Session 1: k-mer histograms and genome profiling

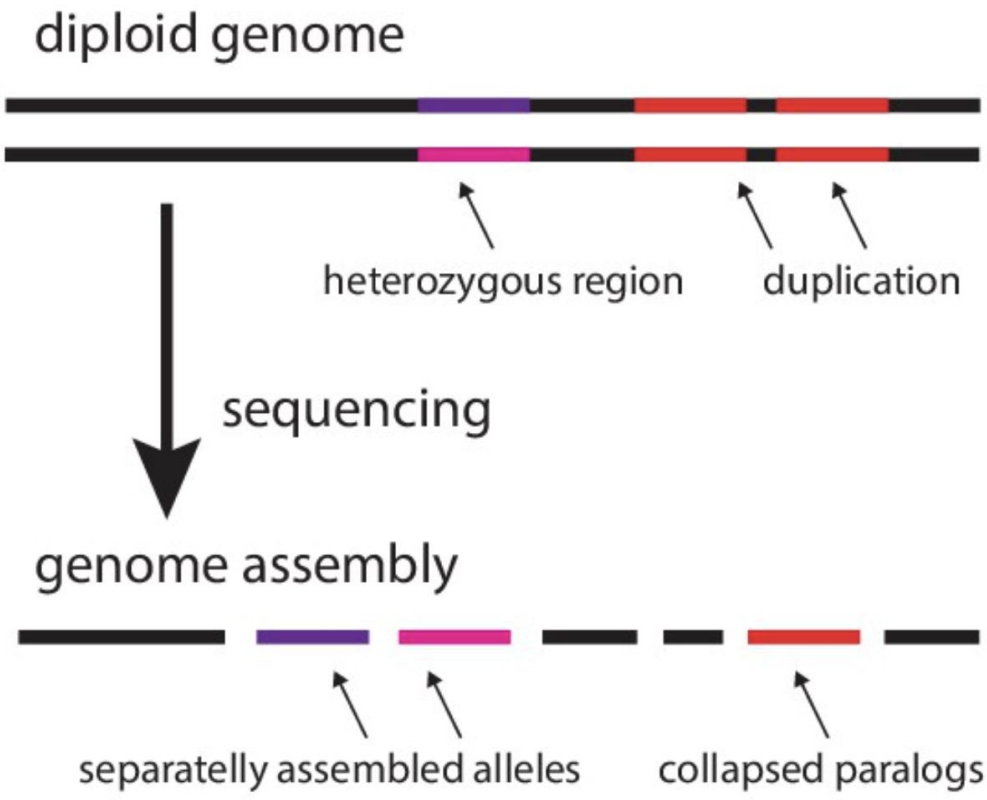
Kamil S. Jaron

2nd June 2025

Genome sequencing

assembly

Assembly problems



Genome sequencing

assembly

genome profiling

13e6 individual plants in the harvest;
each weighted 843g and had 234 kernels on average

Part 1: Understanding *k*-mer histograms

for all reads in the bag…

ATCTAAACGATCGATCGATCGA read

ATCTAAA

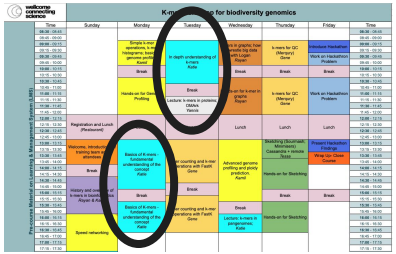
TCTAAAC

CTAAACG

TAAACGA

kmers (k = 7)

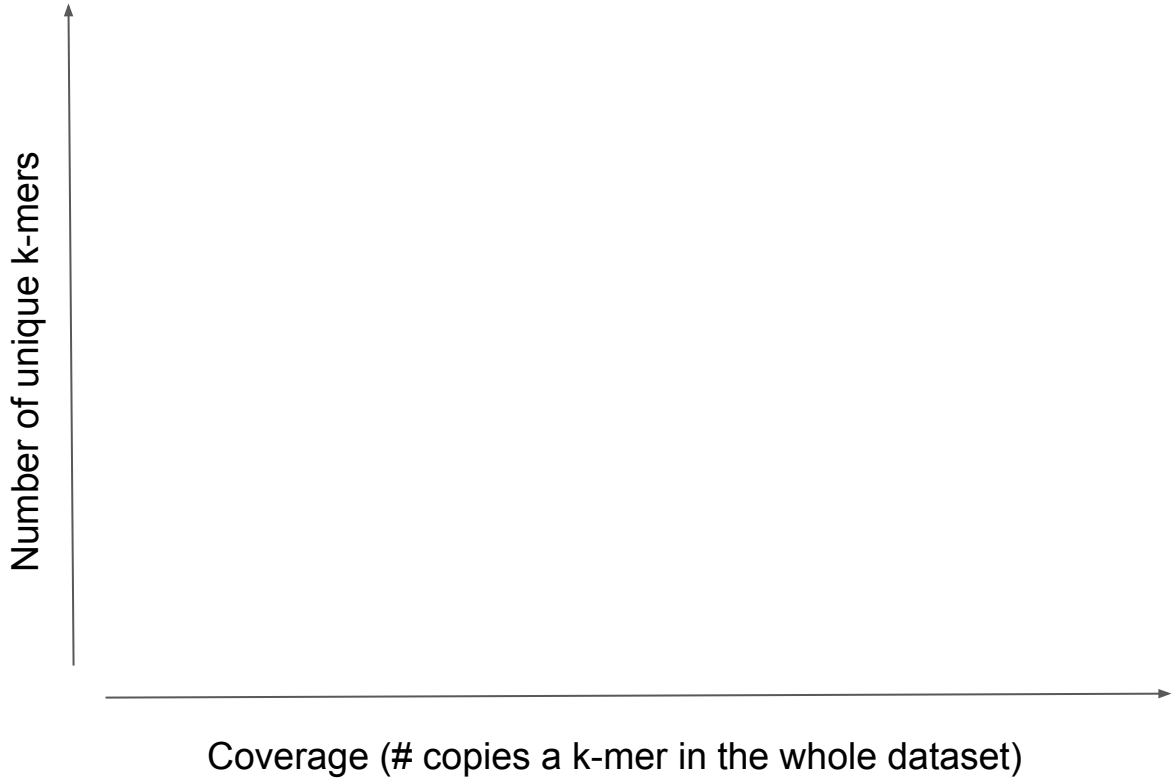
. . .



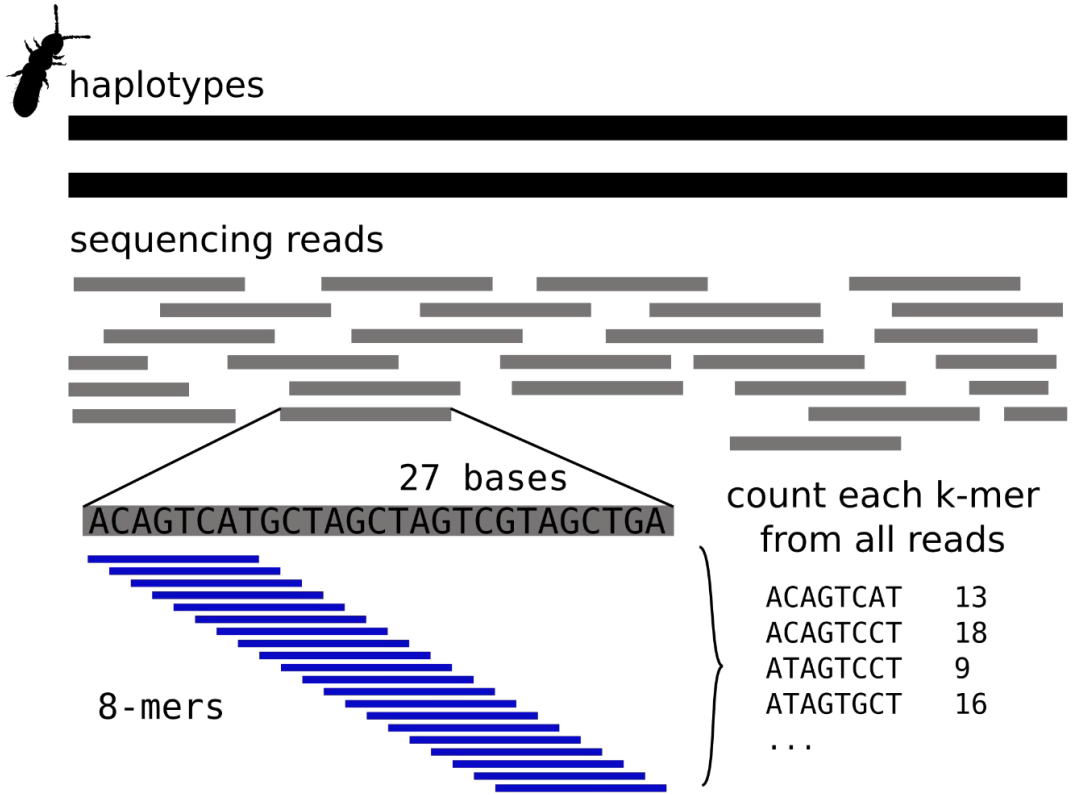
Counting frequency of all k-mers in all sequencing reads

kmer	coverage
AAAAAAAAAACAACGT	28
AAAAATAACACAACGT	31
AAAAATAACACAACGG	1
AAAAATAACGCAACGT	47
AAAATTTACGCAACGT	2
AAAATTTACGCAACGA	17
...	...

