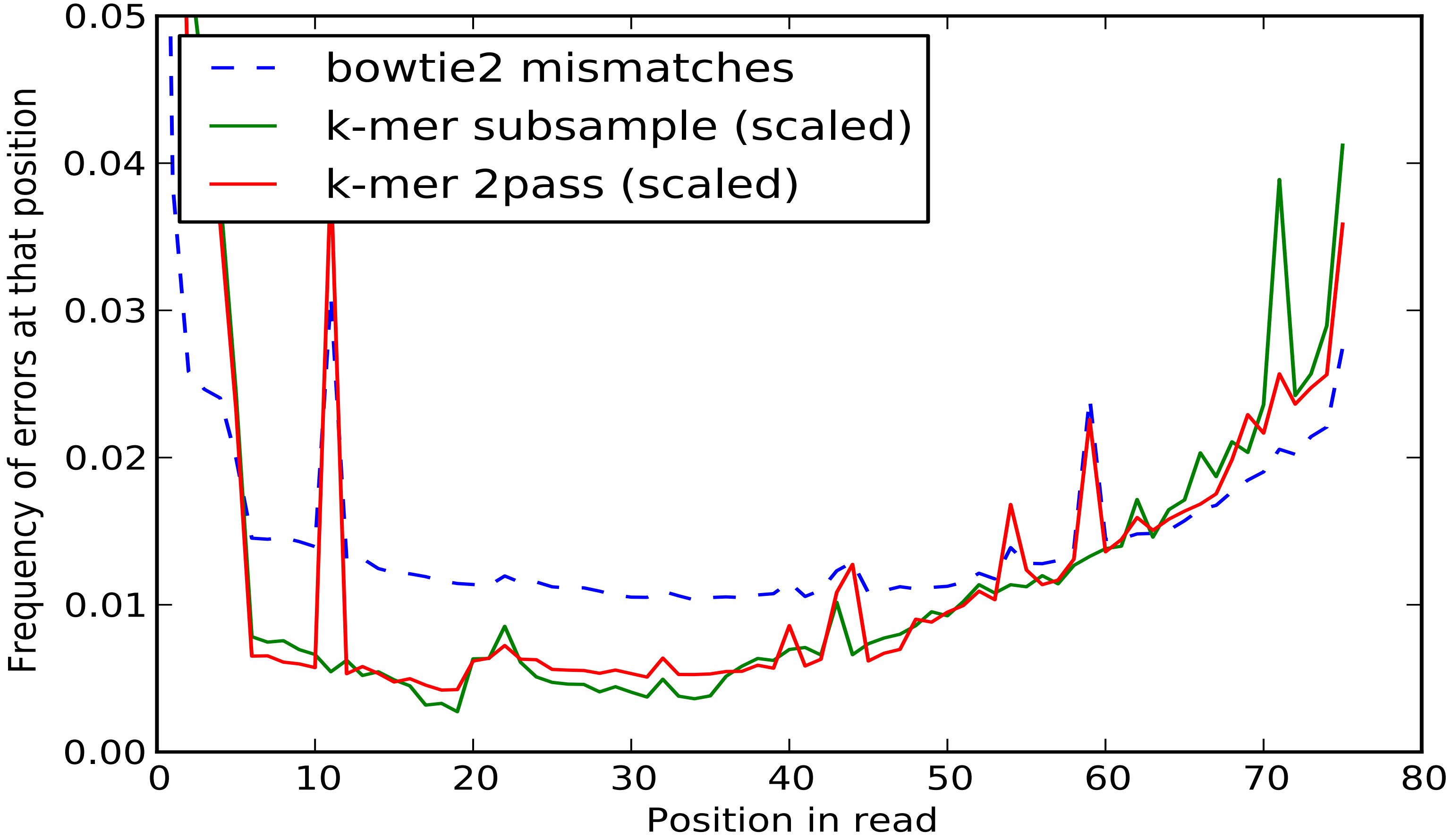
http://khmer-recipes.readthedocs.org/en/latest/

Reference & quality-score independent approaches (k-mers)

