Counting kmers for fun and profit

Will Trimble
Argonne National Laboratory

9 February 2016

Counting is good for you.

- Counting kmers – how
- Counting – what you can see
- Visualizations and case studies
- Error correction options

"Why is kmer error correction better than quality-score error correction?"

Definition

kmers are *overlapping,*
fixed-length
oligonucleotide sequences

```
AGCTTTTCATTCTGACTGCAACGGGCAATATGTCTCTGTGTGGATTAAAAAAAGAGT

AGCTTTTCATTCTGACTGCAA
 GCTTTTCATTCTGACTGCAAC
  CTTTTCATTCTGACTGCAACG
   TTTTCATTCTGACTGCAACGG
    TTTCATTCTGACTGCAACGGG
     TTCATTCTGACTGCAACGGGC
      TCATTCTGACTGCAACGGGCA
       CATTCTGACTGCAACGGGCAA
        ATTCTGACTGCAACGGGCAAT
         TTCTGACTGCAACGGGCAATA
          TCTGACTGCAACGGGCAATAT
                ..........
```

Definition

```
ATCGCGAAAAGTCCC 2
AAAAAAAAAAAAAAA 459
AAAAAAAAAAAAAAC 71
AAAATAAAAAAAATA 1
AAAAAAAAAAAAAAG 36
ACATGAAAAACAACT 1
AAAAAAAAAAAAAAT 23
AAAAAAAAAAAAACA 95
GTAGGAAAAGCCCAC 1
AAAAAAAAAAAAACC 7
AAAAAAAAAAAAACG 8
AAAAAAAAAAAAACT 9
AAAAAAAAAAAAAGA 36
AACAAGAAAAACAAA 1
AAAAAAAAAAAAAGC 10
AAATAAAAAAAATAG 1
AACAGAAAAAACACG 1
AAAAAAAAAAAAAGG 2
AAAAAAAAAAAAAGT 6
```

kmers are *overlapping, fixed-length oligonucleotide sequences*

Just a tool—but a super useful one.

Can count for any sequence dataset

Makes sense for **genomes** / **samples** / haplotypes / genes if you want

Programming / plotting puzzle
http://angus.readthedocs.org/en/2016/automation.html

Short kmers

C	1179554
G	1176923
A	1142228
T	1140970
Total	4639675

For MFG, 4639675 sites land in

k=1

$4^1 = 4$

AA	337870
AC	256662
AG	237877
AT	309819
CA	325149
CC	271673
CG	346670
CT	236061
GA	267247
GC	383931
GG	270137
GT	255608
TA	211961
TC	267288
TG	322239
TT	339482

k=2

$4^2 = 16$

AAAAAAA	711
AAAAAAC	722
AAAAAAG	815
AAAAAAT	941
AAAAACA	791
AAAAACC	629
AAAAACG	732
AAAAACT	607
AAAAAGA	665
AAAAAGC	819
AAAAAGG	522
AAAAAGT	481
AAAAATA	805
AAAAATC	770
AAAAATG	799
AAAAATT	665
AAAACAA	522
AAAACAC	344
AAAACAG	657

k=7

$4^7 = 16384$

AAAAAAAAAA	0
AAAAAAAAAC	1
AAAAAAAAAG	2
AAAAAAAAAT	4
AAAAAAAACA	12
AAAAAAAACC	11
AAAAAAAACG	4
AAAAAAAACT	5
AAAAAAAAGA	10
AAAAAAAAGC	23
AAAAAAAAGG	4
AAAAAAAAGT	4
AAAAAAAATA	13
AAAAAAAATC	12
AAAAAAAATG	11
AAAAAAAATT	7
AAAAAAACAA	13
AAAAAAACAC	7
AAAAAAACAG	18

k=10

$4^{10} = 1M$

Long kmers

AAAAAAAAAAAAAAAAAAAA	0
AAAAAAAAAAAAAAAAAAAC	0
AAAAAAAAAAAAAAAAAAAG	0
...	
AAAAAAAAACCTCTTTTTTT	0
AAAAAAAAACCTGAAAAAAA	1
AAAAAAAAACCTGAAAAAAC	0
...	
AAAAAAAAAGAAAGGTAACG	0
AAAAAAAAAGAAAGGTAACT	1
AAAAAAAAAGAAAGGTAAGA	0
...	
AAAAAAAAAGCCAGCACCCC	0
AAAAAAAAAGCCAGCACCCG	1
AAAAAAAAAGCCAGCACCCT	0

k=20

$4^{20} = 10^{12}$

When there are many more possible observations (4^k) than actual observations, the table is sparse.

For a 100Gb dataset, this is 10^{11} pigeons roosting in 10^{12} windowsills.

Most windowsills are empty.

```
00000000000000000000000000000000000000
00000000000000000001000000000000000000
00000010000000000000000000000000000000
00000000000000000000000000000000000000
```

10^{11}
pigeons

10^{12}
holes

Short kmers	Long kmers
Tables of all 4^k symbols can be computed by novice programmers.	Storing $4^k > 10^{12}$ doesn't work very well, so digital representations are either sparse or probabilistic.
Digests are compact	Digests are bulky
Individual short kmers are not interesting; only aggregate patterns matter.	Because of sparseness, **individual kmers can be interesting**. (kmers map.)
$1 \leq k < 16$	$k > 20$

Genome kmers vs. data kmers

When sequencing a genome, some of your reads will perfectly represent the genome, and some will contain errors.

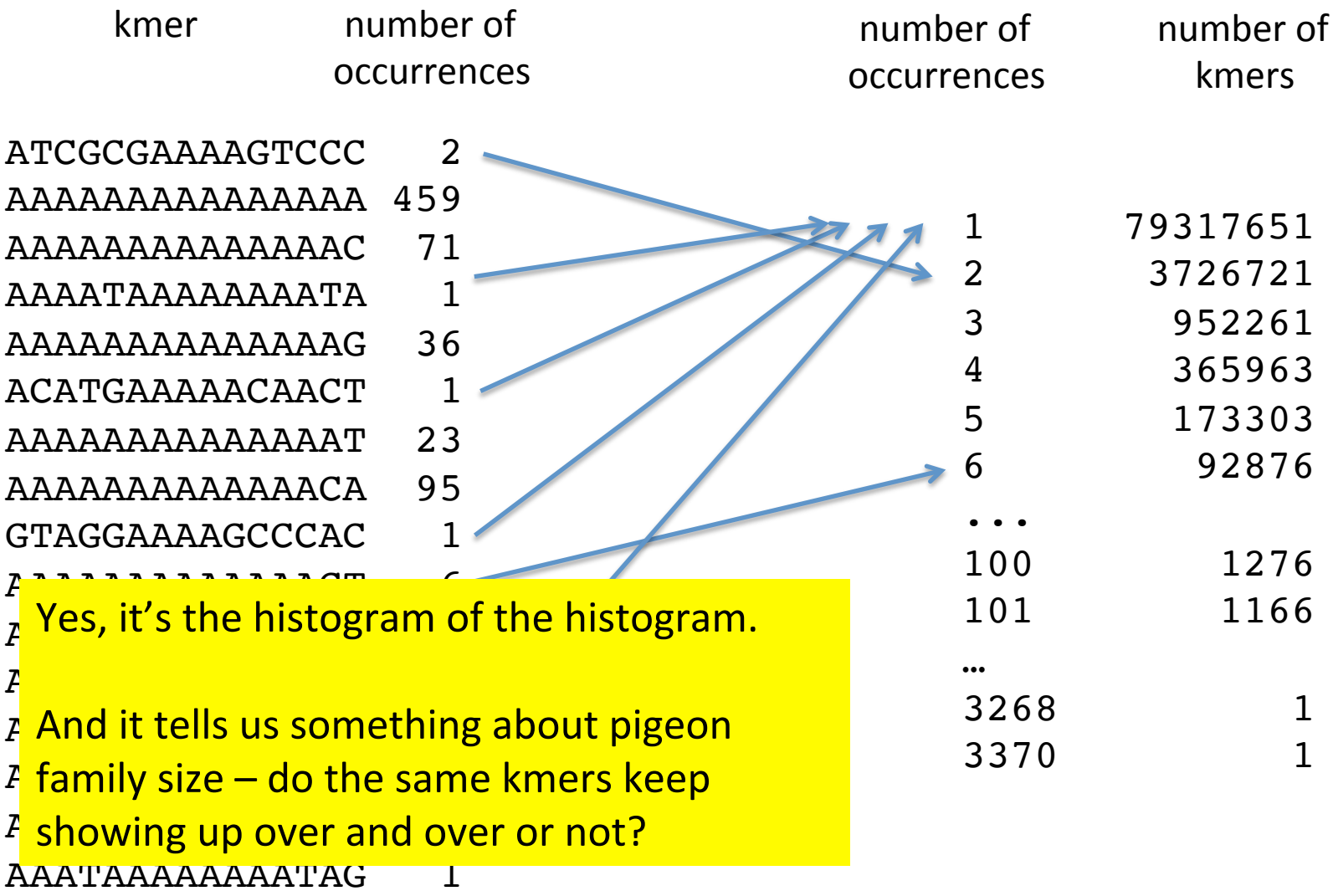
But if you sequence a genome to 100x, you will see the **same genome sequence over and over again**, but most errors you will only see once.

kmer table -> kmer spectrum.

kmer	number of occurrences
ATCGCGAAAAGTCCC	2
AAAAAAAAAAAAAAA	459
AAAAAAAAAAAAAAC	71
AAAATAAAAAAAATA	1
AAAAAAAAAAAAAAG	36
ACATGAAAAACAACT	1
AAAAAAAAAAAAAAT	23
AAAAAAAAAAAAACA	95
GTAGGAAAAGCCCAC	1
AAAAAAAAAAAAACC	7
AAAAAAAAAAAAACG	8
AAAAAAAAAAAAACT	9
AAAAAAAAAAAAAGA	36
AACAAGAAAAACAAA	1
AAAAAAAAAAAAAGC	10
AAATAAAAAAAATAG	1

number of occurrences	number of kmers
1	79317651
2	3726721
3	952261
4	365963
5	173303
6	92876
...	
100	1276
101	1166
…	
3268	1
3370	1

