Informed and automated *k*-mer size selection for genome assembly

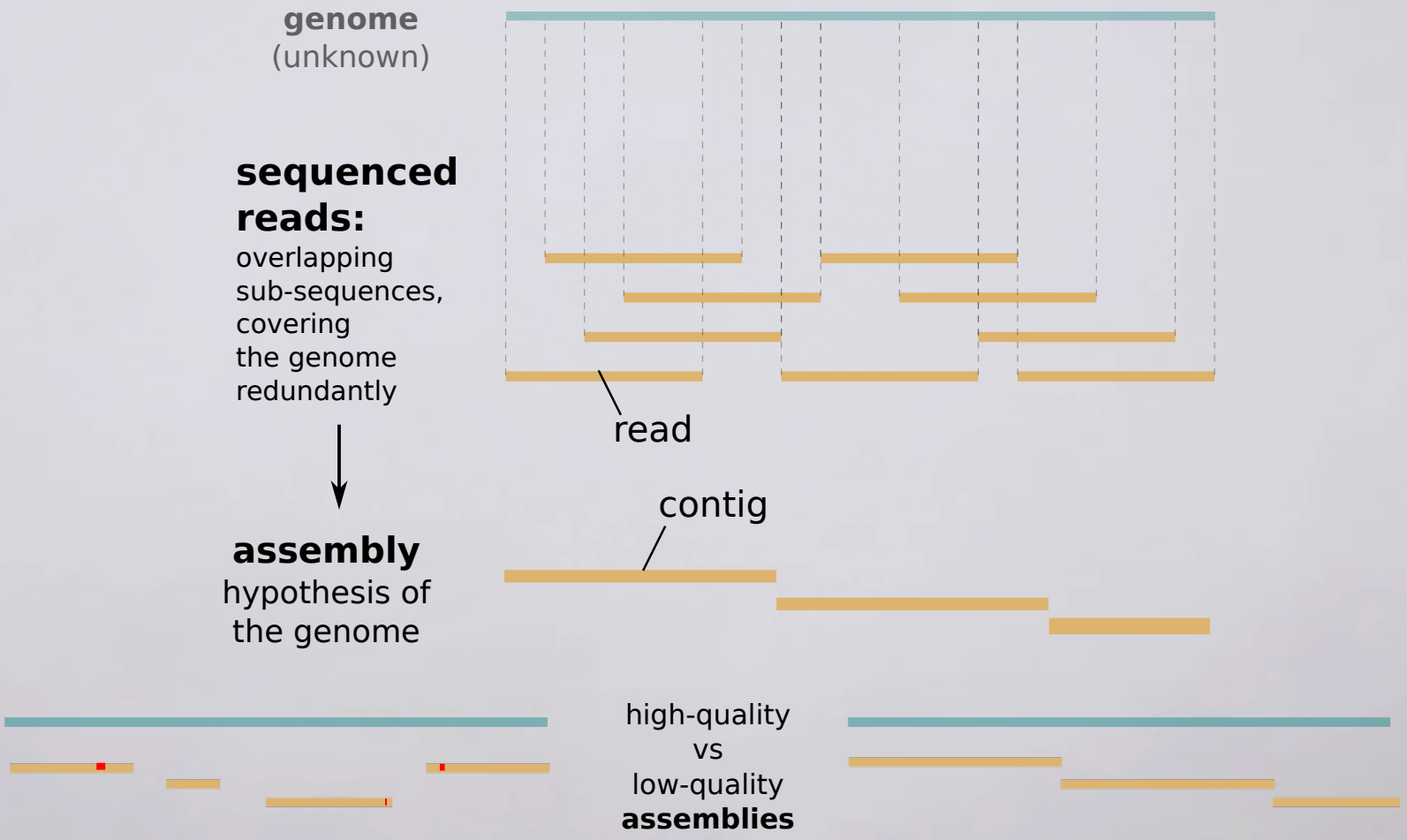
Rayan Chikhi, Paul Medvedev

Pennsylvania State University

HiTSeq - July 2013

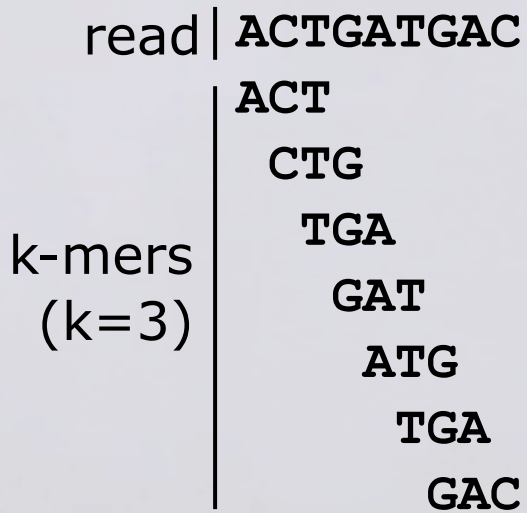
GENOME ASSEMBLY

Genome assembly is the technique used to reconstruct genome sequences from DNA sequencing.



Motivation

Bioinformaticians routinely run assemblers (Allpaths-LG, Soapdenovo2, Velvet, ...) to study novel organisms.
Most assemblers cut reads into *k*-**mers** (de Bruijn graph method).



Practical issue: assemblers rely on the user to set the parameter *k*.

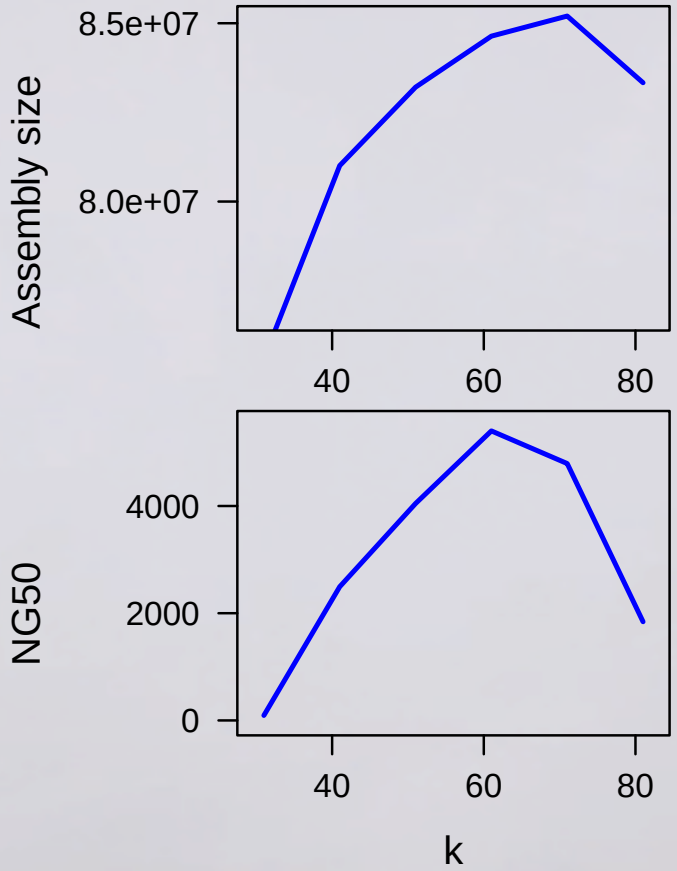
→ *What could go wrong if k is incorrectly set?*

MOTIVATION: OPTIMAL k NEEDED

Total length and contiguity (NG50) of chr. 14 (88 Mbp) assemblies

NG50: maximum ℓ such that $(\sum_{|\text{contig}_i| \geq \ell} |\text{contig}_i|)$ larger than $|\text{genome}|/2$

Illumina 100bp paired-end 70x coverage, assembled by Velvet with several values of k



Fact: Genome assembly is **not robust** with respect to k.

Our motivation: help bioinformaticians obtain the best possible assembly by **finding optimal k automatically**

EXISTING METHODS TO ESTIMATE BEST k

Velvetk: without looking at the data:

$$k_{optim} = argmin_k(|\frac{N_k}{G} - C|)$$

where:
N_k (total number of k-mers in the reads),
G (estimated genome size) and
C (desired target coverage).
Does not know about genome complexity and error rate.

VelvetOptimizer: for a specific assembler (Velvet). Brute-forces all values of k and examines N50.

$$k_{optim} = argmax_k(N50_k)$$

Takes in the order of CPU-years for mammalian genomes.