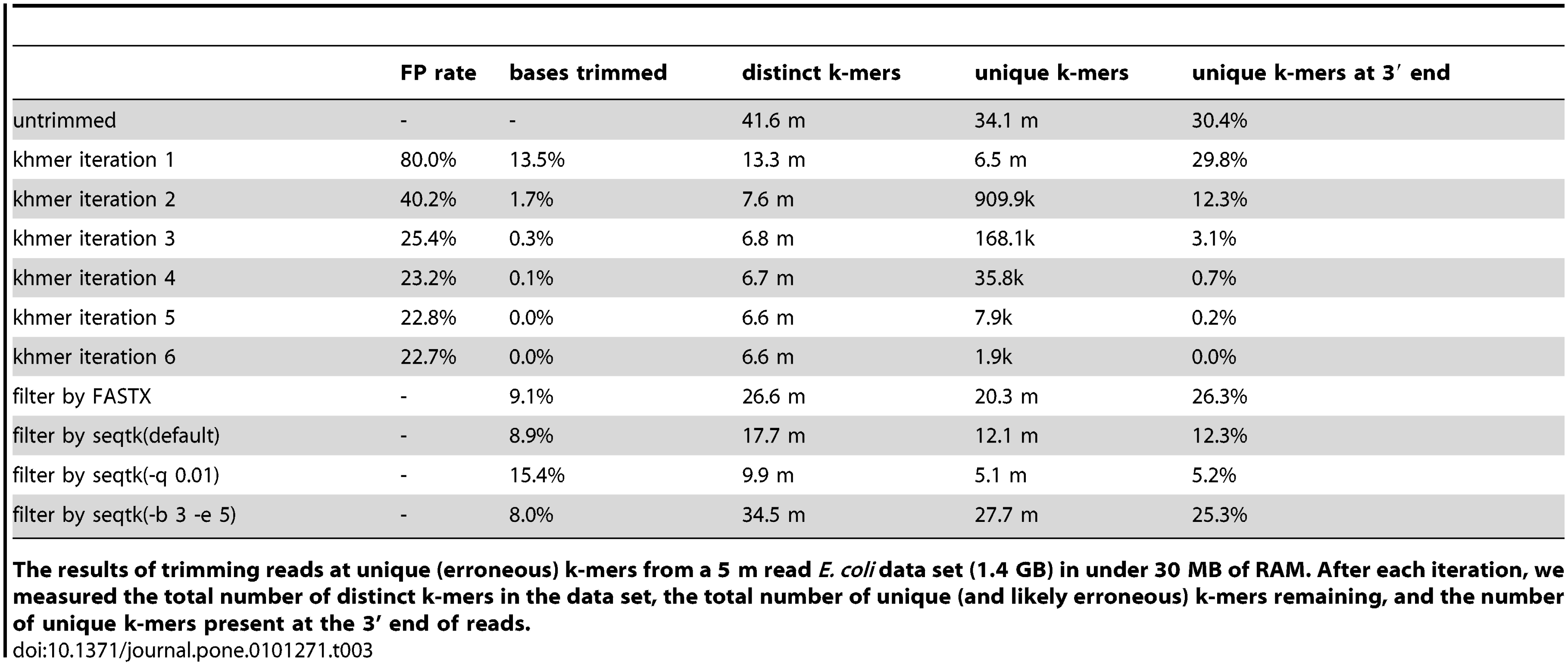


Zhang et al., https://peerj.com/preprints/890/

Mouse mRNAseq



Zhang et al., https://peerj.com/preprints/890/



	FP rate	bases trimmed	distinct k-mers	unique k-mers	unique k-mers at 3′ end
untrimmed	-	-	41.6 m	34.1 m	30.4%
khmer iteration 1	80.0%	13.5%	13.3 m	6.5 m	29.8%
khmer iteration 2	40.2%	1.7%	7.6 m	909.9k	12.3%
khmer iteration 3	25.4%	0.3%	6.8 m	168.1k	3.1%
khmer iteration 4	23.2%	0.1%	6.7 m	35.8k	0.7%
khmer iteration 5	22.8%	0.0%	6.6 m	7.9k	0.2%
khmer iteration 6	22.7%	0.0%	6.6 m	1.9k	0.0%
filter by FASTX	-	9.1%	26.6 m	20.3 m	26.3%
filter by seqtk(default)	-	8.9%	17.7 m	12.1 m	12.3%
filter by seqtk(-q 0.01)	-	15.4%	9.9 m	5.1 m	5.2%
filter by seqtk(-b 3 -e 5)	-	8.0%	34.5 m	27.7 m	25.3%

The results of trimming reads at unique (erroneous) k-mers from a 5 m read *E. coli* data set (1.4 GB) in under 30 MB of RAM. After each iteration, we measured the total number of distinct k-mers in the data set, the total number of unique (and likely erroneous) k-mers remaining, and the number of unique k-mers present at the 3′ end of reads.

doi:10.1371/journal.pone.0101271.t003

K-mer abundance trimming removes errors effectively!