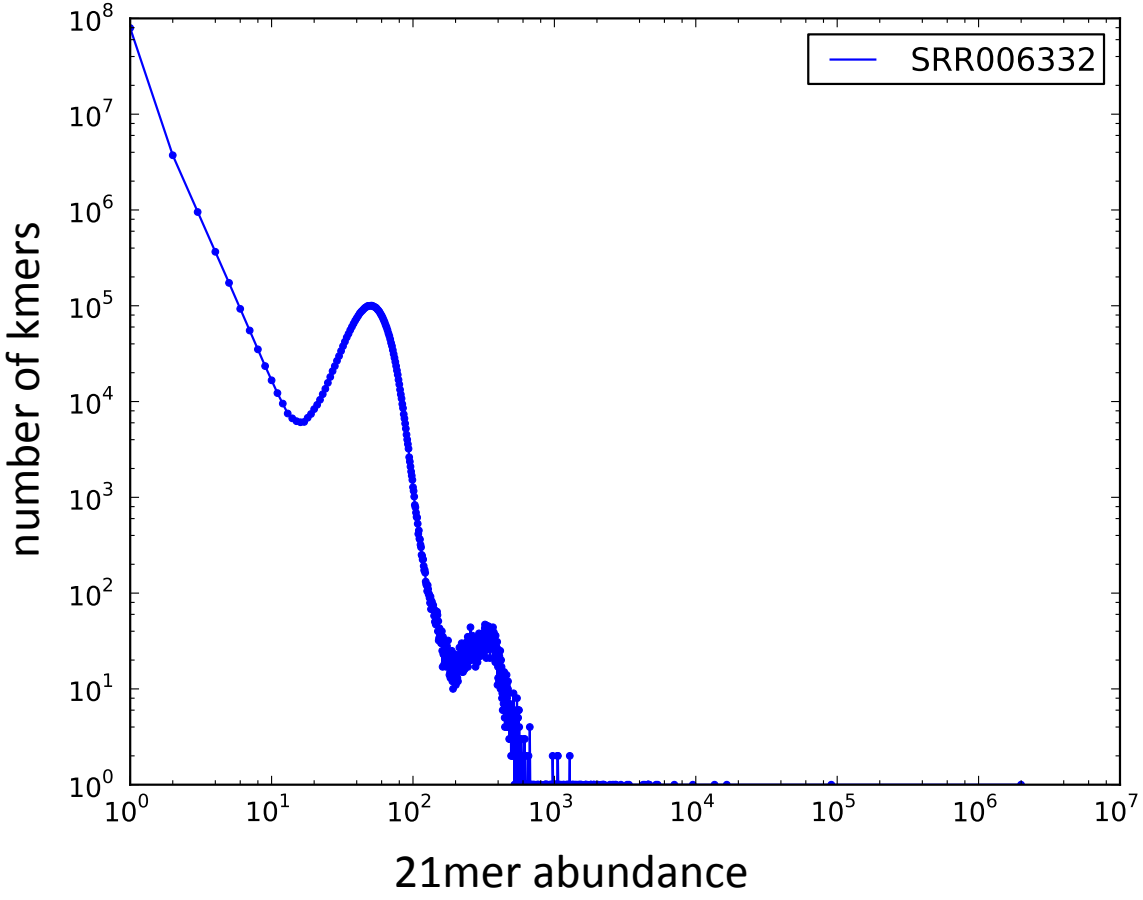
kmer table -> kmer spectrum.



5 million holes
>50 pigoens

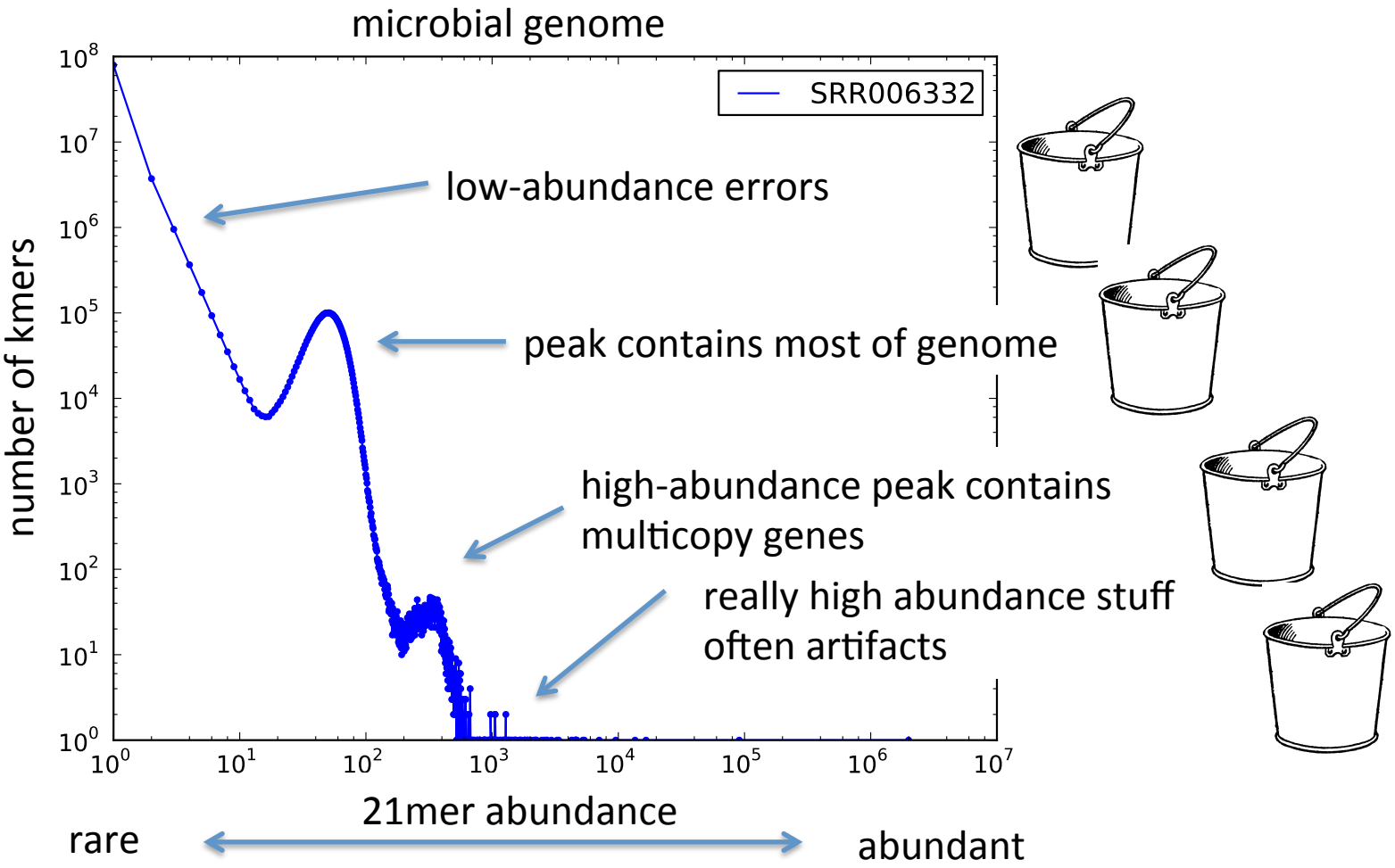
100 million holes
1 pigeon each

The kmer spectrum.