EXISTING METHODS TO ESTIMATE BEST k

Velvetk: without looking at the data:

$$k_{optim} = argmin_k(|\frac{N_k}{G} - C|)$$

where:
N_k (total number of k-mers in the reads),
G (estimated genome size) and
C (desired target coverage).
Does not know about genome complexity and error rate.

VelvetOptimizer: for a specific assembler (Velvet). Brute-forces all values of k and examines N50.

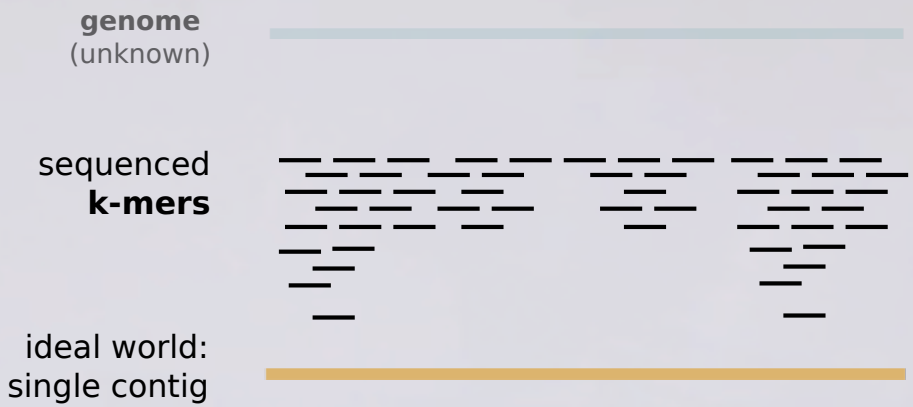
$$k_{optim} = argmax_k(N50_k)$$

Takes in the order of CPU-years for mammalian genomes.

Actually, most of the time:

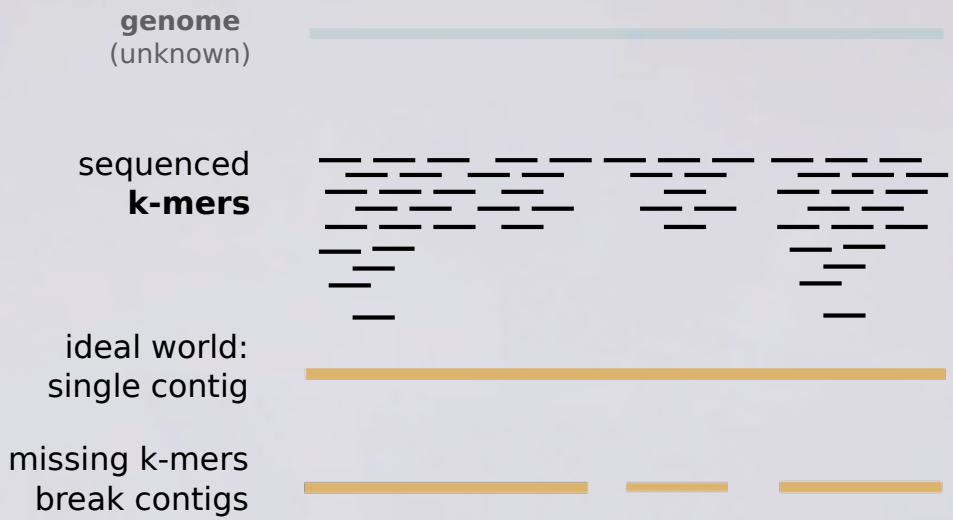
- Bioinformaticians run [assembler] many times with $k = 21, \ldots, 91$, or
- *"Our colleagues had good results with $k = 51$ on [some other bacterial dataset]"*.