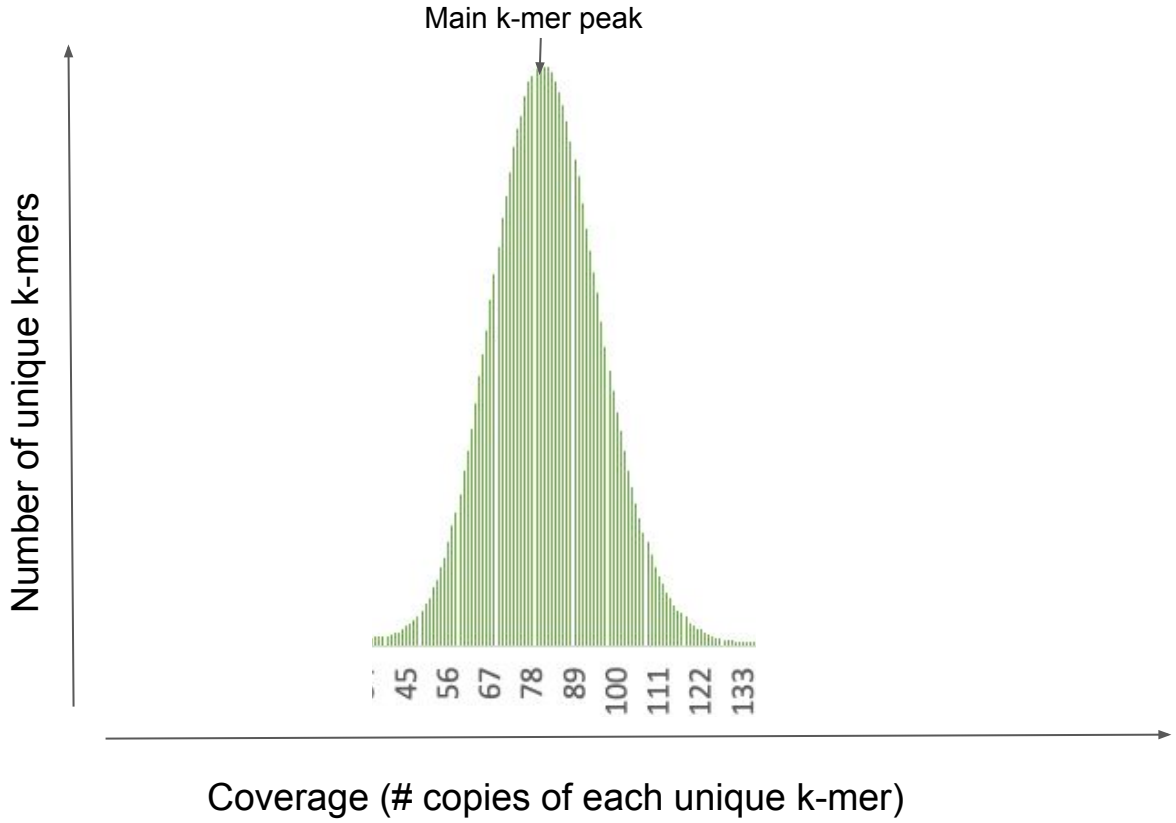
Plotting a k-mer histogram



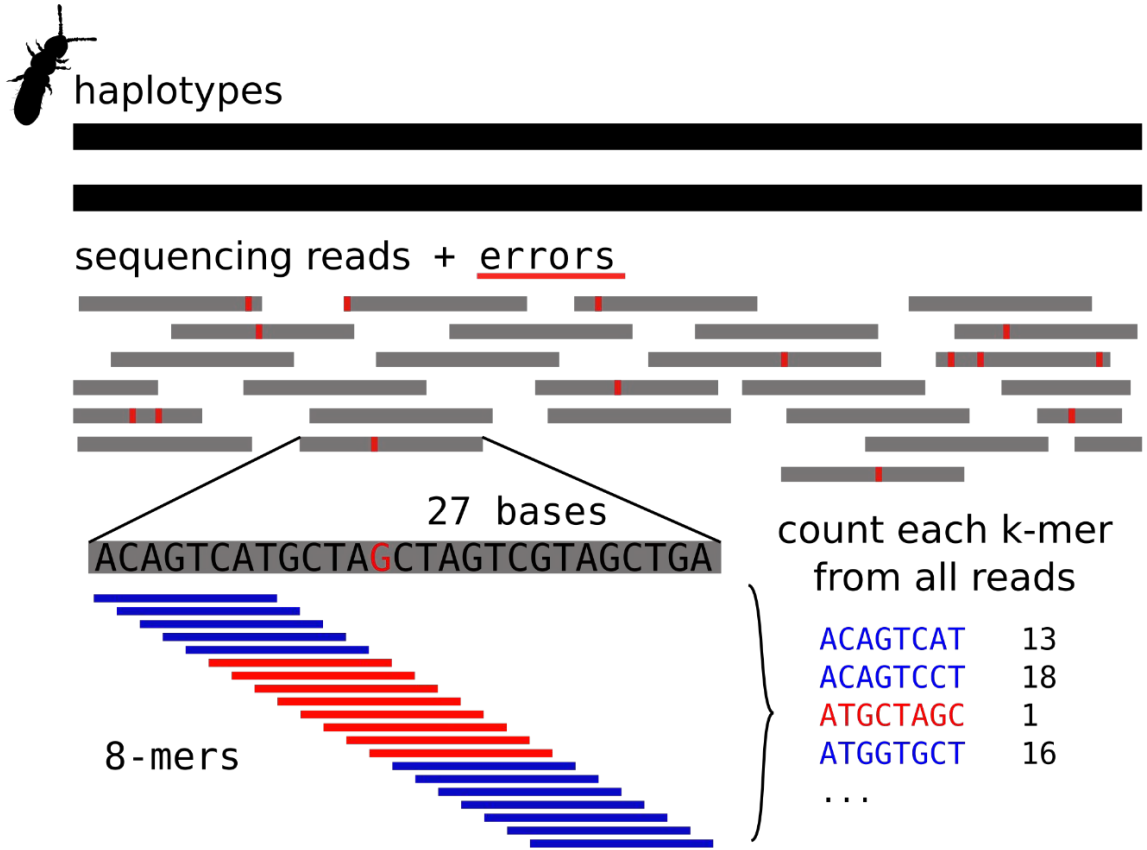
Plotting a perfect k-mer histogram



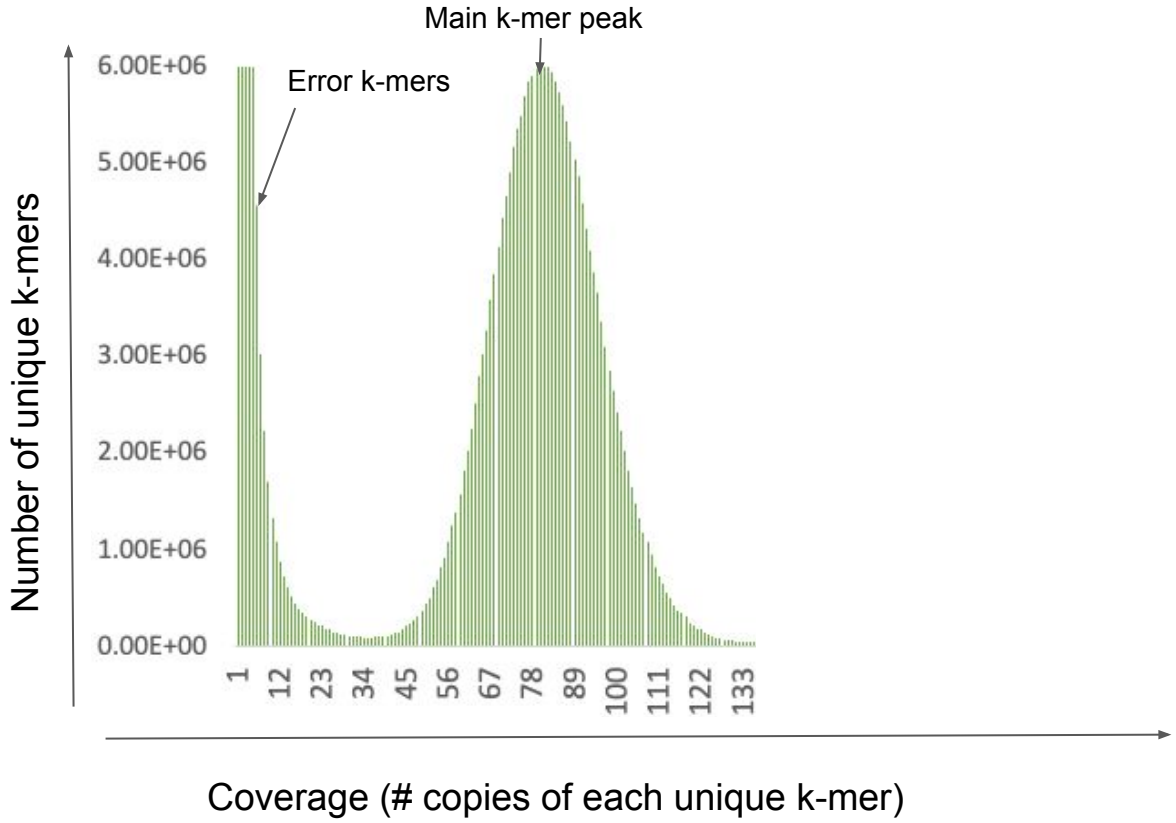
The perfect k-mer histogram



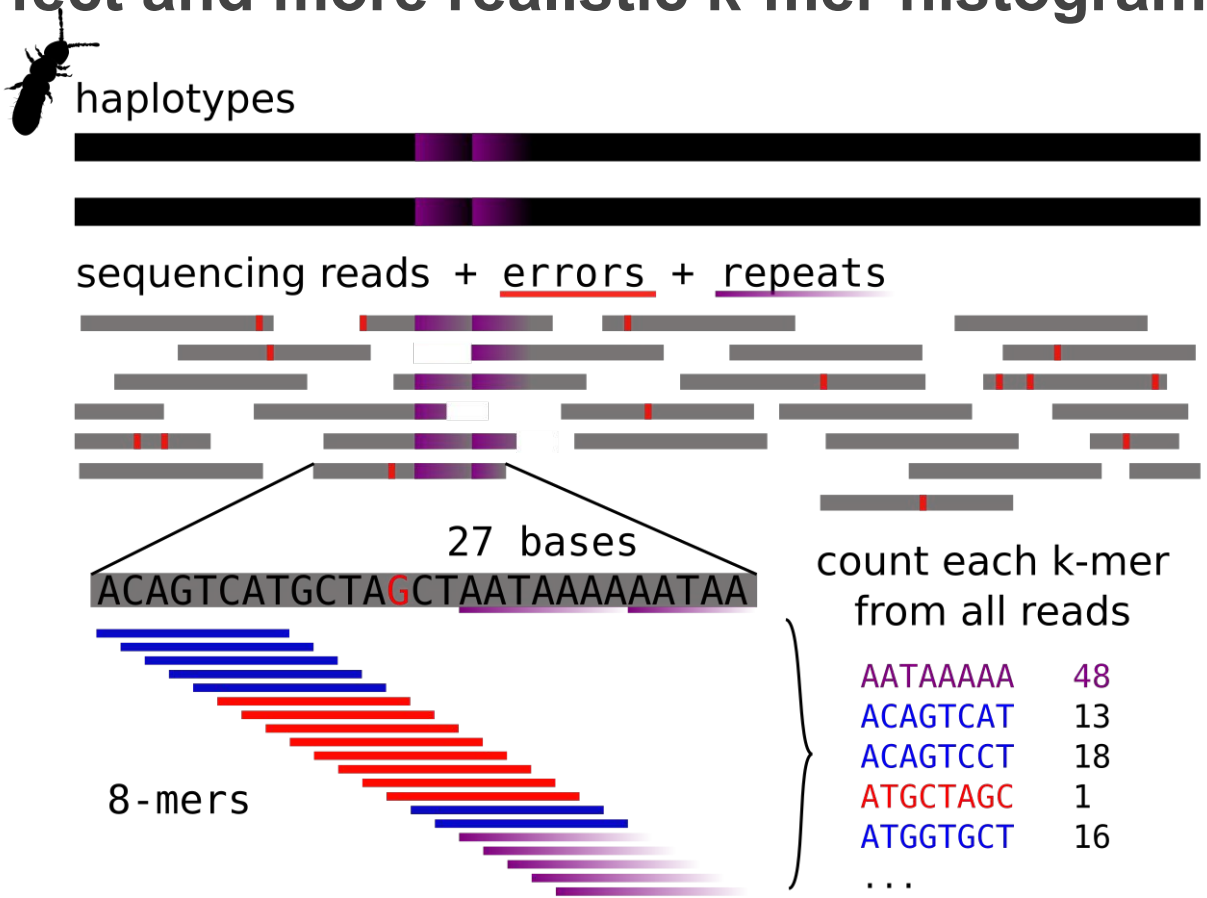
Creating an imperfect k-mer histogram



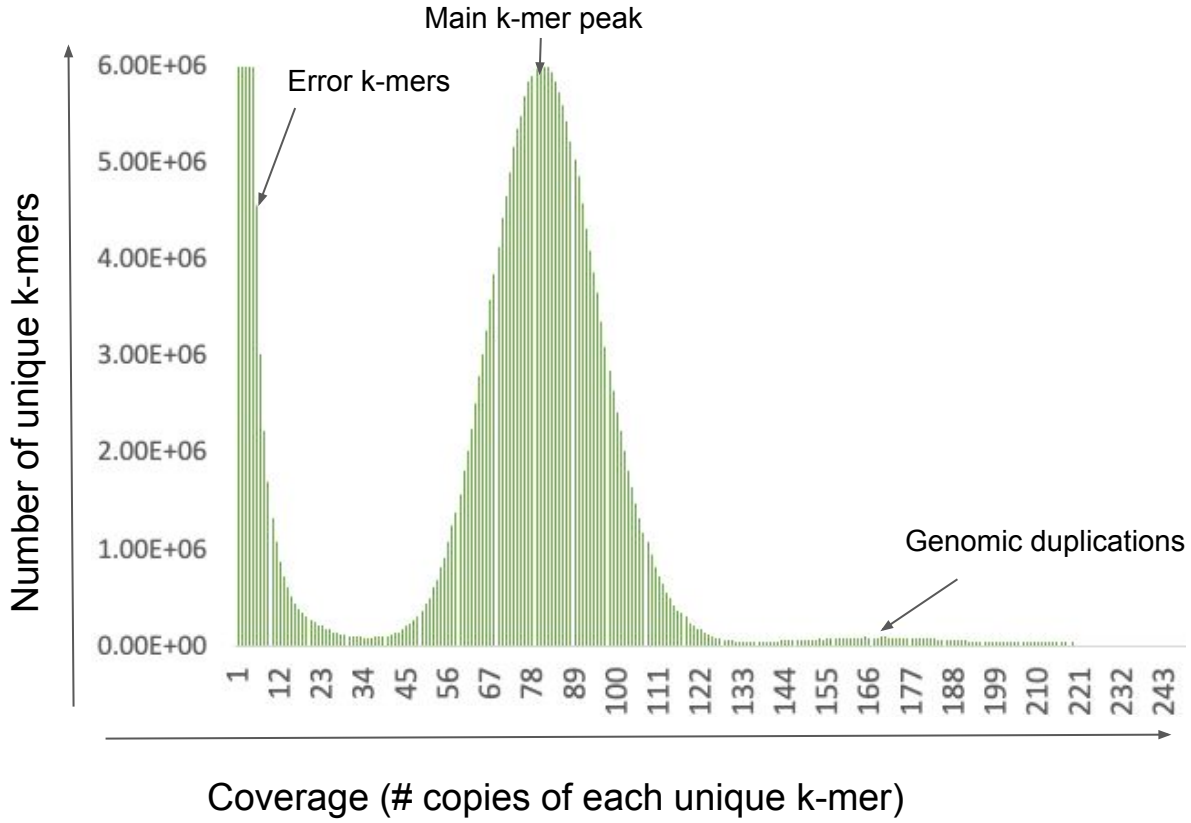
An imperfect histogram



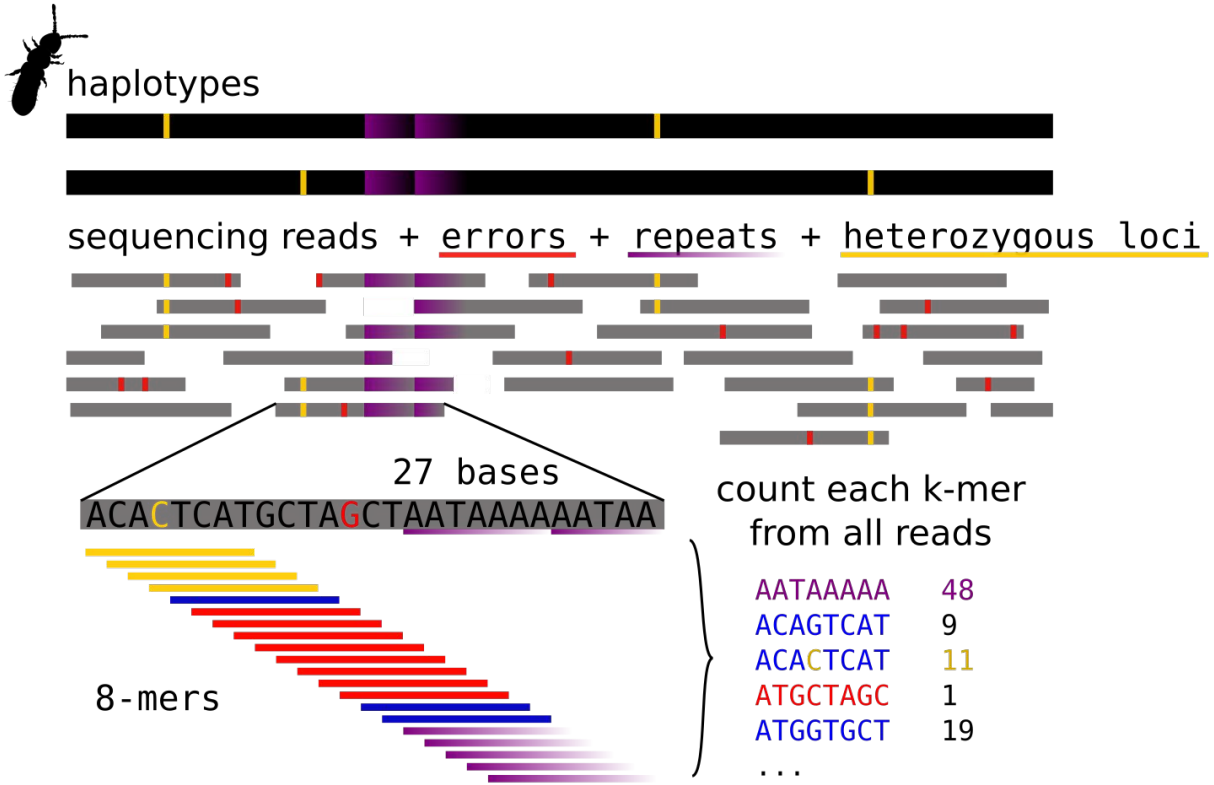
An imperfect and more realistic k-mer histogram



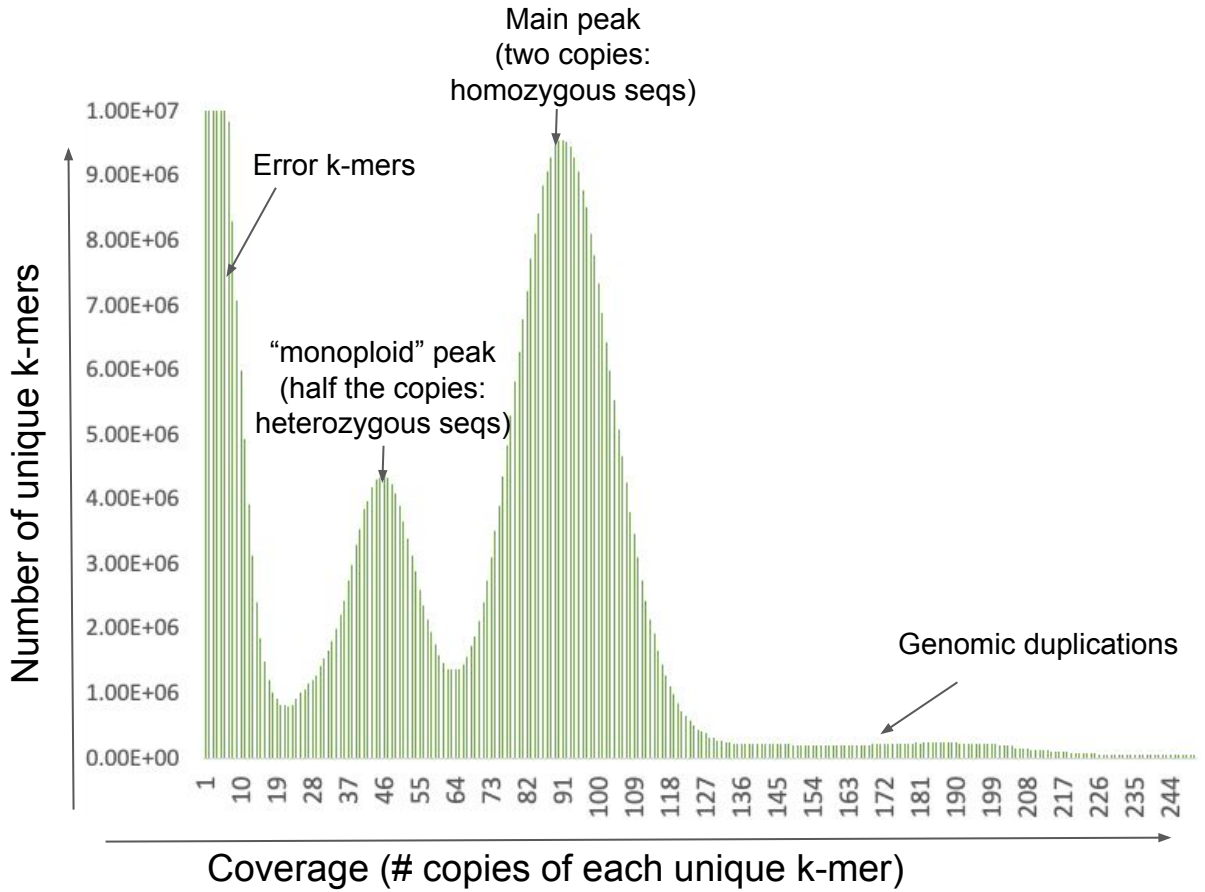
An imperfect and more realistic k-mer histogram