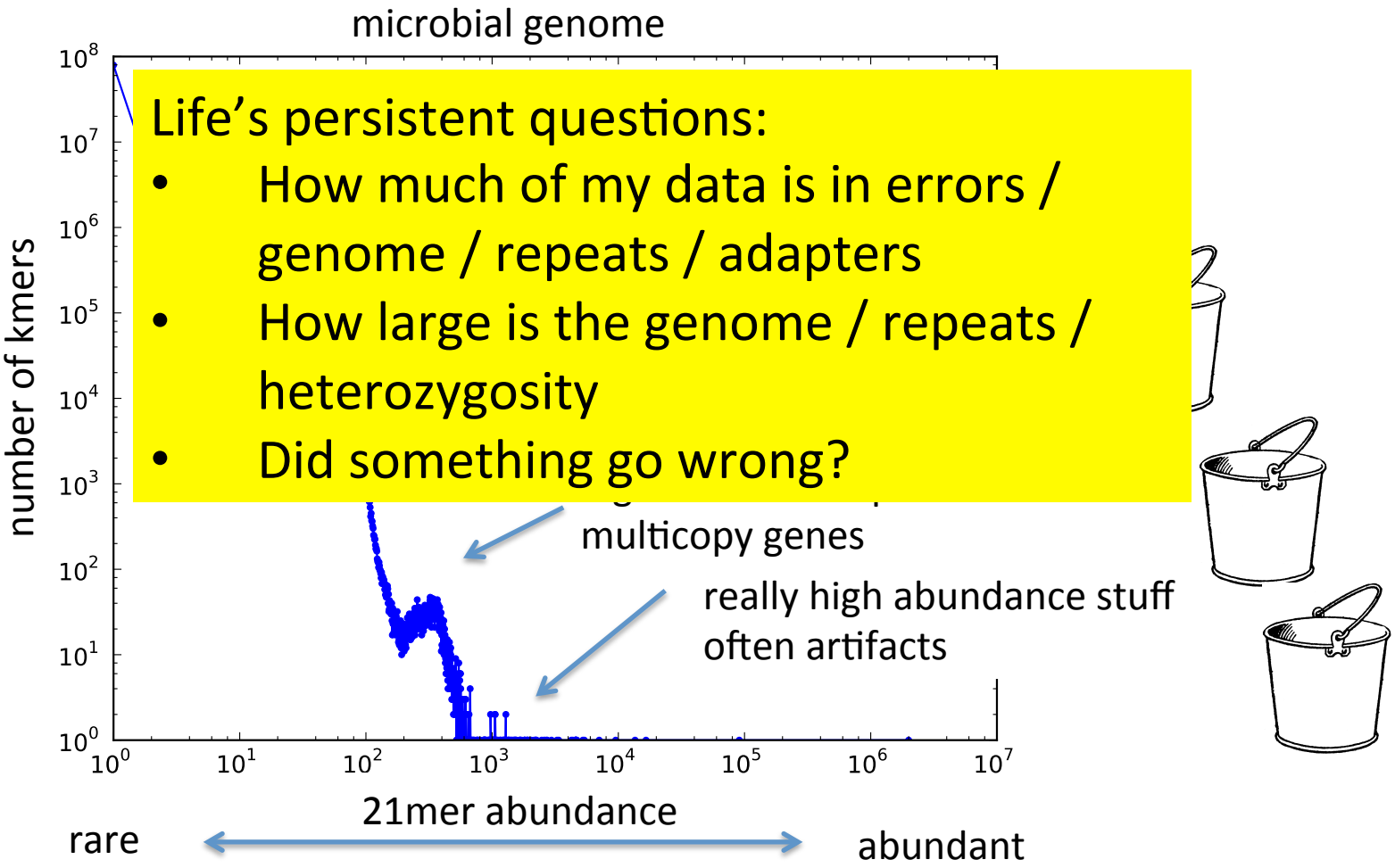


microbial genome

The kmer spectrum.



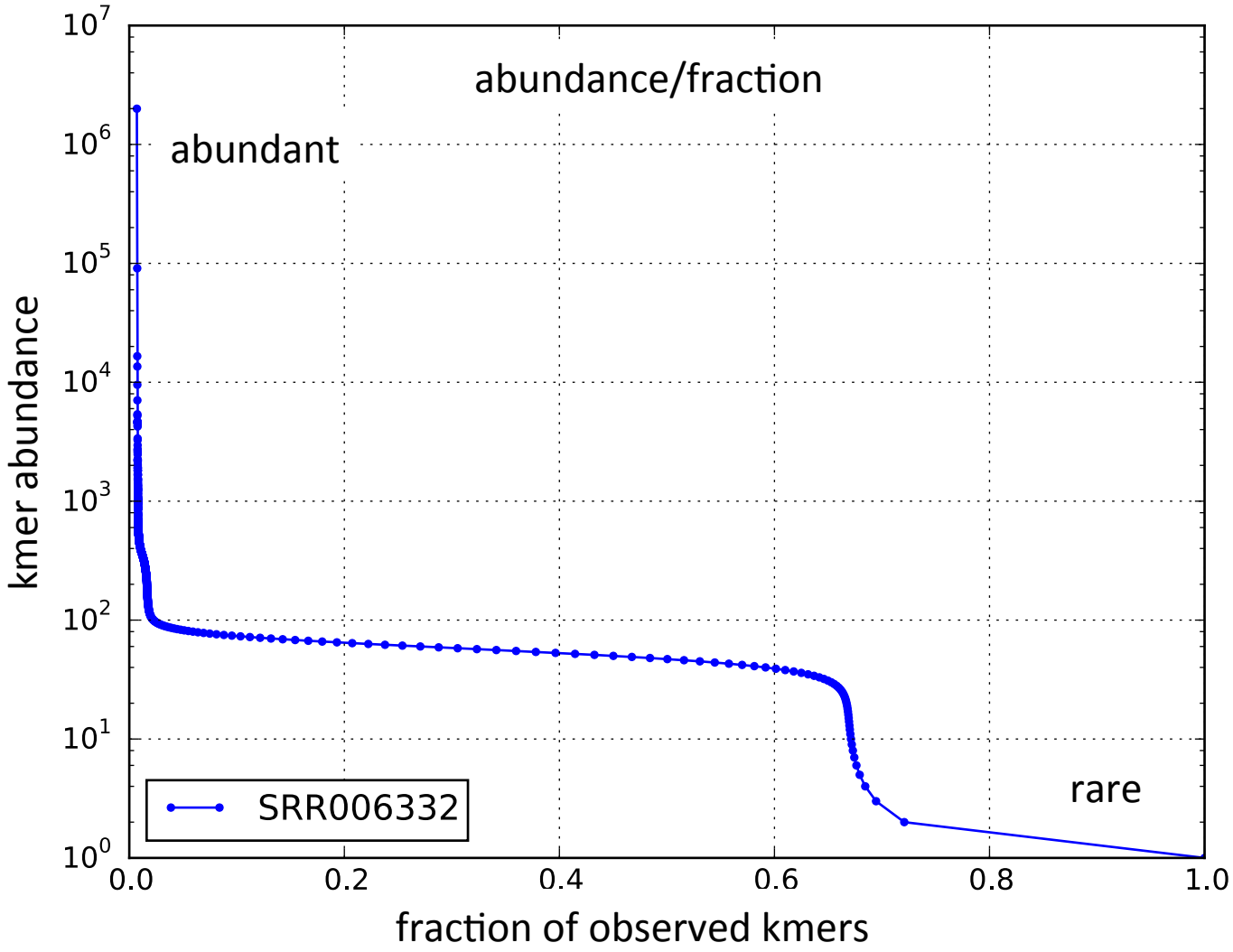
The kmer spectrum.



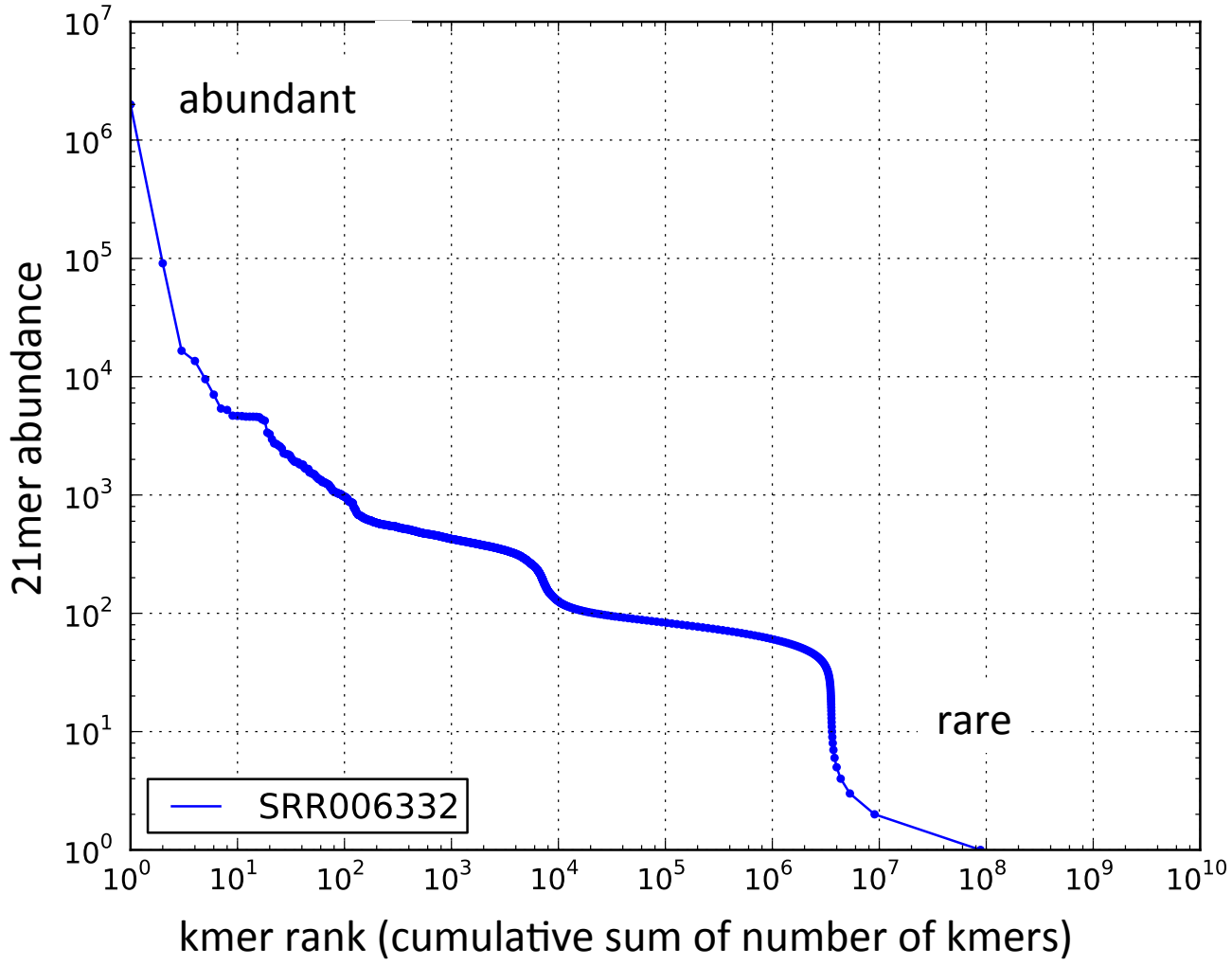
Life's persistent questions:

- How much of my data is in errors / genome / repeats / adapters
- How large is the genome / repeats / heterozygosity
- Did something go wrong?