HYPOTHESIS FOR THE OPTIMAL *k*



In DNA/RNA/metaDNA/metaRNA assembly:

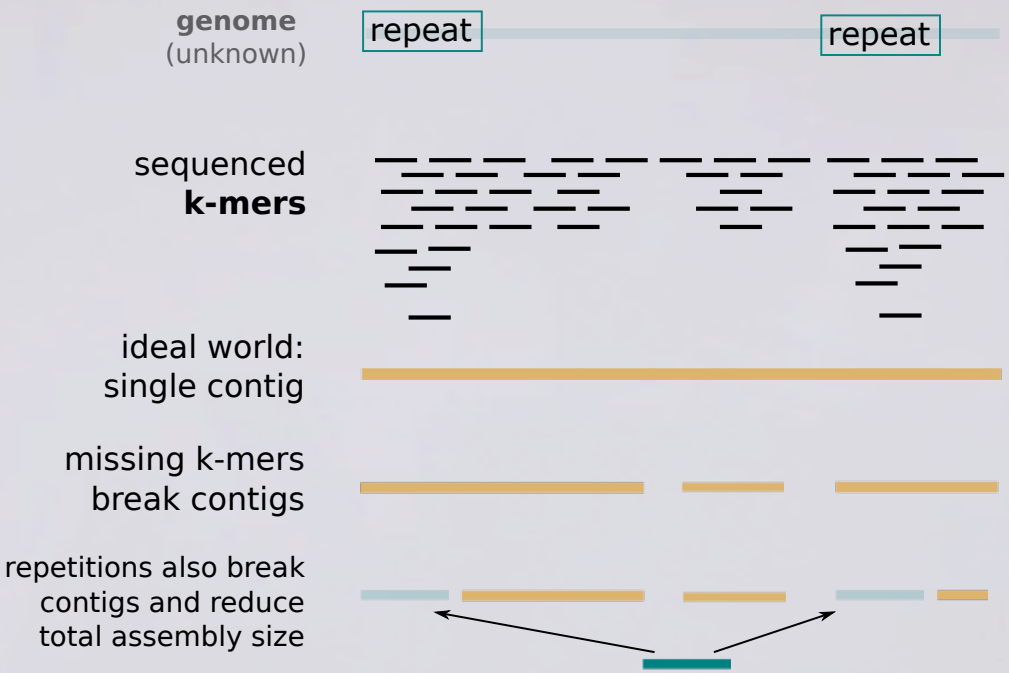
HYPOTHESIS FOR THE OPTIMAL k



In DNA/RNA/metaDNA/metaRNA assembly:

- small k: less chance of missing k-mers

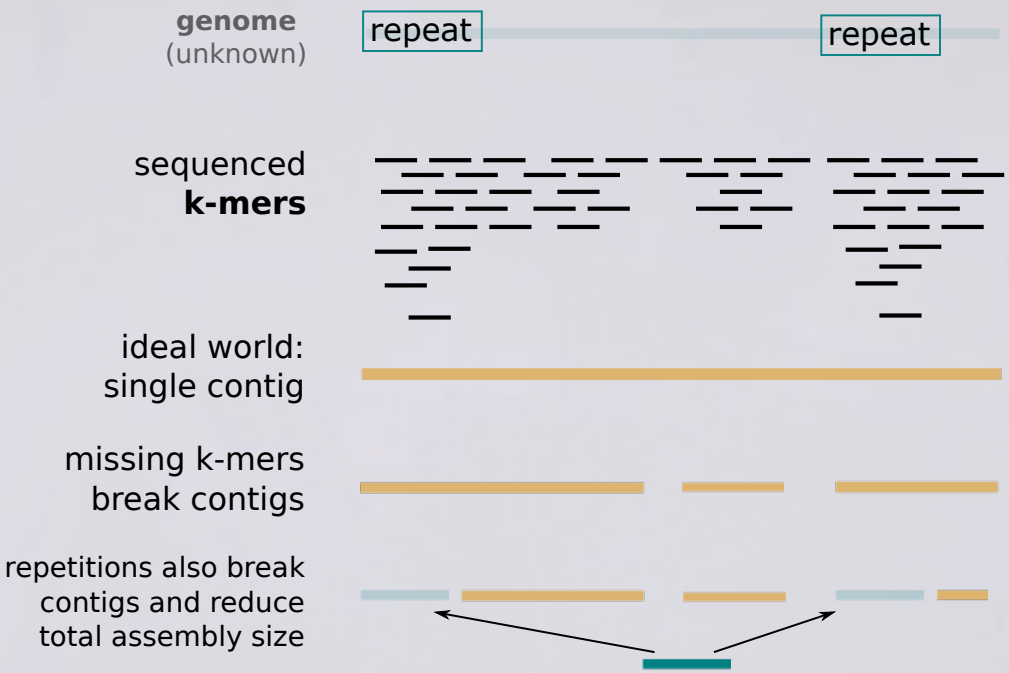
HYPOTHESIS FOR THE OPTIMAL k



In DNA/RNA/metaDNA/metaRNA assembly:

