

# KBase Variation Services

Overview and Demo

Michael Schatz, James Gurtowski  
Cold Spring Harbor Laboratory

1. Introduction to KBase
2. Resequencing and variation calling theory
3. KBase services for variation calling
4. Live Demo
5. Additional Resources

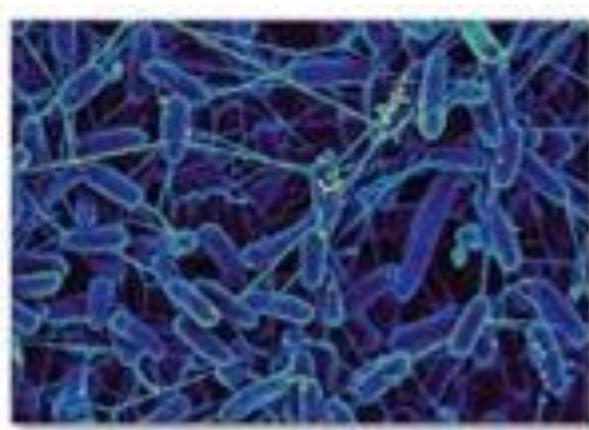


1. Introduction to KBase
2. Resequencing and variation calling theory
3. KBase services for variation calling
4. Live Demo
5. Additional Resources



## ***Knowledgebase*** enabling ***predictive*** systems biology.

- Powerful ***modeling*** framework.
- ***Community-driven***, extensible and scalable ***open-source*** software and application system.
- Infrastructure for integration and reconciliation of ***algorithms*** and ***data sources***.
- Framework for standardization, search, and ***association*** of data
- Resources to enable ***experimental design*** and ***interpretation*** of results.



Microbes

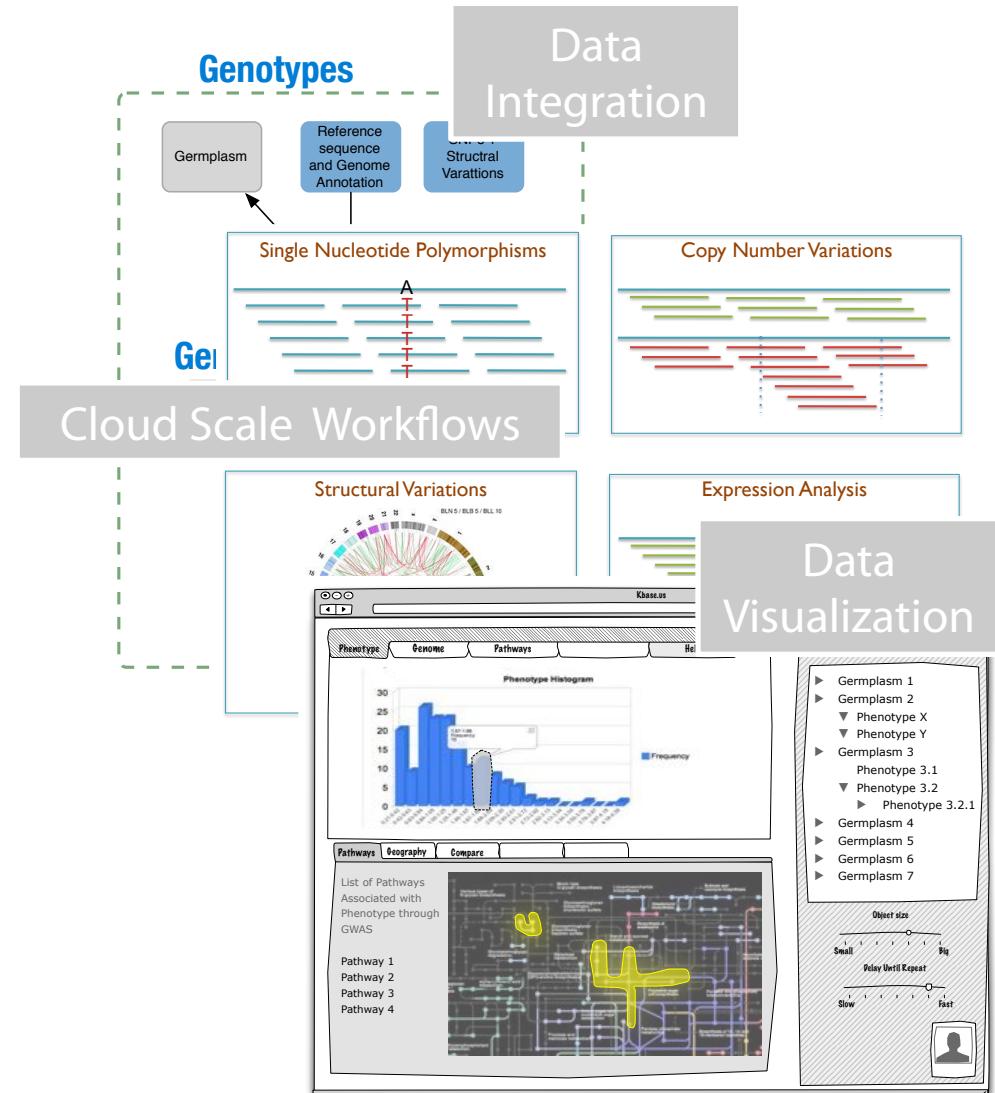
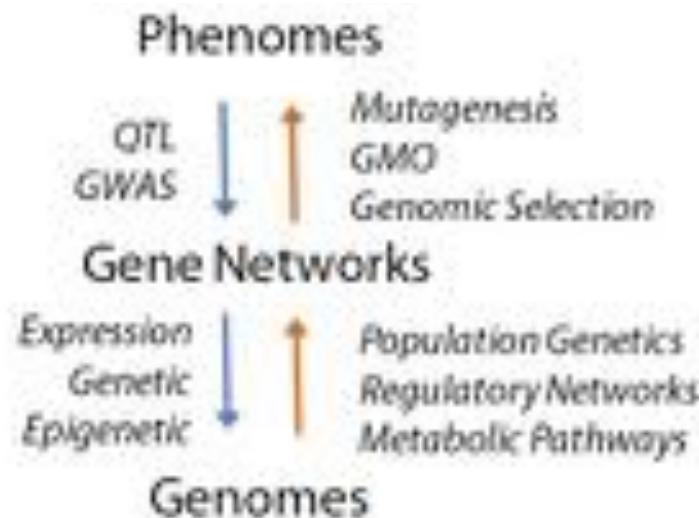


Communities



Plants

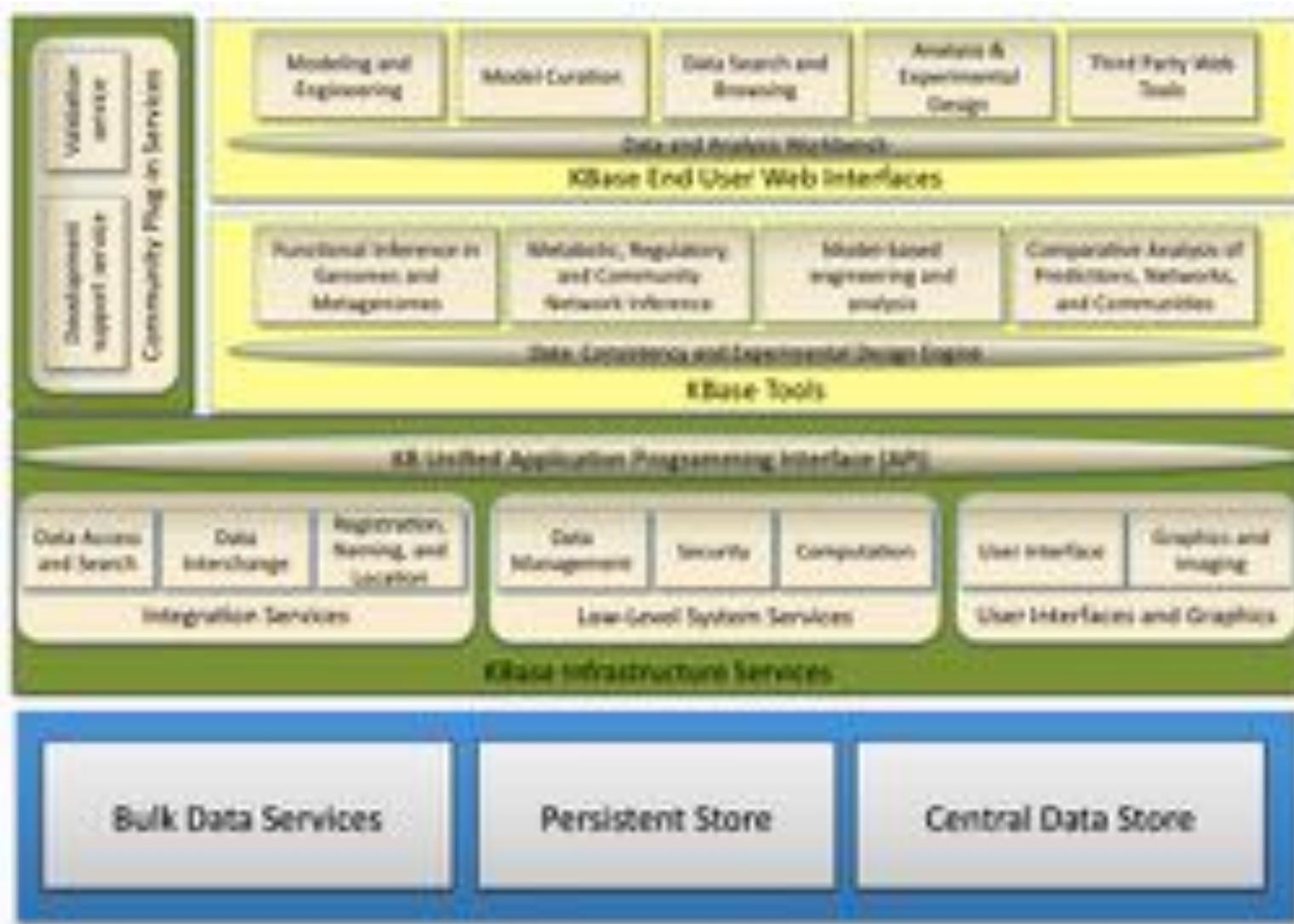
# Model development Hypothesis testing Knowledge Synthesis





DOE Systems Biology Knowledgebase

# KBase Infrastructure and Services





DOE Systems Biology Knowledgebase

# Variation Services: Samples to Discoveries

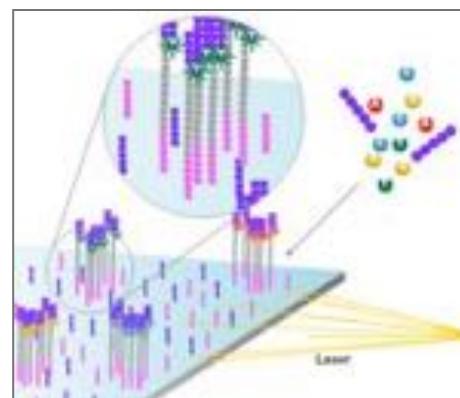
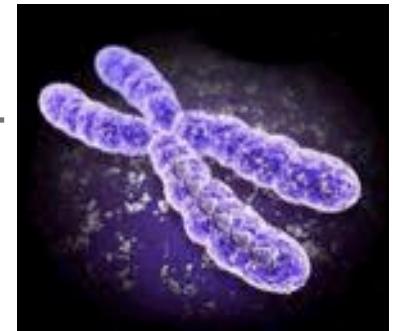