Counting k-mers…

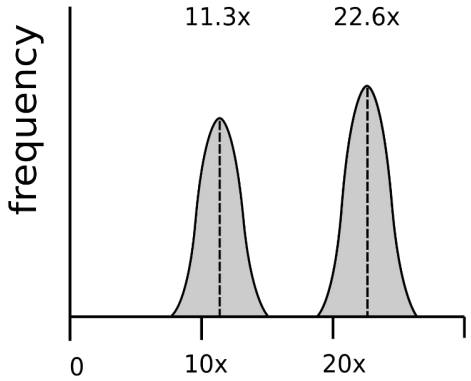


A realistic imperfect and non-inbred k-mer histogram

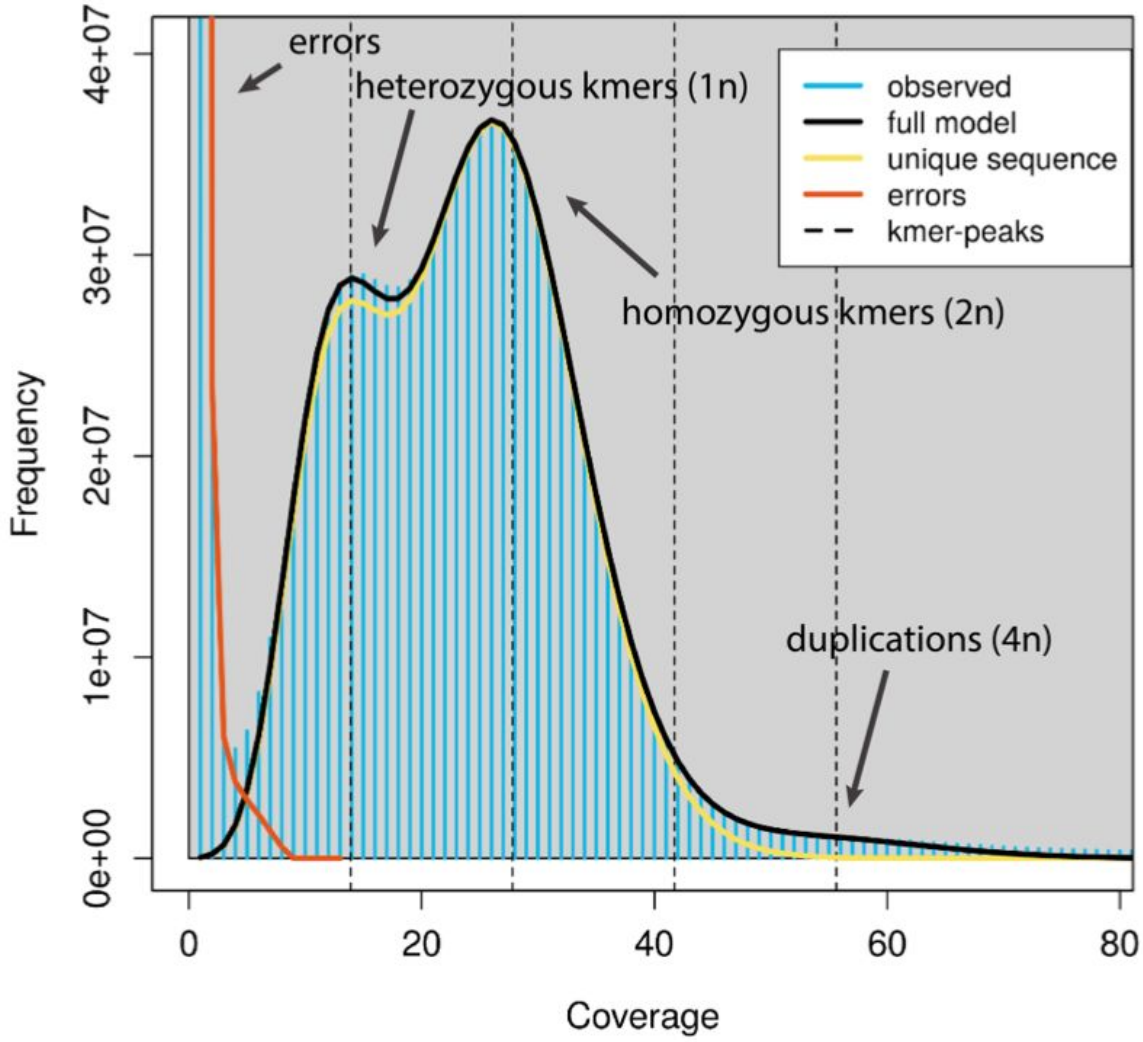


The principle of stoichiometry

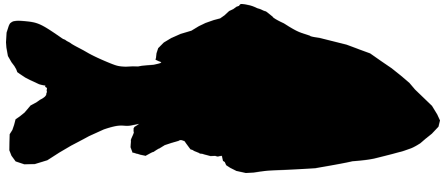
The *k*-mers that occur in a single copy in the genome (e.g. heterozygous in diploid) will have on average ½ coverage of the the *k*-mers that are present in two copies (homozygous in diploid)



There are millions of *k*-mers in each datasets, the deviations are not statistical but technical or biological!!!


errors
heterozygous kmers (1n)
homozygous kmers (2n)
duplications (4n)
observed
full model
unique sequence
errors
kmer-peaks
Frequency
0e+00
1e+07
2e+07
3e+07
4e+07
0
20
40
60
80
Coverage

Common bream



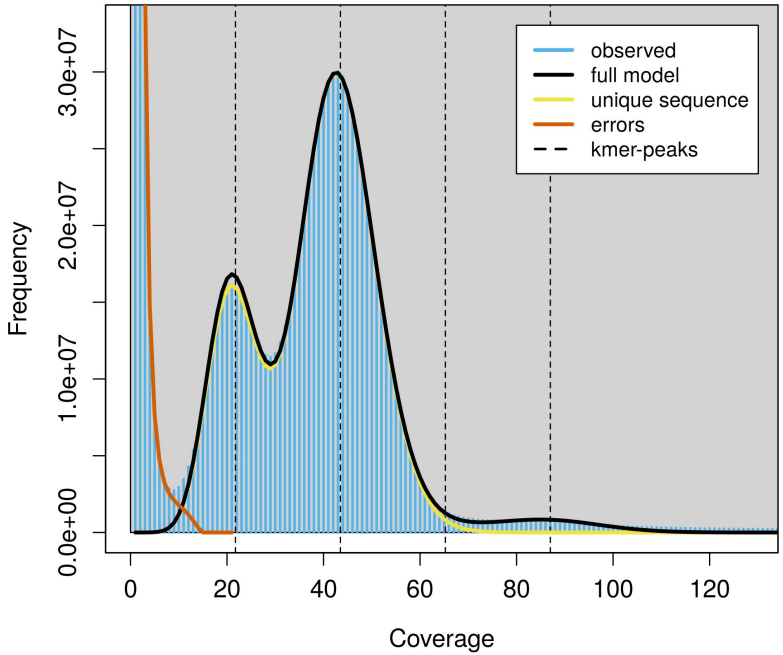
Do we have enough coverage? Did sequencing run go well?

Rule of thumb - do you see clear genomic peak(s)?

Yes: Probably

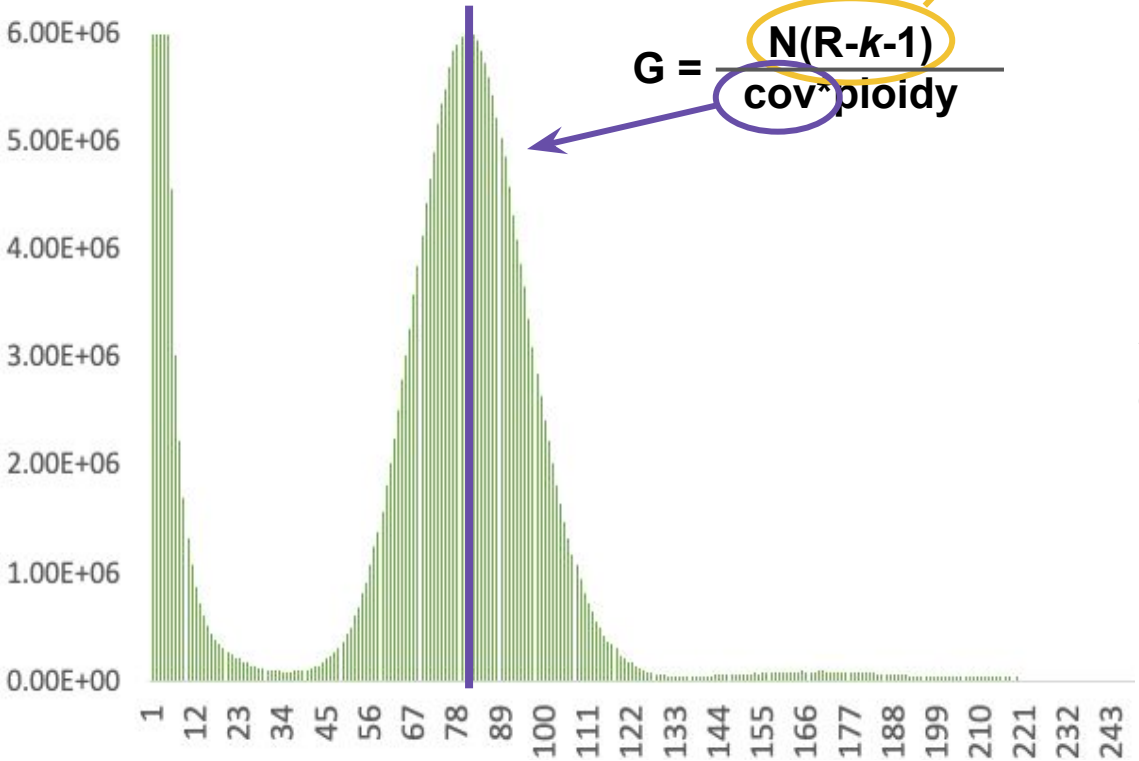
Kind of: Possibly

No: Most likely no!



Part 2: Genome profiling

Modeling genome size



$$G = \frac{N(R-k-1)}{cov * ploidy}$$

∑ freq (y-axis) * cov (x-axis)

G = genome size (bp)

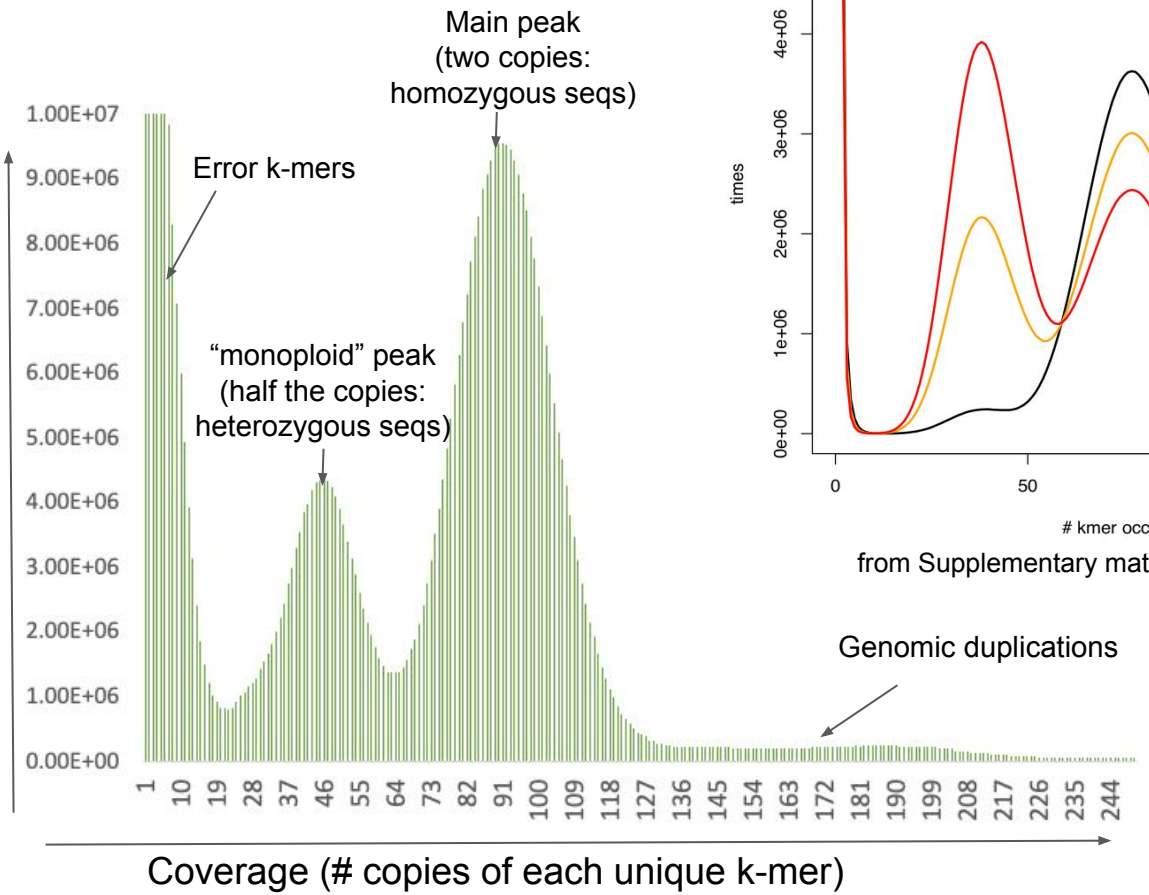
N = number of reads

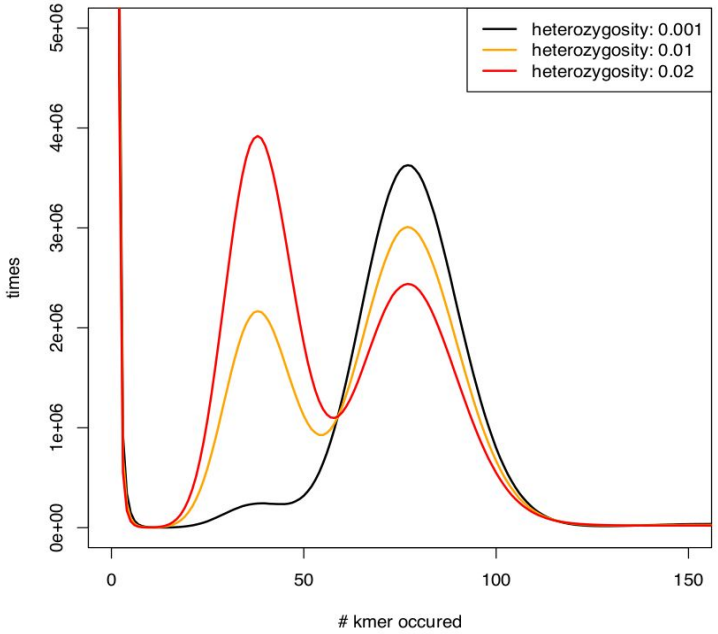
R = read length

k = k-mer length

cov = peak coverage

Modeling heterozygosity





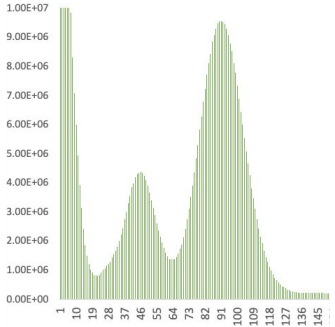
from Supplementary materials of Vurture et al. 2017

The genome profiling workflow

sequencing reads

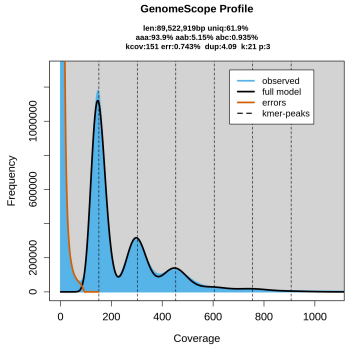
counting k-mers

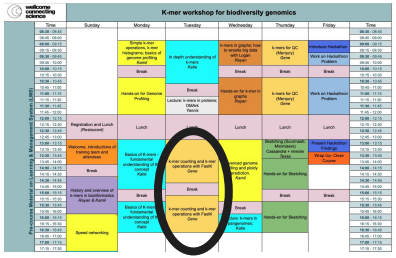
k-mer spectrum



fitting a model

“genome profiling”





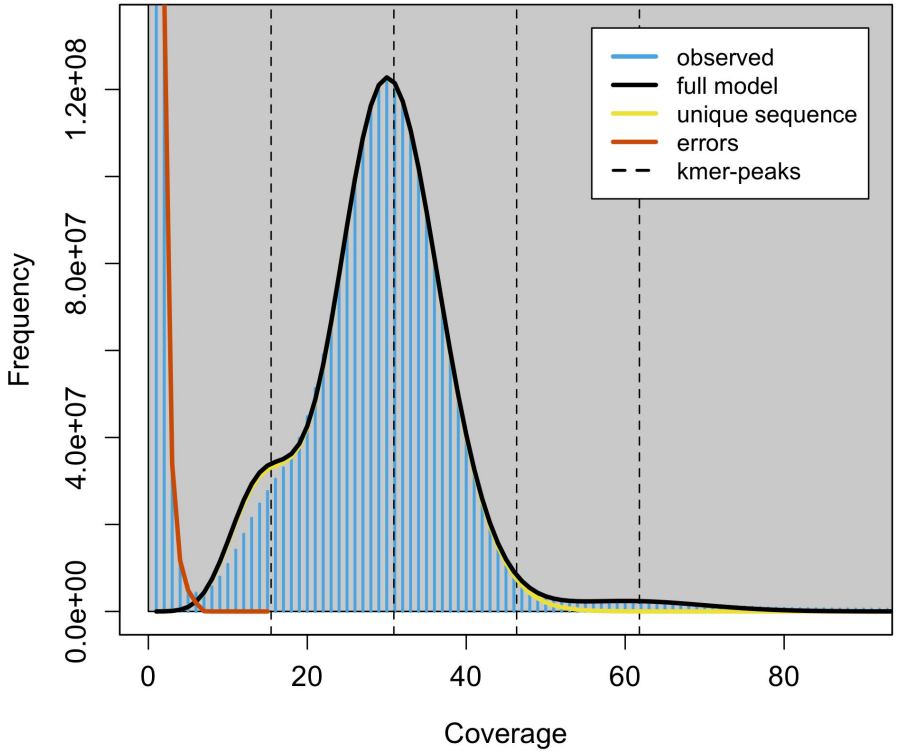
k-mer counters:

