DeNovo assembly process

Gonzalo garcia Accinelli
gonzalo.garcia@earlham.ac.uk

Earlham Institute

Bernardo Clavijo
Group leader

Jon Wright
Wheat genomics

Luis Yanes
Scientific programming and HPC

Ben Ward
Tool development / Pop-genomics / Variation

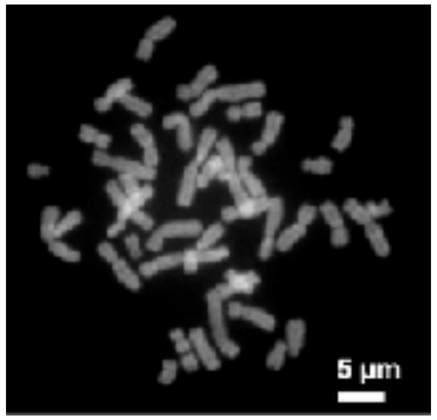
Camilla Ryan
Ph.D. Student

Kat Hodgkinson
Visiting Ph.D. Student

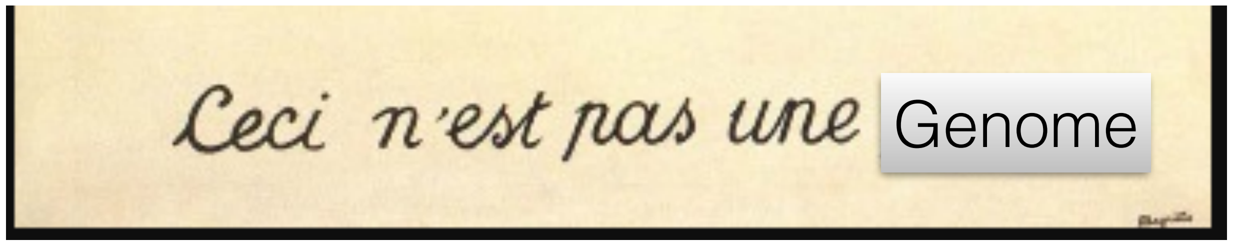
Earlham Institute, Norwich Research Park, Norwich, Norfolk, NR4 7UZ, UK
www.earlham.ac.uk

Earlham Institute
Decoding Living Systems

From genome to assembly



$$\begin{tikzcd}
\text{Sampling} \arrow[d] \\
\text{Lib preping} \arrow[d] \\
\text{Sequencing} \arrow[d] \\
\text{Assembling}
\end{tikzcd}$$

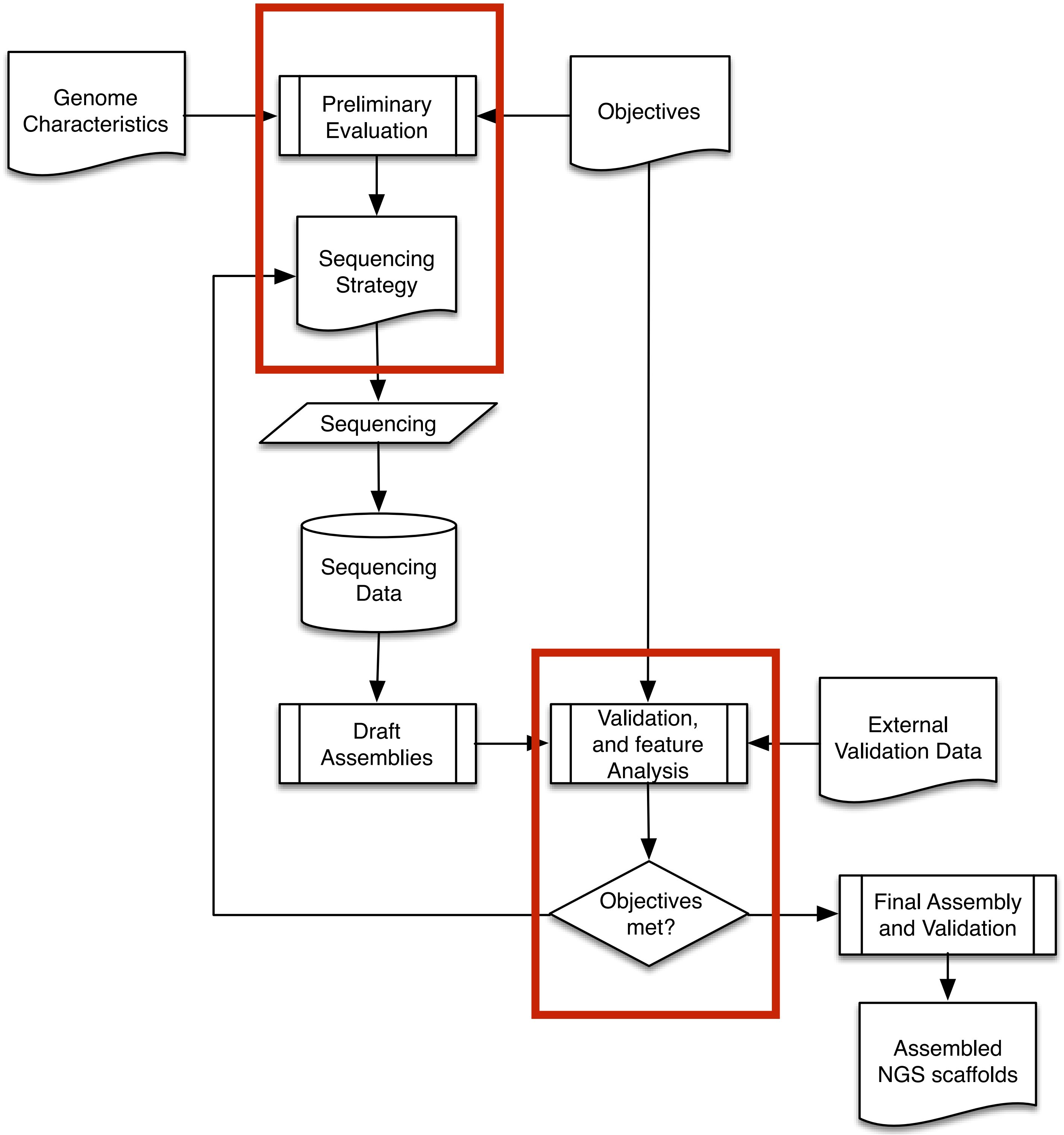


The assembly is a model of the genome.

A tool for hypothesis testing.

Tool to understand the cell/organism/population/ etc.

Have an objective and work towards it



All the information is already present in the experimental results.

Information can not be created
if it’s not in the sampled data is not recoverable

The assembly process is a reduction exercise

The assembly is just a probabilistic model of a genome, condensing the information from the experimental evidence.

Conditions for a correct assembly

The right *motifs,*

the correct number of times,

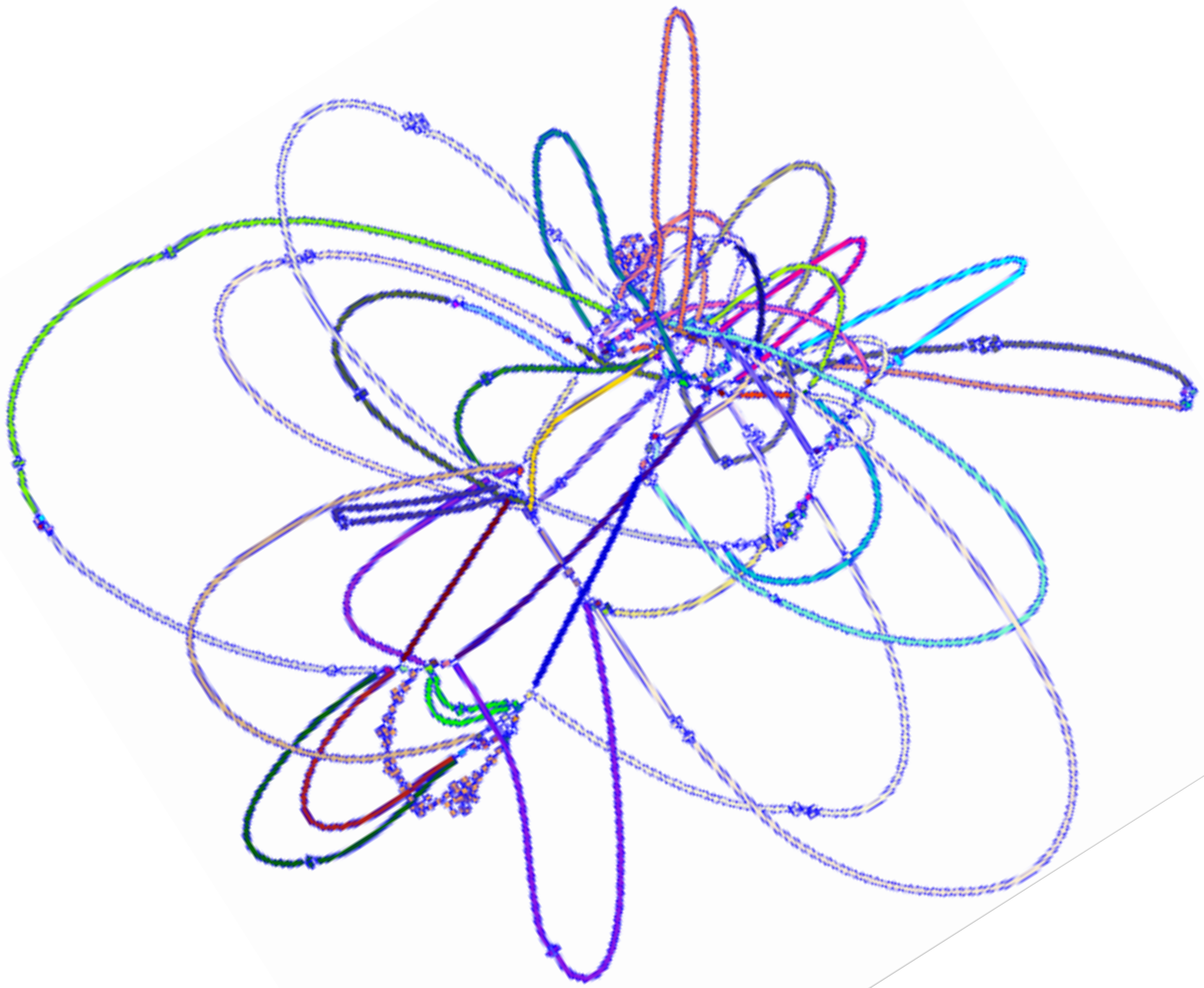
in correct order and position.