Powered by KBase

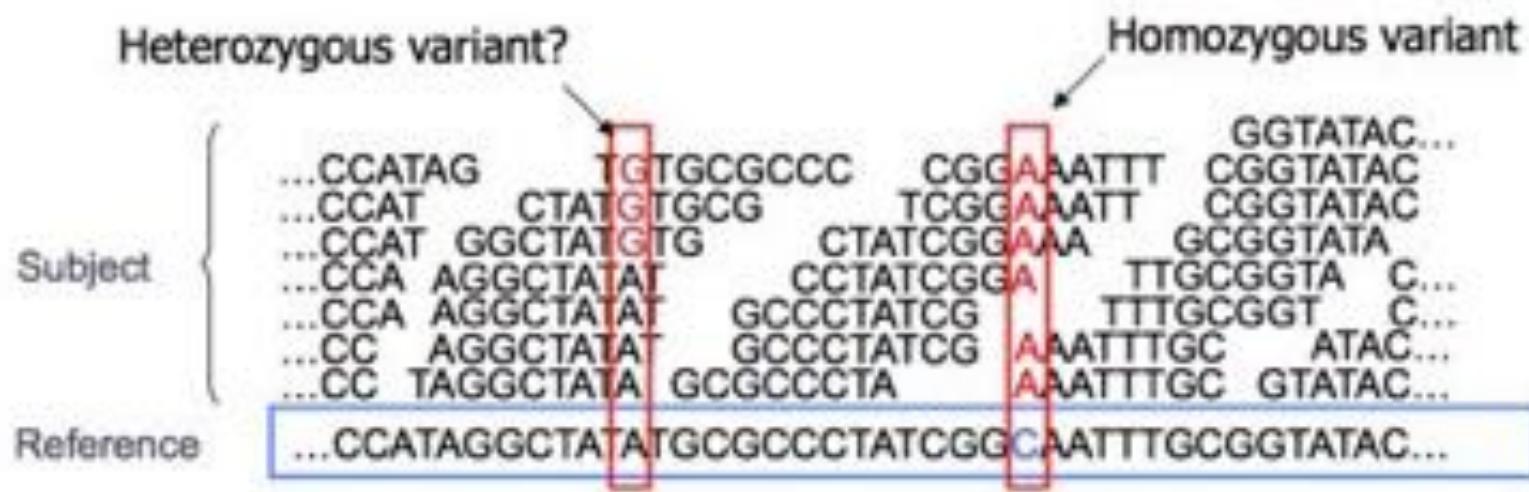
1. Introduction to KBase
2. Resequencing and variation calling theory
3. KBase services for variation calling
4. Live Demo
5. Additional Resources



## How does your sample compare to the reference?

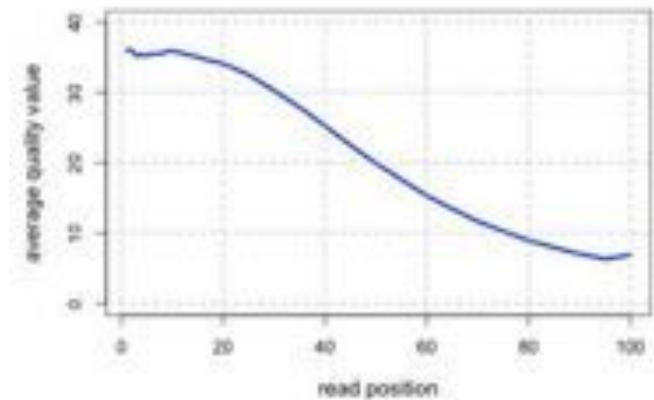


PlantHeight —  
Drought Resistance —  
Biomass production —



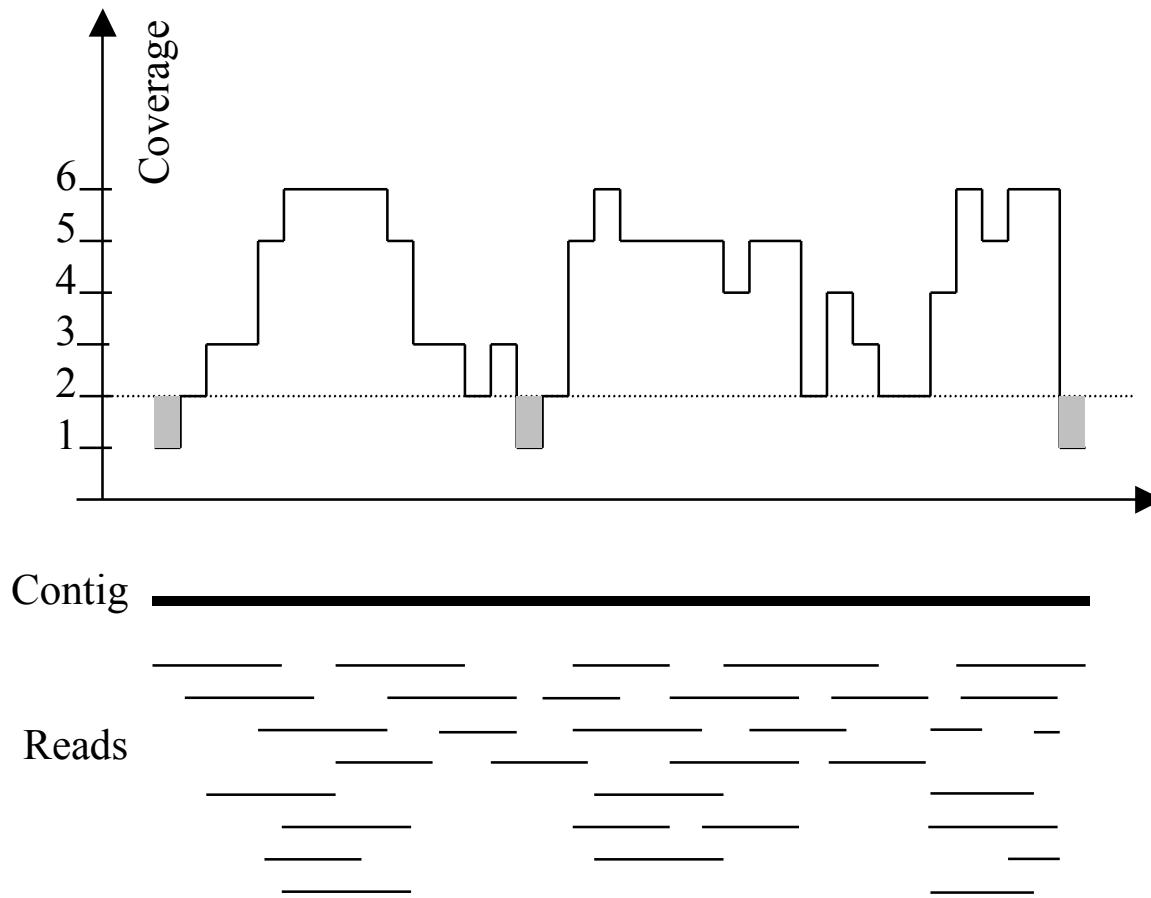
- Sequencing instruments make mistakes
  - Quality of read decreases over the read length
  
- A single read differing from the reference is probably just an error, but it becomes more likely to be real as we see it multiple times
  - Often framed as a Bayesian problem of more likely to be a real variant or chance occurrence of N errors
  - Accuracy improves with deeper coverage