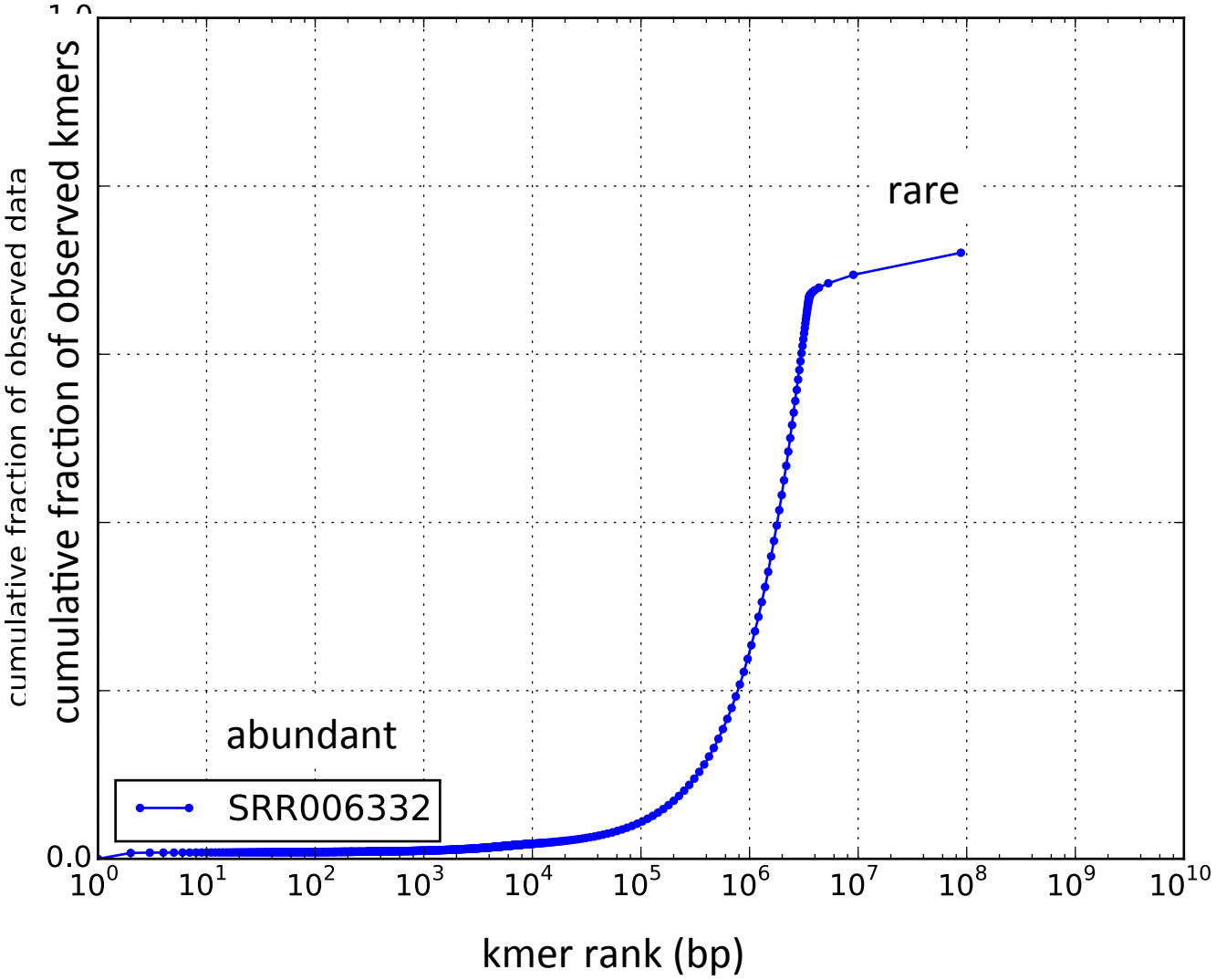
Abundance /fraction



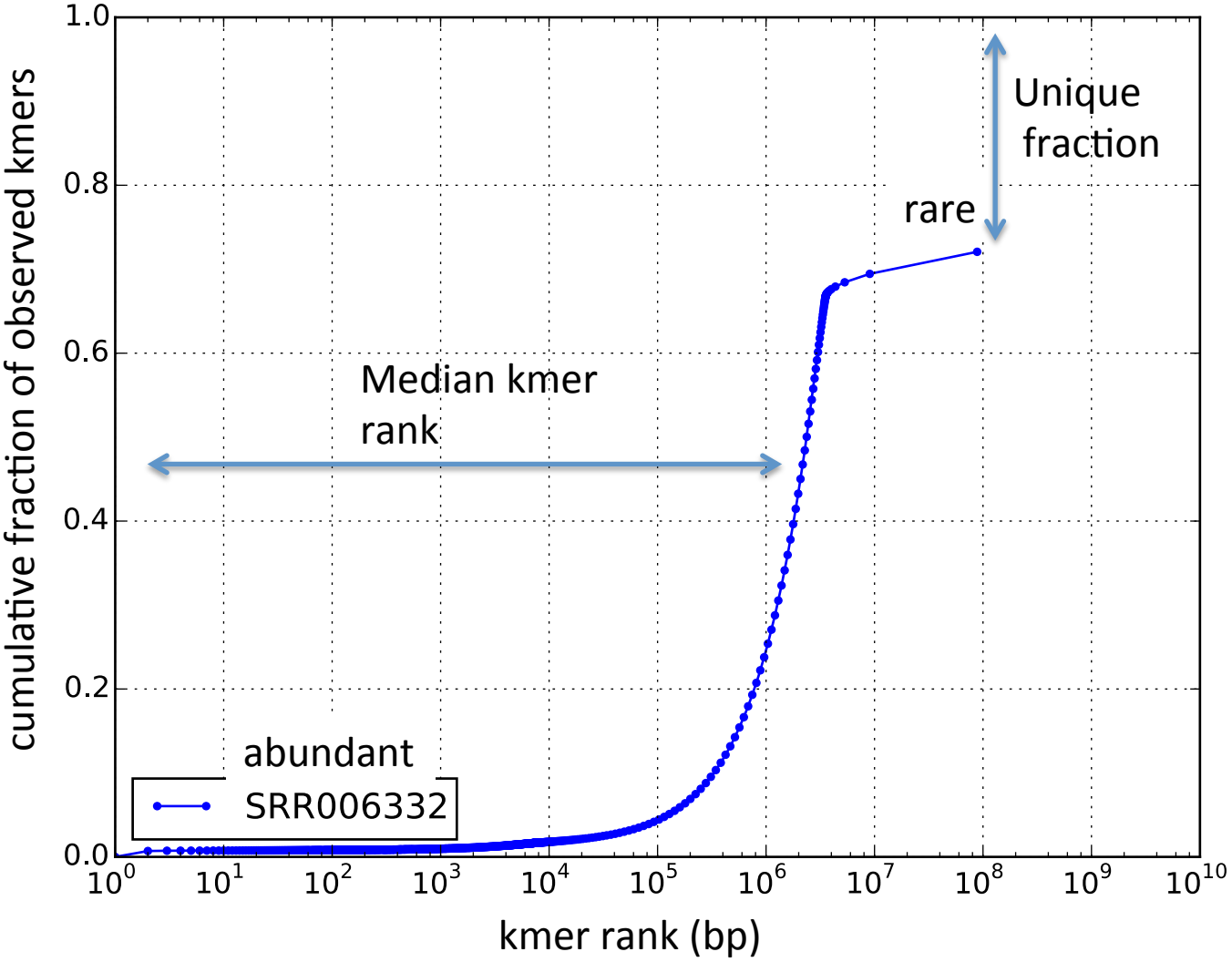
Cumulative ranked kmer spectrum



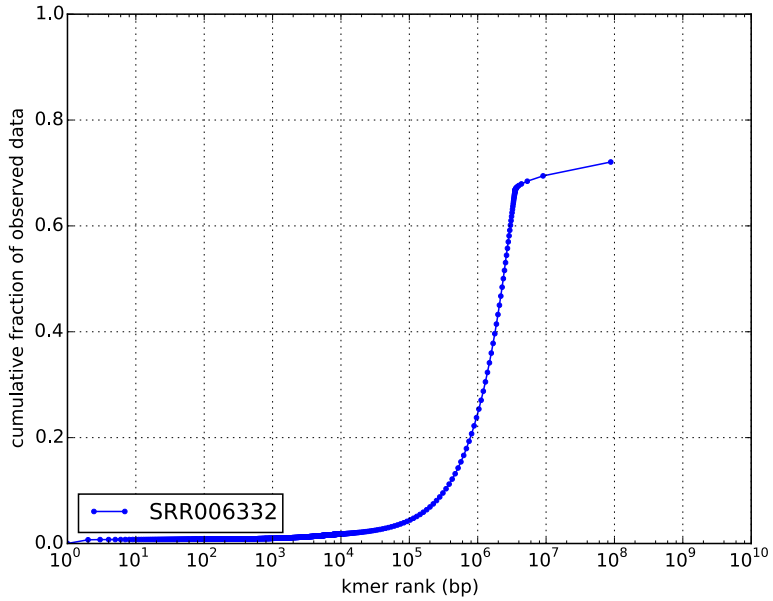
Ranked kmers consumed



Ranked kmers consumed



This graph does not change shape with sampling !!!!!!!