Earlham Institute

Graphs, contigs and scaffolds

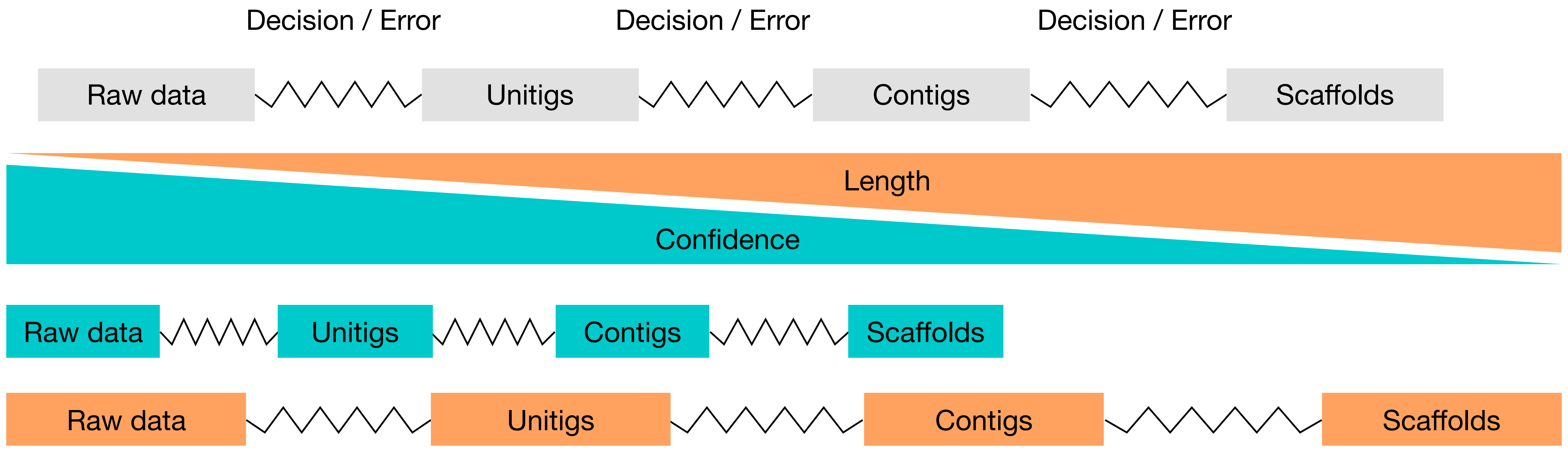
	Sequence origin	Expected quality	Main quality driver
Unitig	1 element in the graph	Very high	Sequence data, cleanup, overlap detection
Contig	suported chain in the graph	High	+ graph complexity, single-read mapping & entropy
Scaffold	external-link group of contigs	Variable	+ pair reliability, parametrisation

- Graphs: assembler's representation
  - More information
  - Allow some back-tracking
  - Can encode support/ambiguity

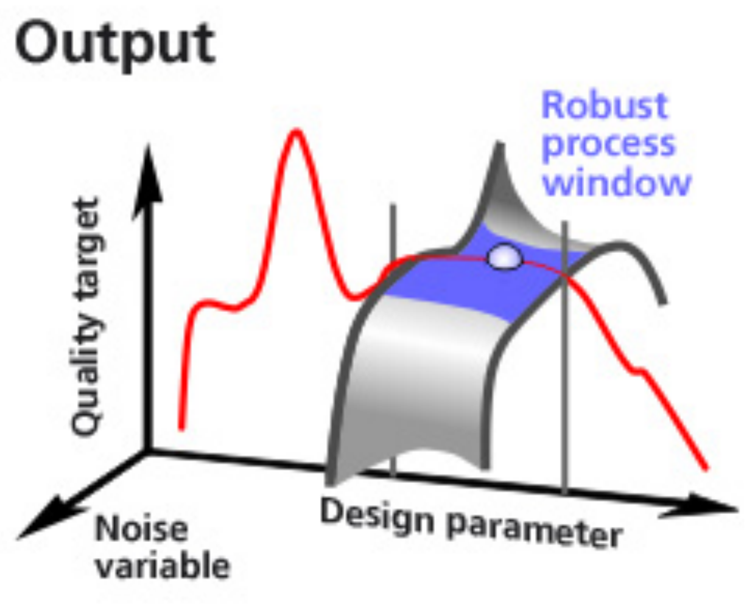


Visualization of a w2rap-contigger GFA for an *E. coli* dataset assembly
Rendered using Bandage (Wick R.R. et al., Bioinformatics, 31(20), 3350-3352)

Assembly process and tradeoffs



How much can we stretch the spring while trusting the results



Approaches for assembly

Paper assembly (?)

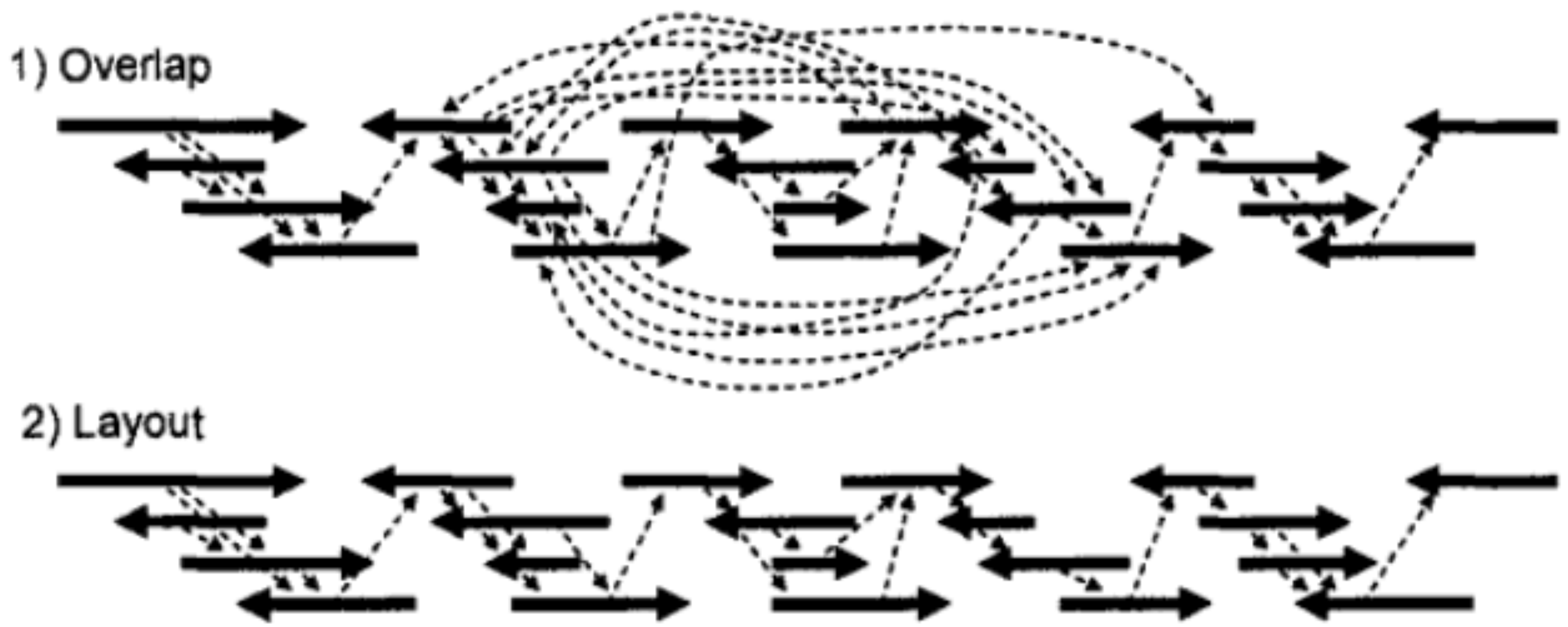
Overlap Layout Consensus

De Bruijn Graphs (DBG)

Approaches for assembly

Overlap Layout Consensus

Overlap - Layout - Consensus



Finding defining and finding overlaps is key.

The layout can be quite difficult.

3) Consensus

```
          CCTATG-TAGTCAGTCG
             ATGCTAGTCAG
               GCTAGTCGGTCGATCTACC
                     CAGTCGATCTGCCGGT
                   GTCAGTC-ATCTAC-GGTTAGCATTGC
Consensus CCTATGCTAGTCAGTCGATCTACCGGTTAGCATTGC
```

The consensus is constructed from the reads.

The method tracks every read.

Short kmer introduction

The kmers of a sequence are all possible k-size subsequences of a sequence

```
start
for each position in sequence:
    kmer = sequence[position:position+k]
stop
```

L-k+1 kmers in a sequence

ATCGATCACCTAGT 3-mers

```
$\boxed{ATC}$GATCACCTAGT   3-mers
ATC************   ATC
```

Earlham Institute

```
ATCGATCACCTAGT   3-mers
ATC***********    ATC
*TCG**********    TCG
```

```
ATCGATCACCTAGT   3-mers
ATC***********   ATC
*TCG**********   TCG
**CGA*********   CGA
```

```
ATCGATCACCTAGT   3-mers
ATC***********   ATC
*TCG**********   TCG
**CGA*********   CGA
***GAT********   GAT
```

```
ATCGATCACCTAGT   3-mers
ATC***********    ATC
*TCG**********    TCG
**CGA*********    CGA
***GAT********    GAT
****ATC*******    ATC
```

```
ATCGATCACCTAGT   3-mers
ATC***********   ATC
*TCG**********   TCG
**CGA*********   CGA
***GAT********   GAT
****ATC*******   ATC
*****TCA******   TCA
```

```
ATCGATCACCTAGT   3-mers
ATC***********   ATC
*TCG**********   TCG
**CGA*********   CGA
***GAT********   GAT
****ATC*******   ATC
*****TCA******   TCA
******CAC*****   CAC
```

```
ATCGATCACCTAGT  3-mers
ATC***********  ATC
*TCG**********  TCG
**CGA*********  CGA
***GAT********  GAT
****ATC*******  ATC
*****TCA******  TCA
******CAC*****  CAC
*******ACC****  ACC
```

```
ATCGATCACCTAGT   3-mers
ATC***********   ATC
*TCG**********   TCG
**CGA*********   CGA
***GAT********   GAT
****ATC*******   ATC
*****TCA******   TCA
******CAC*****   CAC
*******ACC****   ACC
********CCT***   CCT
```

Earlham Institute

```
ATCGATCACCTAGT   3-mers
ATC***********   ATC
*TCG**********   TCG
**CGA*********   CGA
***GAT********   GAT
****ATC*******   ATC
*****TCA******   TCA
******CAC*****   CAC
*******ACC****   ACC
********CCT***   CCT
*********CTA**   CTA
```

Earlham Institute

```
ATCGATCACCTAGT   3-mers
ATC***********   ATC
*TCG**********   TCG
**CGA*********   CGA
***GAT********   GAT
****ATC*******   ATC
*****TCA******   TCA
******CAC*****   CAC
*******ACC****   ACC
********CCT***   CCT
*********CTA**   CTA
**********TAG*   TAG
```