$$Q_{\text{sanger}} = -10 \log_{10} p$$



## Coverage

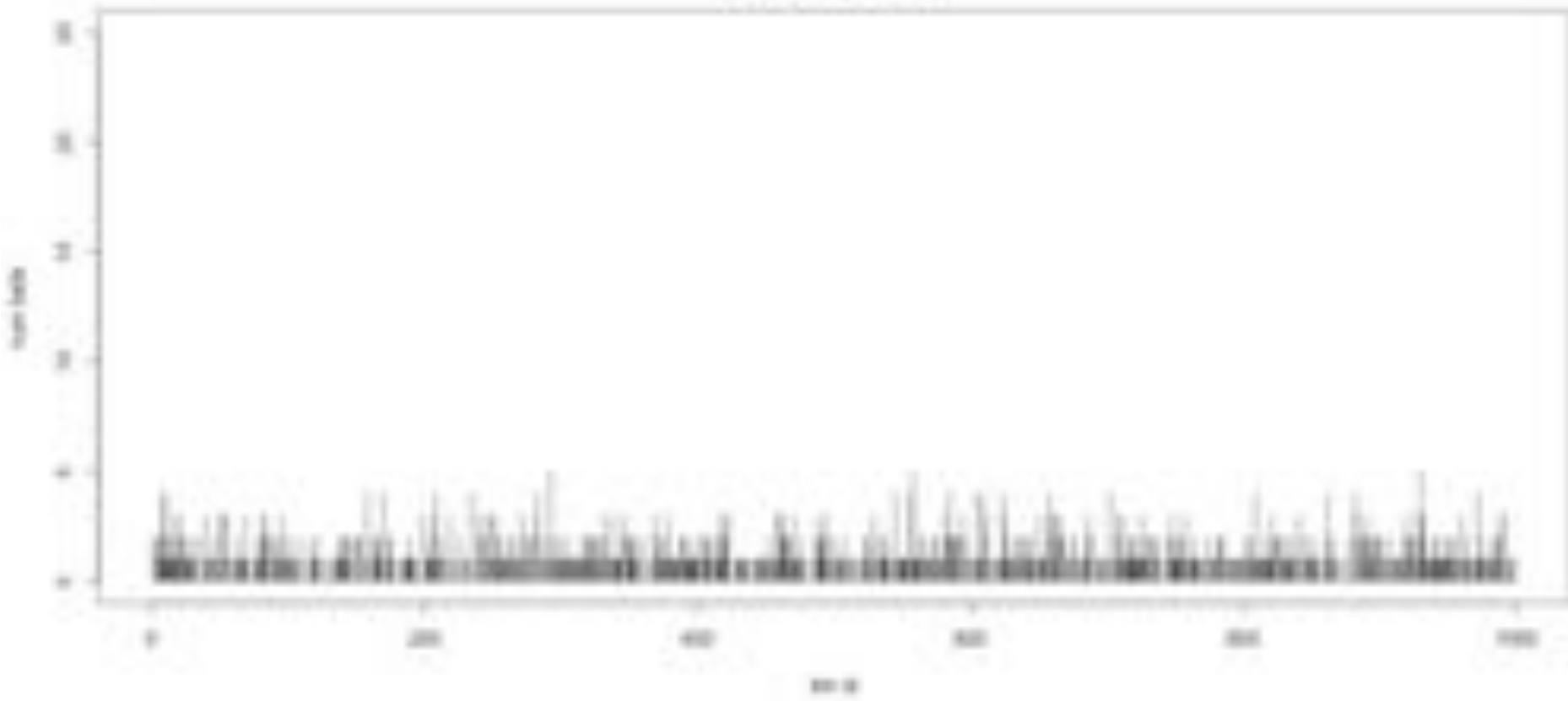
# Typical contig coverage



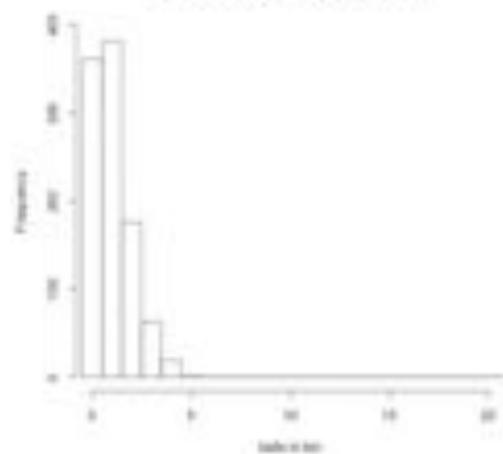
Imagine raindrops on a sidewalk

# Ix Sequencing

Balls in Boxes  
Total balls: 10000

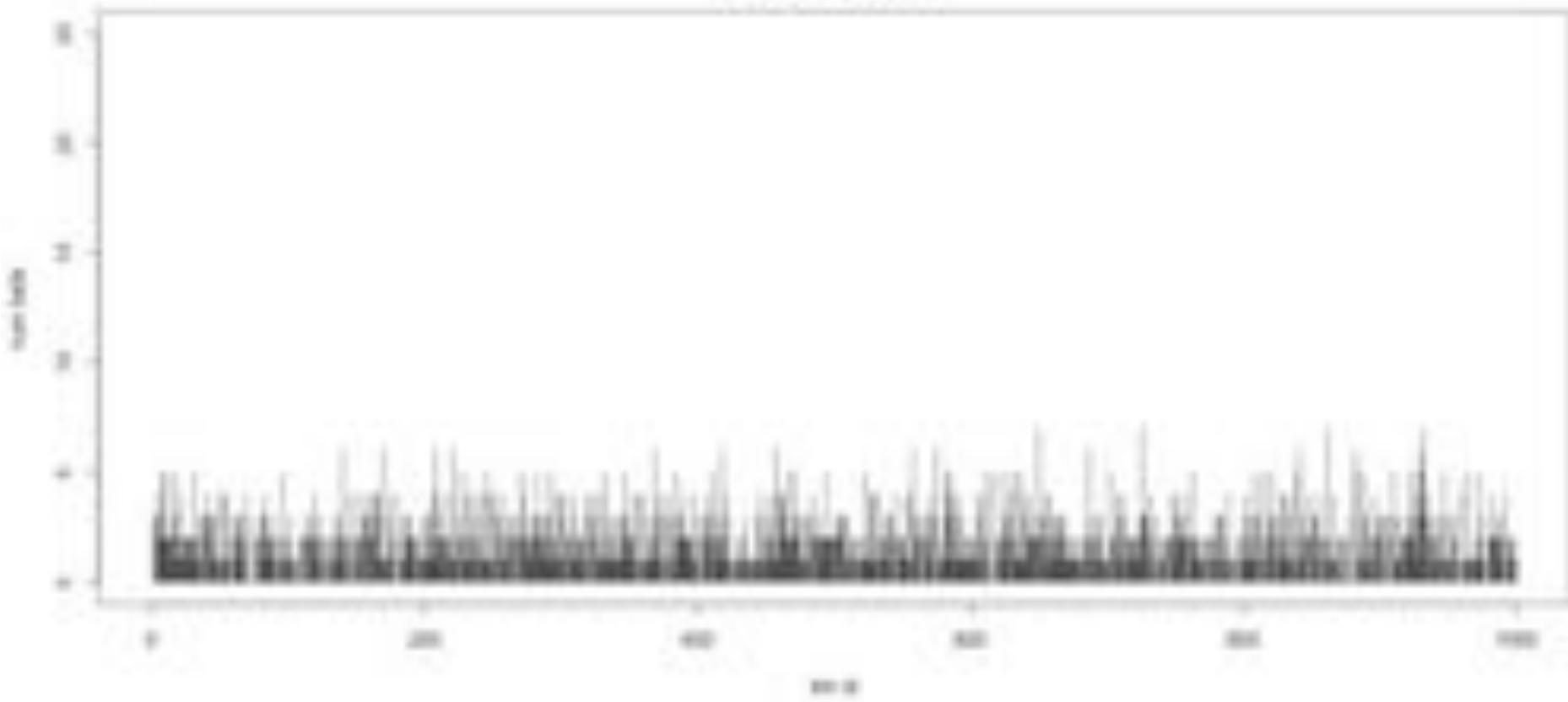


Histogram of balls in each box  
Total balls: 10000. Empty boxes: 301

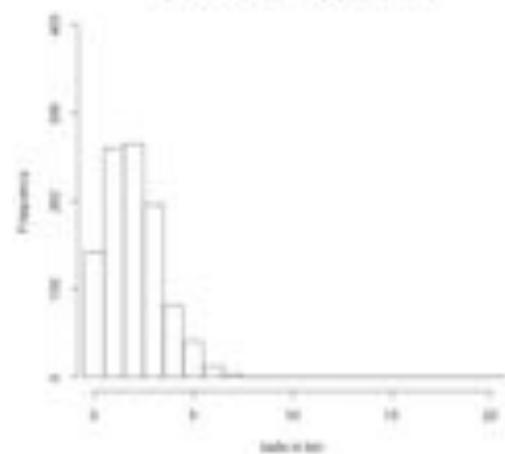


# 2x Sequencing

Reads in bins  
Total reads: 2000

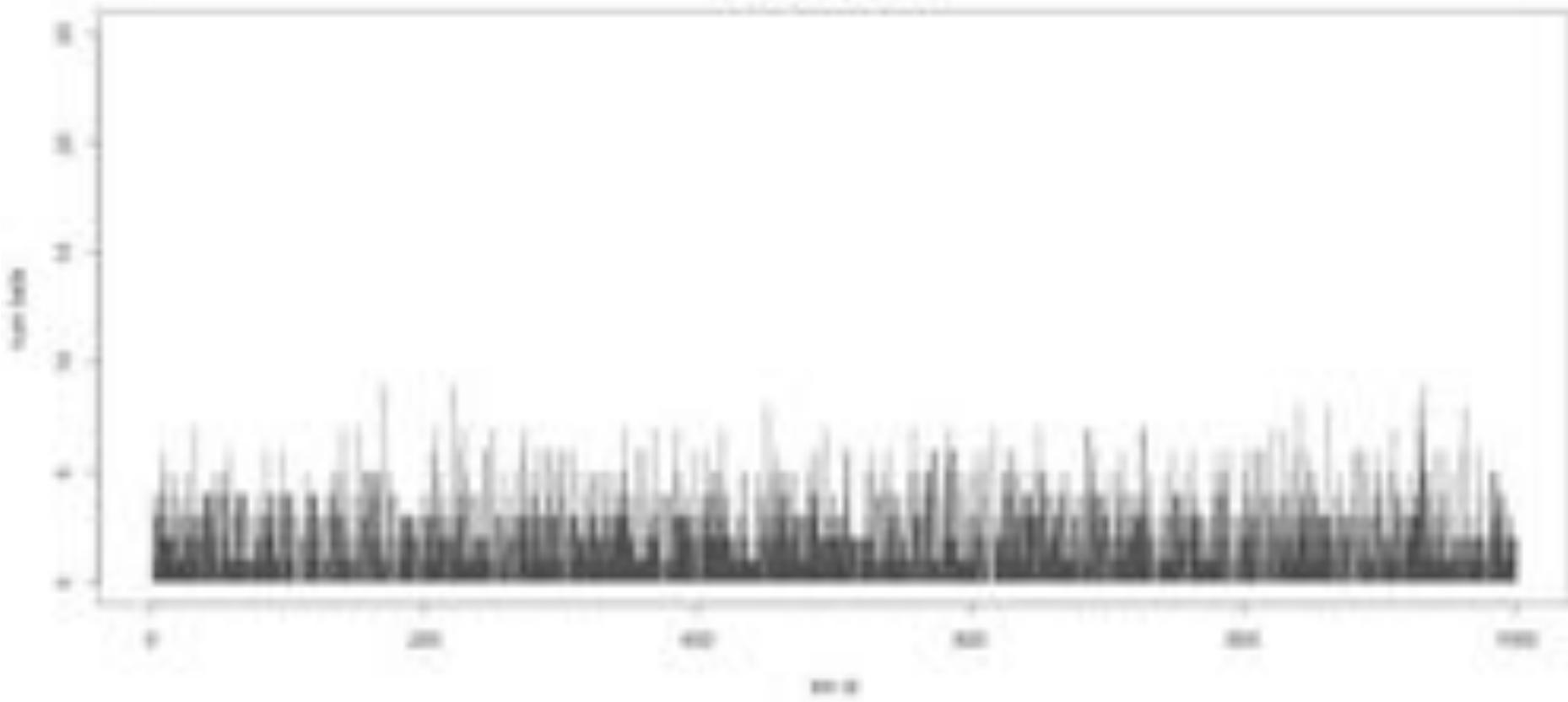


Histogram of bins in each bin  
Total bins: 2000. Empty bins: 142

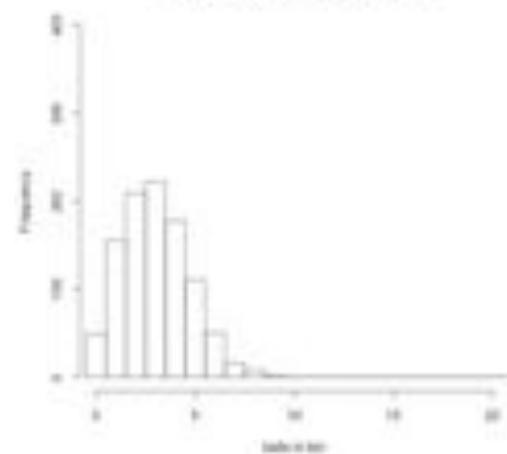


# 3x Sequencing

Reads in bins  
Total reads: 30000

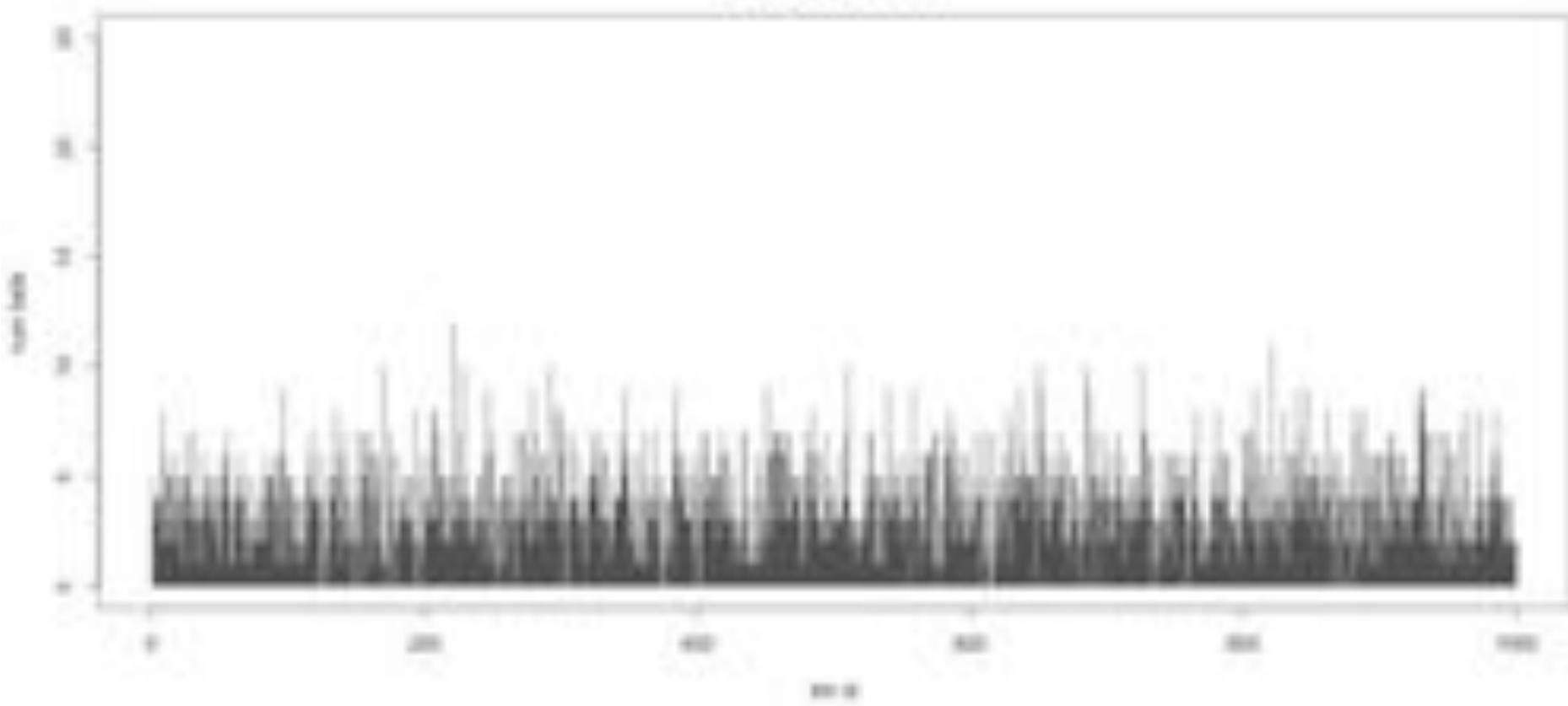


Histogram of reads in each bin  