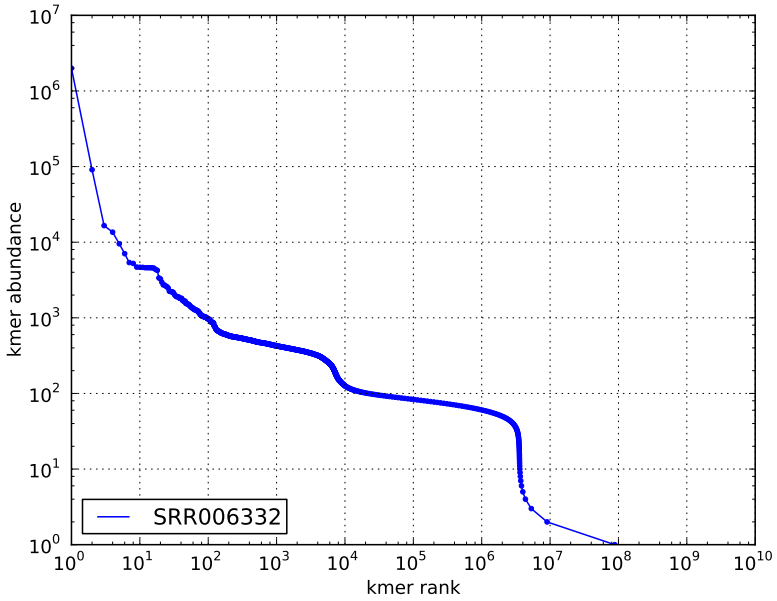


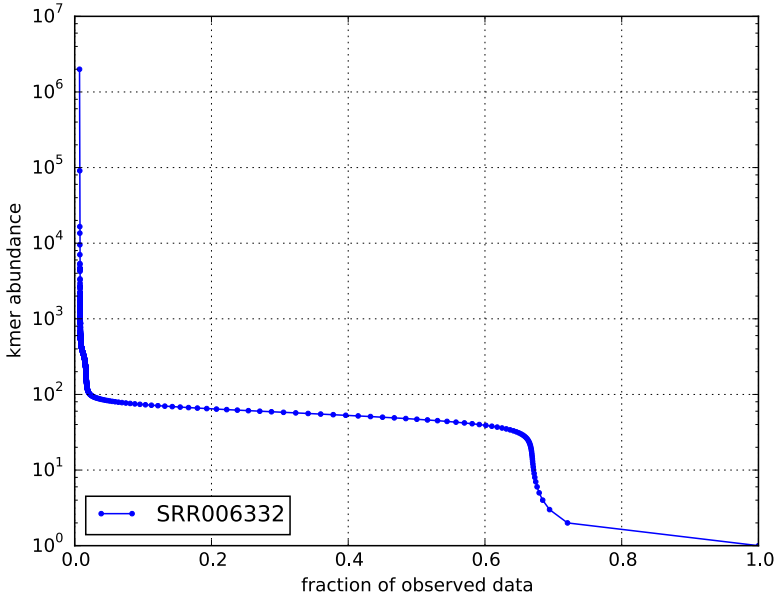
adapter 0.8% 35

repeats 3% 300x 8000

genome 61% 40x 3.5M

rare 35% 1x 79M



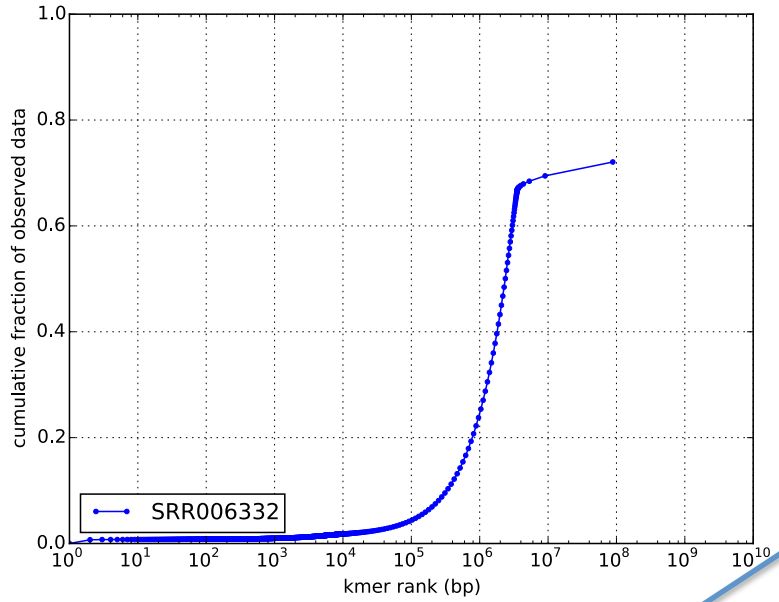


Yes, all these graphs are necessary

Genome size ~ number of kmers (M)

depth ~ abundance (C)

DATA FRACTION (F)