Earlham Institute

```
ATCGATCACCTAGT  3-mers
ATC***********   ATC
*TCG**********   TCG
**CGA*********   CGA
***GAT********   GAT
****ATC*******   ATC
*****TCA******   TCA
******CAC*****   CAC
*******ACC****   ACC
********CCT***   CCT
*********CTA**   CTA
**********TAG*   TAG
***********AGT   AGT
```

Earlham Institute

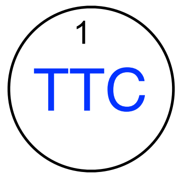
Back to DBG

Assembling a DBG

```
>seq1
TTCTAAGT
>seq2
CGATTCTA
```

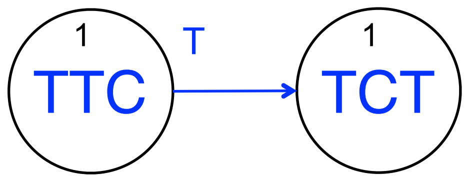
Assembling a DBG

```
>seq1
TTCTAAGT
>seq2
CGATTCTA
```



Assembling a DBG

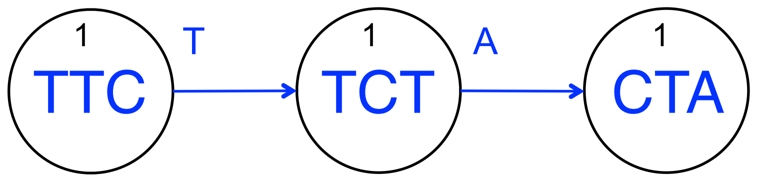
```
>seq1
TTCTAAGT
>seq2
CGATTCTA
```



Earlham Institute

Assembling a DBG

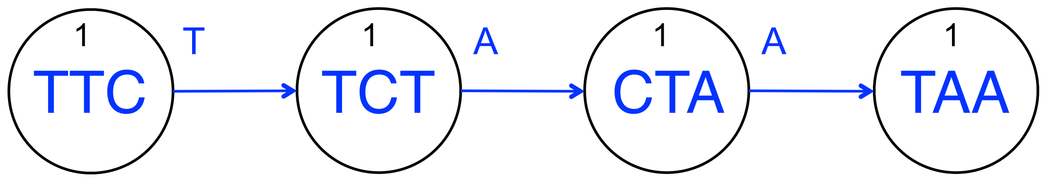
```
>seq1
TTCTAAGT
>seq2
CGATTCTA
```



Earlham Institute

Assembling a DBG

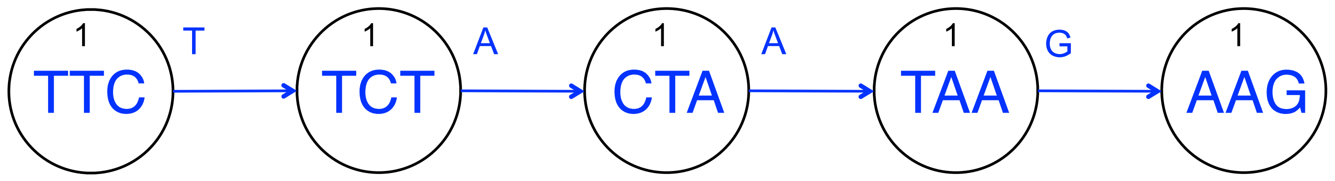
```
>seq1
TTCTAAGT
>seq2
CGATTCTA
```



Earlham Institute

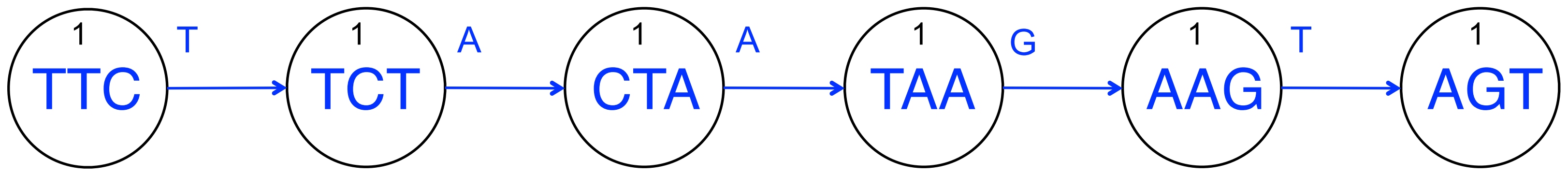
Assembling a DBG

```
>seq1
TTCTAAGT
>seq2
CGATTCTA
```



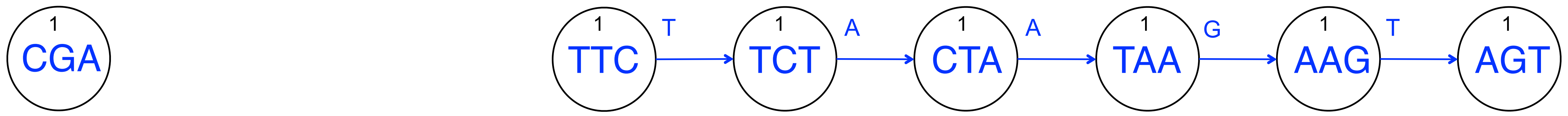
Assembling a DBG

```
>seq1
TTCTAAGT
>seq2
CGATTCTA
```



Assembling a DBG

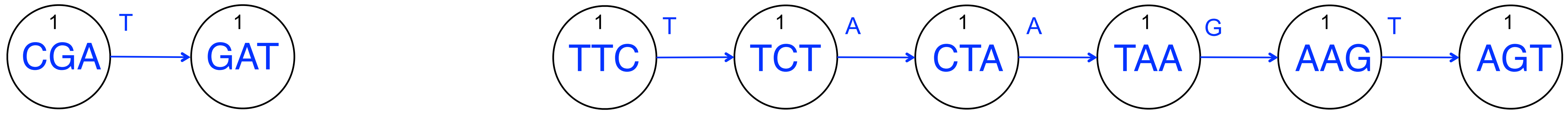
```
>seq1
TTCTAAGT
>seq2
CGATTCTA
```



Earlham Institute

Assembling a DBG

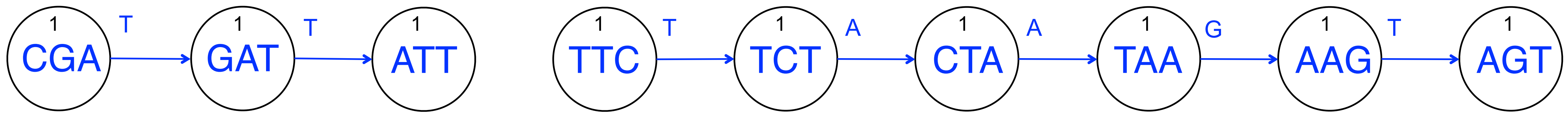
```
>seq1
TTCTAAGT
>seq2
CGATTCTA
```



Earlham Institute

Assembling a DBG

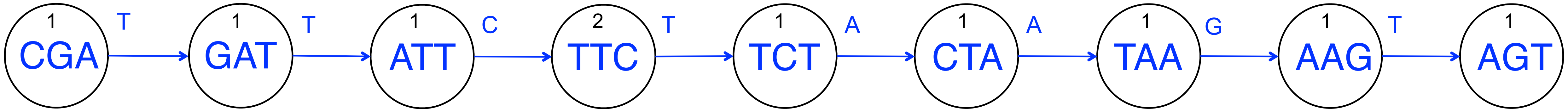
```
>seq1
TTCTAAGT
>seq2
CGATTCTA
```



Earlham Institute

Assembling a DBG

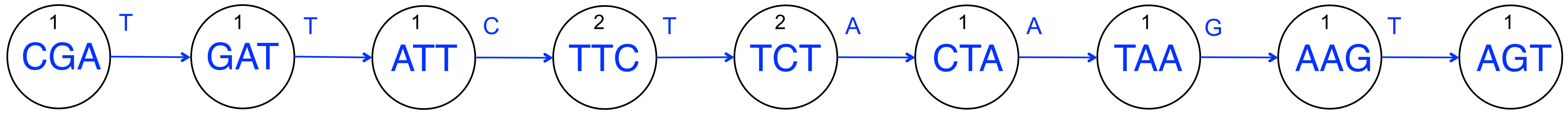
```
>seq1
TTCTAAGT
>seq2
CGATTCTA
```



Earlham Institute

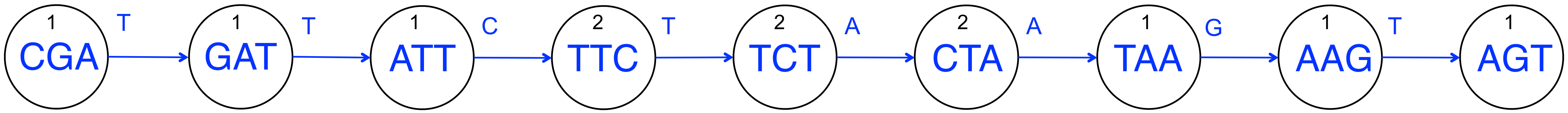
Assembling a DBG

```
>seq1
TTCTAAGT
>seq2
CGATTCTA
```



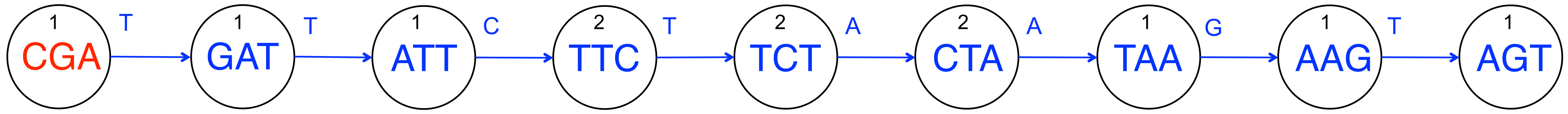
Assembling a DBG

```
>seq1
TTCTAAGT
>seq2
CGATTCTA
```