- small k: less chance of missing k-mers
- large k: less repetitions shorter than k

HYPOTHESIS FOR THE OPTIMAL k



In DNA/RNA/metaDNA/metaRNA assembly:

- small k: less chance of missing k-mers
- large k: less repetitions shorter than k
- Also, larger k-mers: more likely to contain errors (unusable k-mers)

Our hypothesis: use the largest k-mer size possible (to avoid repetitions), such that the genome is sufficiently covered by k-mers.

$\rightarrow$ So, when are sufficiently many (non-erroneous) k-mers seen?

k-MER HISTOGRAMS

Common practice: compute the k-mer abundance histogram.

- x axis: *abundance*
- y axis: number of k-mers having abundance x (seen x times)

Example reads dataset:

```
ACTCA
GTCA
```

3-mers:

```
ACT
CTC
TCA
GTC
TCA
```

Abundance of each distinct 3-mer:

```
ACT: 1
CTC: 1
TCA: 2
GTC: 1
```

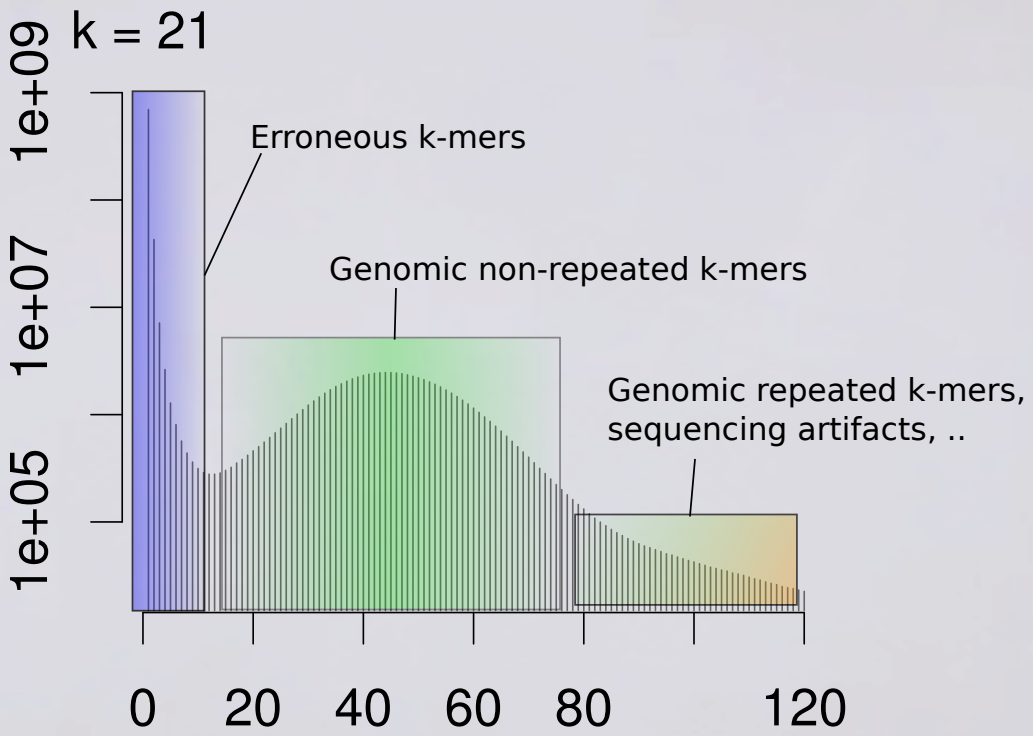
3-mer abundance:

x	y
1	3
2	1
3	0
4	0

For a dataset and a value of k, methods that build histograms already exist (k-mer counting, e.g. Jellyfish, DSK, . . .).