Total reads: 30000 Empty bins: 49

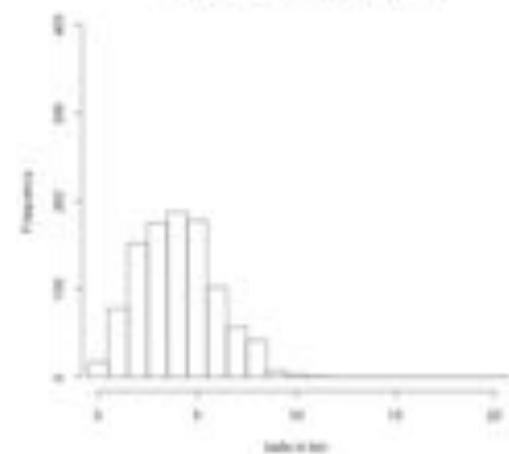


# 4x Sequencing

Reads in bins  
Total reads: 40000

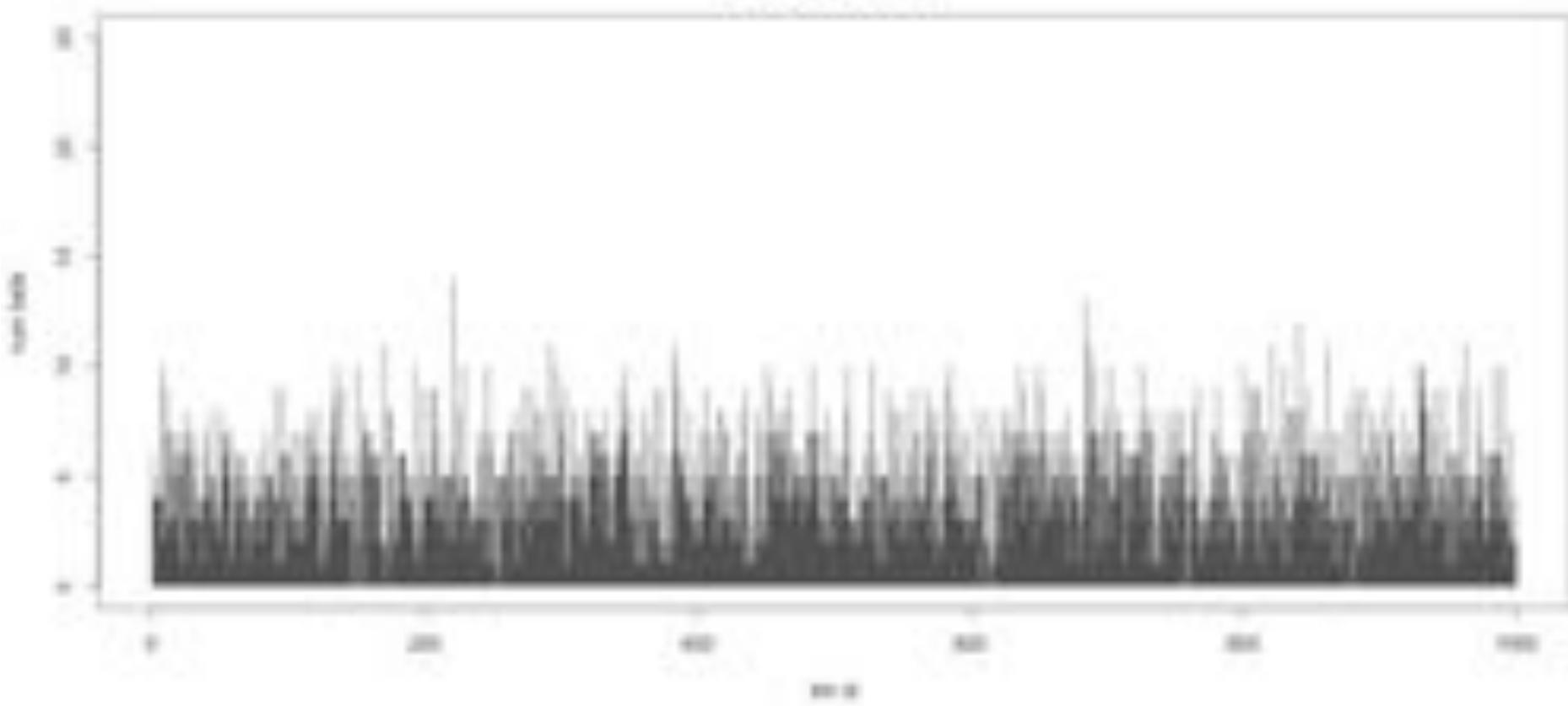


Histogram of reads in each bin  
Total reads: 40000 Empty bins: 17

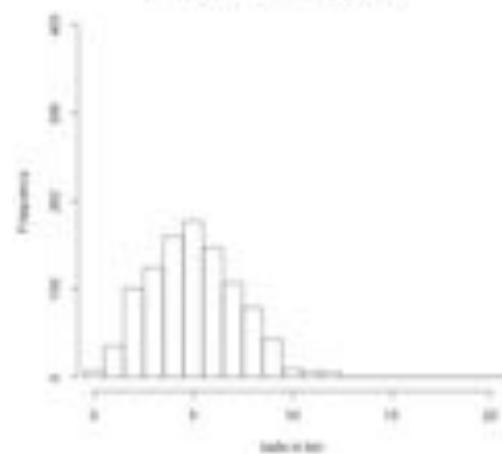


# 5x Sequencing

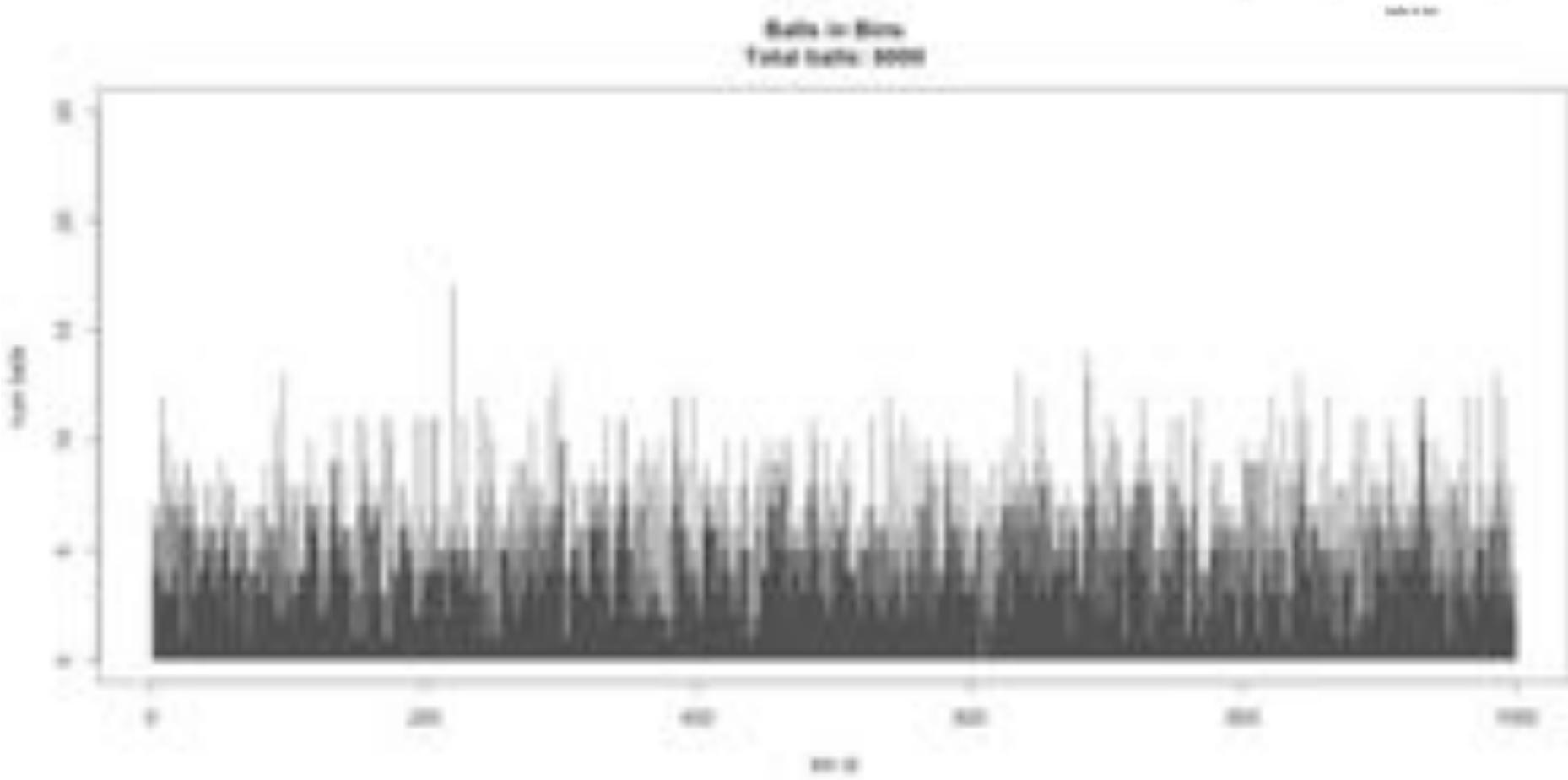
Reads in bins  
Total reads: 9400



Histogram of bins in each bin  
Total bins: 9400 Empty bins: 7



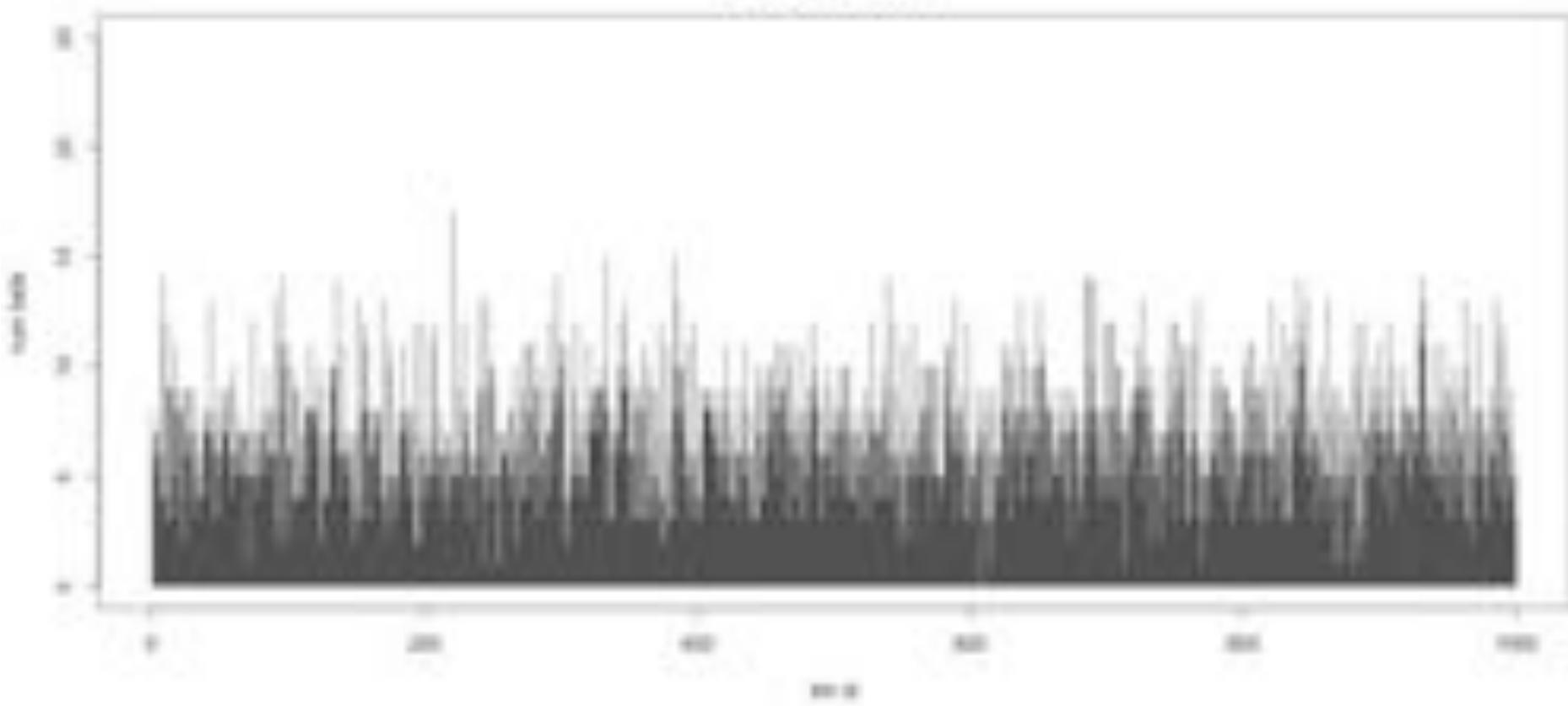
# 6x Sequencing



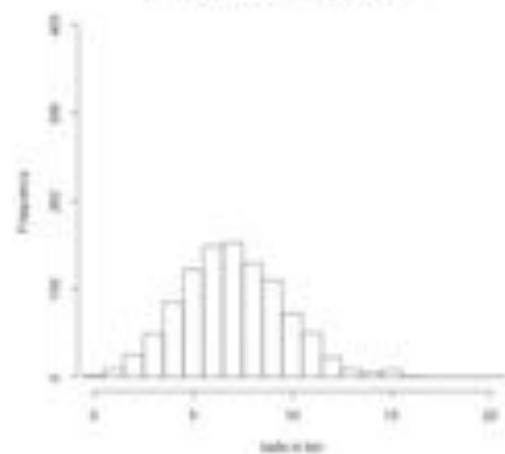
Histogram of bins in each bin  
Total bins: 6000 Empty bins: 2

# 7x Sequencing

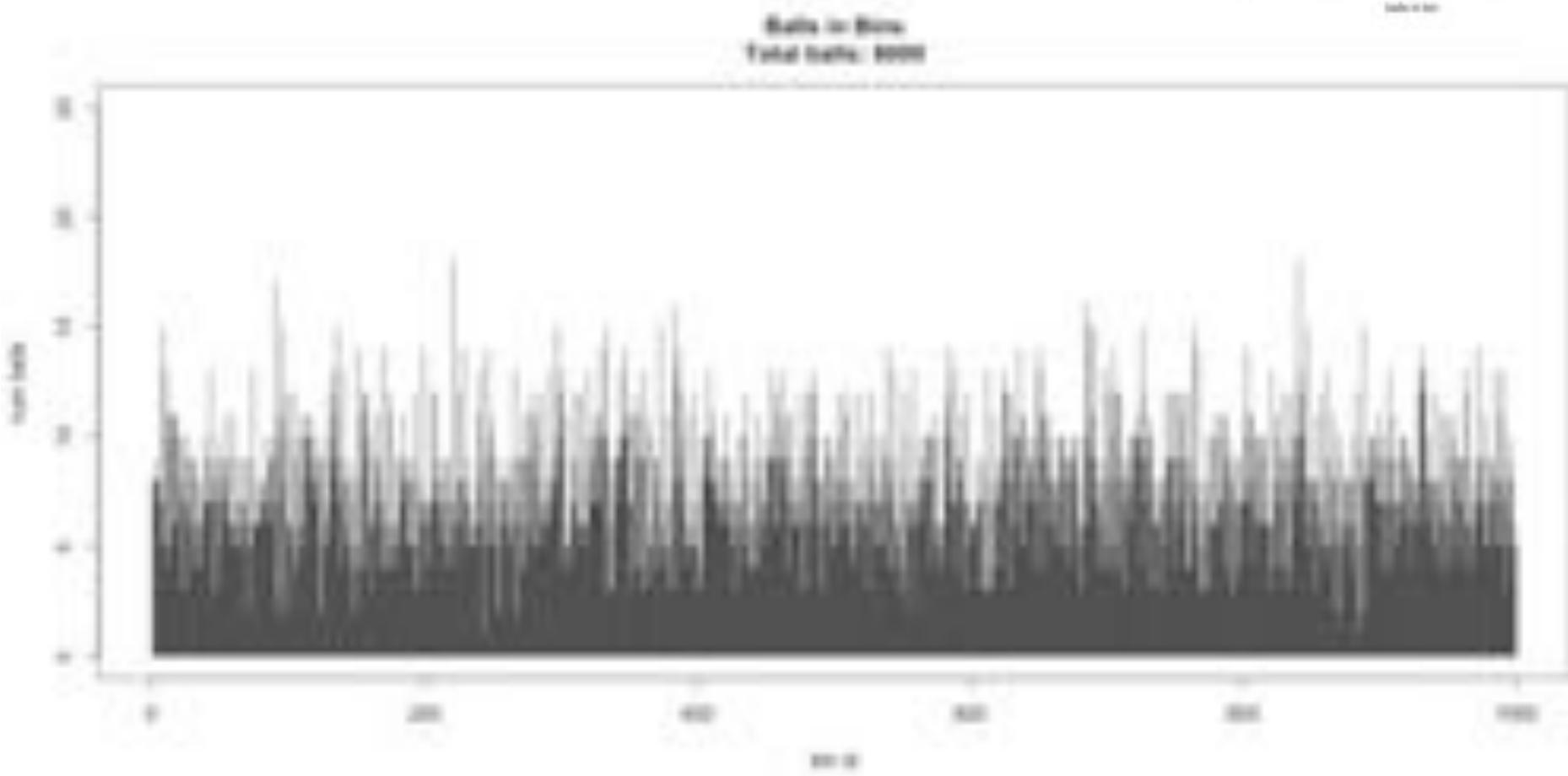
Reads in bins  
Total reads: 7900



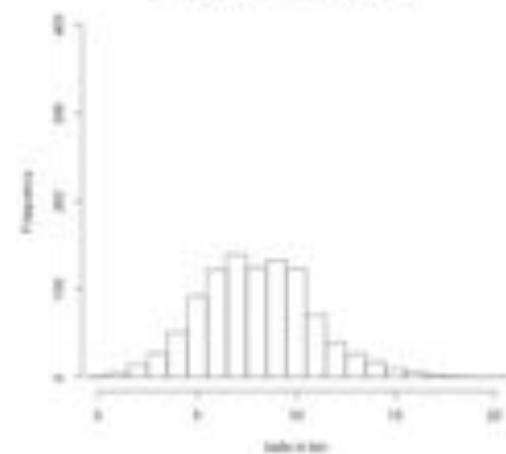