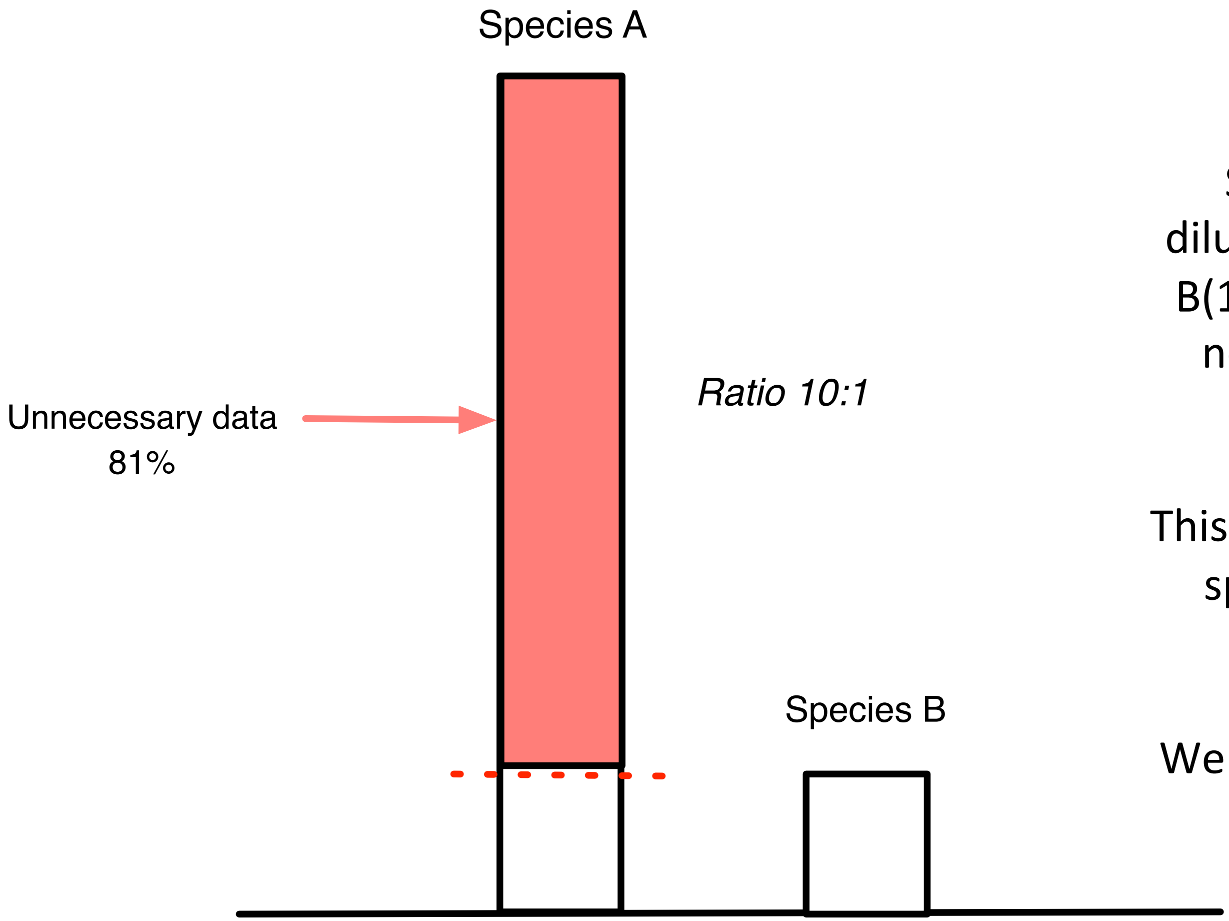
Zhang et al. PLoS One, 2014

CTB research - diginorm

http://arxiv.org/abs/1203.4802

Approach: Digital normalization

(a computational version of library normalization)

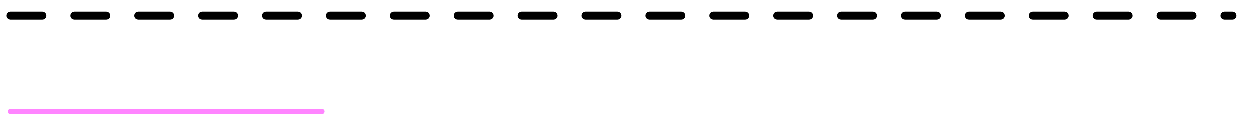


Suppose you have a dilution factor of A (10) to B(1). To get 10x of B you need to get 100x of A! Overkill!!