DISSECTION OF A k-MER HISTOGRAM

Chr 14 ($\approx$ 88 Mbp) GAGE dataset; histogram $k = 21$



Genomic area

$\approx$

number of distinct k-mers covering the genome

$\approx$

size of the assembly

$\rightarrow$ *How to determine exactly this area?*

HISTOGRAM MODEL

We use Quake's model: [DR Kelley 2010]

Erroneous *k*-mers Pareto distribution with shape α,

$$pdf = \frac{\alpha}{x^{\alpha+1}}$$

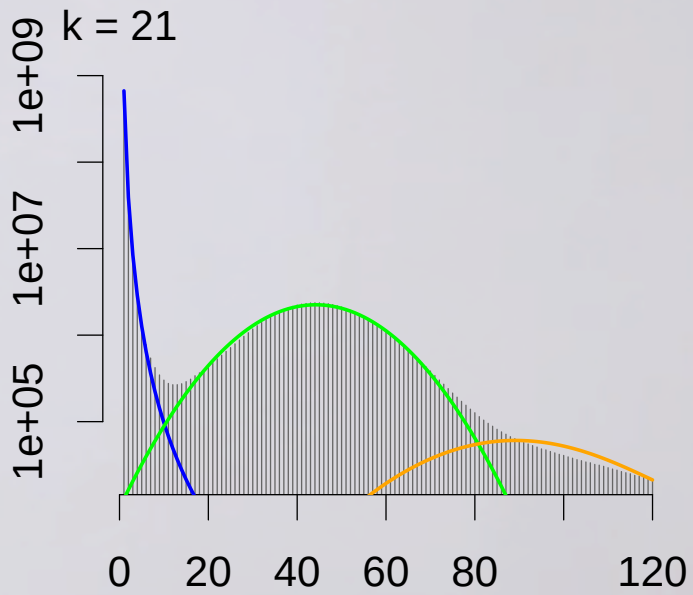
Genomic *k*-mers Mixture of n Gaussians, weighted by a Zeta distribution of shape s:

$$w_1 X_1 + \ldots + w_n X_n$$

$$X_i \sim \mathcal{N}(i\mu_1, (i\sigma_1)^2)$$

$$P(w_i = k) = k^{-s}/\zeta(s)$$

Full model Mixture weighted by $(p_e, 1 - p_e)$.

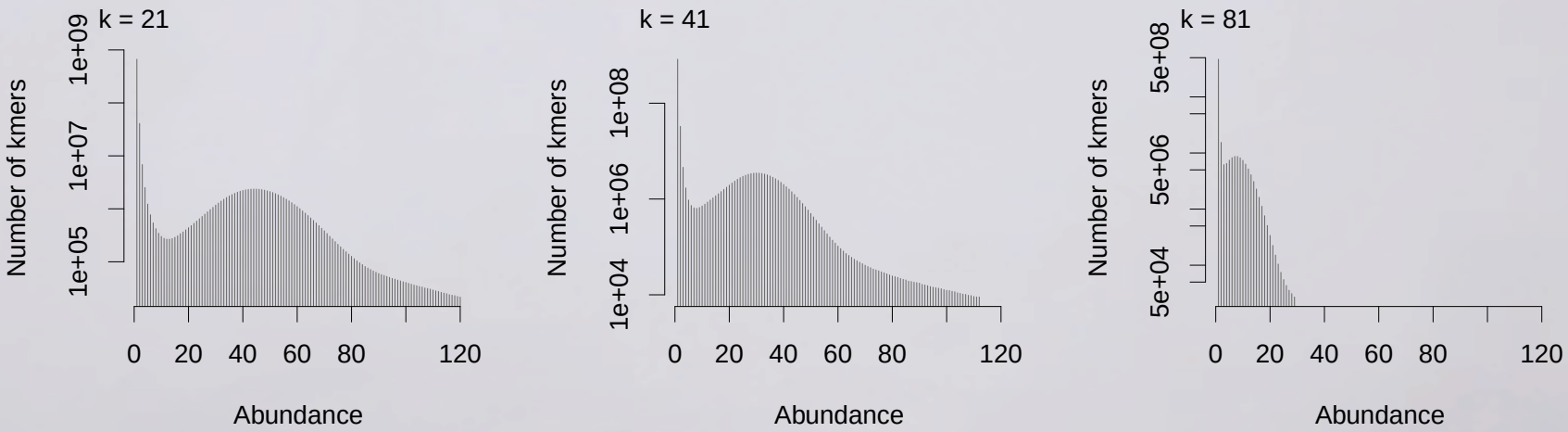


Numerical optimization (R) is used to fit the model to actual histograms.