Histogram of bins in each bin  
Total bins: 7900 Empty bins: 2



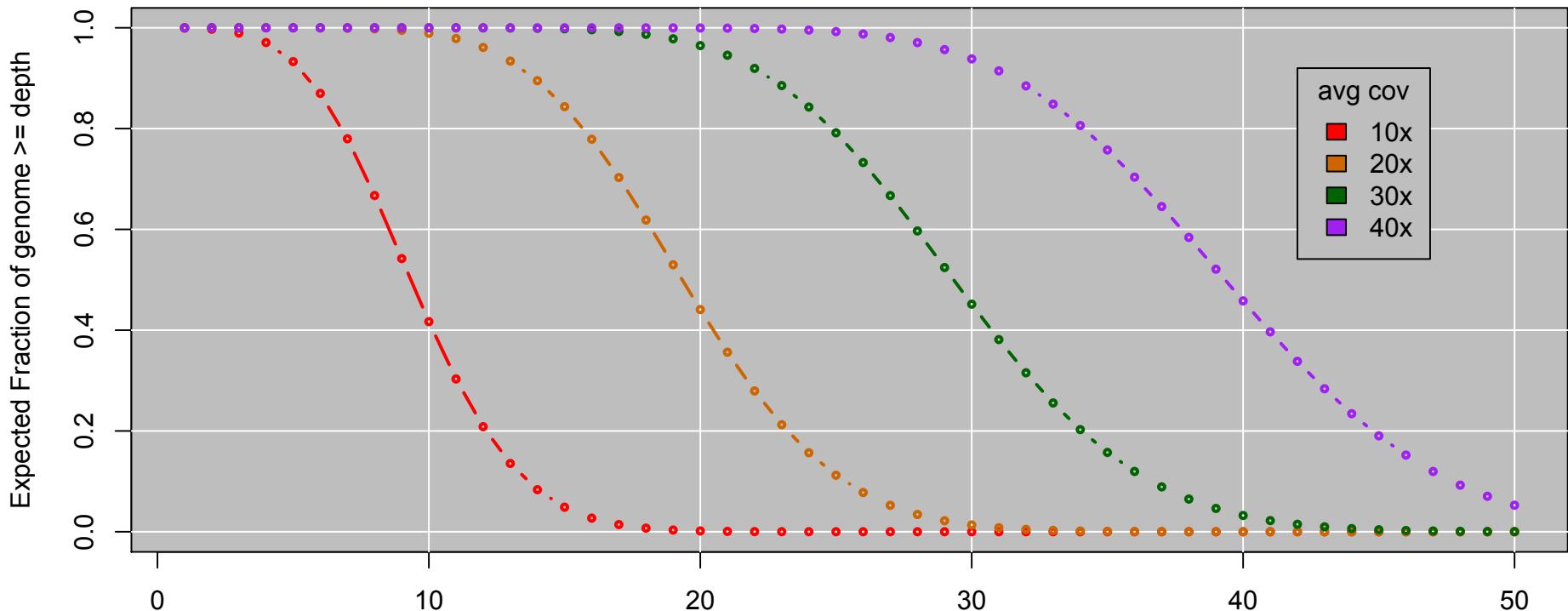
# 8x Sequencing



Histogram of bins in each bin  
Total bins: 8000 Empty bins: 1



# Genome Coverage Distribution



Expect Poisson distribution on depth  
Standard Deviation =  $\sqrt{\text{cov}}$

This is the mathematically model  $\Rightarrow$  reality may be much worse  
Double your coverage for diploid genomes

# Bowtie2 Overview

## 1. Split read into segments

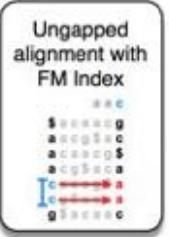
Read  
  
Policy: extract 16 nt seed every 10 nt

Seeds

+, 0: CCAGTAGCTCTCAGGCC	-, 0: TACAGGCCTGGGTA
+, 10: TCAGCCTTATTTTACC	-, 10: GGTAAATAAGGCTGA
+, 20: TTTACCCAGGCCGTGA	-, 20: GGCTGAGAGCTACTGG