Assembling a DBG

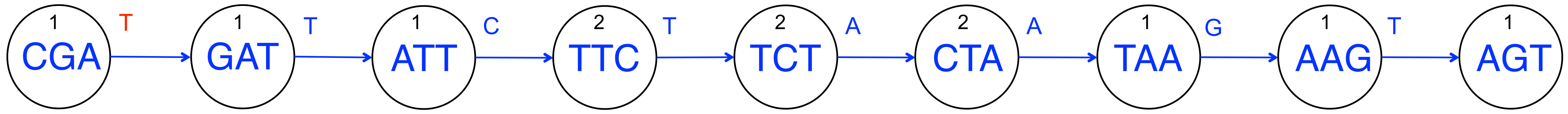
```
>seq1
TTCTAAGT
>seq2
CGATTCTA
```



CGA

Assembling a DBG

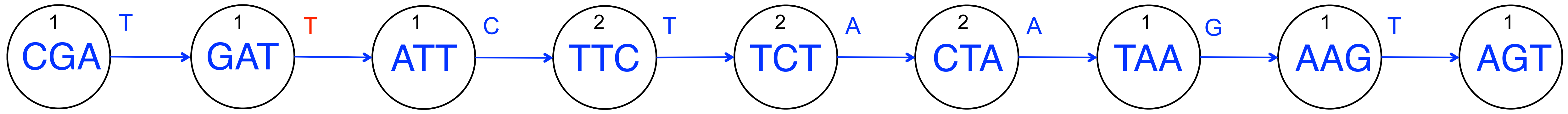
```
>seq1
TTCTAAGT
>seq2
CGATTCTA
```



CGAT

Assembling a DBG

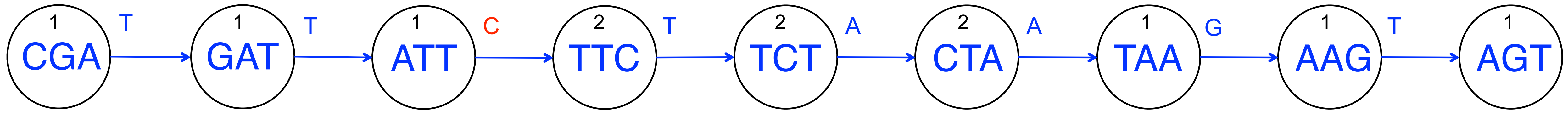
```
>seq1
TTCTAAGT
>seq2
CGATTCTA
```



CGATT

Assembling a DBG

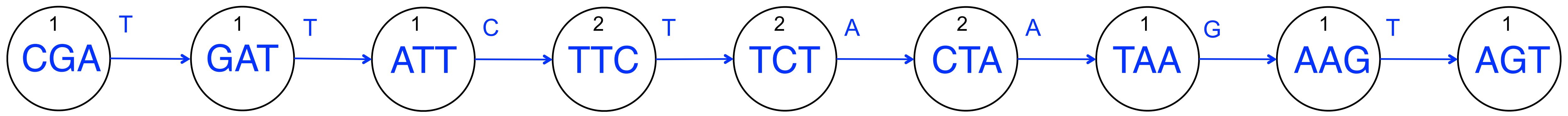
```
>seq1
TTCTAAGT
>seq2
CGATTCTA
```



CGATTC

Assembling a DBG

```
>seq1
TTCTAAGT
>seq2
CGATTCTA
```

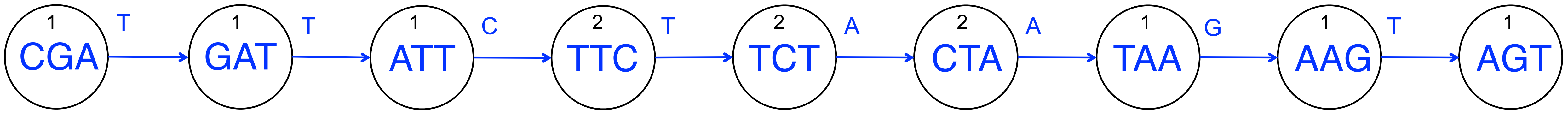


CGATTCTAAGT

Assembling a DBG

```
>seq1
TTCTAAGT
>seq2
CGATTCTA
```

```
>seq3
CGATTGTAAGT
```

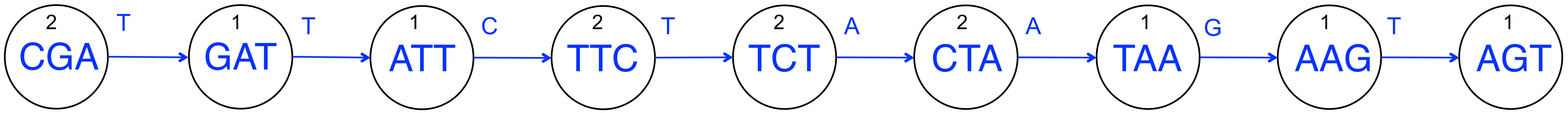


CGATTCTAAGT

Assembling a DBG

```
>seq1
TTCTAAGT
>seq2
CGATTCTA
```

```
>seq3
CGATTGTAAGT
```

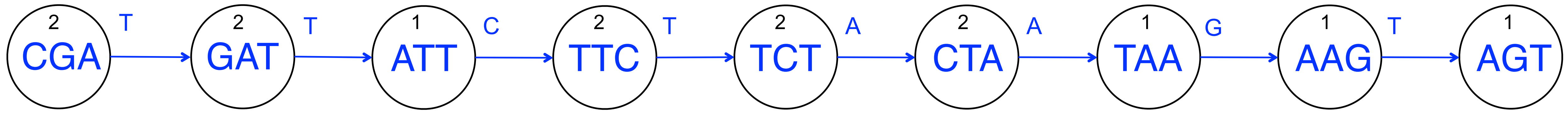


Earlham Institute

Assembling a DBG

```
>seq1
TTCTAAGT
>seq2
CGATTCTA
```

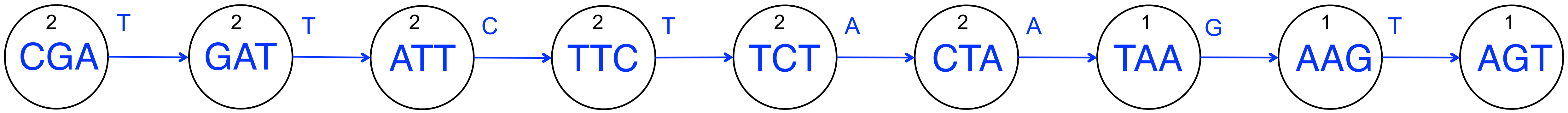
```
>seq3
CGATTGTAAGT
```



Assembling a DBG

```
>seq1
TTCTAAGT
>seq2
CGATTCTA
```

```
>seq3
CGATTGTAAGT
```

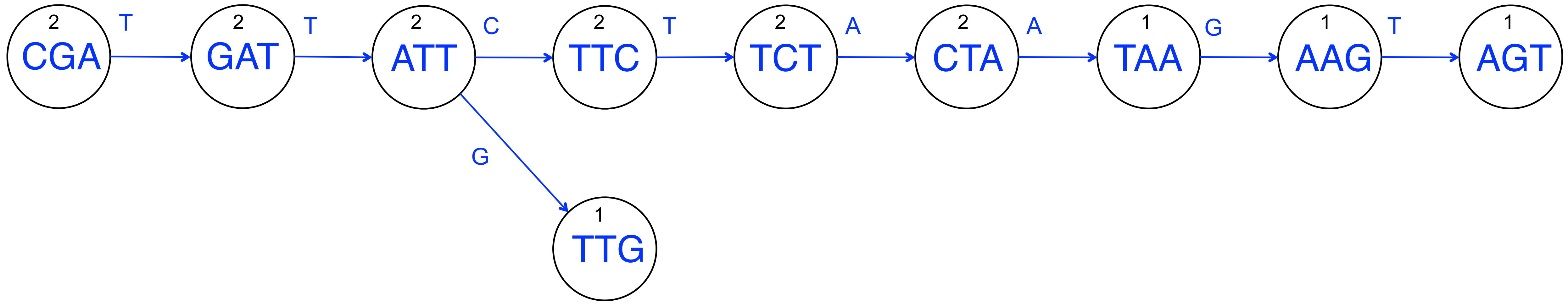


Earlham Institute

Assembling a DBG

```
>seq1
TTCTAAGT
>seq2
CGATTCTA
```

```
>seq3
CGATTGTAAGT
```

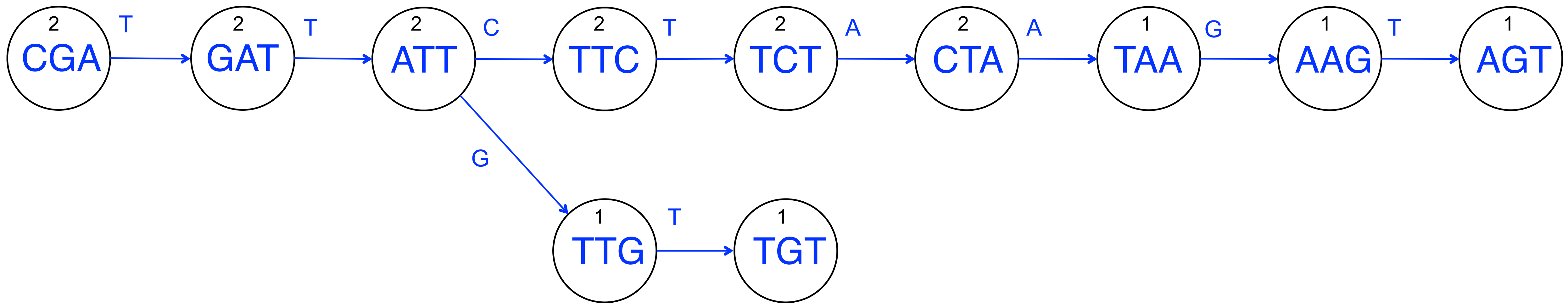


Earlham Institute

Assembling a DBG

```
>seq1
TTCTAAGT
>seq2
CGATTCTA
```

```
>seq3
CGATTGTAAGT
```

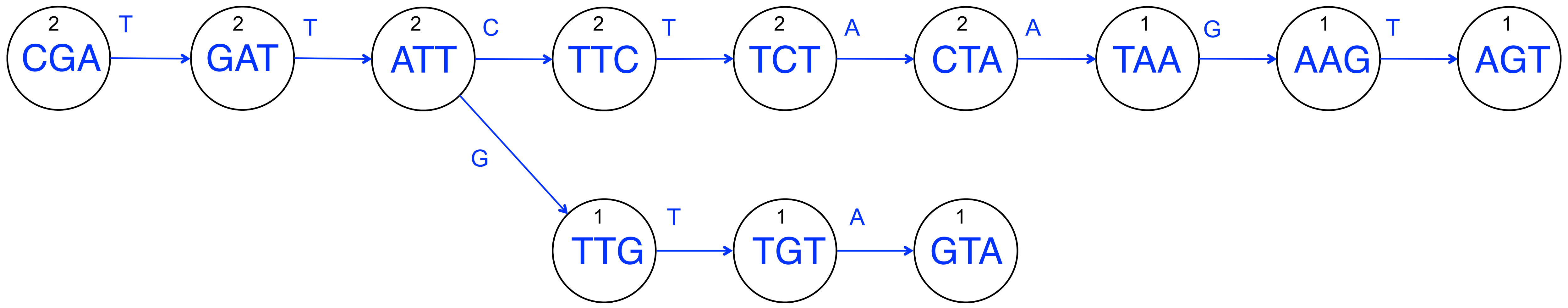


Earlham Institute

Assembling a DBG

```
>seq1
TTCTAAGT
>seq2
CGATTCTA
```

```
>seq3
CGATTGTAAGT
```