SEEN SO FAR

- Genome is sufficiently covered by k-mers $\Longrightarrow$ good k value
- Requires to know the number of genomic k-mers
- Can be estimated with a k-mer histogram and the Quake model

To find the optimal k, one can compare histograms for different values of k.



Chr 14 ($\approx$ 88 Mbp) GAGE dataset; histograms for three values of k

$\rightarrow$ Issue: computing a single histogram (using k-mer counting) is time and memory expensive

SAMPLING HISTOGRAMS

Computing exact k-mer histograms is expensive (= k-mer counting).

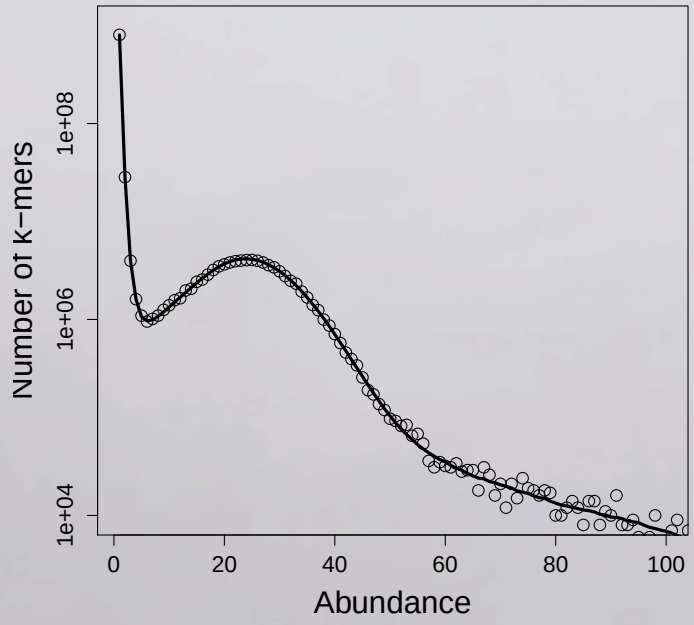
Organism	CPU time per k value
	DSK
S. aureus	2min
chr14	48min
B. impatiens	7.5hour

SAMPLING HISTOGRAMS

Computing exact k-mer histograms is expensive (= k-mer counting).

Organism	CPU time per k value		Memory usage of Sampling method (GB)
	DSK	Sampling method	
S. aureus	2min	11sec	0.1
chr14	48min	7min	0.1
B. impatiens	7.5hour	1.2hour	0.4

We developed a fast and memory-efficient histogram sampling technique. Sample 1 k-mer out of r, in k-mer space (the same k-mer seen in two different reads will be either consistently sampled, either consistently ignored)



- Chr 14 ($\approx$ 88 Mbp) $k = 41$
- continuous line $=$ exact histogram
- dots $=$ sampled histogram
- sampling errors are visible for low number of k-mers (log scale)

TOOLS, DATASETS

Software: `KmerGenie` (`http://kmergenie.bx.psu.edu`)