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

We can discard it for you…

Digital normalization



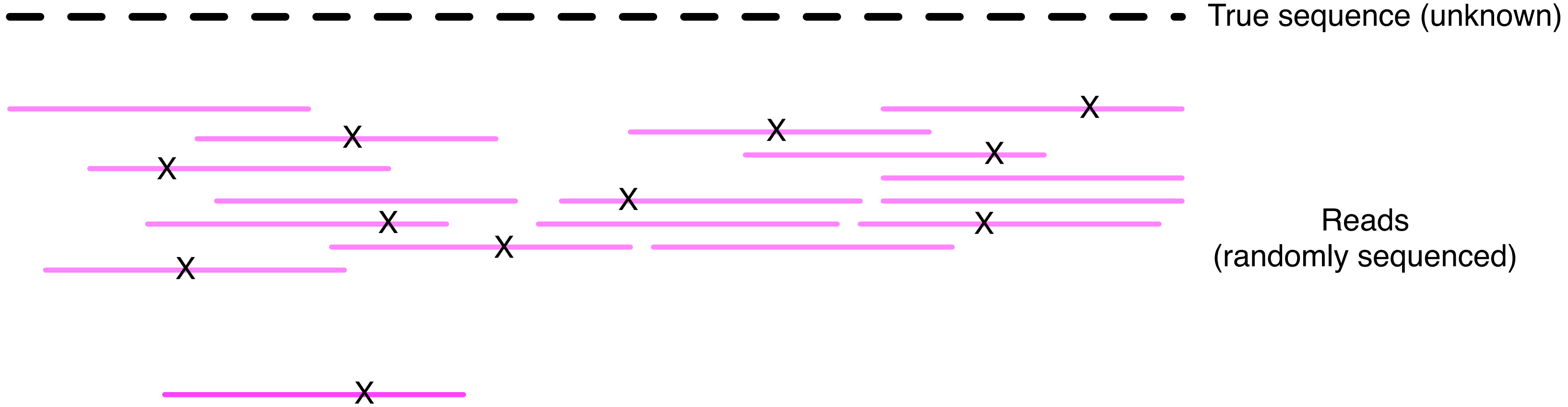
True sequence (unknown)

Reads
(randomly sequenced)

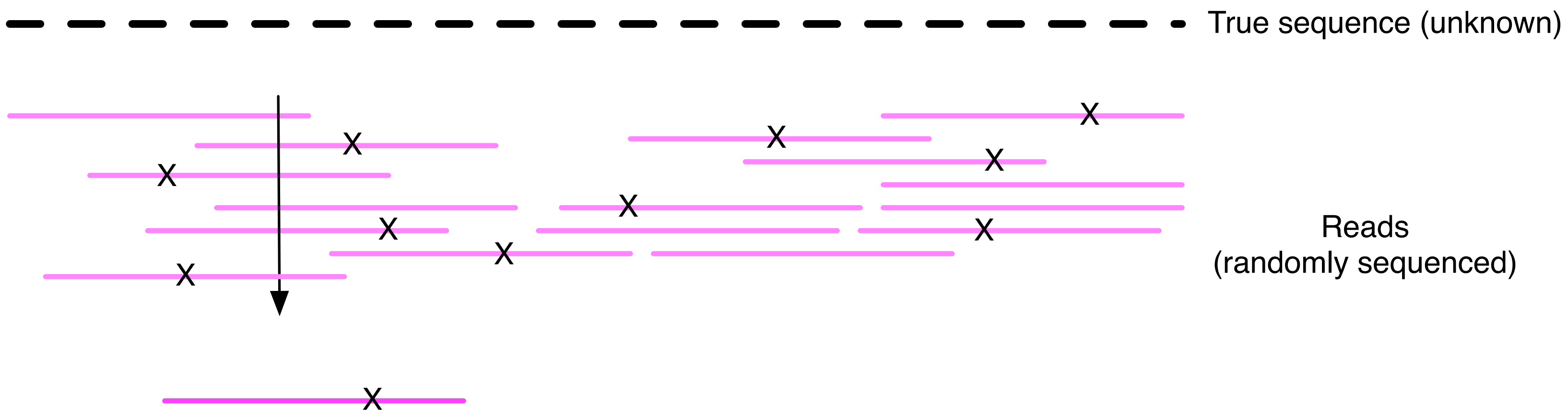
Digital normalization

True sequence (unknown)
X
Reads
(randomly sequenced)