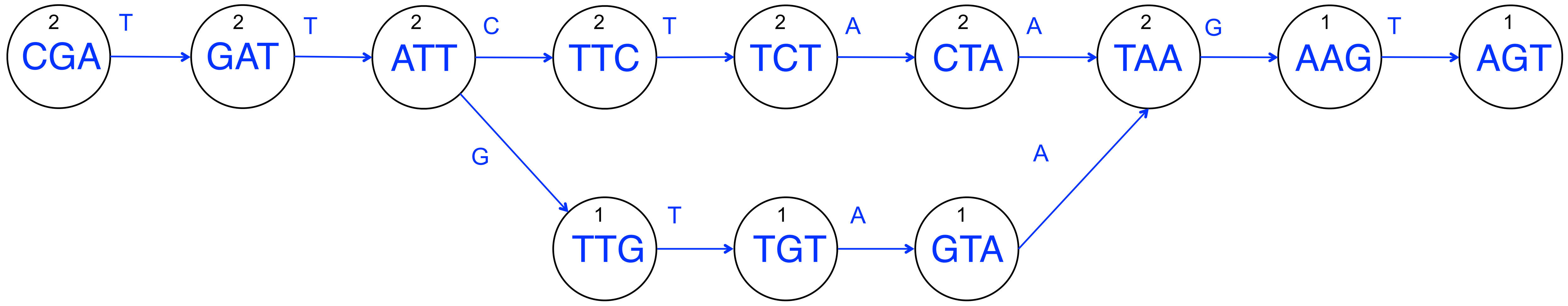


Earlham Institute

Assembling a DBG

```
>seq1
TTCTAAGT
>seq2
CGATTCTA
```

```
>seq3
CGATTGTAAGT
```

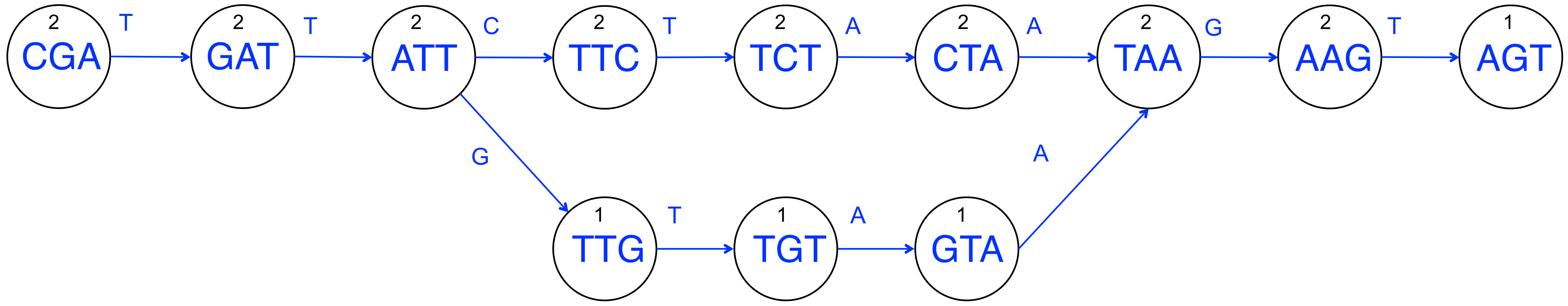


Earlham Institute

Assembling a DBG

```
>seq1
TTCTAAGT
>seq2
CGATTCTA
```

```
>seq3
CGATTGTAAGT
```

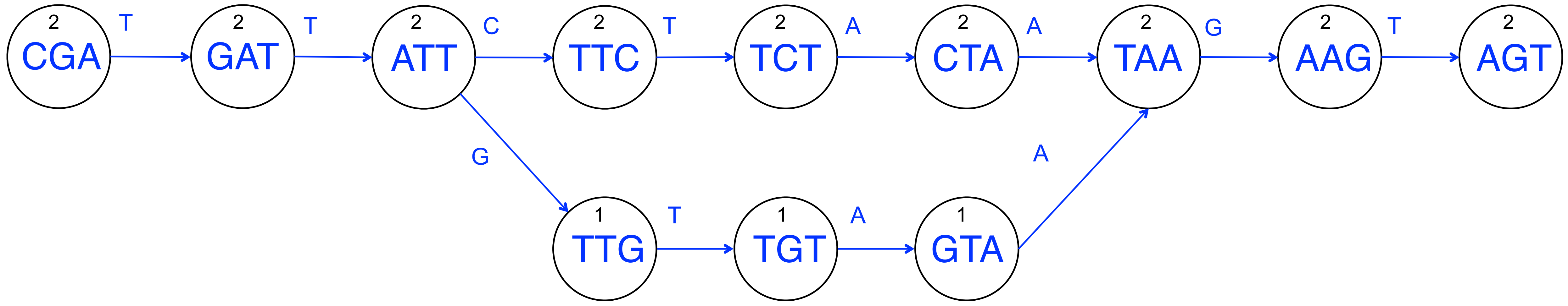


Earlham Institute

Assembling a DBG

```
>seq1
TTCTAAGT
>seq2
CGATTCTA
```

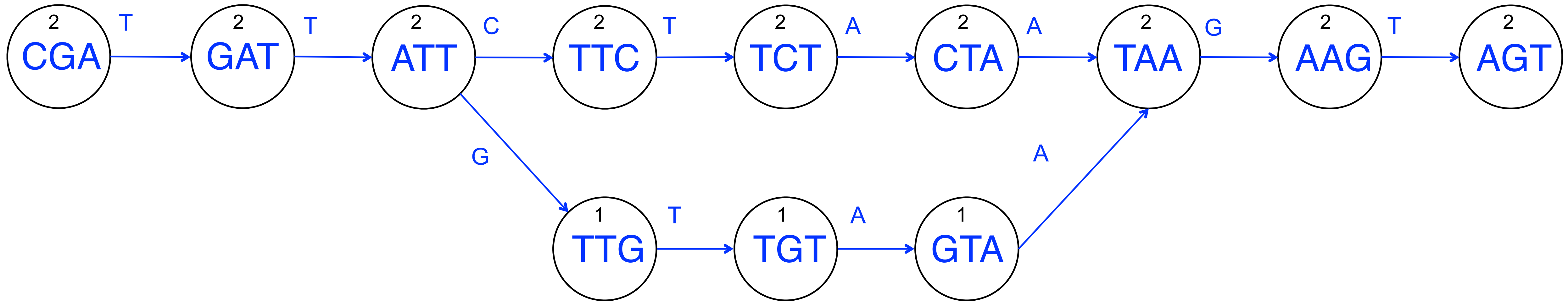
```
>seq3
CGATTGTAAGT
```



Assembling a DBG

```
>seq1
TTCTAAGT
>seq2
CGATTCTA
```

```
>seq3
CGATTGTAAGT
```