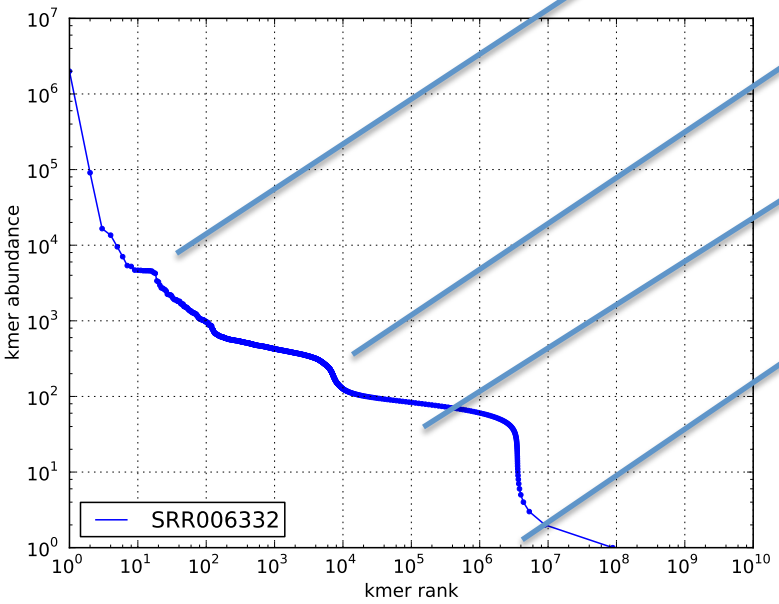


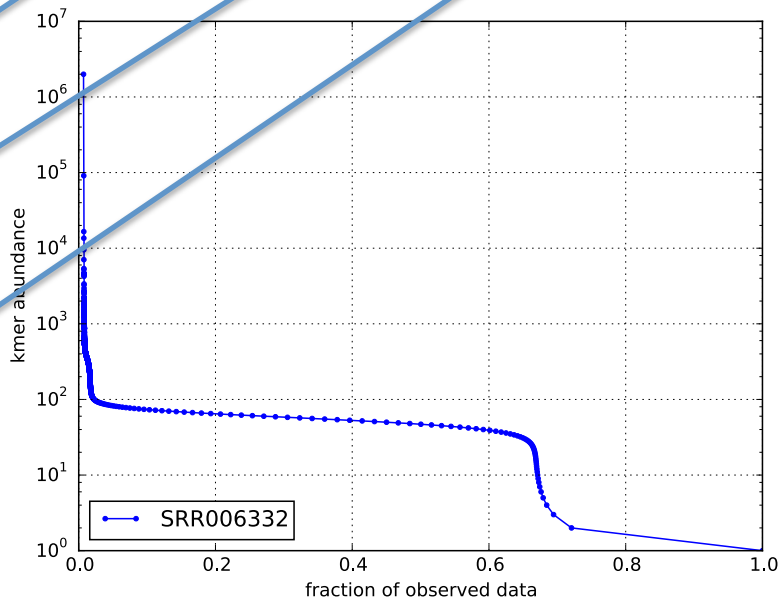
adapter 0.8% 35

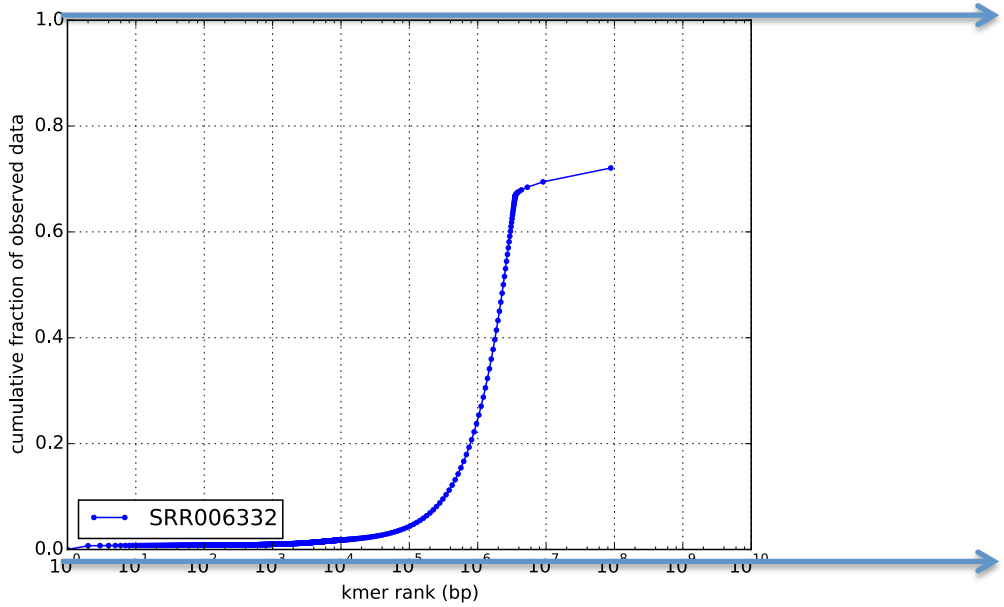
repeats 3% 300x 8000

genome 61% 40x 3.5M

rare 35% 1x 79M





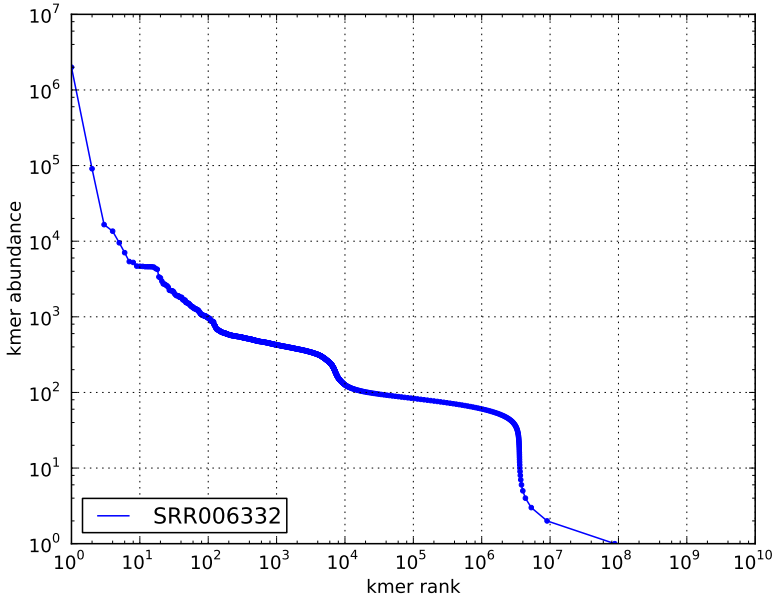


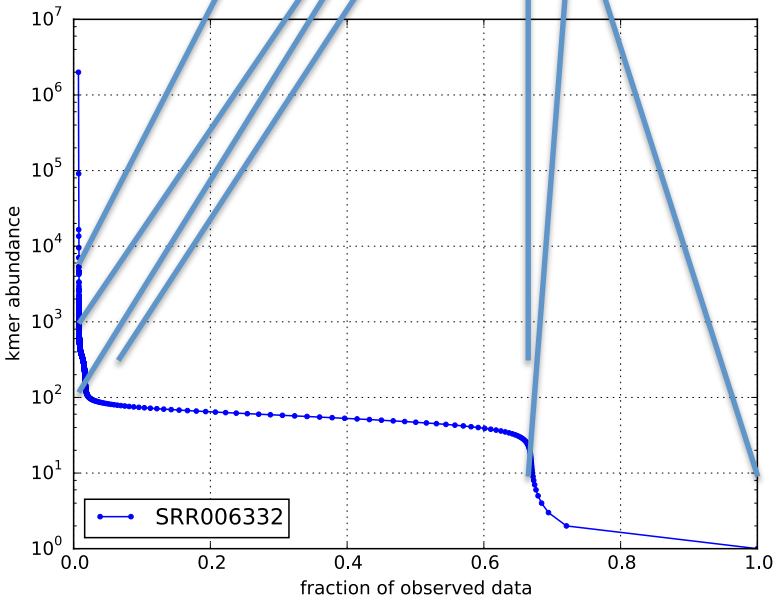
adapter 0.8% 35

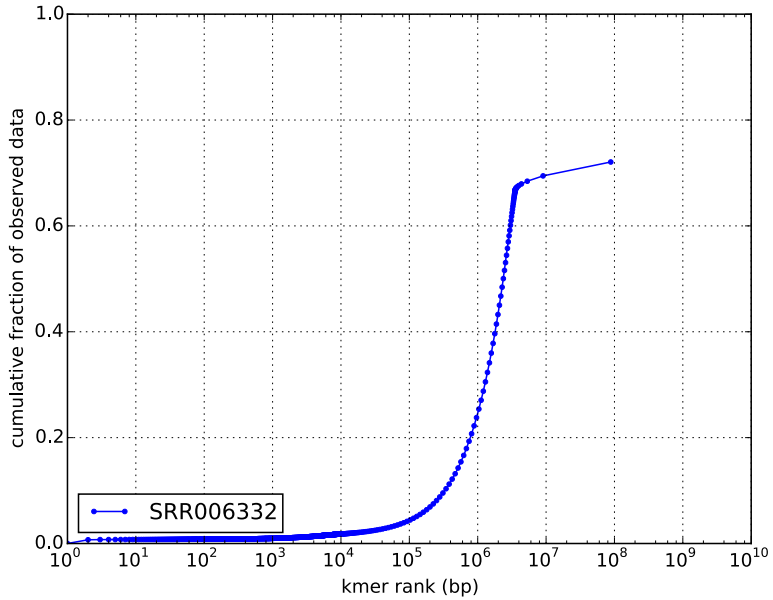
repeats 3% 300x 8000

genome 61% 40x 3.5M

rare 35% 1x 79M



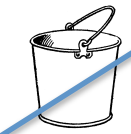




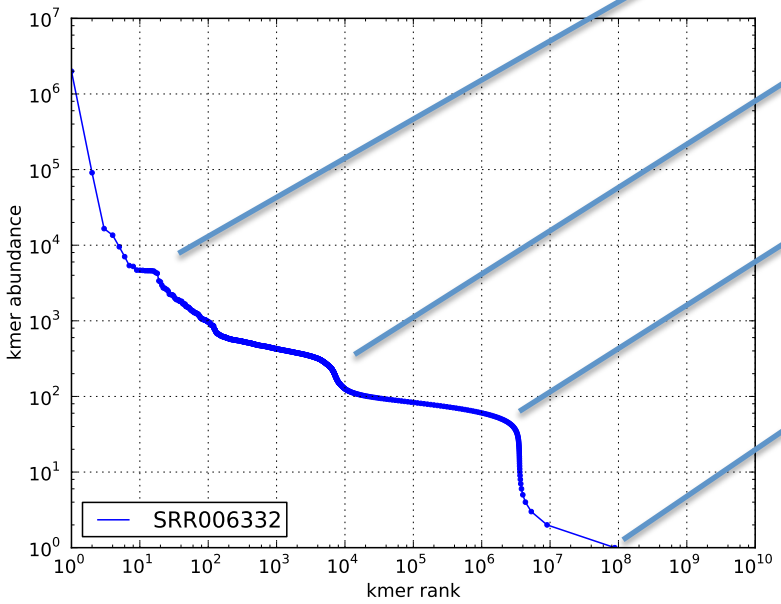
adapter 0.8% 35

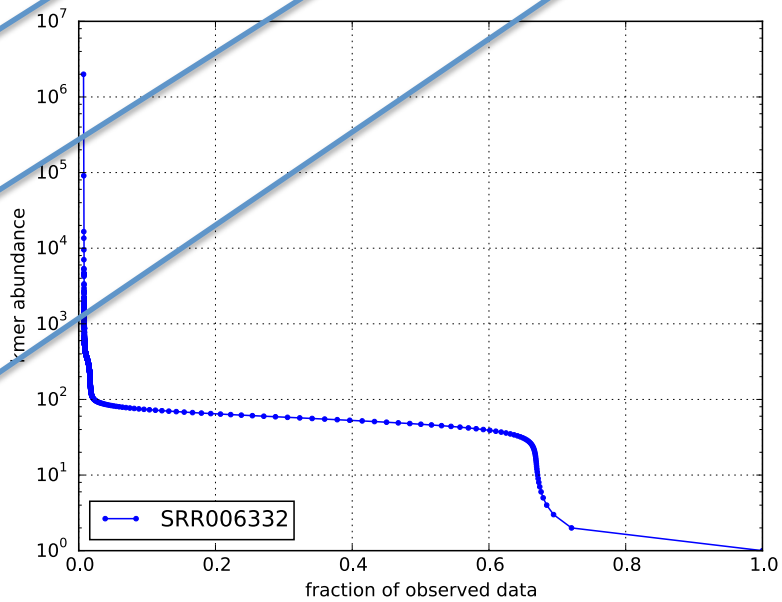
repeats 3% 300x 8000

genome 61% 40x 3.5M

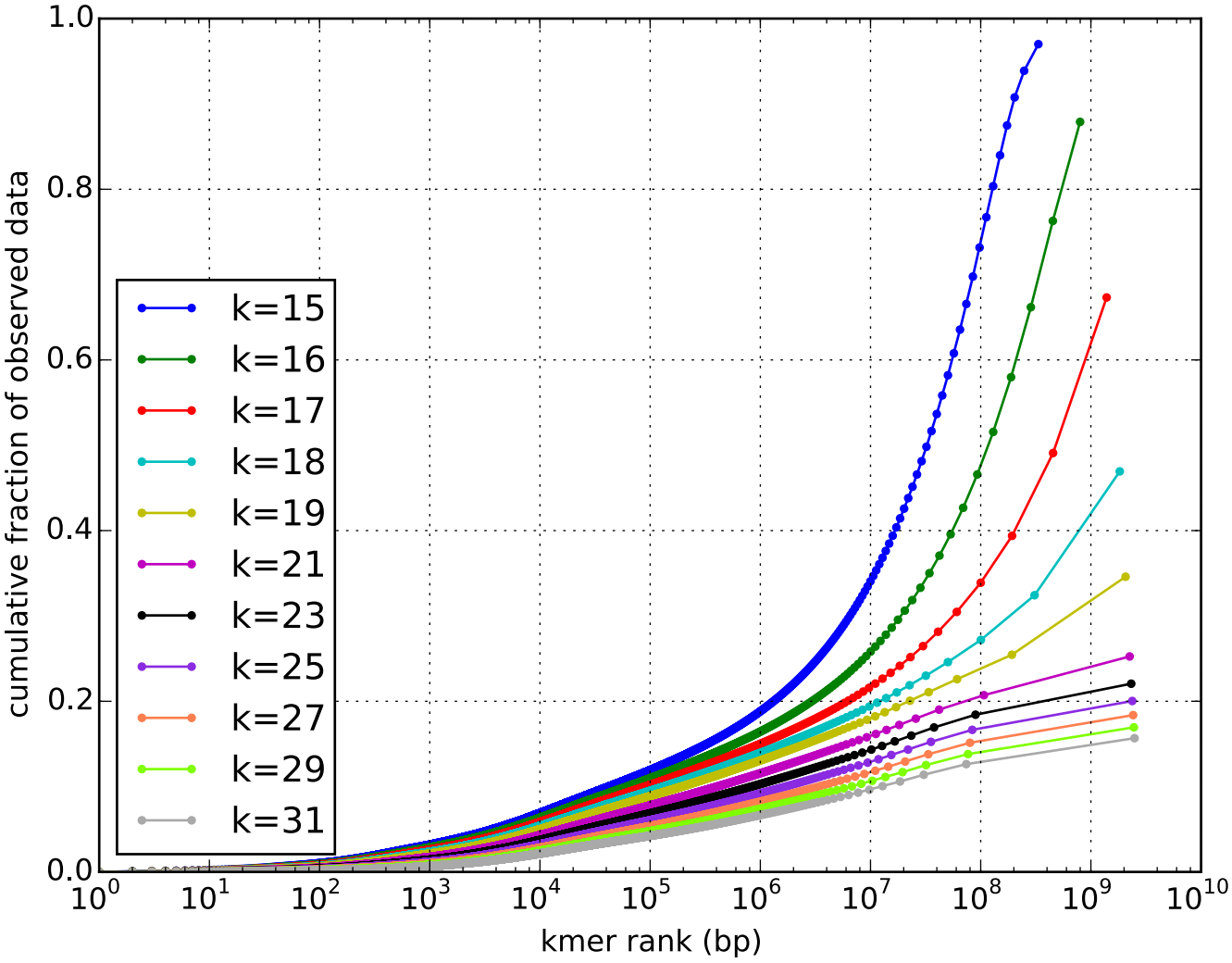


rare 35% 1x 79M





Check: Human reference genome



1 2141383266
2 65007581
3 16171452
4 7542756
5 4250262
6 2791059
7 1931443
8 1390566
9 1018739
10 794306
11 614425
12 507162
13 418706
14 353710
15 311034
16 266206
17 231321
18 202592
19 182020
20 164101

kmer statistical summaries

- H0 kmer richness (VERY BAD)
- H1 Shannon entropy (BAD)
- H2 Reyni entropy / Simpson index (GOOD)

- observation-weighted coverage (BAD)
- observation-weighted size (BAD)
- observation-median coverage (GOOD)
- observation-median rank (GOOD)
- fraction in top 100 kmers (USEFUL)
- fraction unique (OK but requires size correction)

kmer statistical summaries

- H0 kmer richness (VERY BAD)
- (BAD)
- x (GOOD)
- (BAD)
- (BAD)
- (GOOD)
- (GOOD)
- EFUL)
- fraction unique (OK but requires size correction)

Most of these give answers which vary so strongly with sampling depth as to be unusable.

Observation-weighted fraction-of-data metrics behave fairly well. Fractions of the data with particular properties are stable with respect to sampling.

kmer statistical summaries

- H0 kmer richness (VERY BAD)
- H1 Shannon entropy (BAD)
- H2 Reyni entropy / Simpson index (GOOD)

- observation-weighted coverage (BAD)
- observation-weighted size (BAD)
- **observation-median coverage (GOOD) C50**
- **observation-median rank (GOOD) M50**
- **fraction in top 100 kmers (USEFUL) F100**