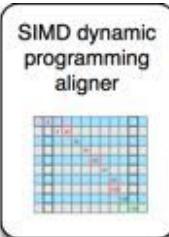
## 2. Lookup each segment and prioritize

Seeds

+, 0: CCAGTAGCTCTCAGGCC	→	Ungapped alignment with FM Index	→	Seed alignments (as B ranges)
+, 10: TCAGCCTTATTTTACC				{ [211, 212], [212, 214] }
+, 20: TTTACCCAGGCCGTGA				{ [653, 654], [651, 653] }
-, 0: TACAGGCCTGGGTA				{ [684, 685] }
-, 10: GGTAAATAAGGCTGA				{ }
-, 20: GGCTGAGAGCTACTGG				{ [624, 625] }

## 3. Evaluate end-to-end match

Extension candidates

SA:684, chr12:1955	→	SIMD dynamic programming aligner	→	SAM alignments
SA:624, chr2:462				r1 0 chr12 1936 0
SA:211: chr4:762				36M * 0 0
SA:213: chr12:1935				CCAGTAGCTCTCAGCCTTATTTTACCCAGGCCGTGA
SA:652: chr12:1945				II

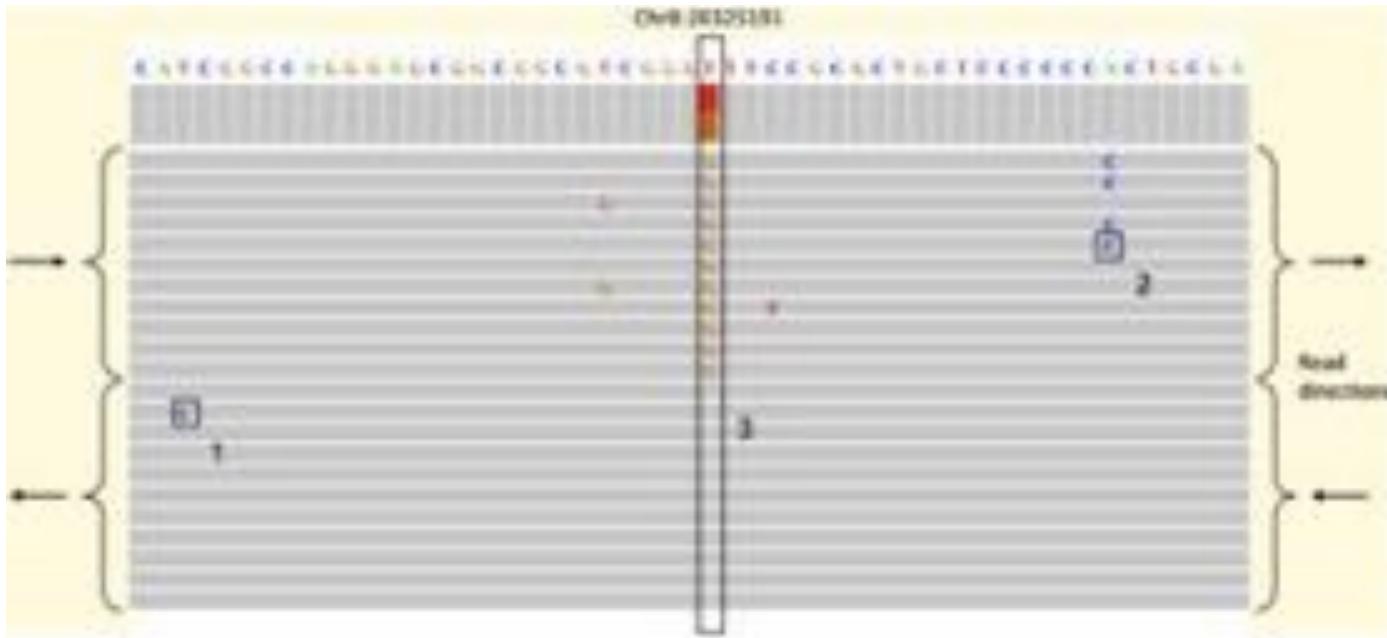
...

**Fast gapped-read alignment with Bowtie 2.**

Langmead B, Salzberg S. Nature Methods. 2012, 9:357-359.

# SNP calling

Beware of (Systematic) Errors



- Distinguishing SNPs from sequencing error typically a likelihood test of the coverage
  - Probability of seeing the data from a heterozygous SNP versus from sequencing error
  - However, some sequencing errors are systematic!

**Identification and correction of systematic error in high-throughput sequence data**

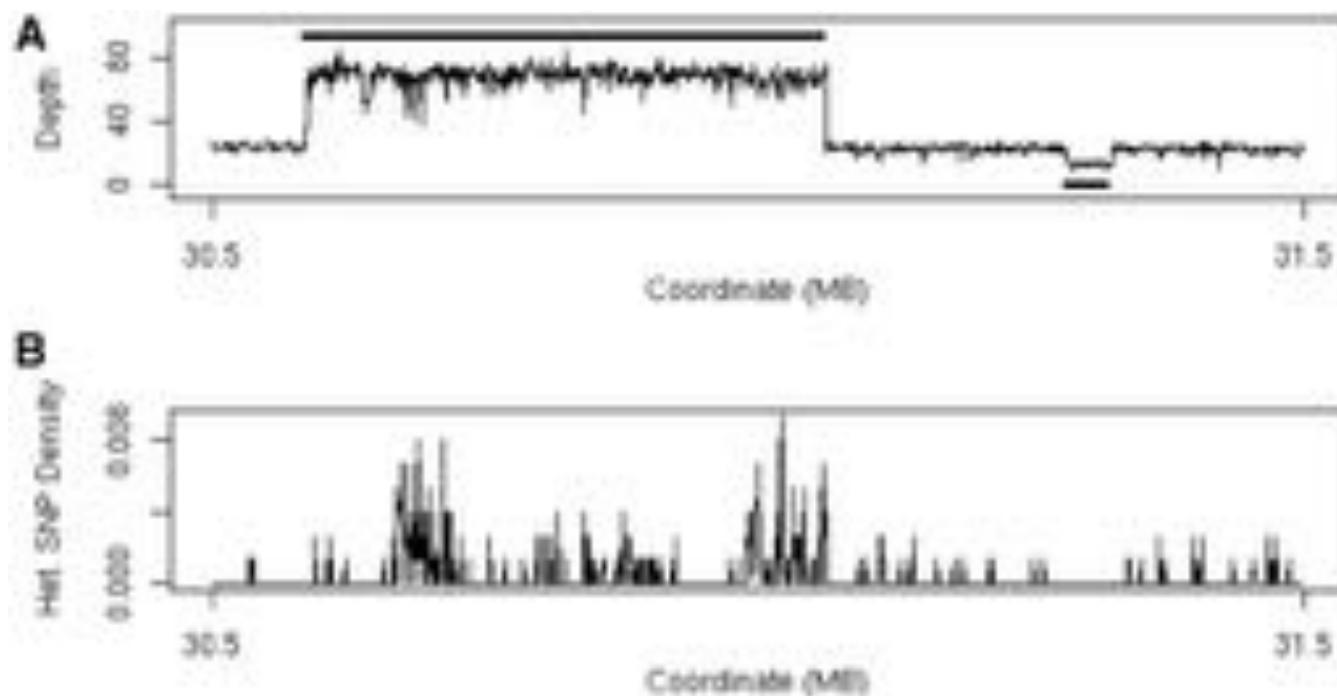
Meacham et al. (2011) *BMC Bioinformatics*. 12:451

**A closer look at RNA editing.**

Lior Pachter (2012) *Nature Biotechnology*. 30:246-247

# CNV calling

Beware of (Systematic) Errors



(A) Plot of sequencing depth across a one megabase region of A/J chromosome 17 clearly shows both a region of 3-fold increased copy number (30.6–31.1 Mb) and a region of decreased copy number (at 31.3 Mb).

Simpson J T et al. Bioinformatics 2010;26:565-567

- Identify CNVs through increased depth of coverage & increased heterozygosity
  - Segment coverage levels into discrete steps
  - Be careful of GC biases and mapping biases of repeats

1. Introduction to KBase
2. Resequencing and variation calling theory
3. KBase services for variation calling
4. Live Demo
5. Additional Resources

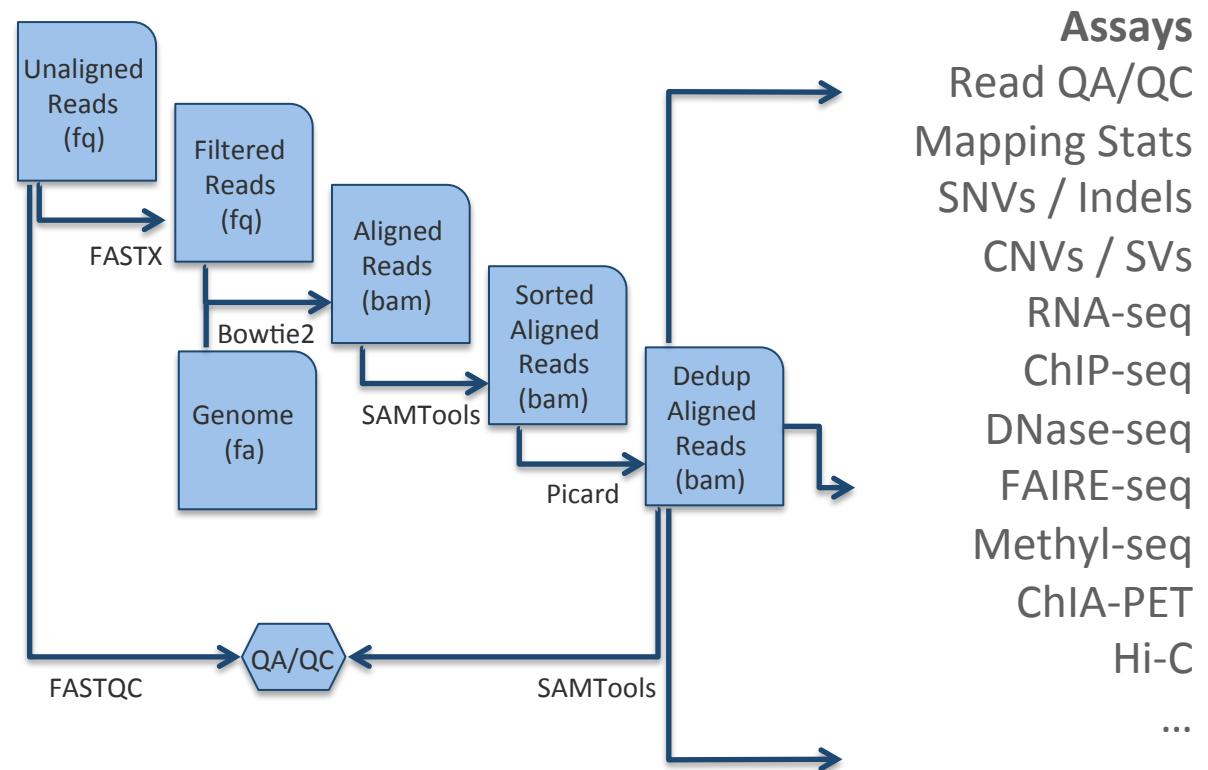


# Sequence to Discovery



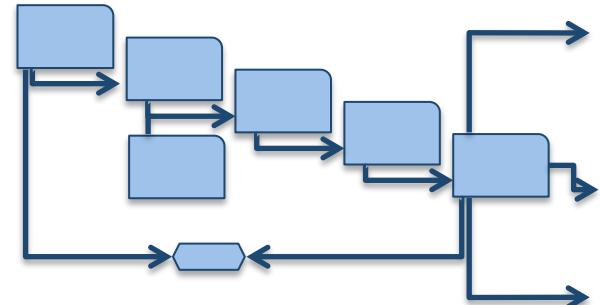
**Illumina HiSeq 2000**  
*Sequencing by Synthesis*