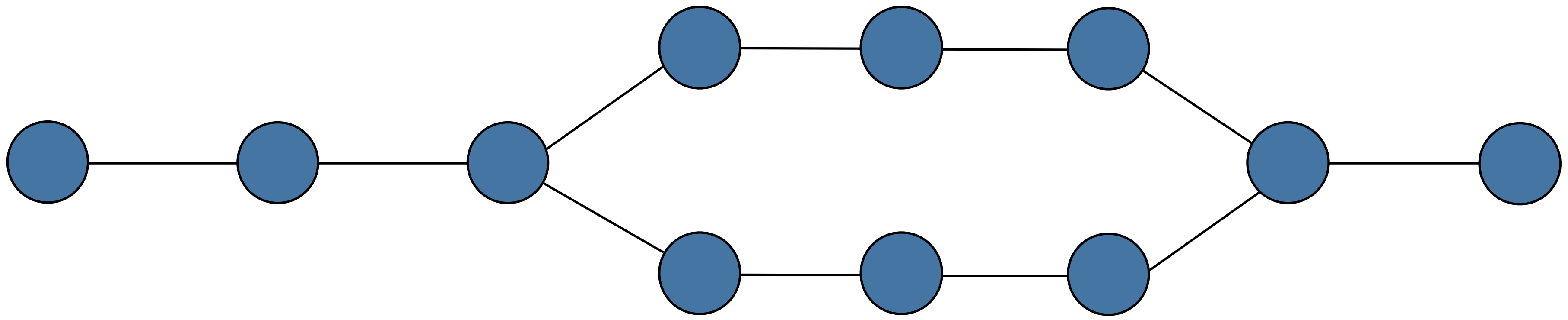


CGATTCTAAGT

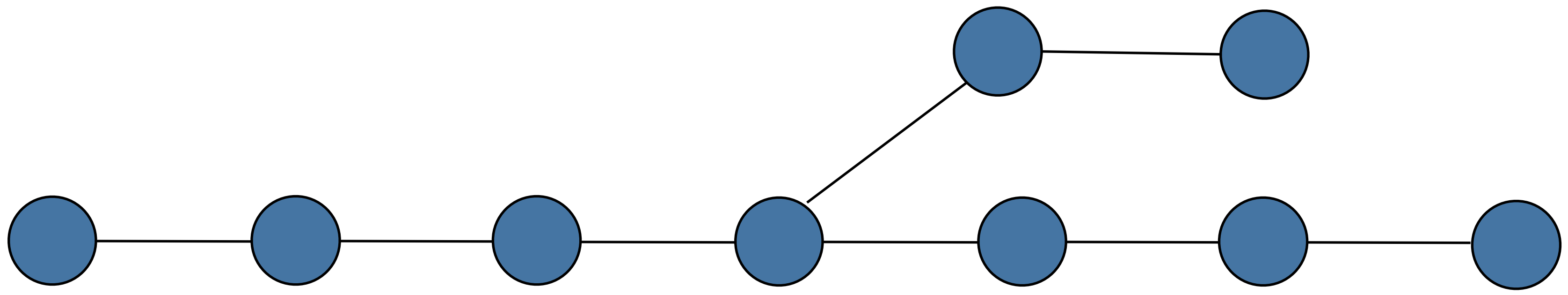
CGATTGTAAGT

Common structures

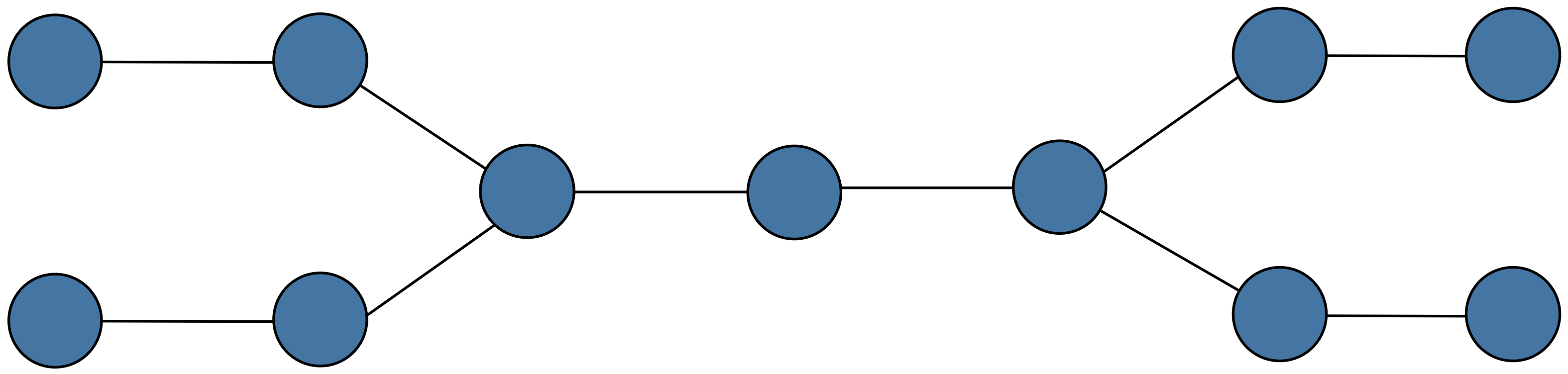
Polymorphism



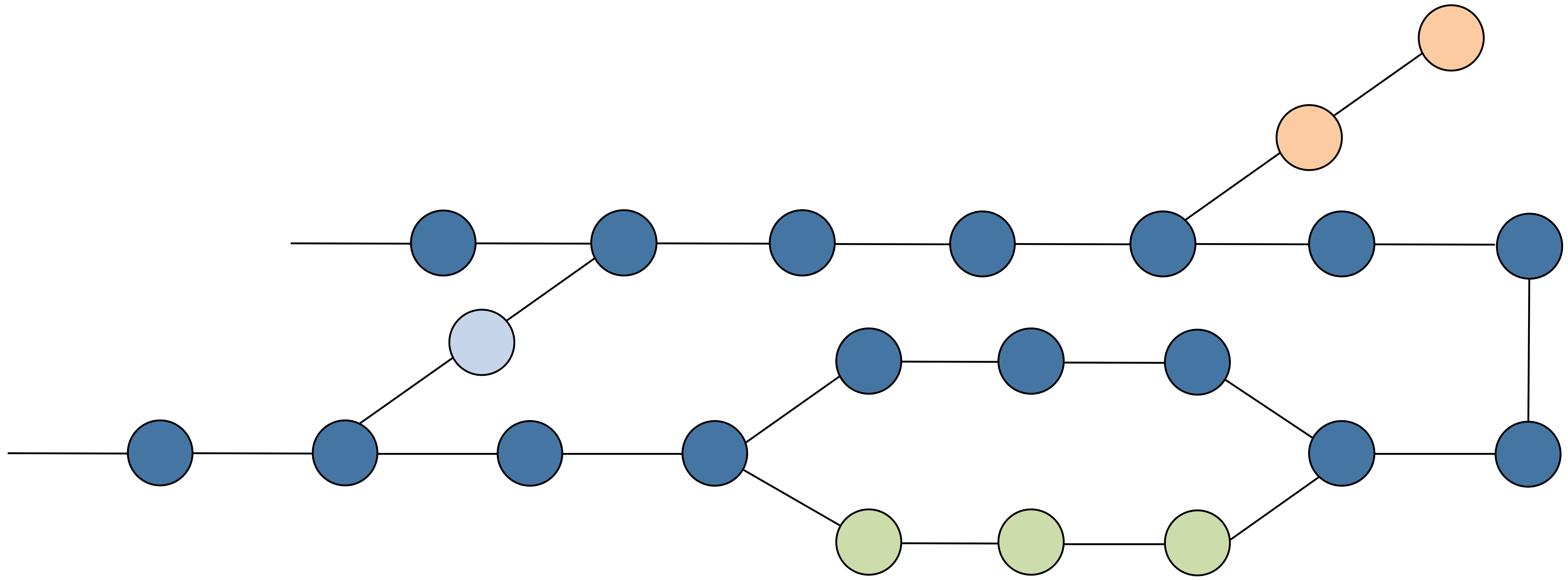
Errant base call



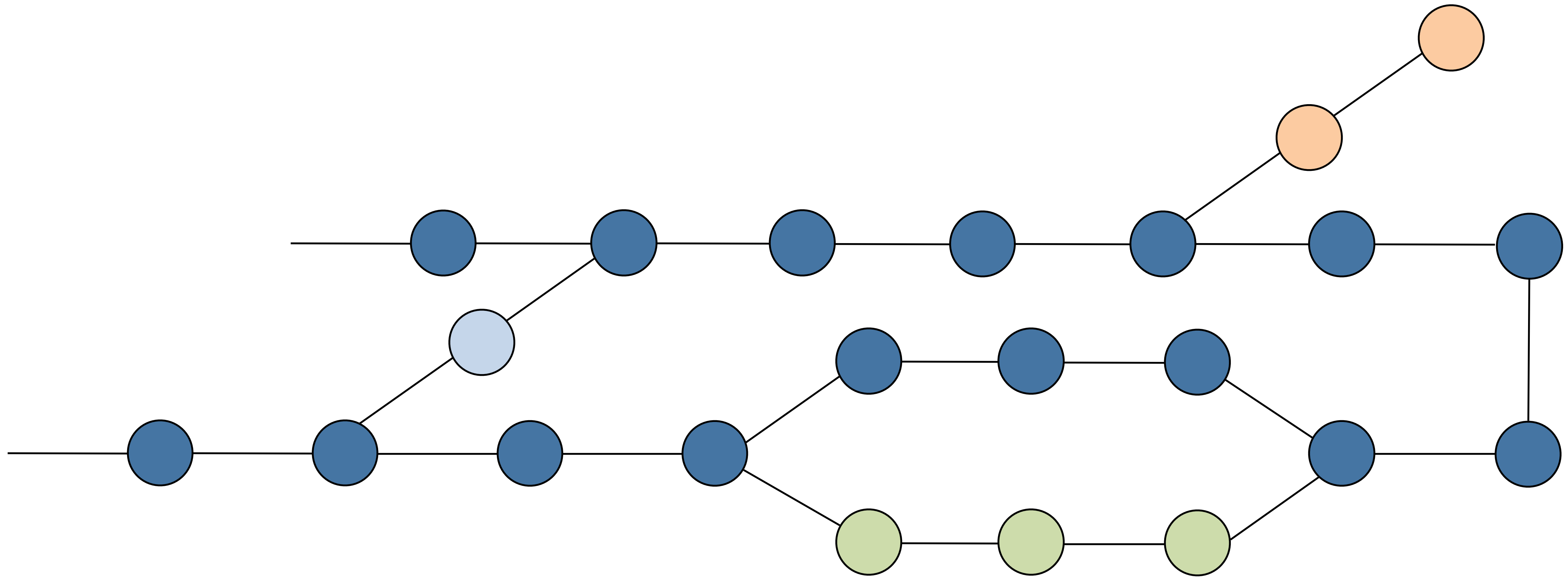
Repeating element



Cleaning the graph

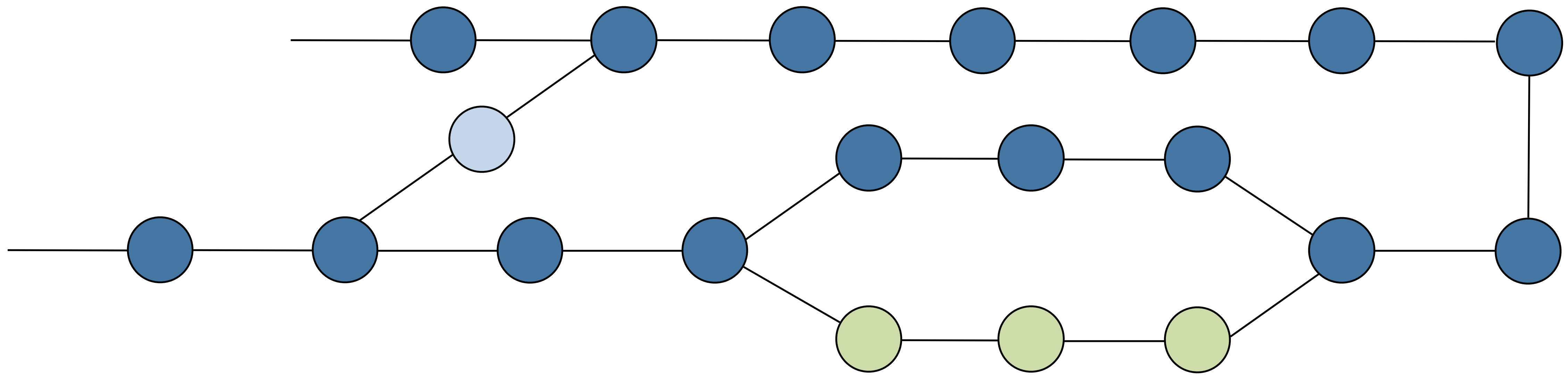


Cleaning the graph



● Clip tips

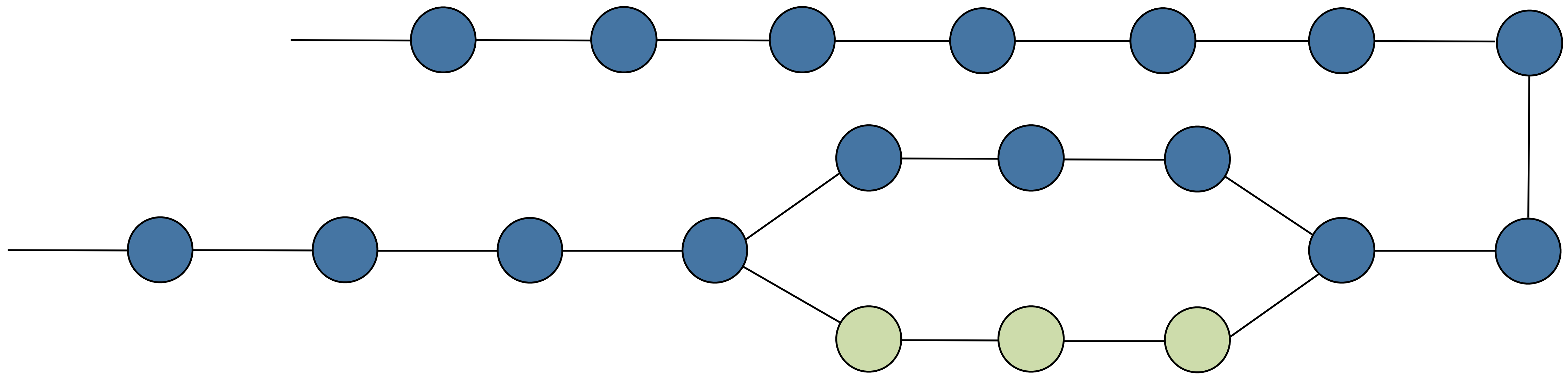
Cleaning the graph



● Clip tips

● Remove low coverage nodes

Cleaning the graph

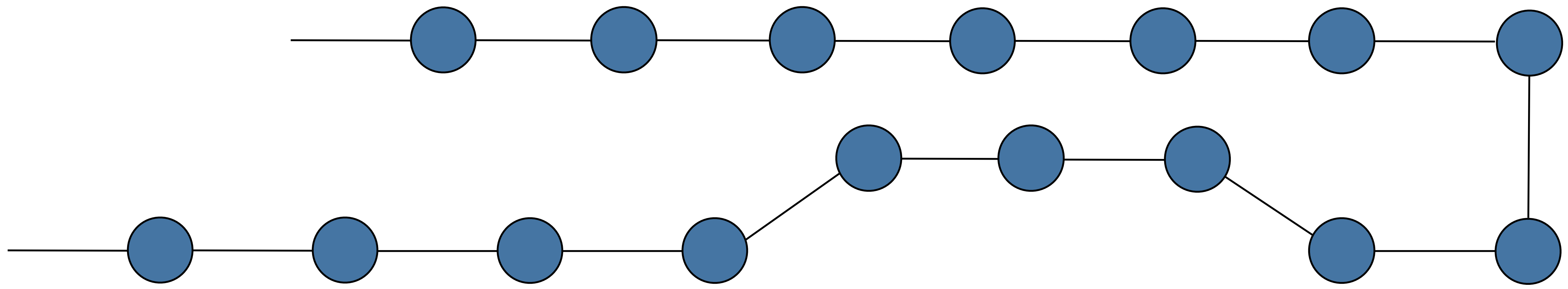


🟧 Clip tips

🟦 Remove low coverage nodes

🟩 Remove bubbles

Cleaning the graph

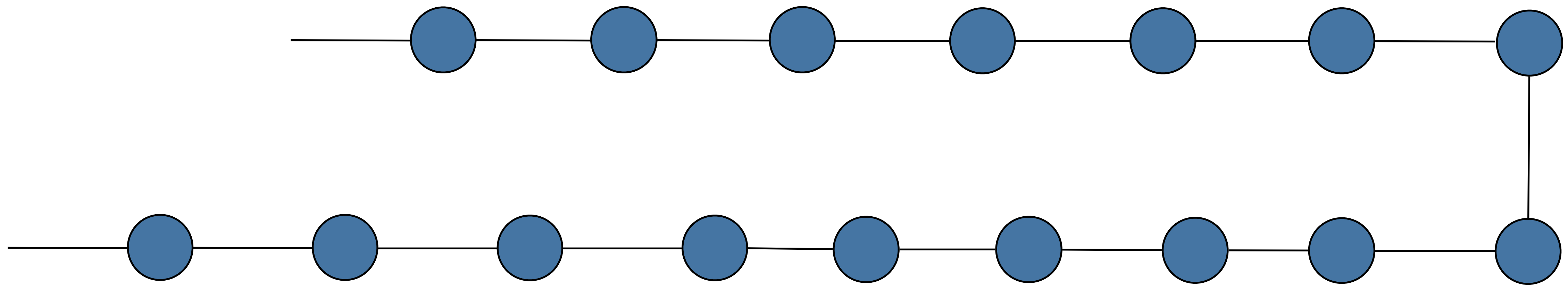


● Clip tips

● Remove low coverage nodes

● Remove bubbles

Cleaning the graph



🟧 Clip tips

🟦 Remove low coverage nodes

🟩 Remove bubbles

DBG assemblers work with frequencies

Earlham Institute

Assembly QC

- How do you QC your assemblies?

Back to Kmers

```
start
for each position in sequence:
    kmer = sequence[position:position+k]
stop
```

ATCGATCACCTAGT 3-mers

$\boxed{\text{ATC}}$GATCACCTAGT

ATC***********

3-mers

ATC

Earlham Institute

```
ATCGATCACCTAGT   3-mers
ATC***********   ATC
*TCG**********   TCG
```

```
ATCGATCACCTAGT   3-mers
ATC***********   ATC
*TCG**********   TCG
**CGA*********   CGA
```

```
ATCGATCACCTAGT    3-mers
ATC***********    ATC
*TCG**********    TCG
**CGA*********    CGA
***GAT********    GAT
```

```
ATCGATCACCTAGT   3-mers
ATC***********    ATC
*TCG**********    TCG
**CGA*********    CGA
***GAT********    GAT
****ATC*******    ATC
```

```
ATCGATCACCTAGT  3-mers
ATC***********  ATC
*TCG**********  TCG
**CGA*********  CGA
***GAT********  GAT
****ATC*******  ATC
*****TCA******  TCA
```

```
ATCGATCACCTAGT   3-mers
ATC***********   ATC
*TCG**********   TCG
**CGA*********   CGA
***GAT********   GAT
****ATC*******   ATC
*****TCA******   TCA
******CAC*****   CAC
```

```
ATCGATCACCTAGT   3-mers
ATC***********   ATC
*TCG**********   TCG
**CGA*********   CGA
***GAT********   GAT
****ATC*******   ATC
*****TCA******   TCA
******CAC*****   CAC
*******ACC****   ACC
```

Earlham Institute