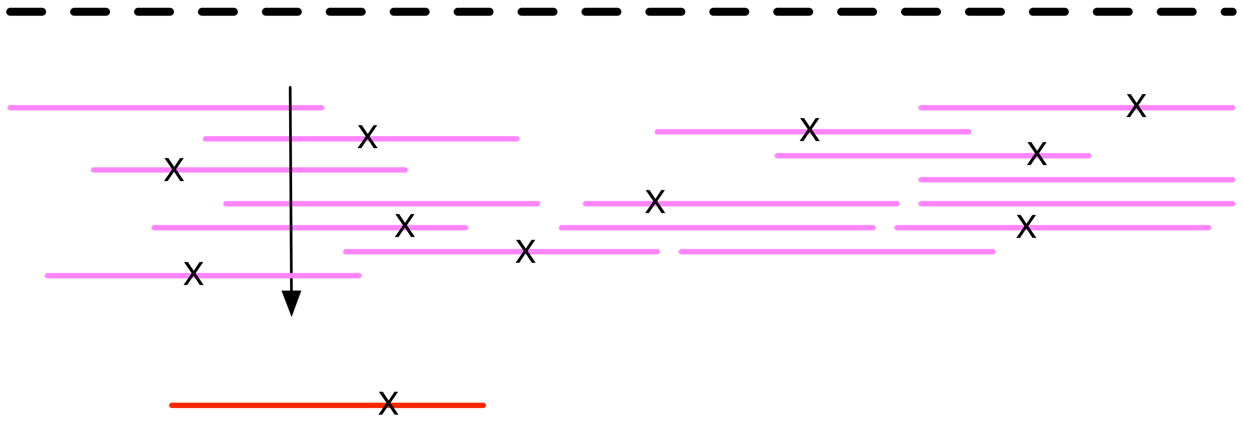
Digital normalization



Digital normalization



Digital normalization

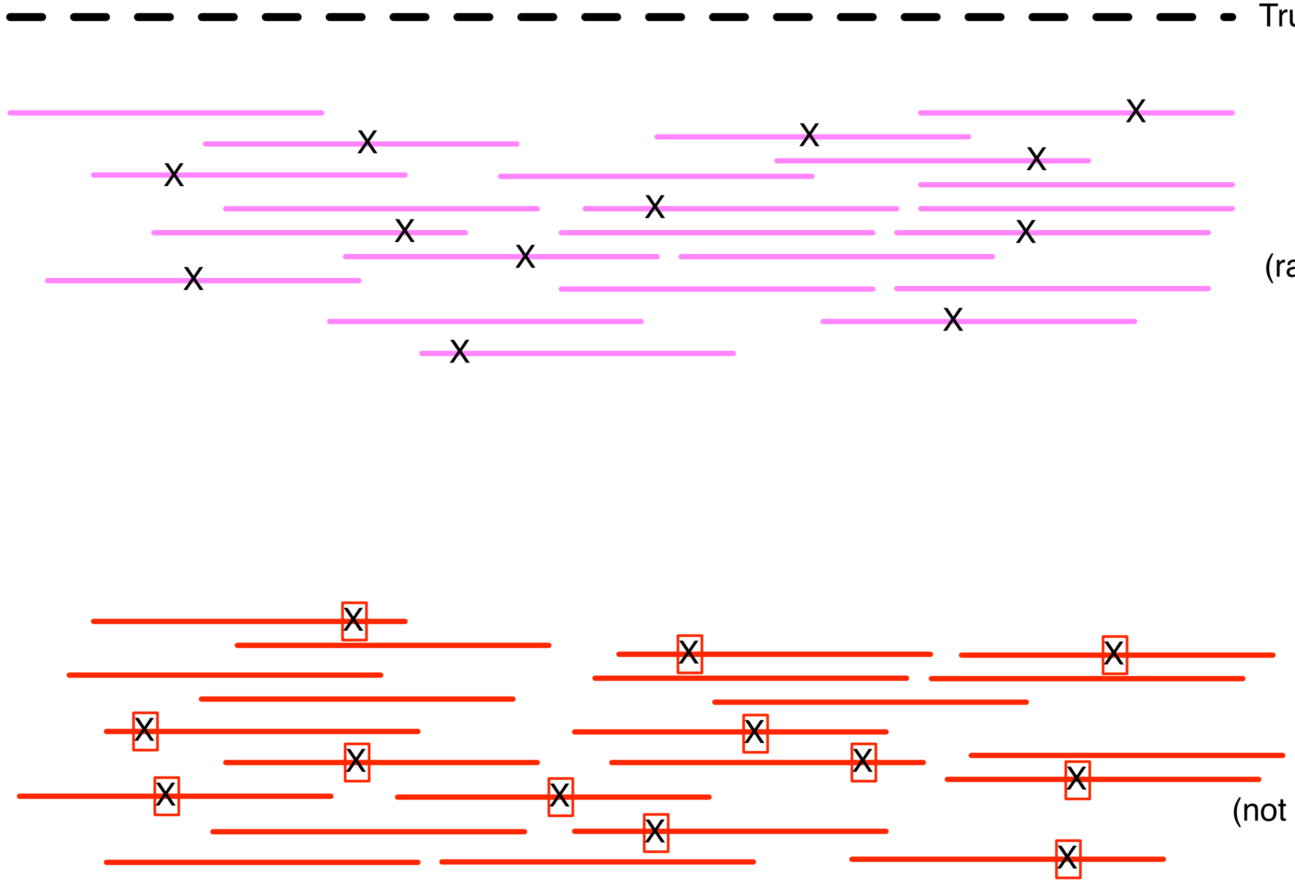


True sequence (unknown)