>60Gbp / day



## Genotyping API

- **Bowtie**: Launch alignment task with Bowtie
- **BWA**: Launch alignment task with BWA
- **SNPCalling**: Launch SNPcalling task with SAMTools
- **SortAlignments**: Launch task to sort by chromosome



## Job API

- **ClusterStatus**: return basic status of cluster (jobs running, nodes available, etc)
- **JobStatus**: Given a JobID, returns current status
- **ListJobs**: List JobID running with a given username
- **KillJob**: Kills a given JobID

## Data API

- **List**: List files in a directory
- **Fetch**: Fetch files from HDFS
- **Put**: Put files into HDFS
- **RM**: Delete files on HDFS
- **FetchBAM**: On-the-fly conversion to BAM
- **PutFastq**: Put reads into HDFS with conversion

### Notes:

- All calls are authenticated with KBase username/password

# Reads to SNPs in Five Easy Steps

## 1. Identify reference genome

```
$ all_entities_Genome -f scientific_name | grep -i 'Populus'
```

## 2. Upload Reads to KBase cloud

```
$ jk_fs_put_pe populus.1.fq.gz populus.2.fq.gz populus
```

## 3. Align Reads with Bowtie2

```
$ jk_compute_bowtie -in=populus.pe -org=populus -out=populus_align
```

## 4. Call SNPs with SAMTools

```
$ jk_compute_samtools.snp -in=populus_align -org=populus -out=populus_snps
```

## 5. Merge and Download VCF files

```
$ jk_compute_vcf_merge -in=populus_snps --alignments=populus_align -out=populus.vcf  
$ jk_fs_get populus.vcf
```



DOE Systems Biology Knowledgebase

# Identify a Reference Genome

```
$ all_entities_Genome -f scientific_name | grep -i 'populus'  
kb|g.3907      Populus trichocarpa
```

S&H Lepidoptera, Genus = F patient ID, Name    Refs = 5 "Saccharomyces"
#S&H.10018 <i>Saccharomyces cerevisiae</i> octosporus yf1286-2
#S&H.10042 <i>Zygosaccharomyces baileya</i> IPO 1738
#S&H.10053 <i>Saccharomyces cerevisiae</i> japonicus
#S&H.21735 <i>Zygosaccharomyces kudriavii</i>
#S&H.10036 <i>Saccharomyces cerevisiae</i> rorim
#S&H.21821 <i>Saccharomyces cerevisiae</i> 5288c
#S&H.10088 <i>Saccharomyces cerevisiae</i> (Baker's yeast)
#S&H.20075 <i>Saccharomyces cerevisiae</i> virus S-A 1111
#S&H.20038 <i>Saccharomyces cerevisiae</i> virus U-BC 4147
#S&H.10029 <i>Saccharomyces cerevisiae</i> octosporus
#S&H.21823 <i>Saccharomyces castellii</i>
#S&H.20015 <i>Saccharomyces</i> 235 RNA pathovirus
#S&H.9729 <i>Saccharomyces cerevisiae</i> japonicus yF5275
#S&H.9158 <i>Saccharomyces cerevisiae</i> rorim 9729-2
#S&H.10048 <i>Zygosaccharomyces baileyi</i>
#S&H.10044 <i>Saccharomyces cerevisiae</i>
#S&H.2799 <i>Saccharomyces cerevisiae</i> rorim
#S&H.9058 <i>Saccharomyces cerevisiae</i>
#S&H.8725 <i>Saccharomyces</i> 285 RNA pathovirus
#S&H.21842 <i>Saccharomyces cerevisiae</i> FRS1-1a 5
#S&H.14013 <i>Saccharomyces pastorianus</i> Weihenstephan 34/78
#S&H.8153 <i>Zygosaccharomyces baileyi</i> virus Z
#S&H.8481 <i>Saccharomyces cerevisiae</i> killer virus RL

## Select the proper KBase ID

## Identify reference genome

```
$ all_entities_Genome -f scientific_name | grep -i 'Populus'
```



DOE Systems Biology Knowledgebase

# Upload Reads to KBase Cloud



## User Workstation



KBase Cloud

## Upload Reads to KBase cloud

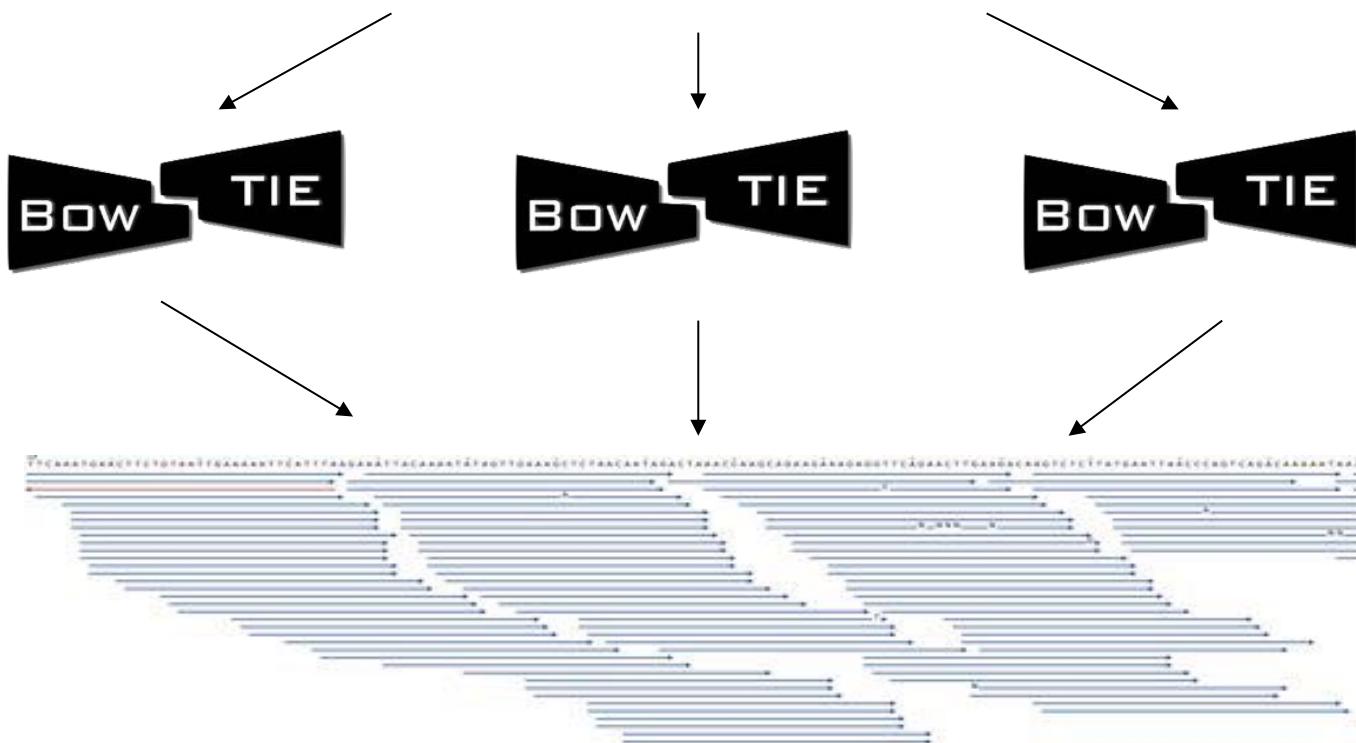
```
$ jk_fs_put_pe populus.1.fq.gz populus.2.fq.gz populus
```



DOE Systems Biology Knowledgebase

## Align Reads with Bowtie2

## Raw Fastq Reads



# Bowtie2 Aligner

## Alignments

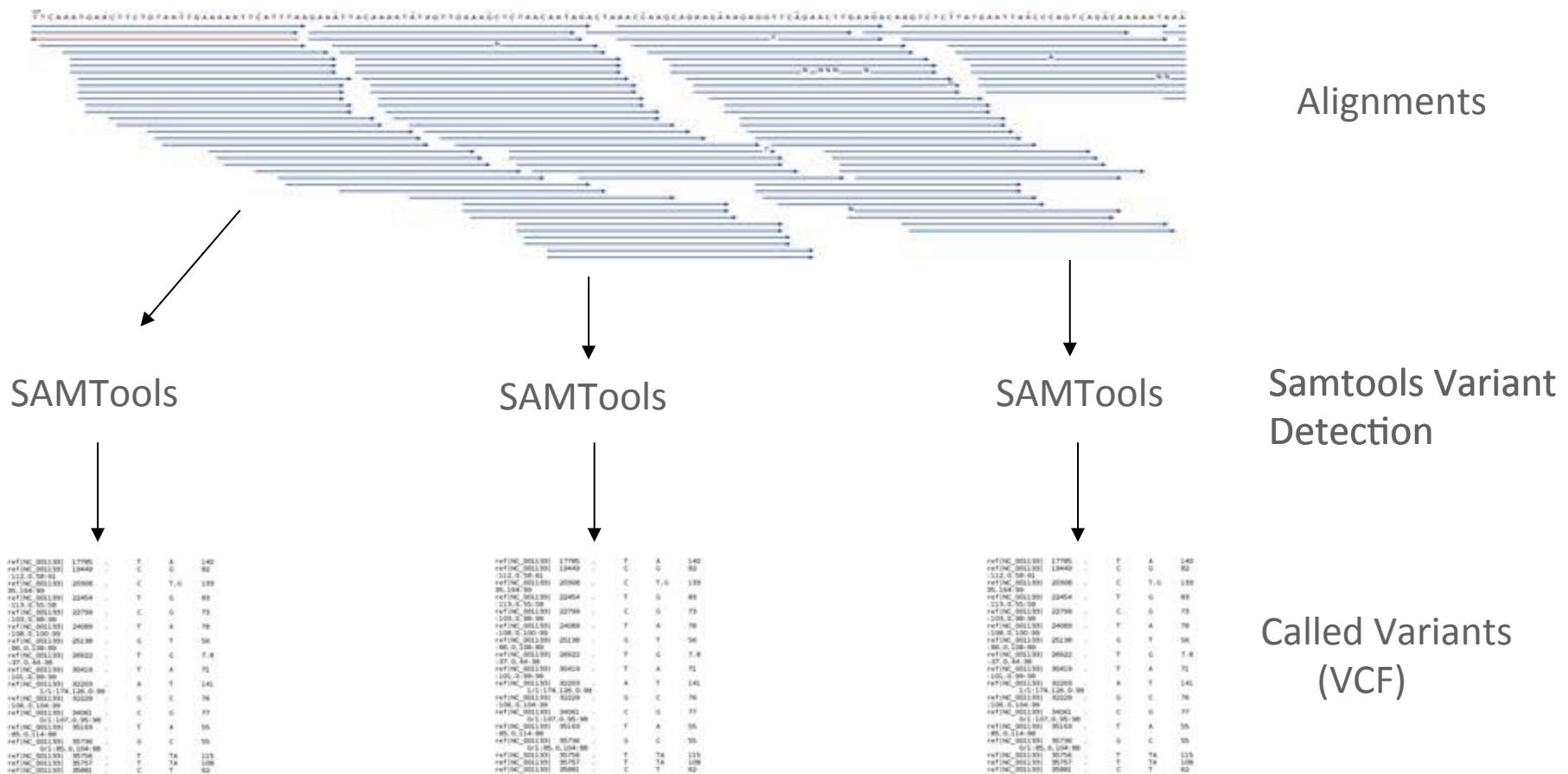
## Align Reads with Bowtie2

```
$ jk_compute_bowtie -in=populus.pe -org='kb|g.3907' -out=populus_align
```