- **fraction unique (OK but requires size correction)**

Generalities from the kmer counting mines

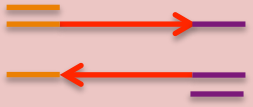
- Many datasets have as much as 5-45% of the sequence yield in **adapters**.
- **FEW DATASETS have well-separated abundance peaks** (of the sort metavelvet was engineered to find)
- Diverse datasets have a **featureless, geometric relationship** between kmer rank and kmer abundance.
- **Shannon entropy is oversensitive to errors.** *

Three approaches to error correction:

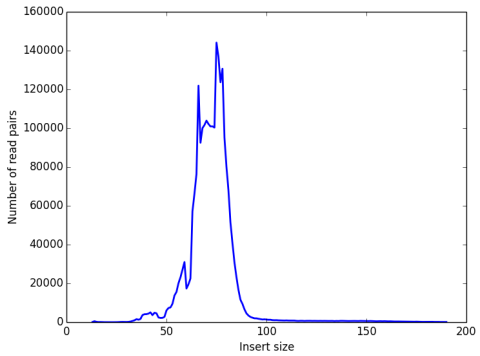
- Filtering (Q-values, tile quality)
- Pair-joining
- Error correction

Choosing insert size

0 < insert size < L
eats into adapters



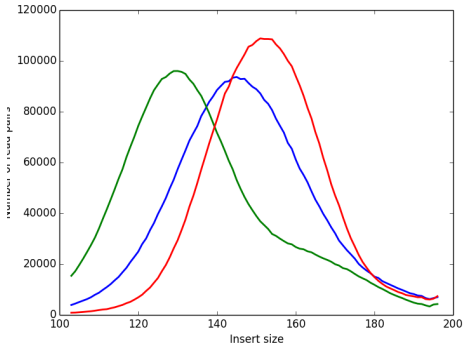
Sequencing adapter contamination



60% of sequencing yield wasted on adapters; half of the remainder is redundant.

overlapping reads
L < insert size < 2L

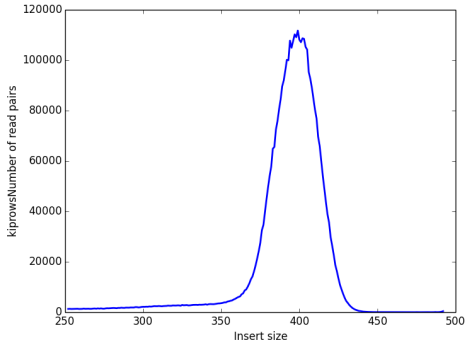
Error-corrected overlapping reads



25% of sequencing yield consumed in error-correcting overlaps. 75% of basepairs in reads 1.5-1.8L

unsequenced insert
2 L < insert size

Unsequenced insert
No error correction

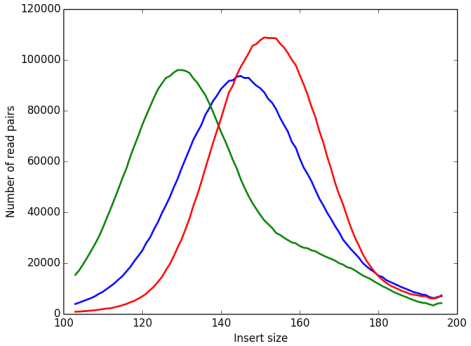


None of the sequencing is lost, but past 200bp the uncorrected quality is very poor.

Overlapping reads

overlapping reads
L < insert size < 2L

Error-corrected overlapping reads



- Careful size selection results in 90+ % of reads with overlapping ends.
- **Read joining is easy, cheap.**
- Produces pile of joined reads and two piles of unjoined reads (and a whole pile of headache keeping read pairs together)
- Error correction when reads differ.
- Overlap operation benignly reduces total coverage

- Size selection may disturb mixture proprtions (“bias”)
- What to do with the qualitatively different unoverlapped pairs?
- Keep them: possible size – length effect
- Discard them: possible size—depth effect

kmer statistical summaries

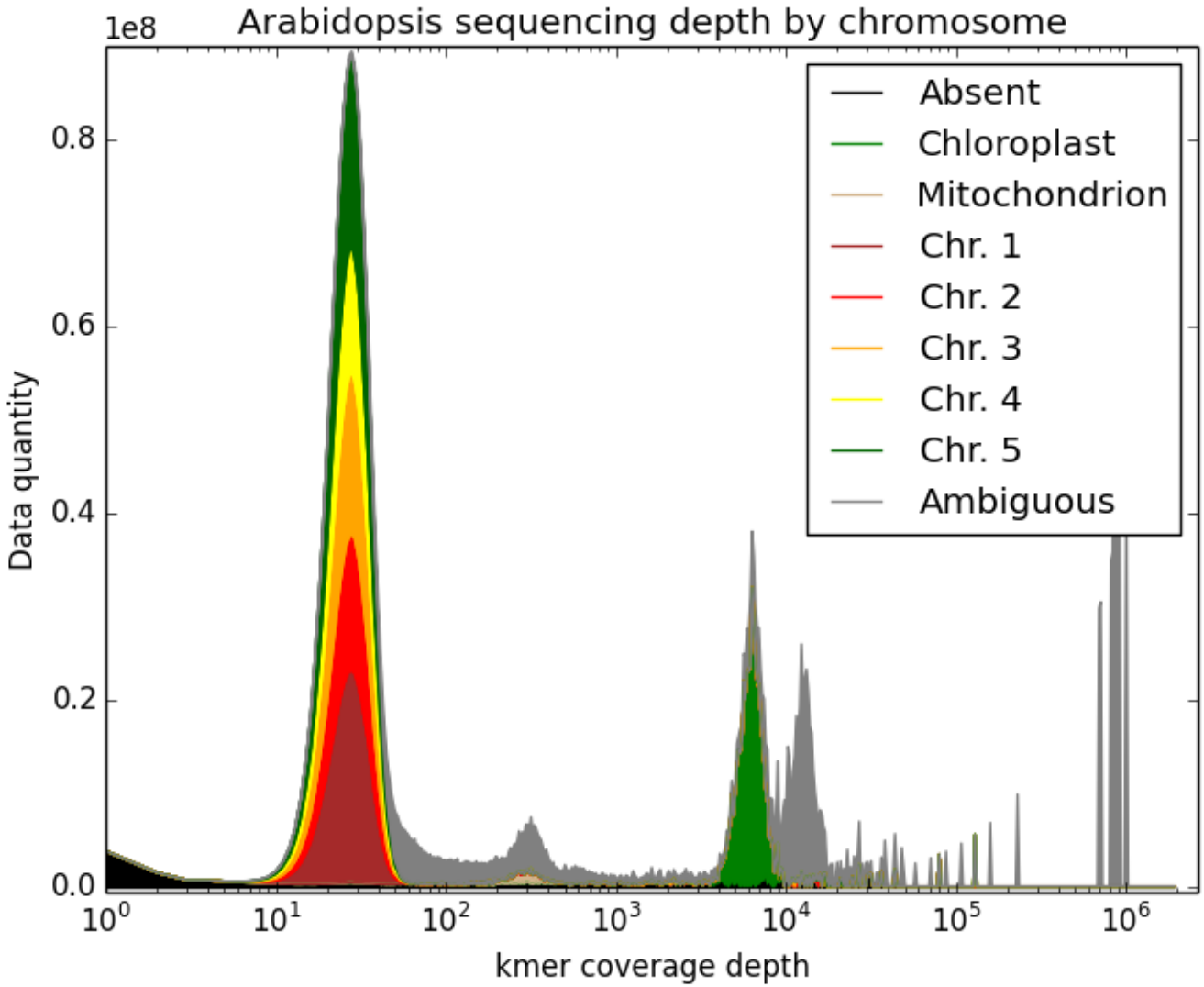
observation-median coverage (C50)

observation-median rank (M50)