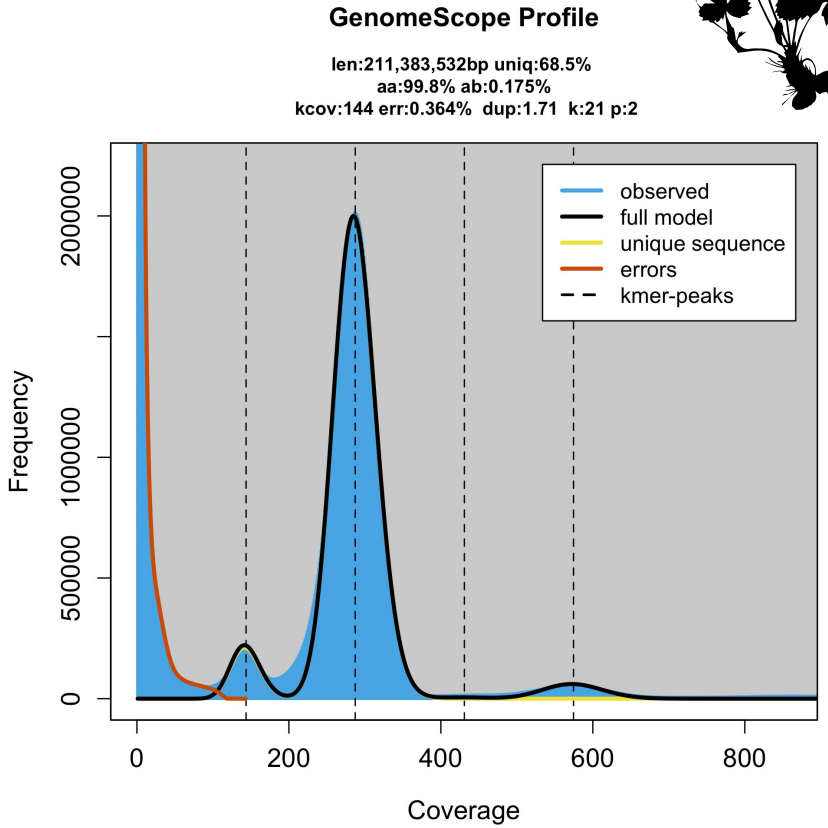
- JellyFish
- KMC
- **FastK**

Genome profiling tools:

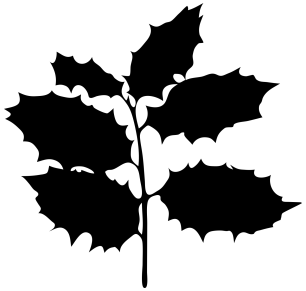
- **GenomeScope**
- RESPECT
- Tetmer

For sane high coverage genomes spectra look good!



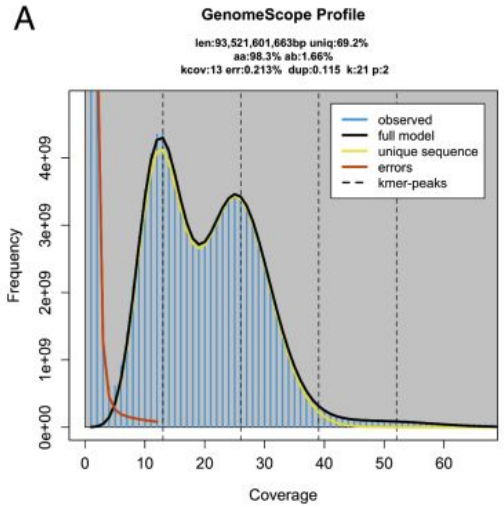


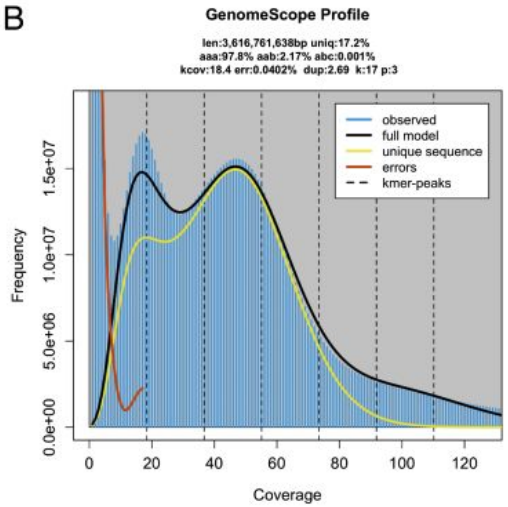
Coverage patterns

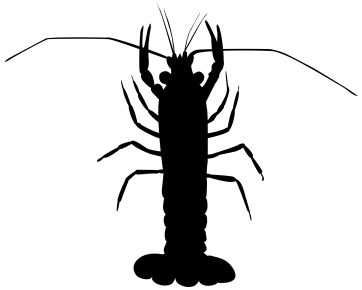


Great

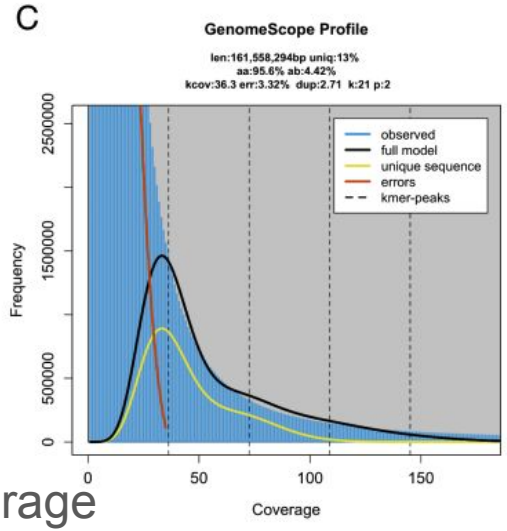
Not enough coverage

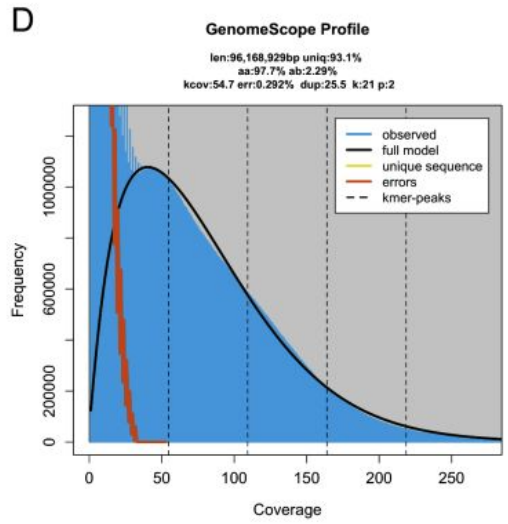


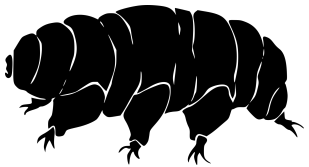




Perhaps ok?







Contamination

Contamination is one and the only exception of possible good run without a clear coverage peak

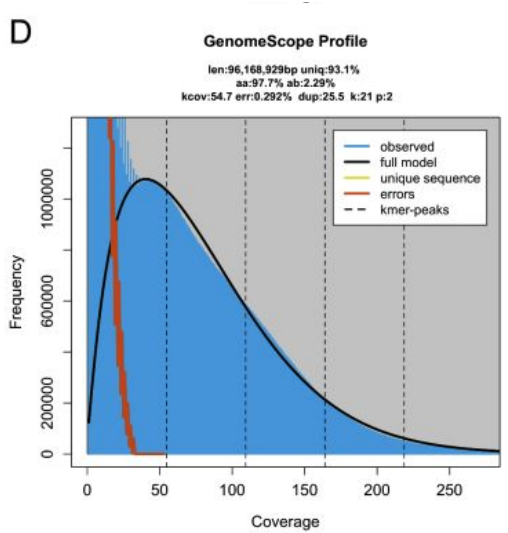
Would apply only for samples where very high levels of contamination can be expected… sequencing guts, environmental or otherwise messy samples…

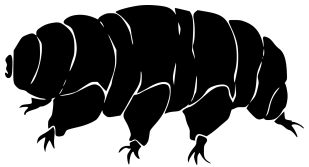
Rule of thumb - do you see clear genomic peak(s)?

No? Is it a tiny messy sample, is the genome size in the right ballpark?

Yes - Might be fine

No - Low coverage!



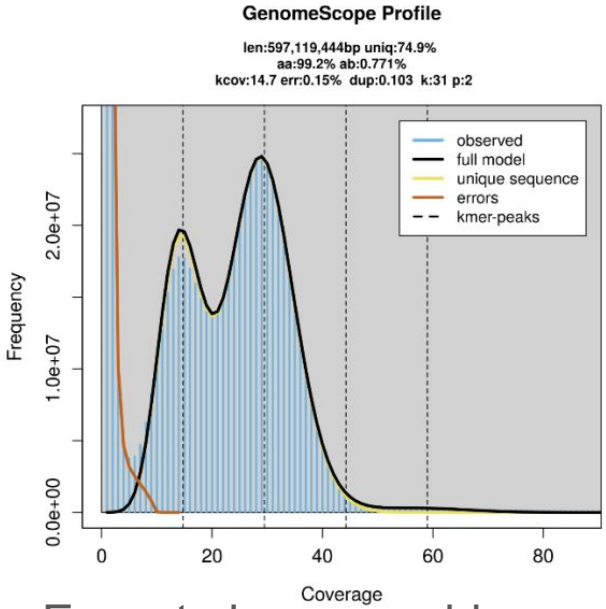


Contamination

Sequencing biases can make “coverage bridge”

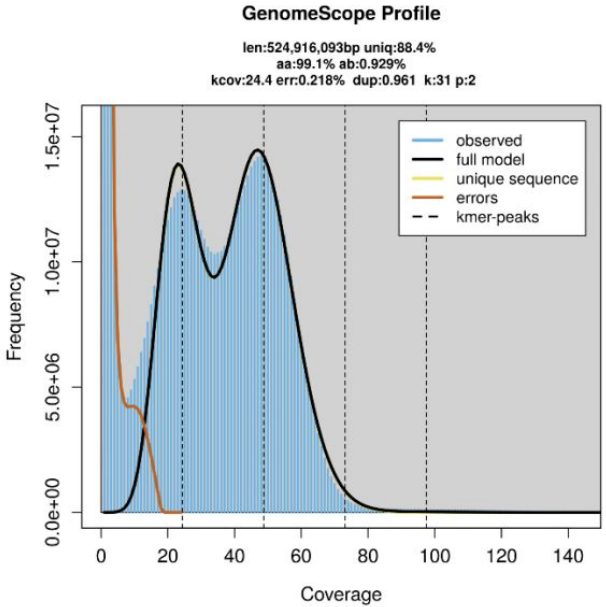
(GA-rich regions)