DOE Systems Biology Knowledgebase

# Call SNPs with SAMTools



## Call SNPs with SAMTools

```
$ jk_compute_samtools_snp -in=populus_align -org='kb|g.3907' -out=populus_snps
```



DOE Systems Biology Knowledgebase

# Merge and Download VCF Files

T	C	A	G	U	T	C	G	A
F	G	140	refseq_0011391	17795				
C	G	82	refseq_0011391	19340				
T	C	159	refseq_0011391	1512.0-1513				
T	G	83	refseq_0011391	20580				
C	G	73	refseq_0011391	22454				
F	A	70	refseq_0011391	22799				
G	T	56	refseq_0011391	24089				
F	G	7.8	refseq_0011391	25100				
F	A	71	refseq_0011391	26922				
A	T	141	refseq_0011391	30458				
S	C	76	refseq_0011391	30210				
C	G	97	refseq_0011391	32219				
F	A	55	refseq_0011391	34043				
S	C	55	refseq_0011391	34167-35-36				
T	H	110	refseq_0011391	35736				
C	T	82	refseq_0011391	35861				

## Merge VCF Files

## Download to Local Workstation



## Merge and Download

```
$ jk_compute_vcf_merge -in=populus_snps -alignments=populus_align -out=populus.vcf  
$ jk_fs_get populus.vcf
```

# Reads to SNPs in Five Easy Steps

## 1. Identify reference genome

```
$ all_entities_Genome -f scientific_name | grep -i 'Populus'
```

## 2. Upload Reads to KBase cloud

```
$ jk_fs_put_pe populus.1.fq.gz populus.2.fq.gz populus
```

## 3. Align Reads with Bowtie2

```
$ jk_compute_bowtie -in=populus.pe -org=populus -out=populus_align
```

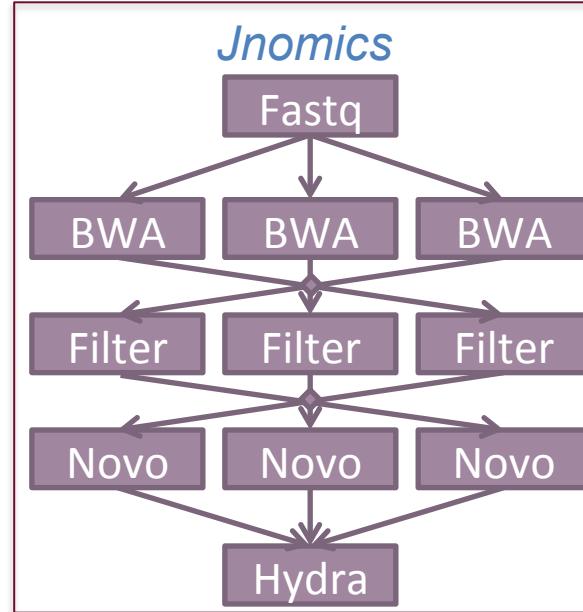
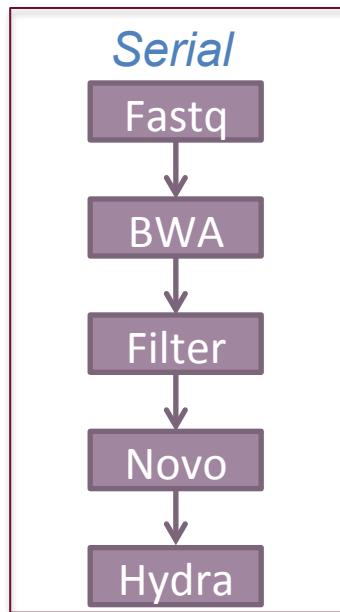
## 4. Call SNPs with SAMTools

```
$ jk_compute_samtools.snp -in=populus_align -org=populus -out=populus_snps
```

## 5. Merge and Download VCF files

```
$ jk_compute_vcf_merge -in=populus_snps --alignments=populus_align -out=populus.vcf  
$ jk_fs_get populus.vcf
```

# Jnomics: Cloud-scale genomics

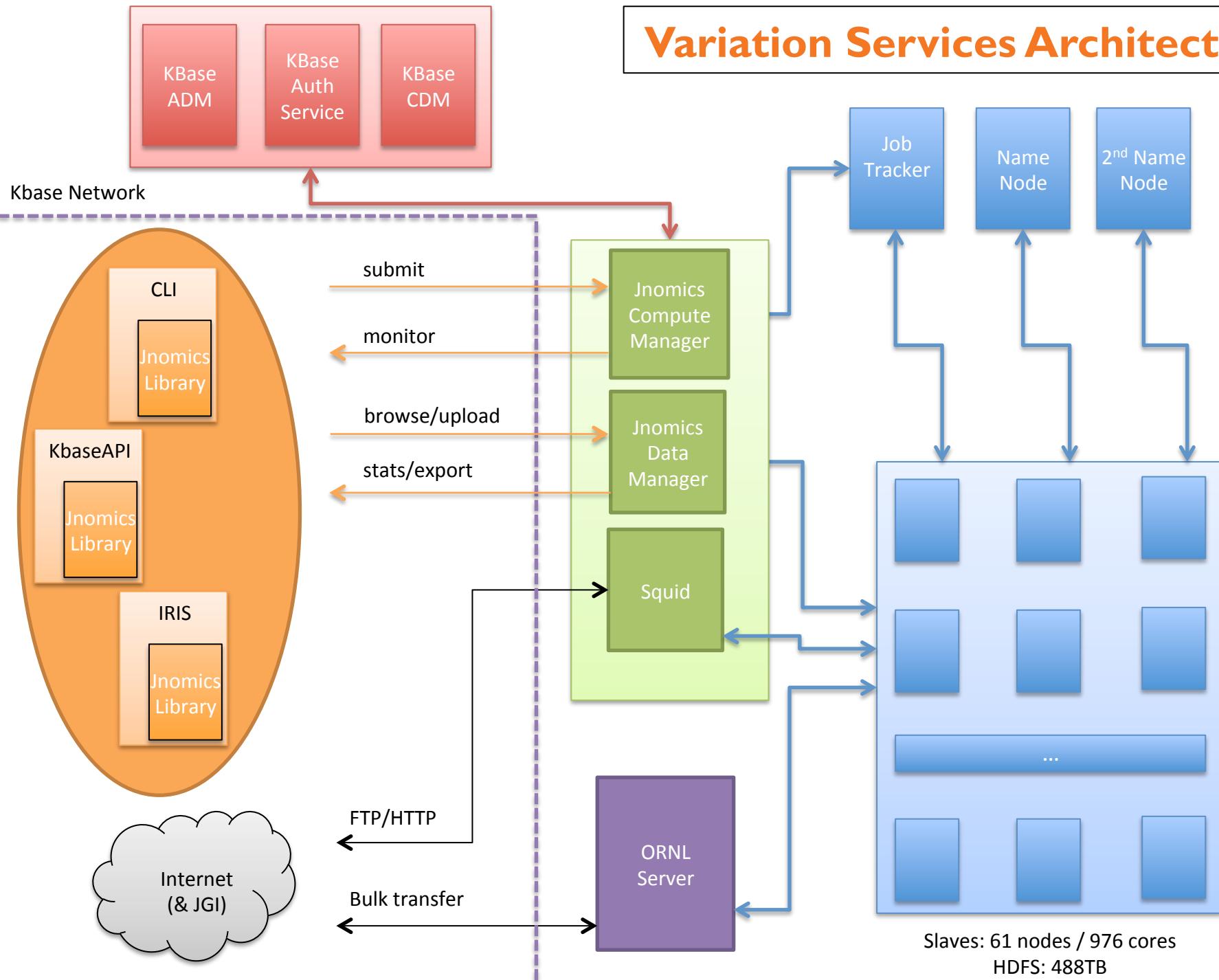


- Rapid parallel execution of data-intensive analysis
  - FASTX, BWA, Bowtie2, Novoalign, SAMTools, Hydra
  - Sorting, merging, filtering, selection, clustering, correlating
  - Supports BAM, SAM, BED, fastq

**Answering the demands of digital genomics**  
 Titmus, MA, Gurtowski, J, Schatz, MC (2012) *Concurrency & Computation*



# Variation Services Architecture



Align & call SNPs from 35M 80bp (14Gbp) reads with maize genome (zmb73v2)  
 Identified 372k high confidence SNPs

	Serial	Multicore	KBase Cloud
Config	1 core (1 node)	44 core (1 node)	118 cores (15 nodes)
Bowtie2	45 h*	1h 10m	23 m
Sort	2 hr	2 hr	N/A
Samtools	2 hr	2 hr	12 m
End-to-End Speedup	50h* 1x	5h 10m 9.6x	35 m 86x

\*estimated time

# Maize Population Analysis

Align & call SNPs from 131 maize samples  
 1TB fastq / 408Gbp input data

	Serial	KBase cloud (small)	KBase Cloud (large)
Config	1 core (1 node)	210 cores (15 nodes)	854 cores (61 nodes)
Bowtie2	1311 hr*	19.5 hr	5 hr
Sort	58 hr*	N/A	N/A
Samtools	58 hr*	3.5 hr	1.5 hr
End-to-End Speedup	1427 hr* 1x	23 hr 62x	6.5 hr 219x

\*estimated time

1. Introduction to KBase
2. Resequencing and variation calling theory
3. KBase services for variation calling
4. Live Demo
5. Additional Resources



## Online Demo

1. Browse to KBase website: <http://kbase.us/>
2. Sign up for KBase account: <https://gologin.kbase.us/SignUp>
3. Download KBase DMG: <http://kbase.us/for-users/get-started/>  
Or use IRIS: <http://kbase.us/services/docs/invocation/Iris/>
4. Variation Services Tutorial:  
<http://kbase.us/for-users/tutorials/analyzing-data/variation-service/>
5. Summarize mutations:  
\$ cat yeast.vcf  
\$ grep -v '^#' yeast.vcf | cut -f1 | sort | uniq -c  
\$ grep -v '^#' yeast.vcf | cut -f 4,5 | sort | uniq -c | sort -nrk1 | head

1. Introduction to KBase
2. Resequencing and variation calling theory
3. KBase services for variation calling
4. Live Demo
5. Additional Resources



## Additional Resources

Resource	URL
KBase	<a href="http://kbase.us/">http://kbase.us/</a>
Getting Started	<a href="http://kbase.us/for-users/user-home/">http://kbase.us/for-users/user-home/</a>
Variation Services	<a href="http://kbase.us/for-users/tutorials/analyzing-data/variation-service/">http://kbase.us/for-users/tutorials/analyzing-data/variation-service/</a>
Bowtie2	<a href="http://bowtie-bio.sourceforge.net/bowtie2/index.shtml">http://bowtie-bio.sourceforge.net/bowtie2/index.shtml</a>
BWA	<a href="http://bio-bwa.sourceforge.net/">http://bio-bwa.sourceforge.net/</a>
SAMTools	<a href="http://samtools.sourceforge.net/">http://samtools.sourceforge.net/</a>
VCF Spec	<a href="http://www.1000genomes.org/wiki/Analysis/Variant%20Call%20Format/vcf-variant-call-format-version-40">http://www.1000genomes.org/wiki/Analysis/Variant%20Call%20Format/vcf-variant-call-format-version-40</a>
SNPeff	<a href="http://snpeff.sourceforge.net/">http://snpeff.sourceforge.net/</a>
KBase Contact	<a href="http://kbase.us/contact-us/">http://kbase.us/contact-us/</a>
***Survey***	<a href="https://www.surveymonkey.com/s/KB-user-info">https://www.surveymonkey.com/s/KB-user-info</a>

Questions?

# Thank You!

<http://schatzlab.cshl.edu>  
@mike\_schatz / @DOEKBase