```
ATCGATCACCTAGT   3-mers
ATC***********   ATC
*TCG**********   TCG
**CGA*********   CGA
***GAT********   GAT
****ATC*******   ATC
*****TCA******   TCA
******CAC*****   CAC
*******ACC****   ACC
********CCT***   CCT
```

Earlham Institute

```
ATCGATCACCCTAGT   3-mers
ATC************   ATC
*TCG***********   TCG
**CGA**********   CGA
***GAT*********   GAT
****ATC********   ATC
*****TCA*******   TCA
******CAC******   CAC
*******ACC*****   ACC
********CCT****   CCT
*********CTA**    CTA
```

Earlham Institute

```
ATCGATCACCTAGT   3-mers
ATC***********   ATC
*TCG**********   TCG
**CGA*********   CGA
***GAT********   GAT
****ATC*******   ATC
*****TCA******   TCA
******CAC*****   CAC
*******ACC****   ACC
********CCT***   CCT
*********CTA**   CTA
**********TAG*   TAG
```

```
ATCGATCACCTAGT   3-mers
ATC***********    ATC
*TCG**********    TCG
**CGA*********    CGA
***GAT********    GAT
****ATC*******    ATC
*****TCA******    TCA
******CAC*****    CAC
*******ACC****    ACC
********CCT***    CCT
*********CTA**    CTA
**********TAG*    TAG
***********AGT    AGT
```

Types of kmers

3-mers

ATC
TCG
CGA
GAT
ATC
TCA
CAC
ACC
CCT
CTA
TAG
AGT

- Total: total amount of kmers extracted from a sequence.
- Distinct: all different deduplicated kmers extracted from a sequence
- Unique: total of kmers that appear once in the counted set

kmer	Total #	Distinct #	Unique #
ATC	2	1	0
TCG	1	1	1
CGA	1	1	1
GAT	1	1	1
TCA	1	1	1
CAC	1	1	1
ACC	1	1	1
CCT	1	1	1
CTA	1	1	1
TAG	1	1	1
AGT	1	1	1
Total	**12**	**11**	**10**

Reverse complement and canonical

The canonical form of each kmer is the lexicographically smaller of the forward and reverse-complement representations.

kmer	Revcomp	Distinct #
ATC	GAT	ATC
TCG	CGA	CGA
CGA	TCG	CGA
GAT	ATC	ATC
ATC	GAT	ATC
TCA	TGA	TCA
CAC	GTG	CAC
ACC	GGT	ACC
CCT	AAG	AGG
CTA	TAG	CTA
TAG	CTA	CTA
AGT	ACT	ACT

Earlham Institute

Canonical count effect in types

kmer	Total #	Distinct #	Unique #
ATC	3	1	0
CGA	2	1	0
TCA	1	1	1
CAC	1	1	1
ACC	1	1	1
AGG	1	1	1
CTA	2	1	0
ACT	1	1	1
Total	**12**	**8**	**5**

Canonical

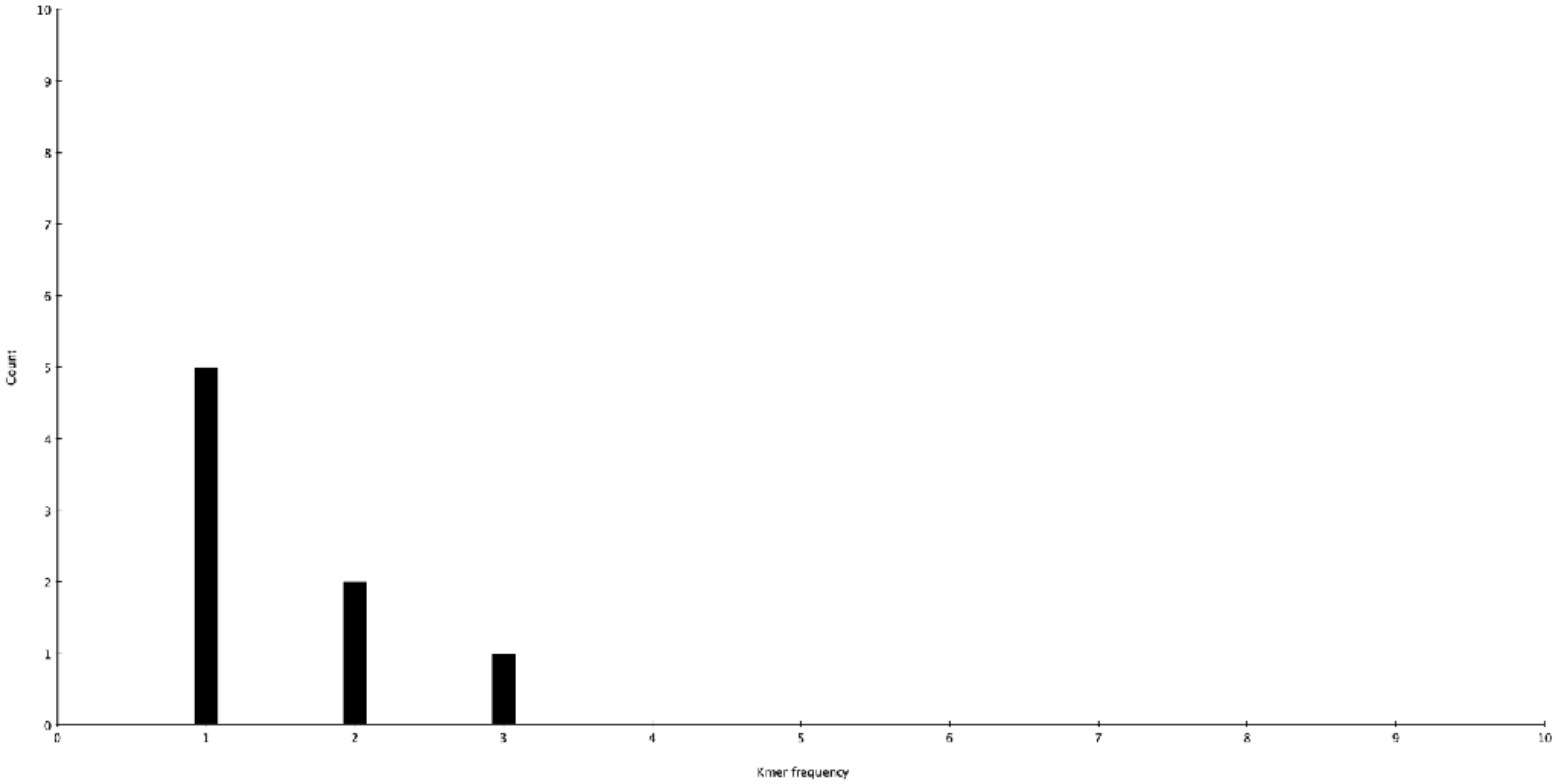
ATC
CGA
CGA
ATC
ATC
TCA
CAC
ACC
AGG
CTA
CTA
ACT