A. stone loach

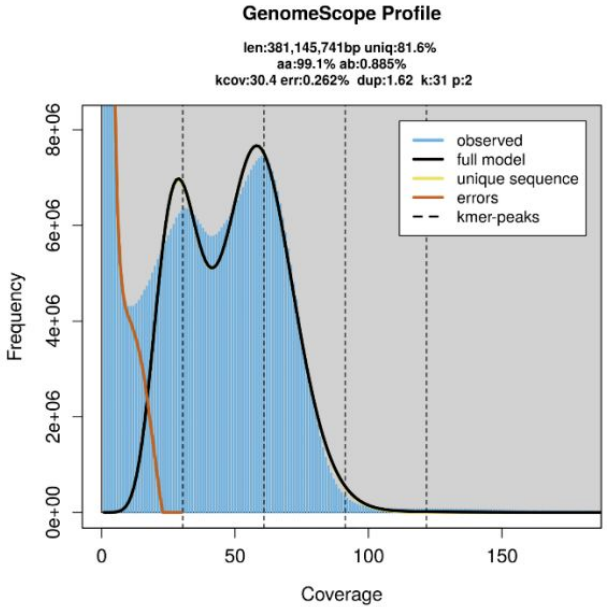


Expect nice assembly

B. European flounder

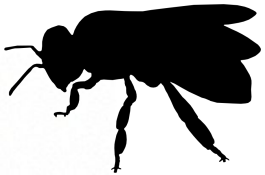


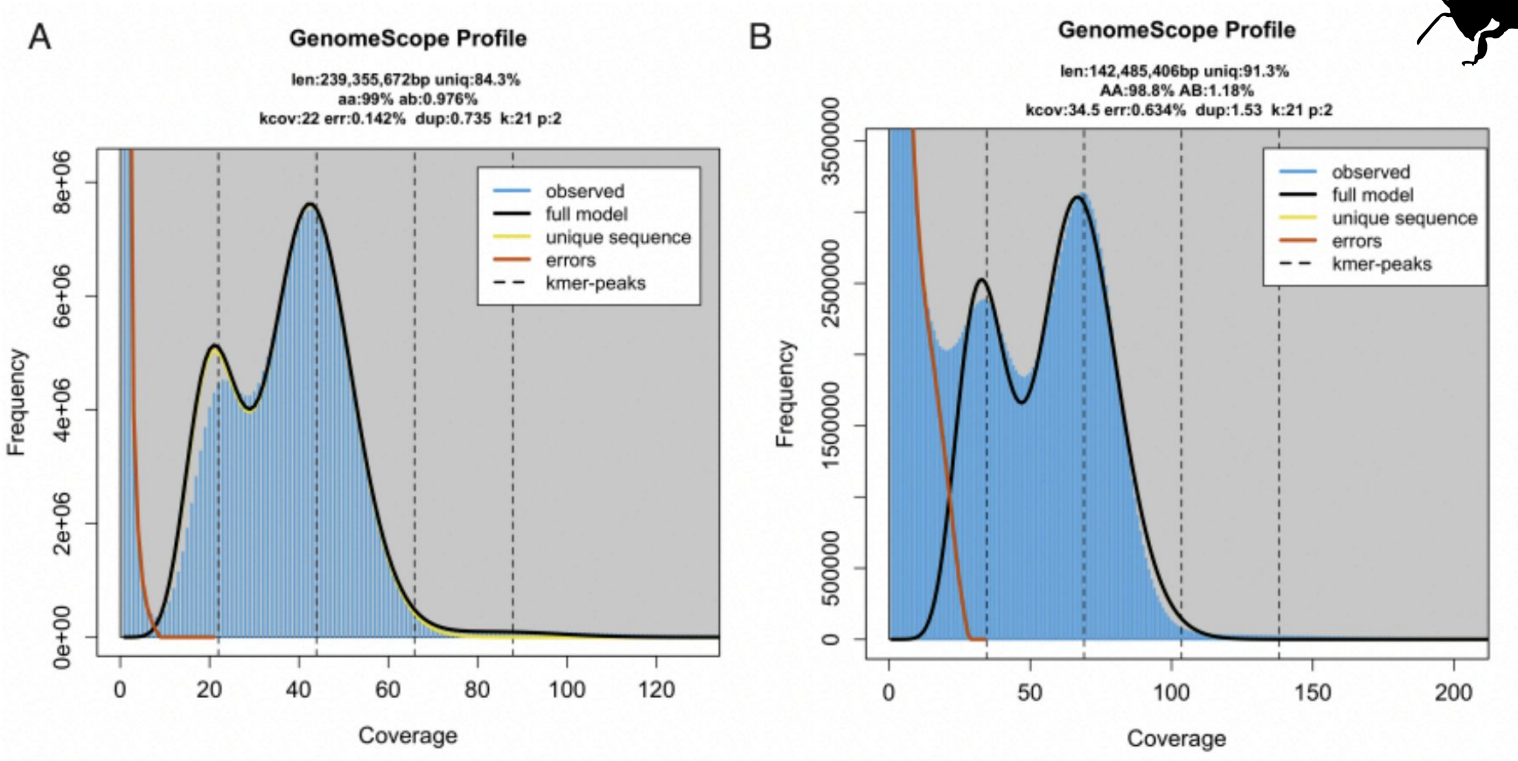
C. nine-spined stickleback



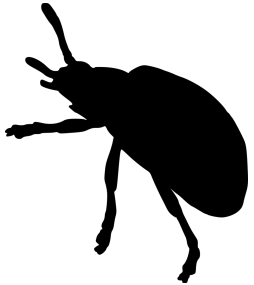
Expect fragmented assembly

“Coverage bridge” can be due to poor sample handling



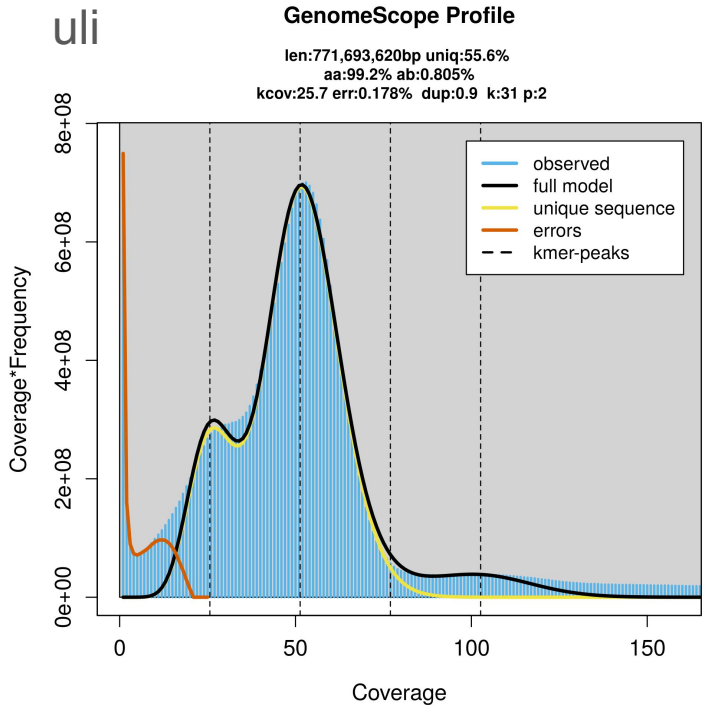


“Coverage bridge” can be due amplification



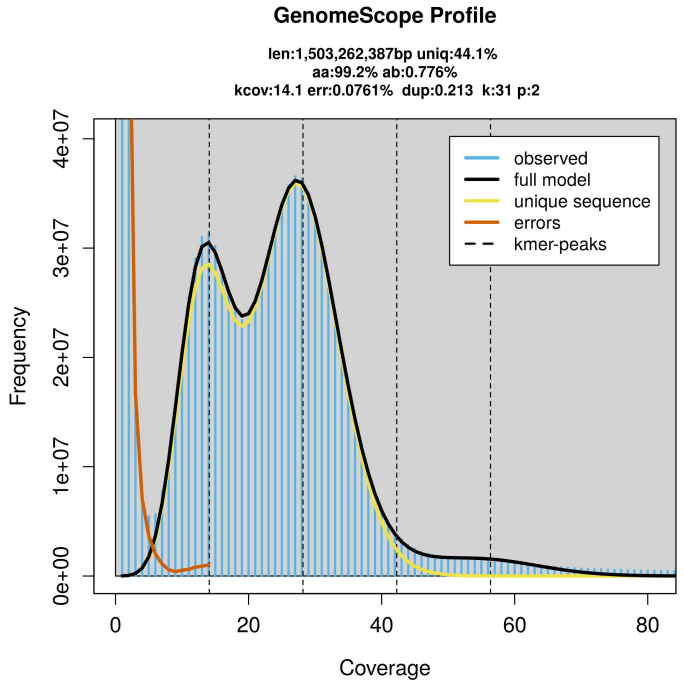
Chrysolina herbacea

uli



Chrysolina graminis

li

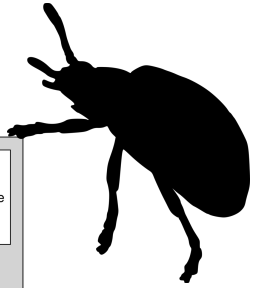


Example by Ksenia Krasheninnikova

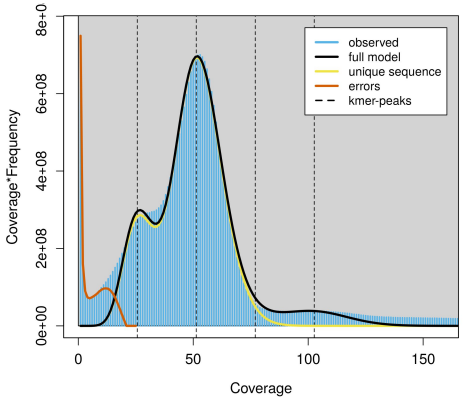
Chrysolina herbacea

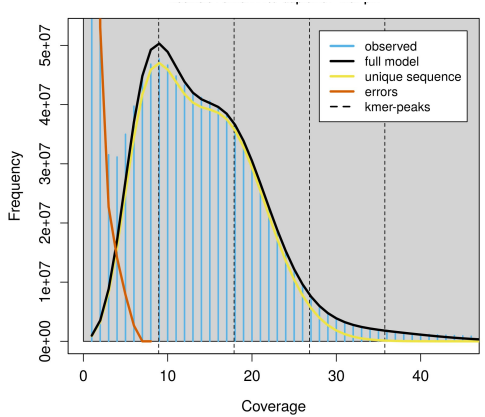
Chrysolina graminis

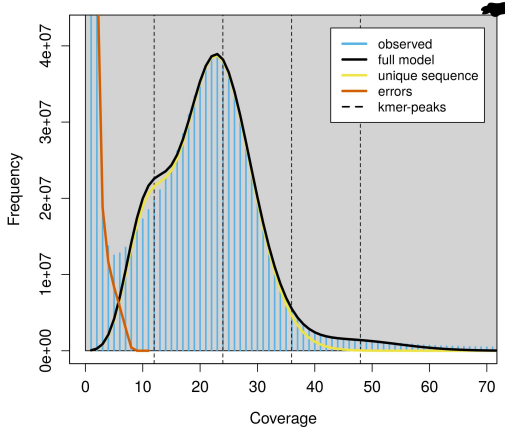
Chrysolina oricalcia



uli

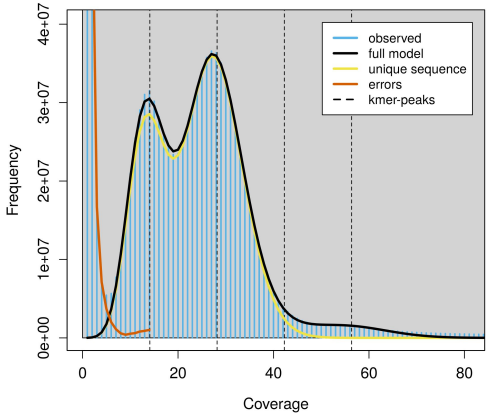


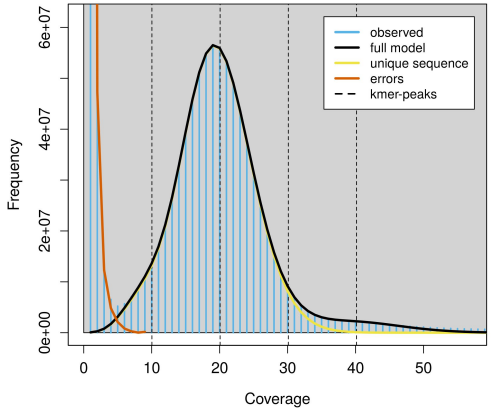




All bridges

li





Ok

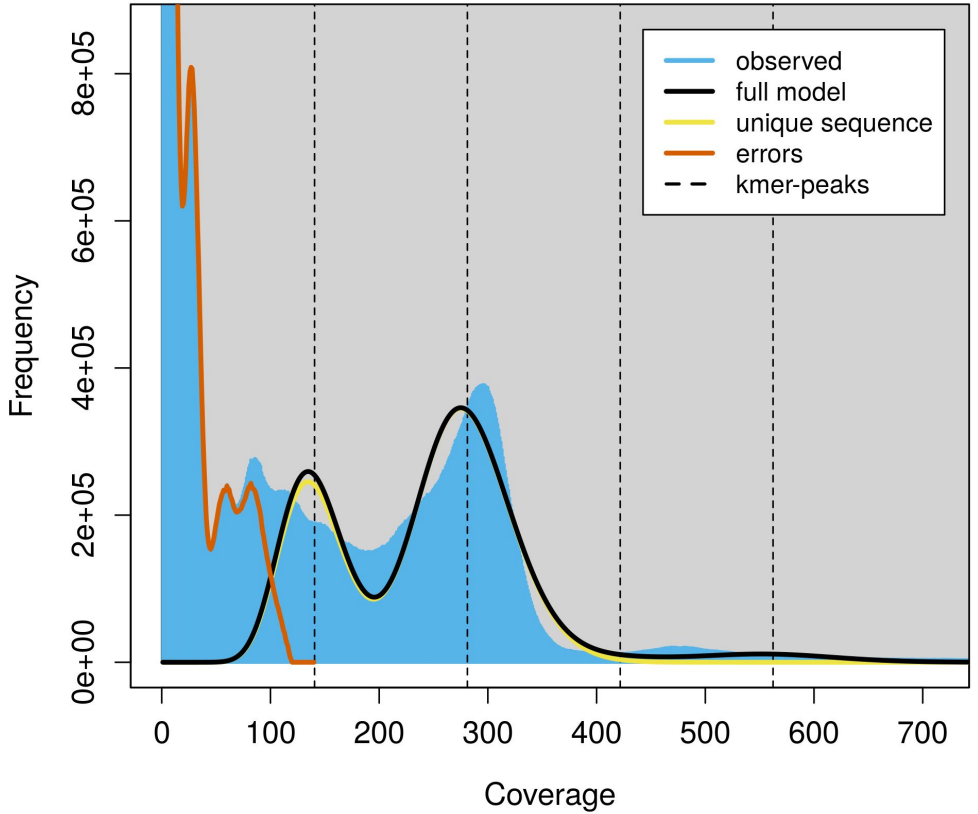
Example by Ksenia Krasheninnikova

“Pooled” sequencing also make bridges

Stylops aterrimus

- One clear genomic peak
- Lots of minor peaks that do not see to have clear stochasticity to the main peak

> looks like pool of genotypes, possibly just because the sequenced female was mated?



Running GenomeScope on a server

http://genomescope.org/genomescope2.0/

GenomeScope 2.0

Estimate genome heterozygosity, repeat content, and size from sequencing reads using a kmer-based statistical approach.

Run GenomeScope

Click or drop .histo file here to upload

Description	my sample
K-mer length	21
Ploidy	2
Max k-mer coverage	-1
Average k-mer coverage for polyploid genome	-1

Submit

Instructions

Upload results from running Jellyfish or KMC. Example: inputk21.hist

Instructions for running Jellyfish:

1. Download and install jellyfish from: http://www.genome.umd.edu/jellyfish.html#Release
2. Count k-mers using jellyfish:

```
$ jellyfish count -C -m 21 -s 1000000000 -t 10 *.fastq -o reads.jf
```

Note you should adjust the memory (-s) and threads (-t) parameters according to your server. This example will use 10 threads and 1GB of RAM. The k-mer length (-m) may need to be scaled if you have low coverage or a high error rate. You should always use "canonical k-mers" (-C).

3. Export the k-mer count histogram

```
$ jellyfish histo -t 10 reads.jf > reads.histo
```

Again the thread count (-t) should be scaled according to your server.

4. Upload reads.histo to GenomeScope

Instructions for running KMC:

1. Download and install KMC from: http://sun.aei.polsl.pl/REFRESH/index.php?page=projects&project=kmc&subpage=download
2. Count k-mers using KMC:

```
$ mkdir tmp
```

```
$ ls *.fastq > FILES
```

```
$ kmc -k21 -t10 -m64 -ci1 -cs10000 @FILES reads tmp/
```

Note you should adjust the memory (-m) and threads (-t) parameters according to your server. This example will use 10 threads and 64GB of RAM. The k-mer length (-k) may need to be scaled if you have low coverage or a high error rate. The lower (-ci) and upper (-cs) bounds exclude k-mers with counts outside these boundaries. FILES is a file with a list of input files.

3. Export the k-mer count histogram

k-mer histogram

1	1264991662
2	128715689
3	35706231
4	14690890
5	7658823
6	4750866
7	3423518
8	2858068
9	2737239
10	2944294
11	3473705
12	4288452
13	5379104
14	6746789
15	8374781
16	10152558
17	11869712
18	13391733
19	14663647
20	15457083
21	15833391
22	15731512
23	15224706
24	14487391
25	13626577

Mike Schatz

Running GenomeScope in terminal

```
[kj11@mib117963s:~/Projects/kmer_stuff/genomescope2.0$ genomescope.R --help
usage: /Users/kj11/bin/genomescope.R [-h] [-v] [-i INPUT] [-o OUTPUT]
                                     [-p PLOIDY] [-k KMER_LENGTH]
                                     [-n NAME_PREFIX] [-l LAMBDA]
                                     [-m MAX_KMERCOV] [--verbose]
                                     [--no_unique_sequence] [-t TOPOLOGY]
                                     [--initial_repetitiveness INITIAL_REPETITIVENESS]
                                     [--initial_heterozygosities INITIAL_HETEROZYGOSITIES]
                                     [--transform_exp TRANSFORM_EXP]
                                     [--testing] [--true_params TRUE_PARAMS]
                                     [--trace_flag] [--num_rounds NUM_ROUNDS]
                                     [--fitted_hist]
                                     [--start_shift START_SHIFT]
                                     [--typical_error TYPICAL_ERROR]
```

https://github.com/tbenavi1/genomescope2.0

The workflow for this morning

FastK

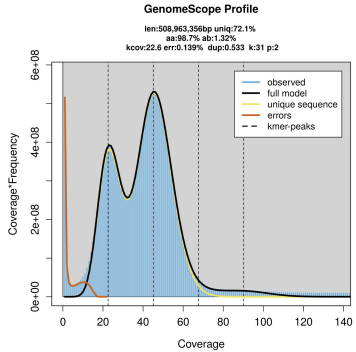
creates a *k*-mer database and

kmer	coverage
AAAAAAAAAACAACGT	28
AAAAATAACACAACGT	31
AAAAATAACACAACGG	1
AAAAATAACGCAACGT	47
AAAATTTACGCAACGT	2
AAAATTTACGCAACGA	17
...	...

k-mer histogram

1	1264991662
2	128715689
3	35706231
4	14690890
5	7658823
6	4750866
7	3423518
8	2858068
9	2737239
10	2944294
11	3473705
12	4288452
13	5379104
14	6746789
15	8374781
16	10152558
17	11869712
18	13391733
19	14663647
20	15457083
21	15833391
22	15731512
23	15224706
24	14487391
25	13626577