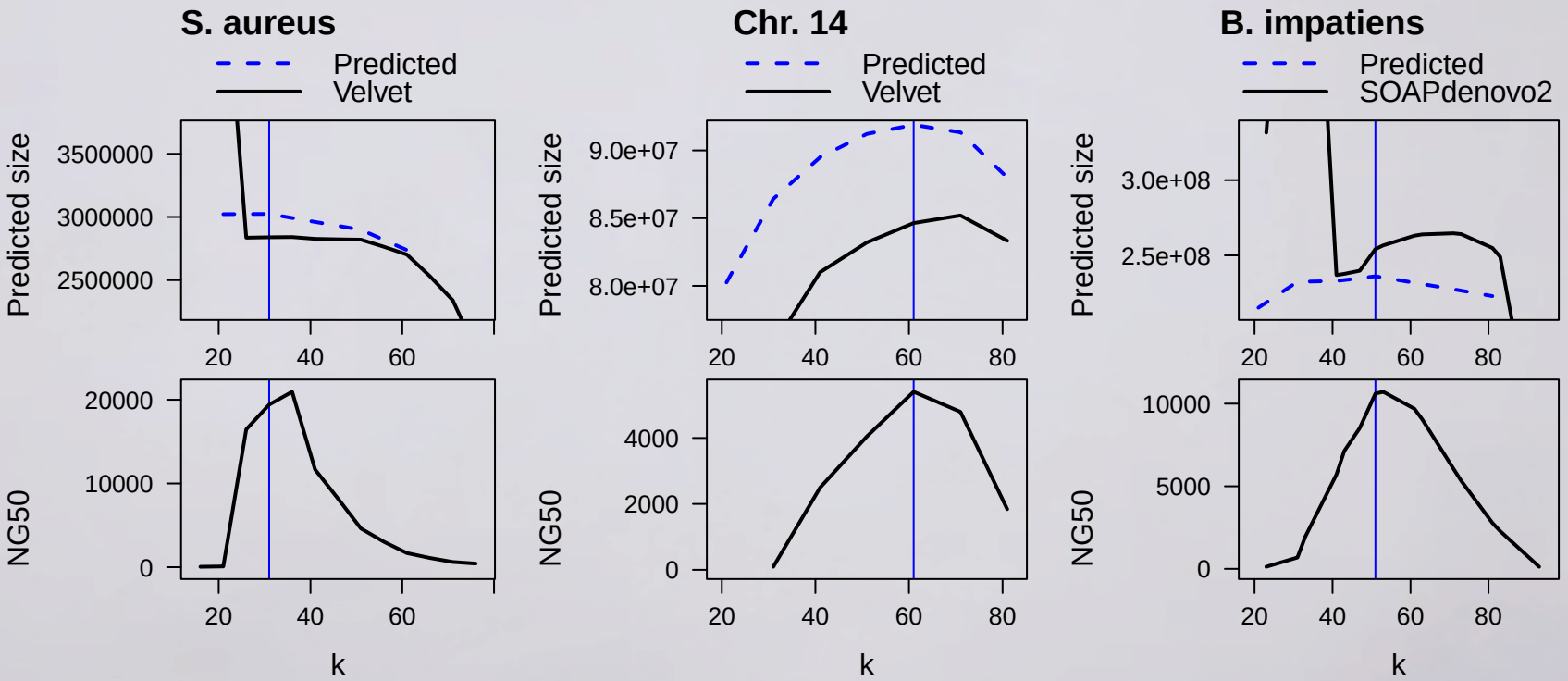
Evaluation on actual datasets from GAGE (assembly benchmark):
[Salzberg 2011]

Dataset	**S. aureus**	**human chr 14**	**B. impatiens**
Genome size	2.9 Mbp	88 Mbp	250 Mbp
Coverage	167x	70x	247x
Avg read length	101 bp	101 bp	124 bp

Selected a typical assembler for each dataset, executed $\forall k$:
Velvet and SOAPdenovo2 [Zerbino 2008, Luo 2013]

KmerGenie Results: accuracy

Predicted best *k* and predicted assembly size vs actual assembly size and NG50 for 3 organisms (GAGE benchmark).



vertical lines corresponds to predicted best *k*

Conclusion / perspectives

- `KmerGenie` **helps choose the k-mer size for de novo assembly**
- Experiments: choices are close to the best possible
- Methods:
  - Best k maximizes the number of genomic k-mers
  - Quake's statistical model
  - Efficient k-mer histogram sampling

Perspectives:

- Increase robustness (high-coverage, longer reads)
- Improve statistical model
- Estimation of Velvet's cov_cutoff $\Longrightarrow$ zero-parameter assembler
- Extract information from histograms for transcriptome and meta-genomes

USING KMERGENIE

```
curl http://kmergenie.bx.psu.edu/kmergenie-1.5397.tar.gz | tar xz
cd kmergenie-1.5397
make
```

Usage for a single file:

```
./kmergenie reads.fastq
```

Usage for a list of files:

```
ls -1 *.fastq > list_reads
./kmergenie list_reads
```

It returns:

```
best k: 47
```

As well as a set of kmer histograms to visualize.

Thank you for your attention!