Reads
(randomly sequenced)

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

Digital normalization



True sequence (unknown)

Reads
(randomly sequenced)

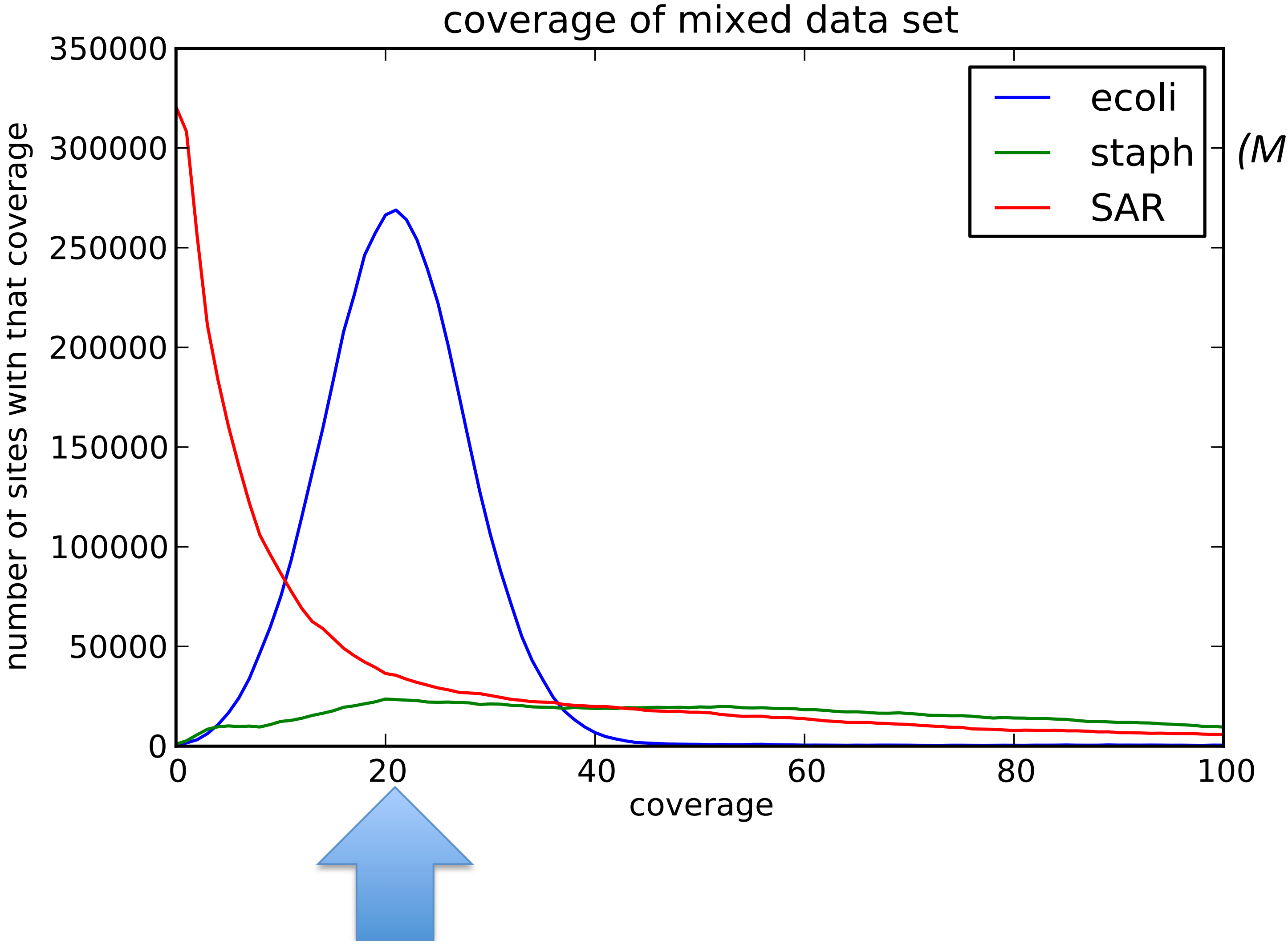
Redundant reads
(not needed for assembly)