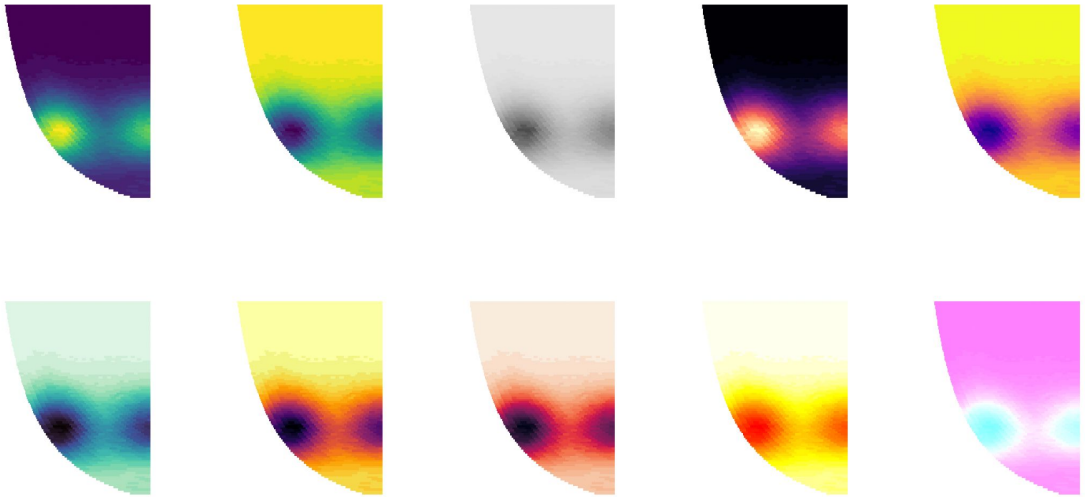
GenomeScope



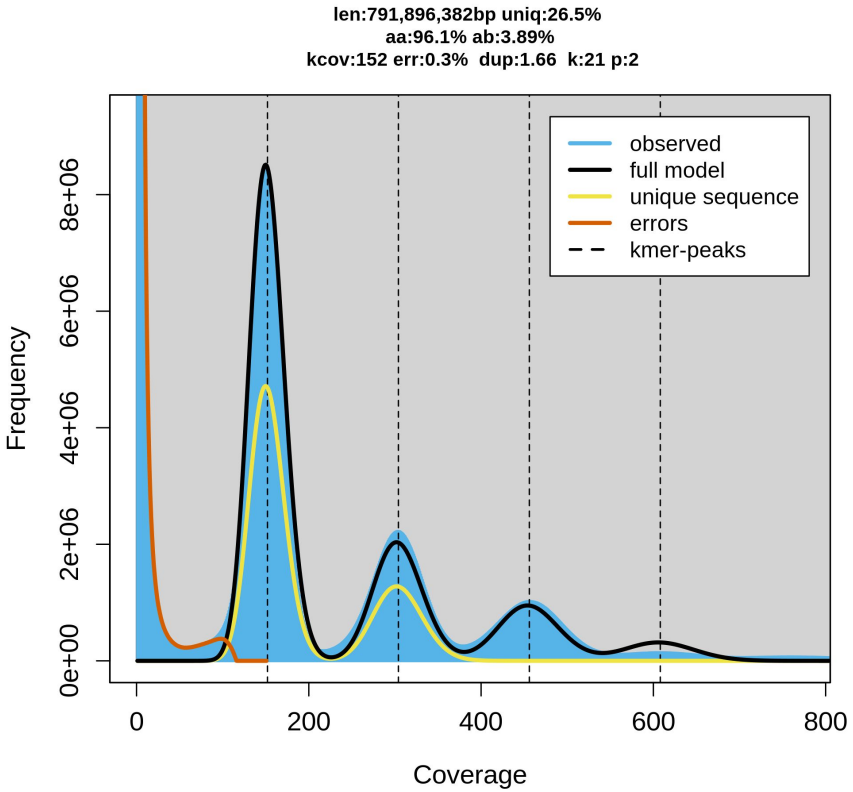
Your turn…

- Build k-mer spectrum
- Fit example GenomeScope model
- Fit many more
  - discuss what do you see / think !!!

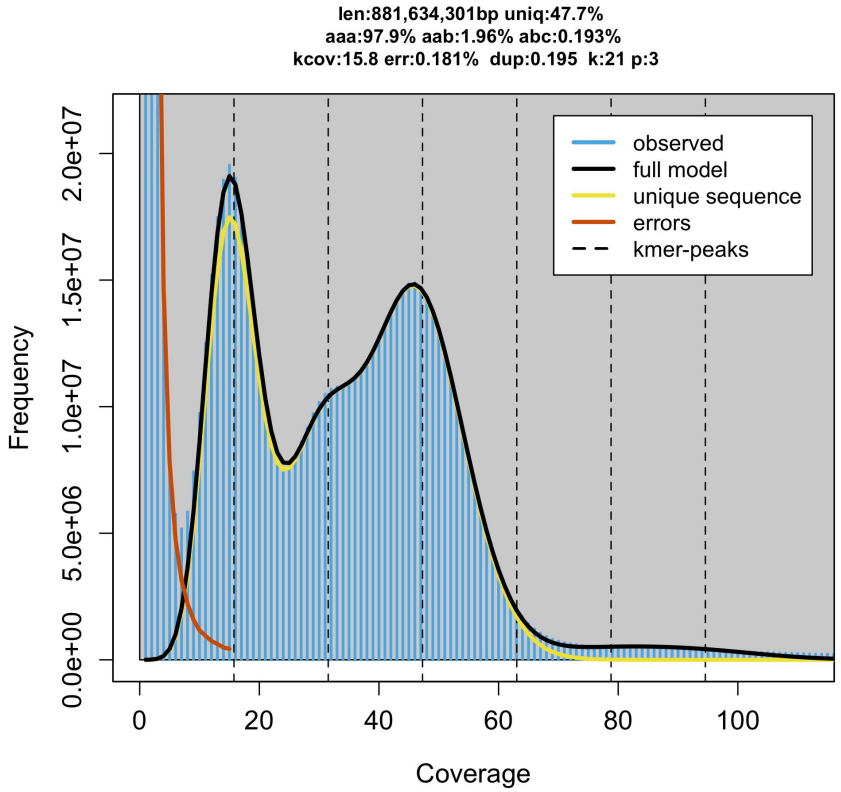
Part 3: Genome profiling of polyploids



Spectra with less obvious interpretation



Posted by João G. Ferreira to smudgeplot Issues



The flea beetle

Find all “hetmers” - **unique** k-mers **distant one nt**

AACTCA
AACTAA — Yaay, a pair

CCCTTA — Not a pair

GGCTCA
GGCTAA
GGCTTA — Not unique

b

Structure	CovA + CovB	$\frac{\text{CovB}}{\text{CovA + CovB}}$
AB	2λ	0.5
AAB	3λ	0.333
AAAB	4λ	0.25
AABB	4λ	0.5
AAAAB	5λ	0.2
AAABB	5λ	0.4
AAAAAB	6λ	0.166
AAAABB	6λ	0.333
AAABBB	6λ	0.5
...	...	...

Find all “hetmers” - **unique** k-mers **distant one nt**

```
smudgeplot hetmers <FastK_kmer_db>
```

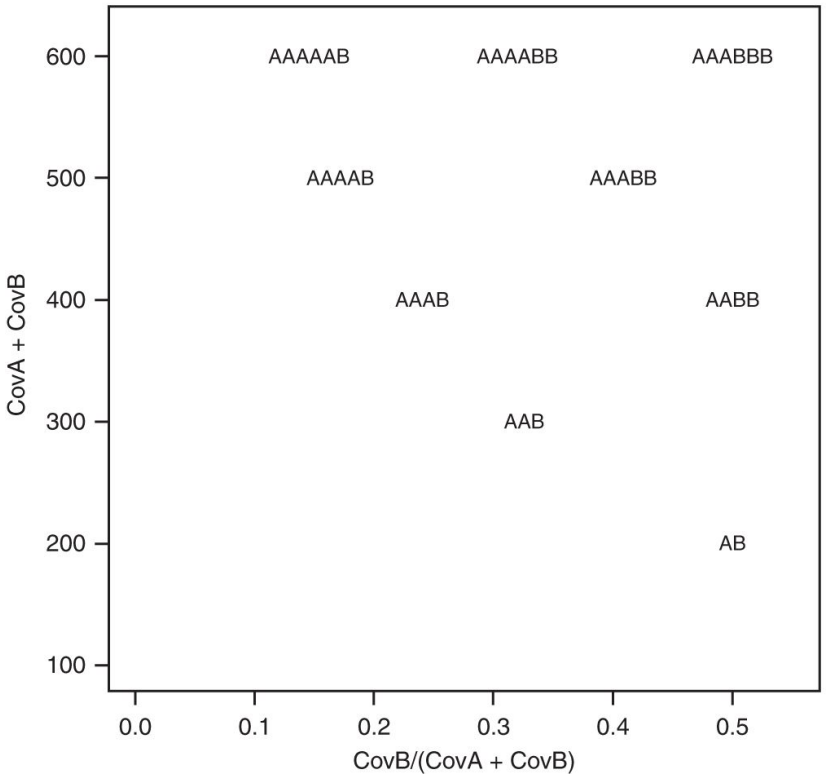
Fast implementation by Gene Myers

```
kj11@ib117963s:~/Projects/kmer_stuff$ head ./smudgeplot/data/Fara2/kmerpairs_text.smu
26      26      6560
26      27      12888
26      28      12584
27      27      6512
26      29      12532
27      28      12764
26      30      11990
27      29      12624
28      28      6110
26      31      11110
```

covB covA freq

Plotting the two creates “smudgeplot”

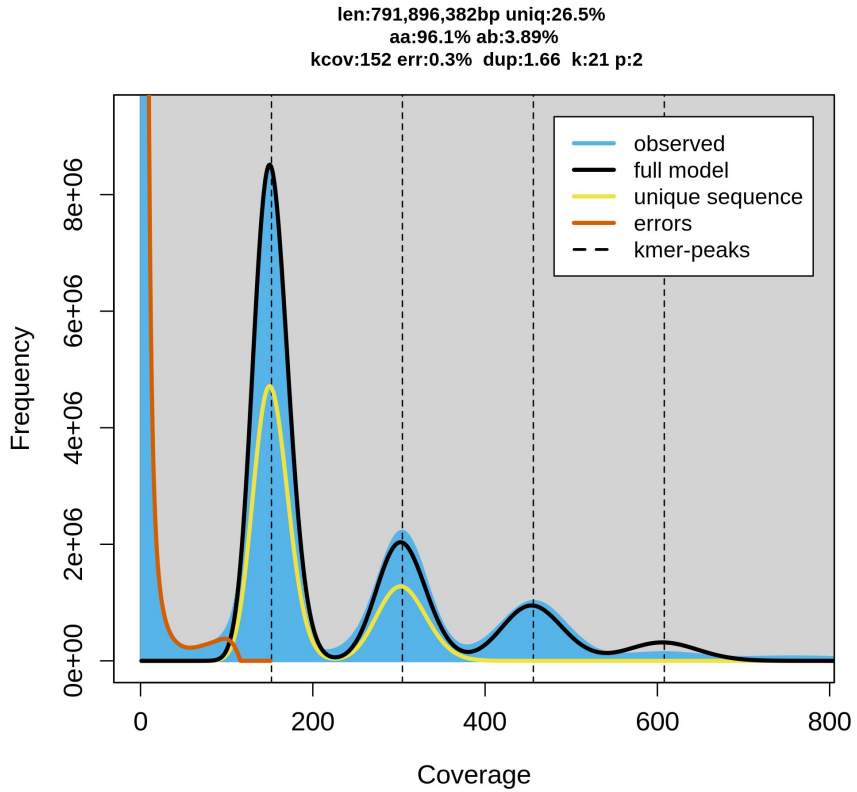
a

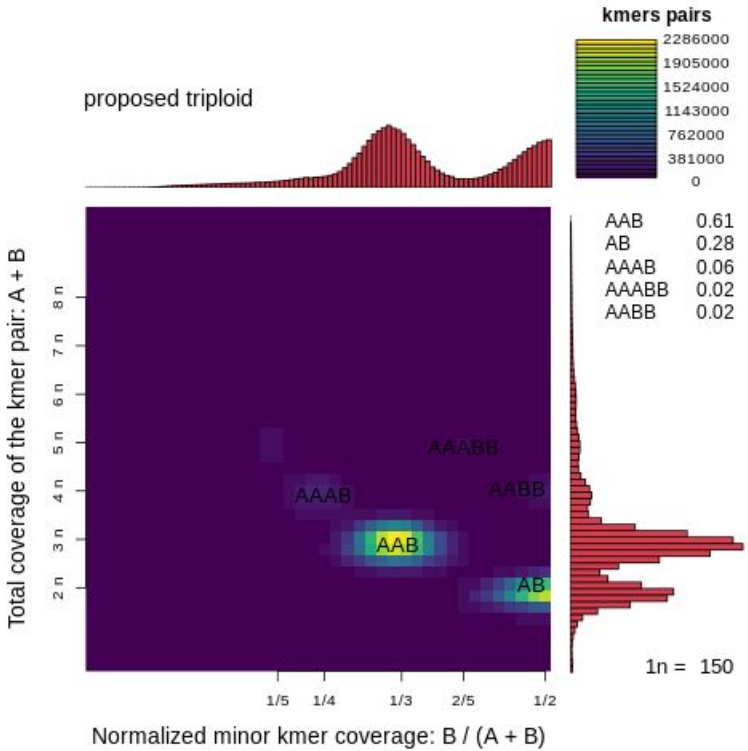


b

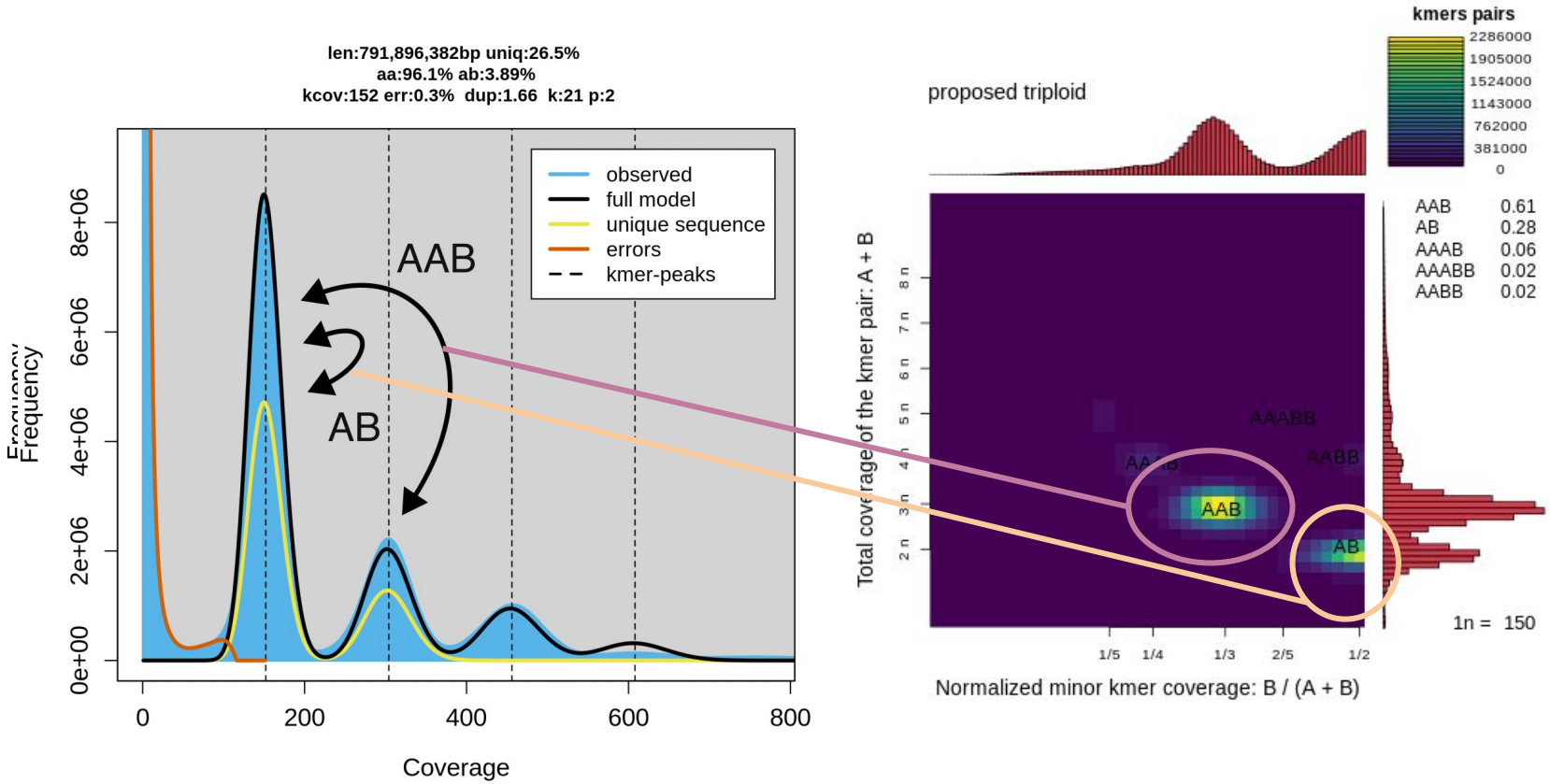
Structure	CovA + CovB	$\frac{\text{CovB}}{\text{CovA + CovB}}$
AB	2λ	0.5
AAB	3λ	0.333
AAAB	4λ	0.25
AABB	4λ	0.5
AAAAB	5λ	0.2
AAABB	5λ	0.4
AAAAAB	6λ	0.166
AAAABB	6λ	0.333
AAABBB	6λ	0.5
...	...	...