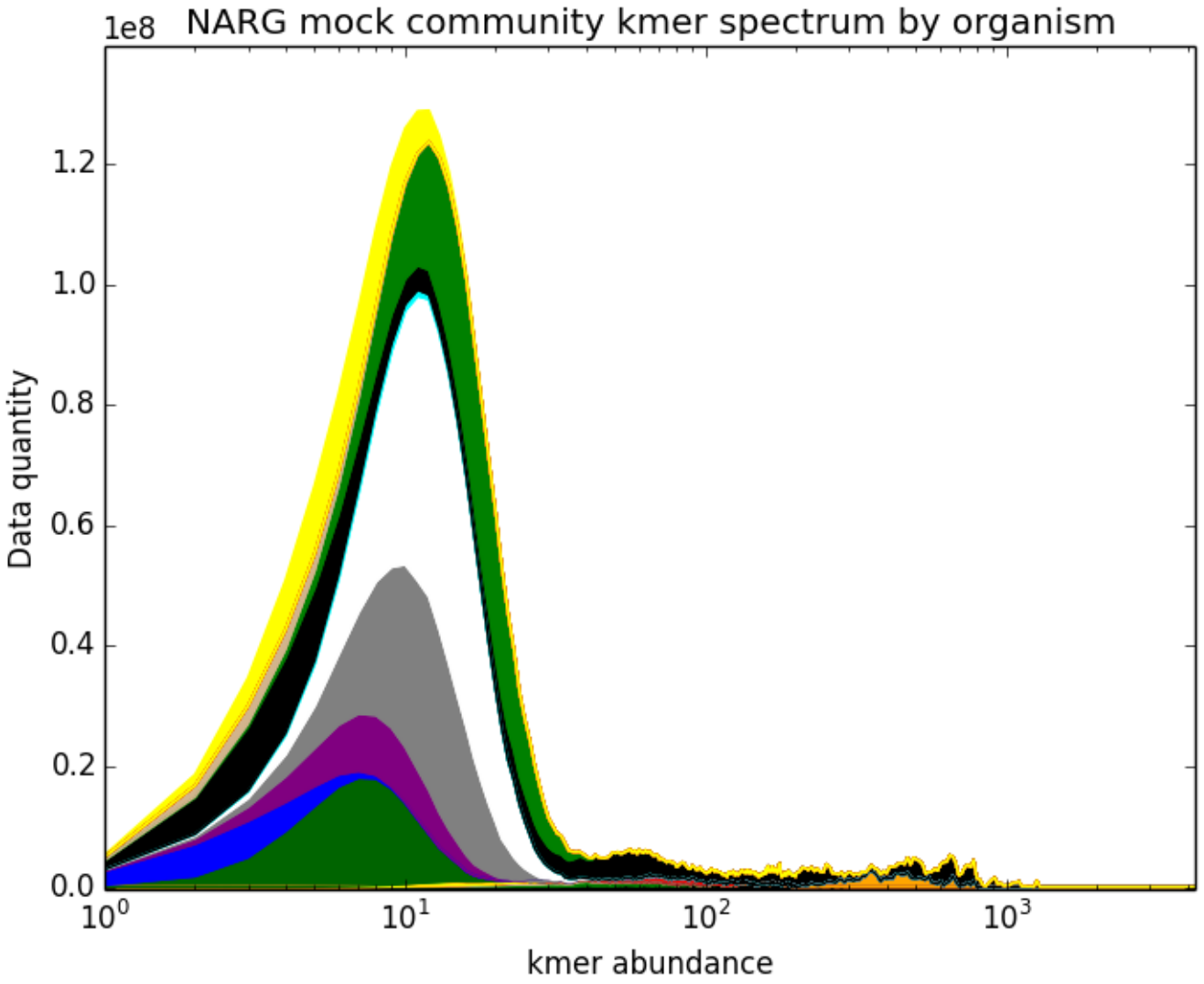
fraction in top 100 kmers (F100)

fraction unique

Straightforward abundance distribution: plant



Complex abundance distribution: mock community



The double-counting quirk

```
ATCGCGAAAAGTCCC 2
AAAAAAAAAAAAAAA 459
AAAAAAAAAAAAAAC 71
AAAATAAAAAAAATA 1
AAAAAAAAAAAAAAG 36
ACATGAAAAACAACT 1
AAAAAAAAAAAAAAT 23
AAAAAAAAAAAAACA 95
GTAGGAAAAGCCCAC 1
AAAAAAAAAAAAACC 7
AAAAAAAAAAAAACG 8
AAAAAAAAAAAAACT 9
AAAAAAAAAAAAAGA 36
AACAAGAAAAACAAA 1
AAAAAAAAAAAAAGC 10
AAATAAAAAAAATAG 1
AACAGAAAAAACACG 1
AAAAAAAAAAAAAGG 2
AAAAAAAAAAAAAGT 6
```

There is a sense in which
AAAAAAAAAAAAAAA
and
TTTTTTTTTTTTTTT
are the same.

2 approaches:

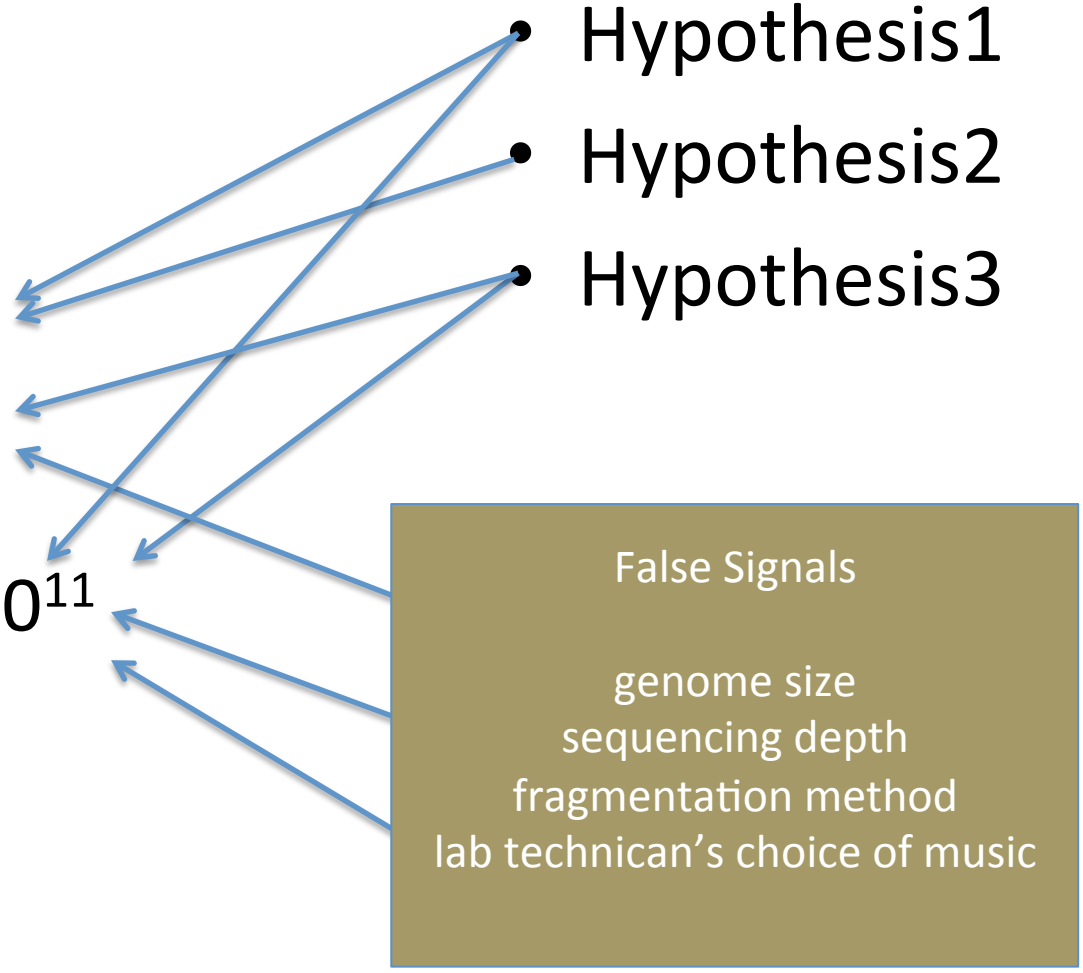
- double-count everything

- only count once

We don't care about the left side!

- Observation 1
- Observation 2
- …
- Observation $1+10^{11}$

- Hypothesis1
- Hypothesis2
- Hypothesis3



Generic experiment