Digital normalization approach

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

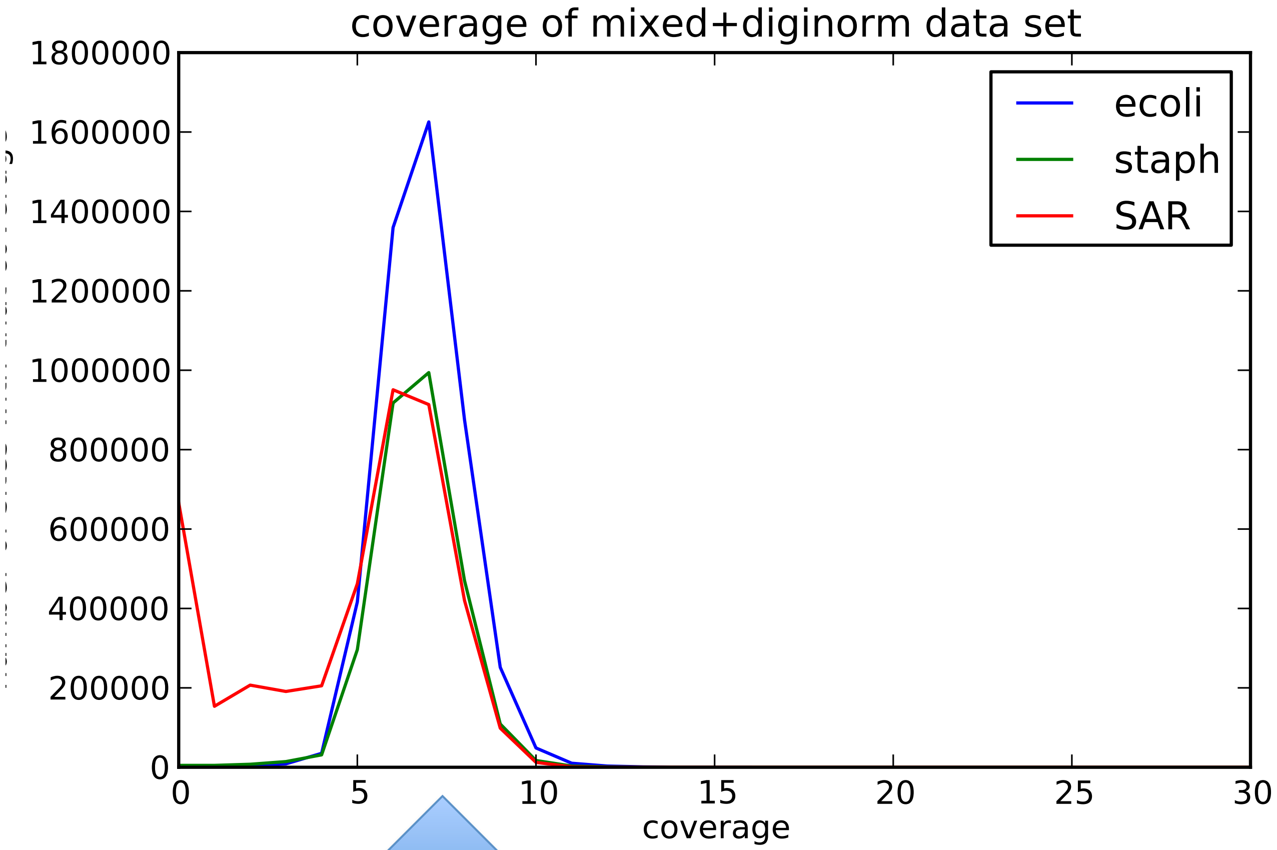
- Is single pass: looks at each read only once;
- Does not “collect” the majority of errors;
- Keeps all low-coverage reads;
- Smooths out coverage of regions.

Coverage before digital normalization:



(MD amplified)

Coverage after digital normalization:



Normalizes coverage

Discards redundancy

Eliminates majority of errors

Scales assembly dramatica

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

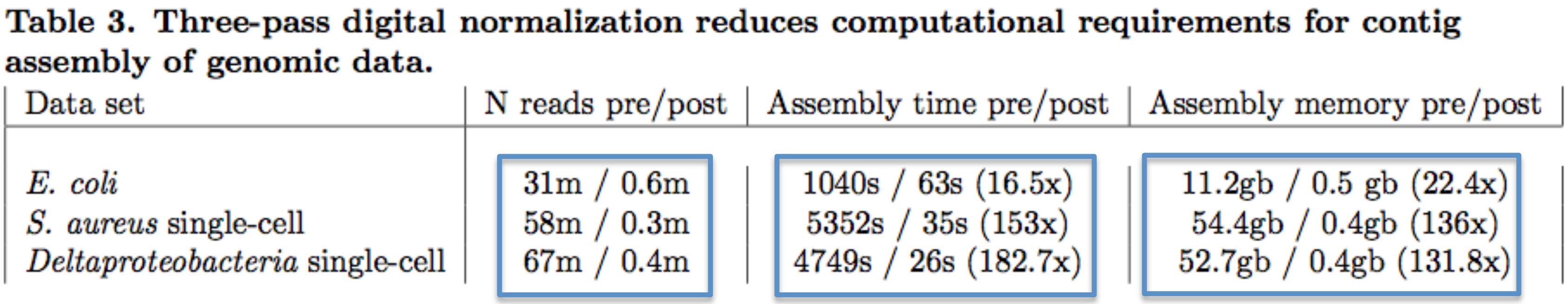


Table 3. Three-pass digital normalization reduces computational requirements for contig assembly of genomic data.