Smudgeplot indicates triploidy



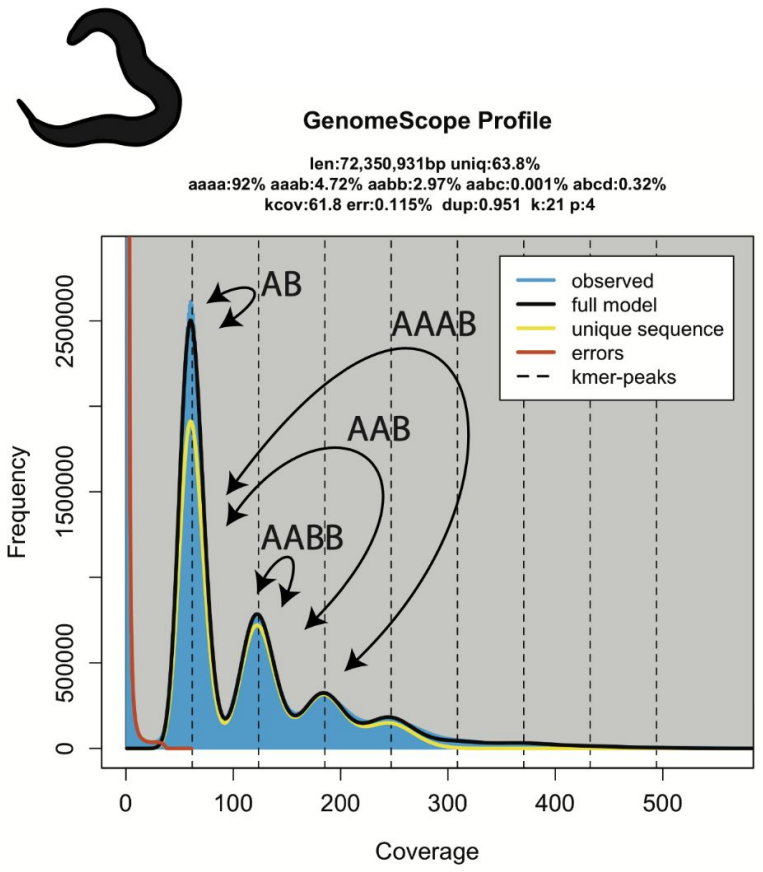


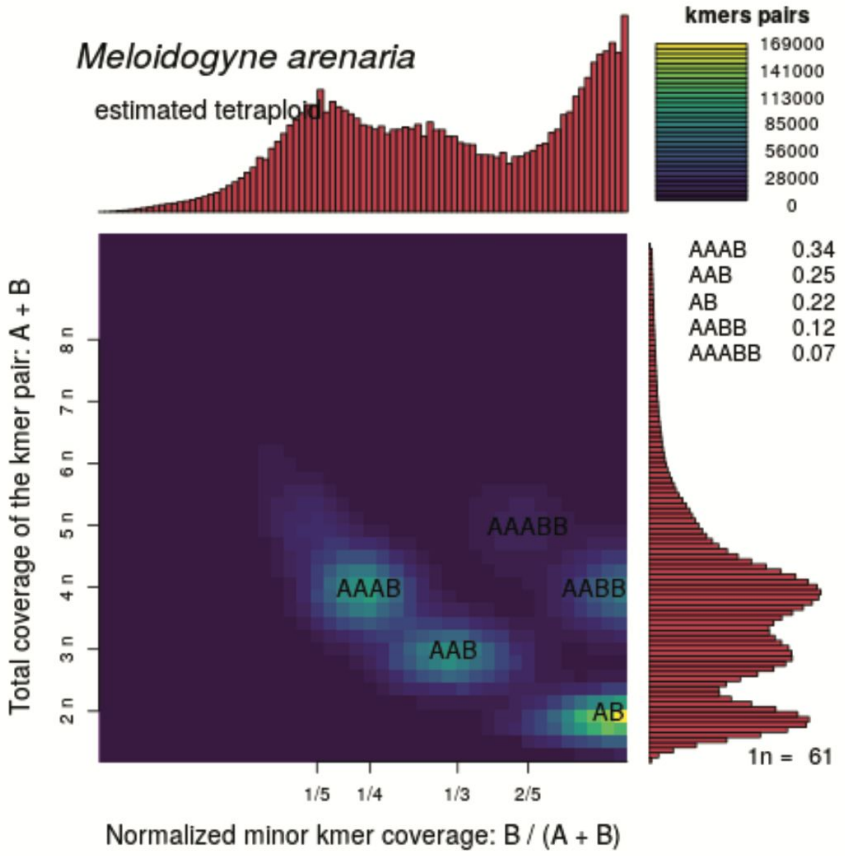
Joint interpretation of smudgeplot and genomescope



Posted by João G. Ferreira to smudgeplot Issues

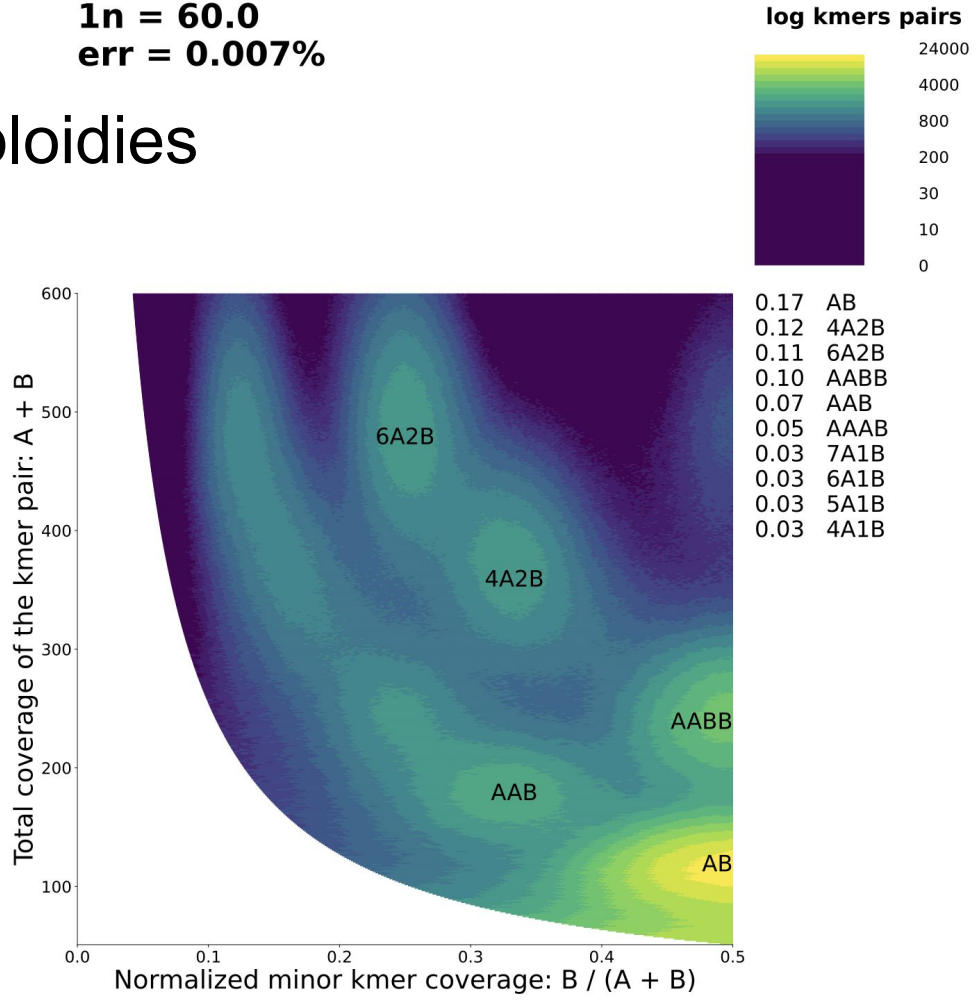
Works for tetraploids too!



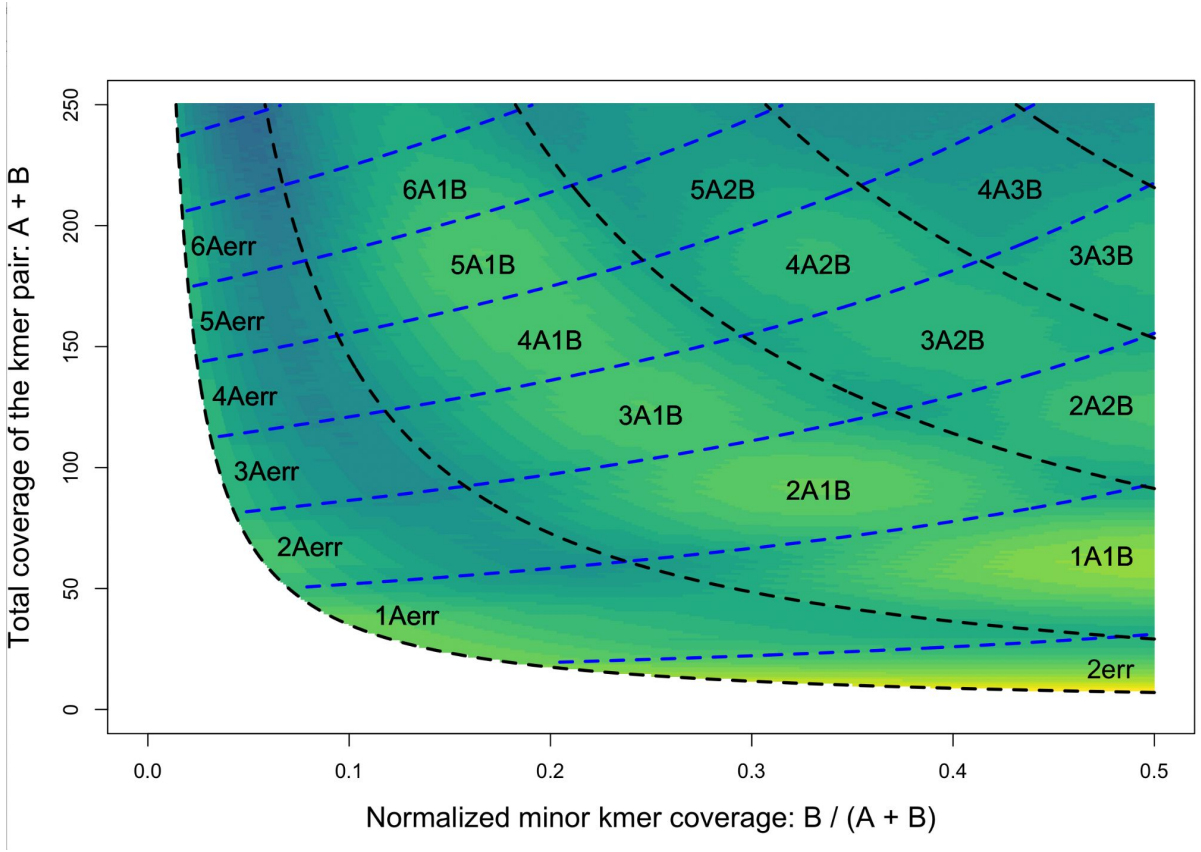


Smudgeplot works for high ploidies

octoploid strawberry

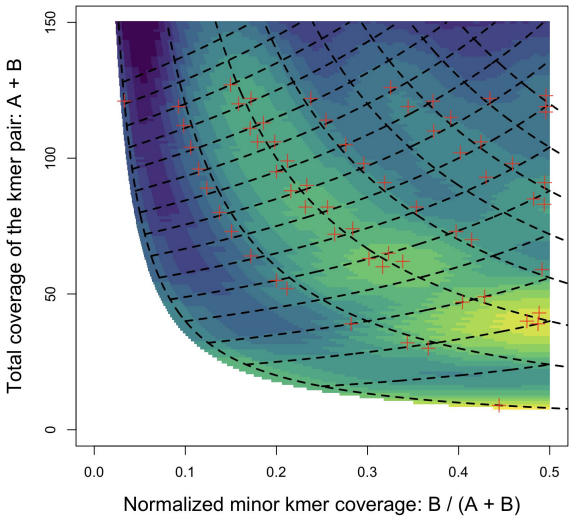


Coverage inference by “fishnet”



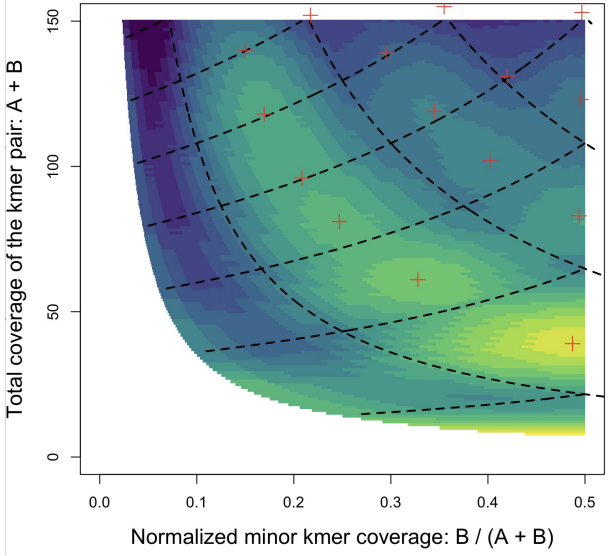
We can test each coverage

cov = 8



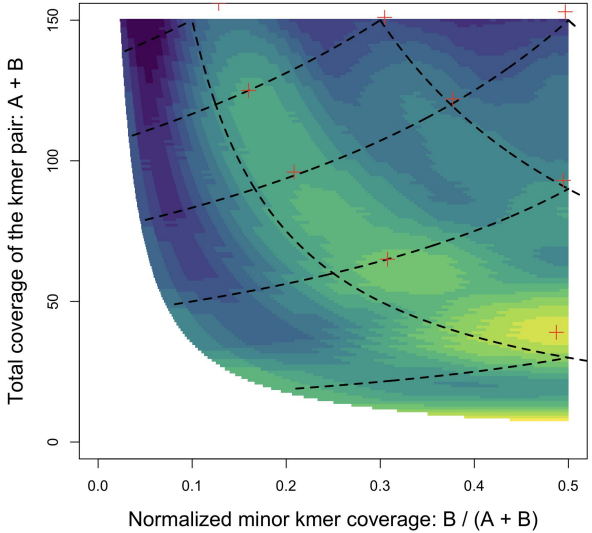
centrality: 0.8

cov = 21.2



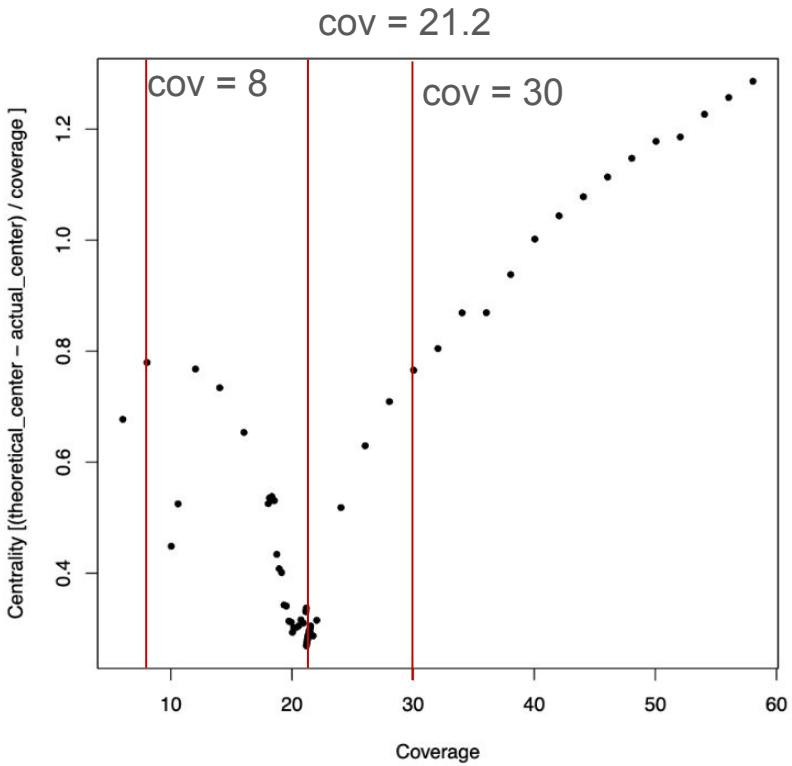
centrality: 0.31

cov = 30



centrality: 0.78

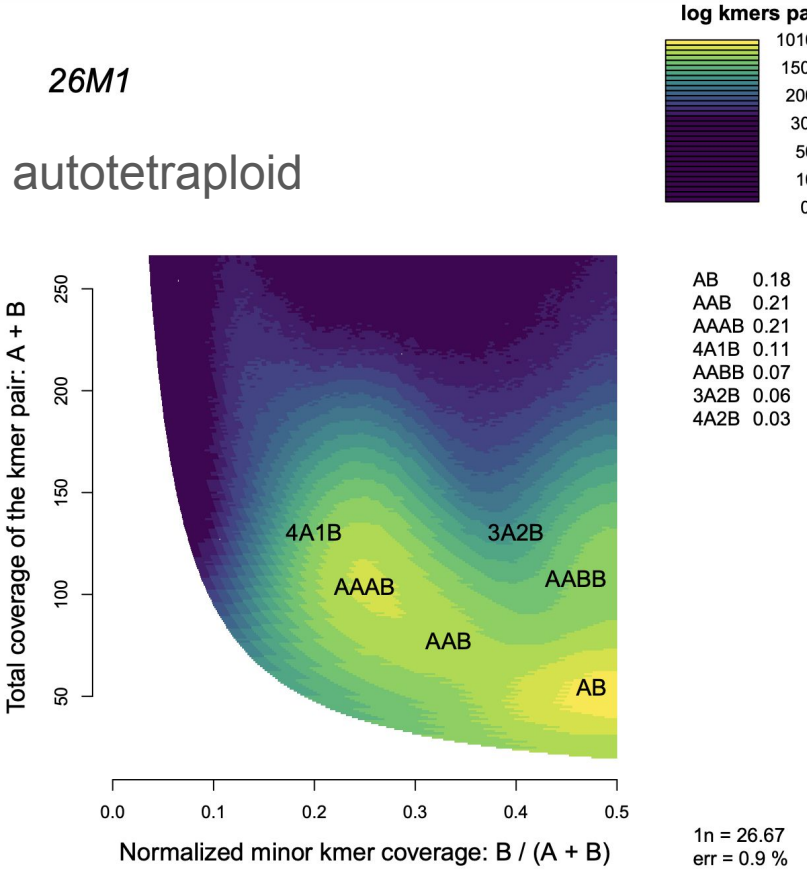
For each coverage, “cast the fishnet” and calculate “centrality”



willow

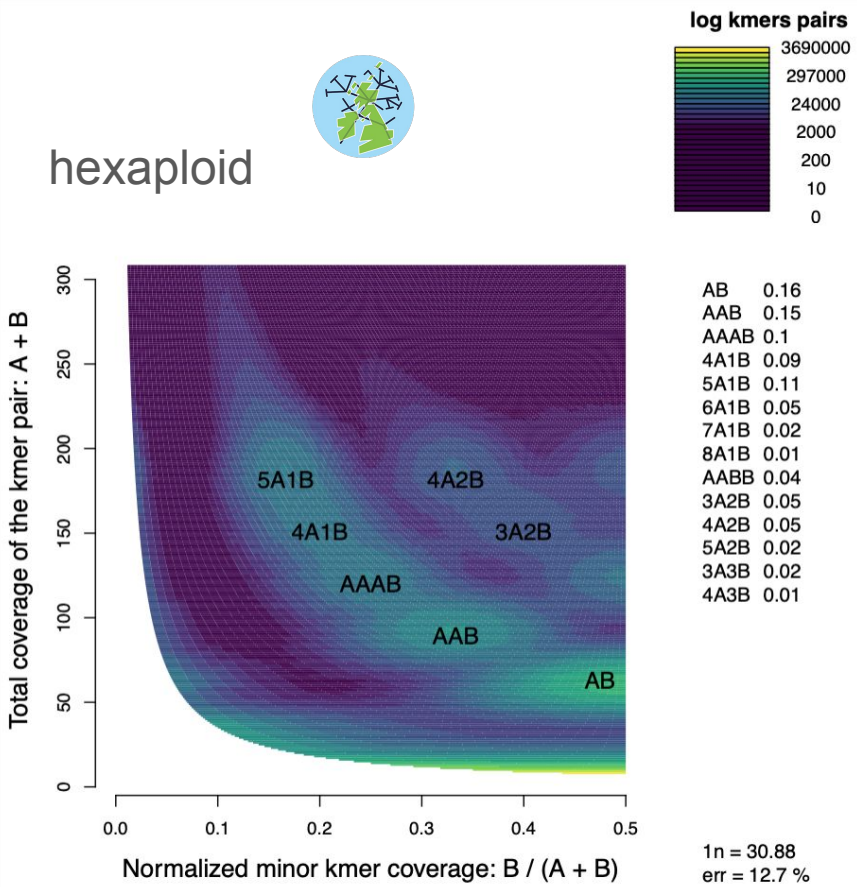
26M1

autotetraploid



sea wormwood

hexaploid

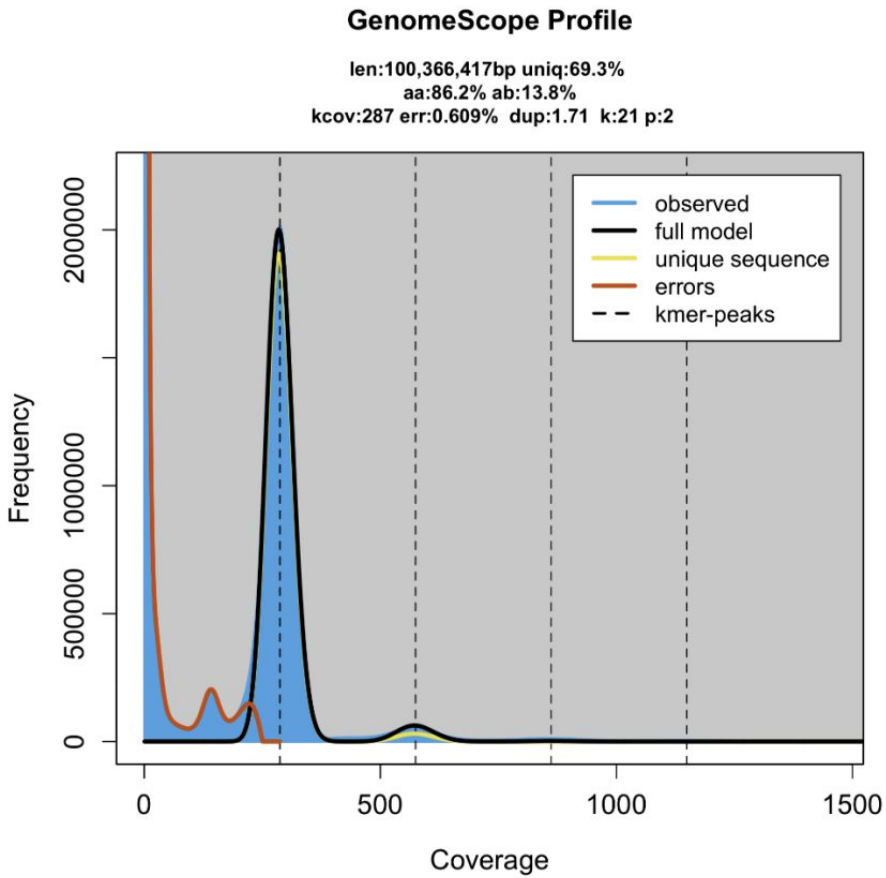


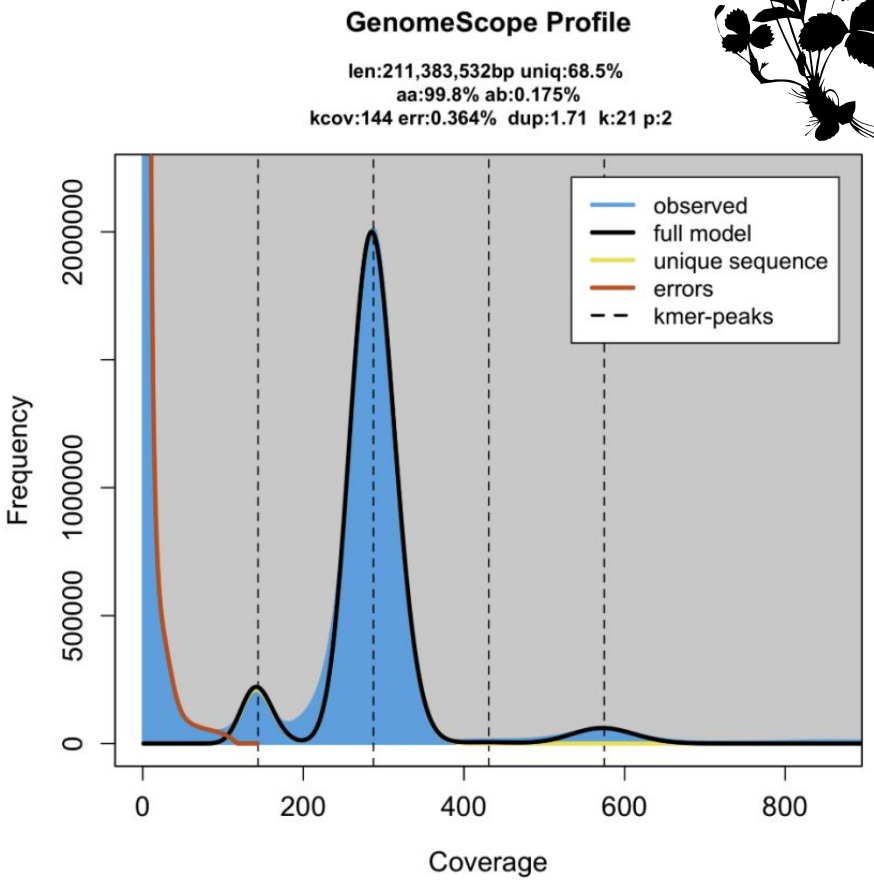
Smudgeplot by Vinciane Mossion

What to pay attention when genome profiling

- Find all possible “genome models”
  - Combinations of genomic features that would generate patterns observed by GenomeScope and Smudgeplot
- Make sure your model fits the data
- Make sure that all evidence makes sense together
  - E.g. Is the 1n coverage in smudgeplot and genomescope consistent?
- Make sure that your interpretation is somewhat compatible with known biology of the species
- Confront your model to the genome assembly
  - Est genome size vs assembled
  - Is the assembly with collapsed or uncollapsed haplotypes? Would you expect it given estimated heterozygosity?
  - Is the assembly ploidy consistent with smudgeplot?

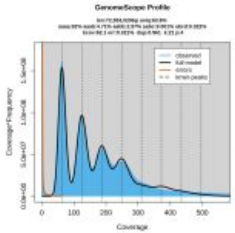
Convergence problems - strawberry

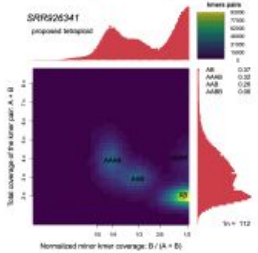




The whole genome profiling pipeline

ENA
European Nucleotide Archive





Gitpod

software & data

FastK
-k31

kmer database

kmer histogram

smudgeplot hetmers
lower end coverage threshold
-L 40

kmer pairs

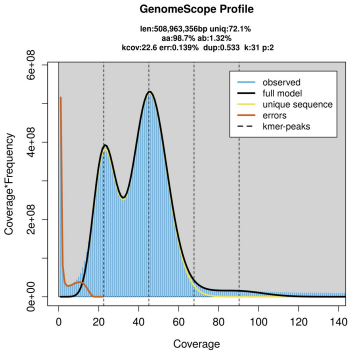
smudgeplot plot
expected coverage
-n 120

There is a place for all the freaks…

all data are available on ToLQC

https://tolqc.cog.sanger.ac.uk/

Shane McCarthy



GenomeScope plots

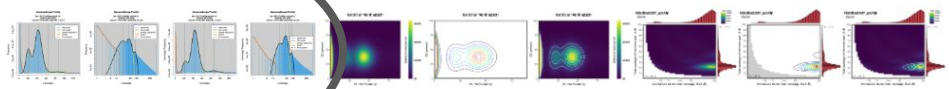
k-mer histogram

```
1       1264991662
2       128715689
3       35706231
4       14690890
5       7658823
6       4750866
7       3423518
8       2858068
9       2737239
10      2944294
11      3473705
12      4288452
13      5379104
14      6746789
15      8374781
16      10152558
17      11869712
18      13391733
19      14663647
20      15457083
21      15833391
22      15731512
23      15224706
24      14487391
25      13626577
```

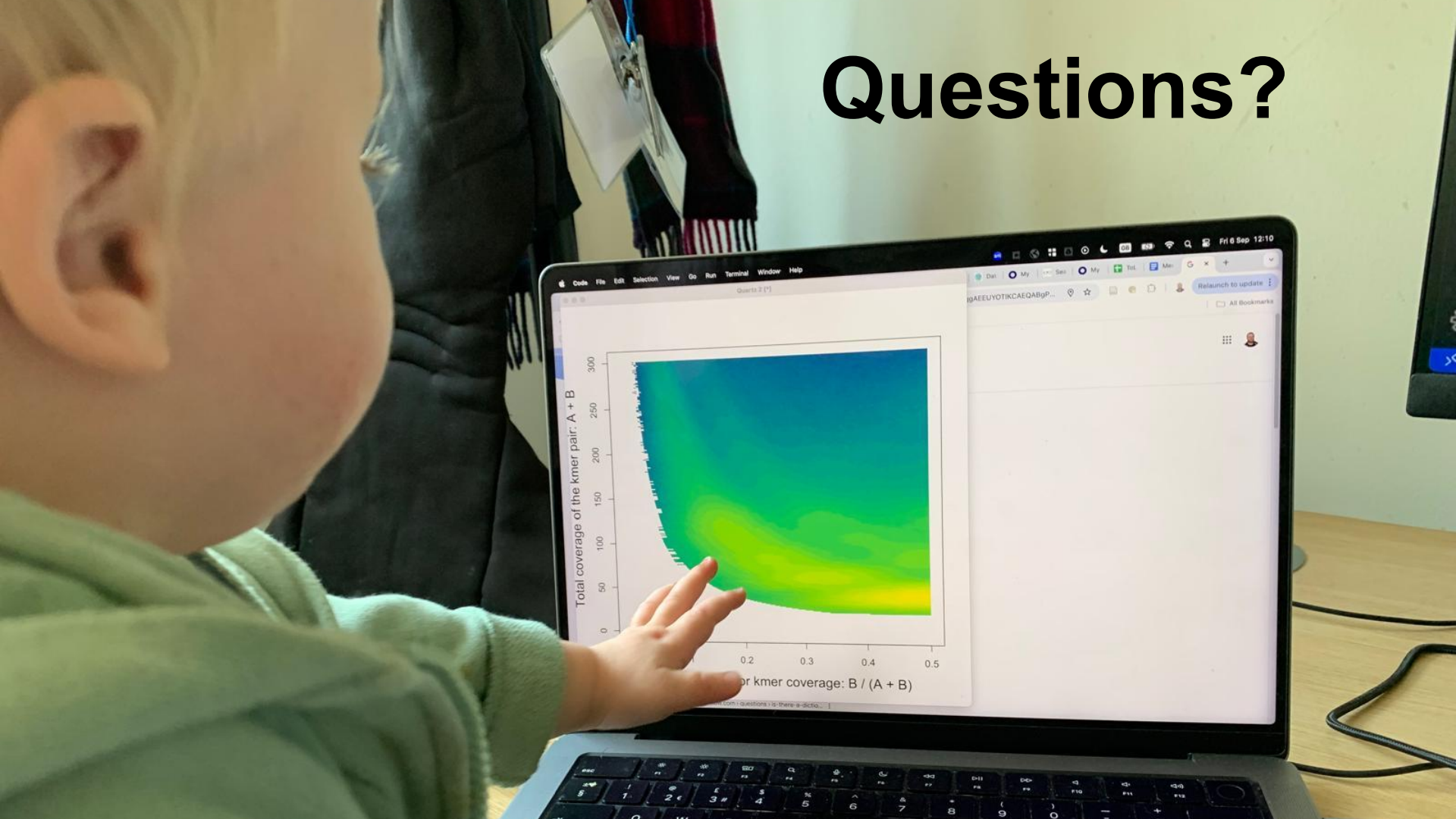
Search

show all plots | hide all plots

source	specimen	*k*-mer	*k*-cov	haploid size	repeat (%)	heterozygosity (%)	model fit (%)	model error (%)	histogram
pacbio	AbrBra2	31	21.75	1,060,723,998	35.95	0.57	97.39	0.13	histogram.txt



Questions?


Total coverage of the kmer pair: A + B
0
50
100
150
200
250
300
0.2
0.3
0.4
0.5
kmer coverage: B / (A + B)