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

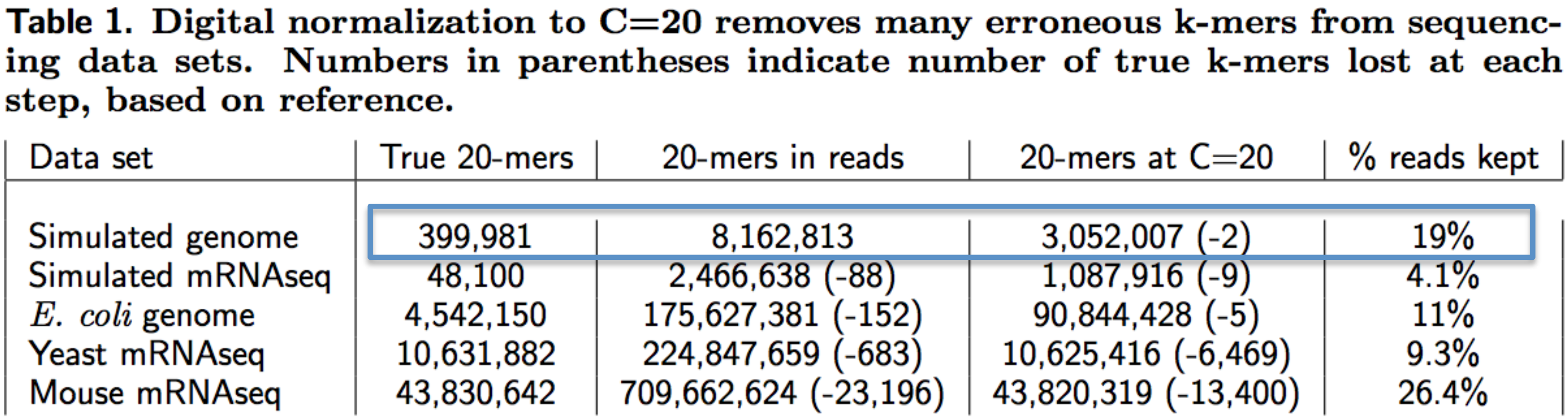


Table 1. Digital normalization to C=20 removes many erroneous k-mers from sequencing data sets. Numbers in parentheses indicate number of true k-mers lost at each step, based on reference.

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%

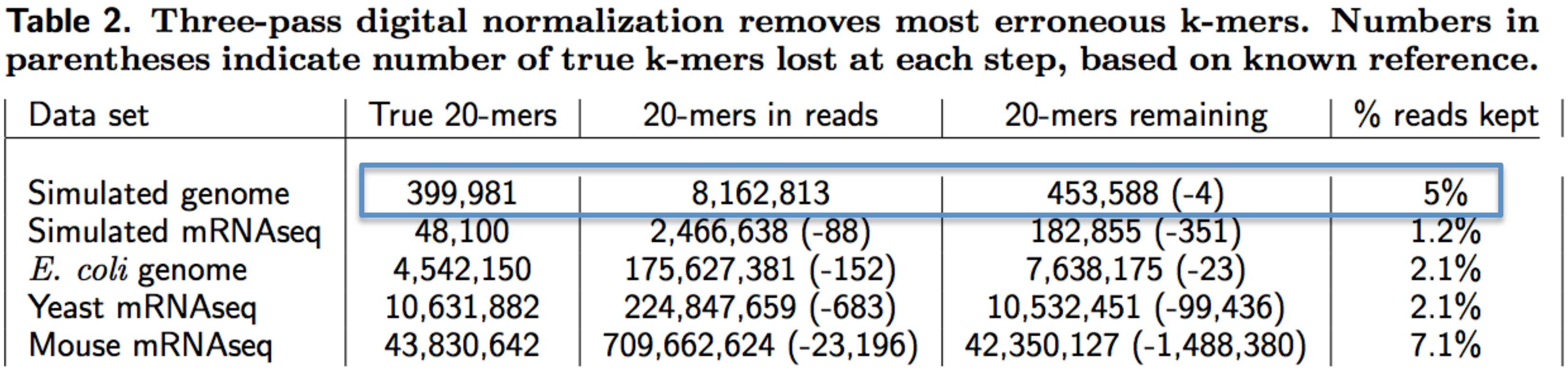


Table 2. Three-pass digital normalization removes most erroneous k-mers. Numbers in parentheses indicate number of true k-mers lost at each step, based on known reference.

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%