kmer	Total #	Distinct #	Unique #
ATC	3	1	0
CGA	2	1	0
TCA	1	1	1
CAC	1	1	1
ACC	1	1	1
AGG	1	1	1
CTA	2	1	0
ACT	1	1	1
Total	**12**	**8**	**5**

Frequency	#
1	5
2	2
3	1

Earlham Institute

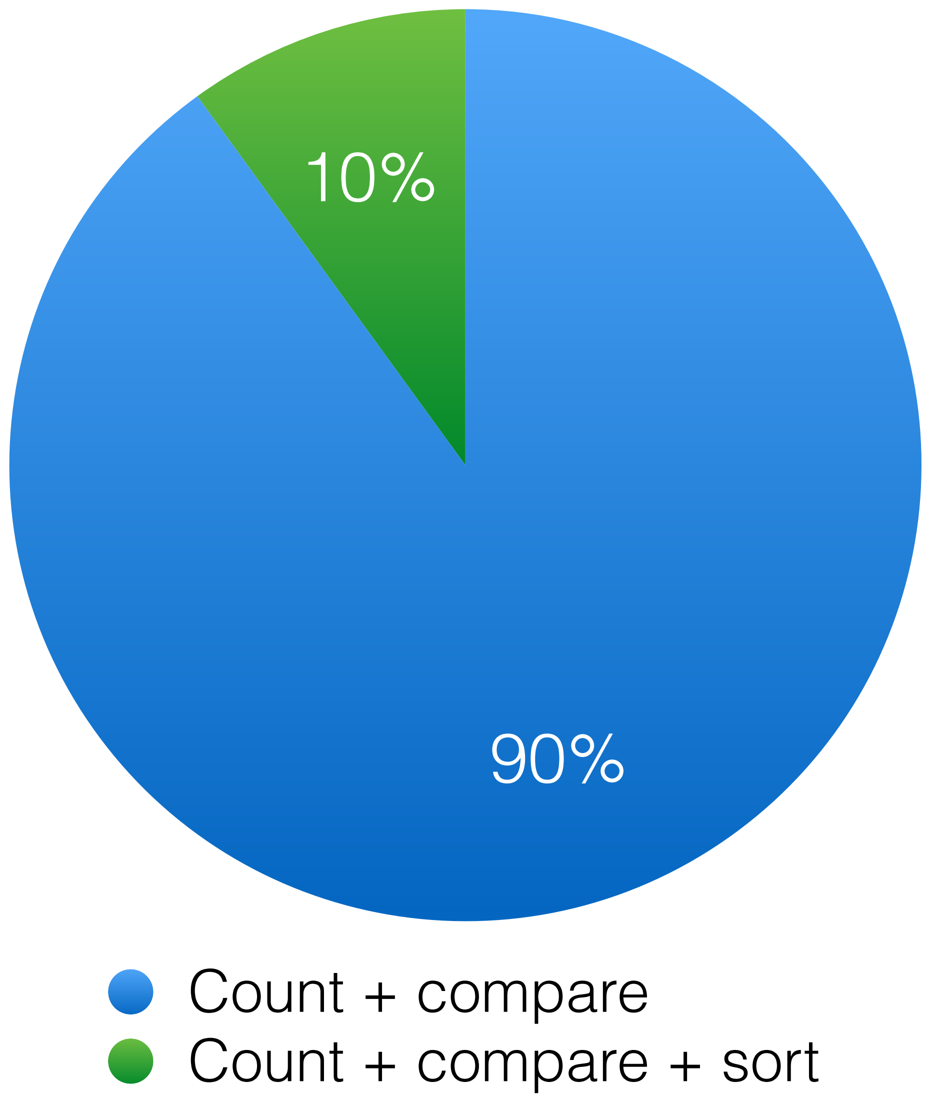
Kmer spectra

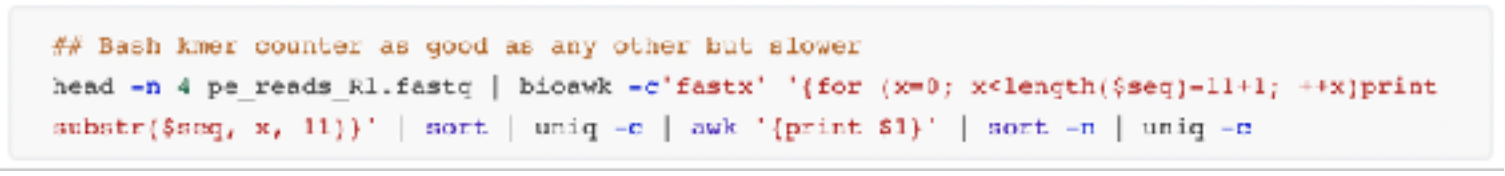


Earlham Institute

Kmer spectra

Counting Kmers is easy… anyone can count





```
## Bash kmer counter as good as any other but slower
head -n 4 pe_reads_R1.fastq | bioawk -c'fastx' '{for (x=0; x<length($seq)-11+1; ++x)print
substr($seq, x, 11)}' | sort | uniq -c | awk '{print $1}' | sort -n | uniq -c
```

Any software that can count and compare is good

https://github.com/TGAC/KAT

A bit of kmer wrangling with KAT to get confortable

PhiX-174

5386 bp

https://www.ncbi.nlm.nih.gov/nuccore/NC_001422.1?report=fasta

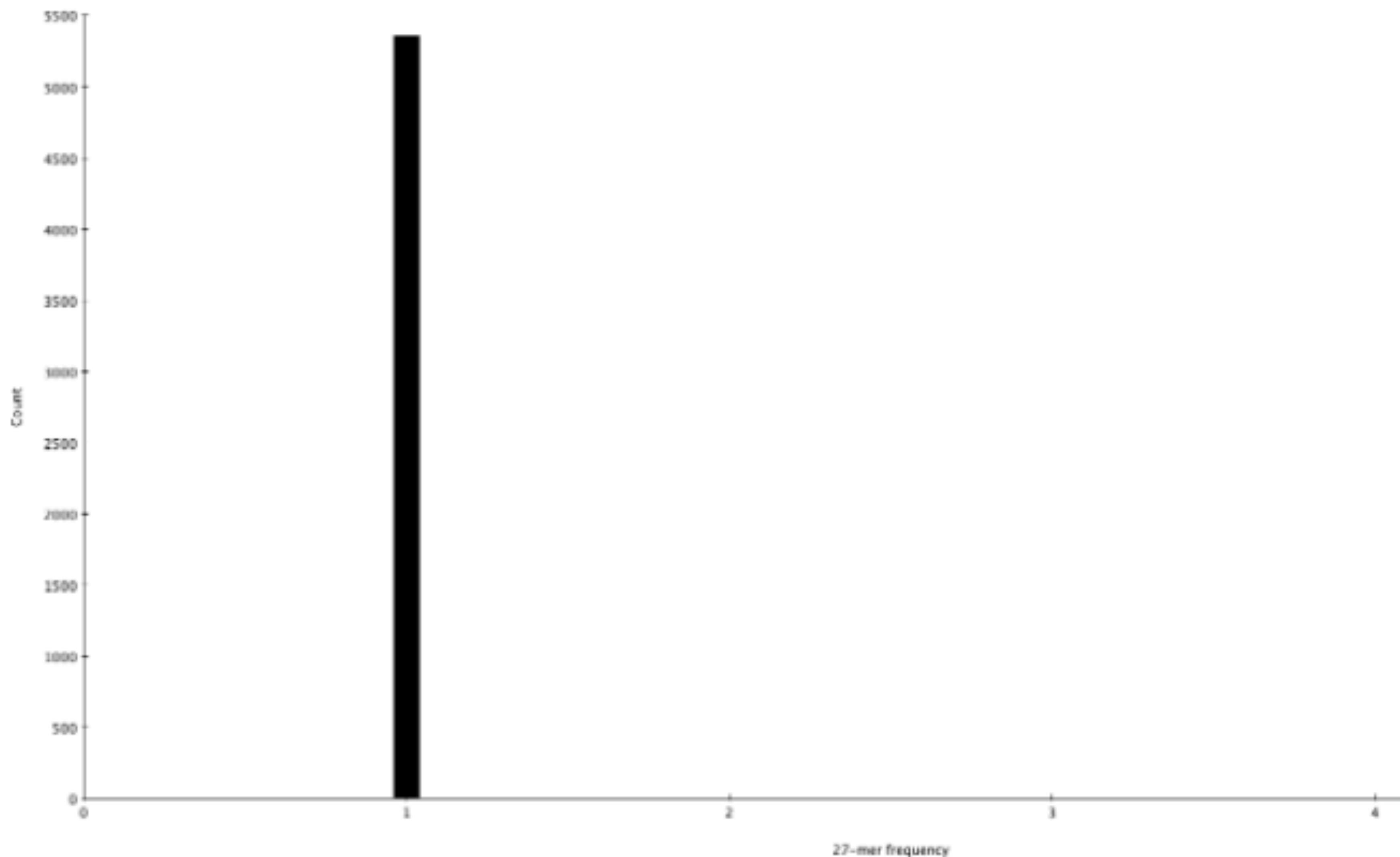
Create a kmer spectra for the phiX genome

```
kat hist --help
```

KAT

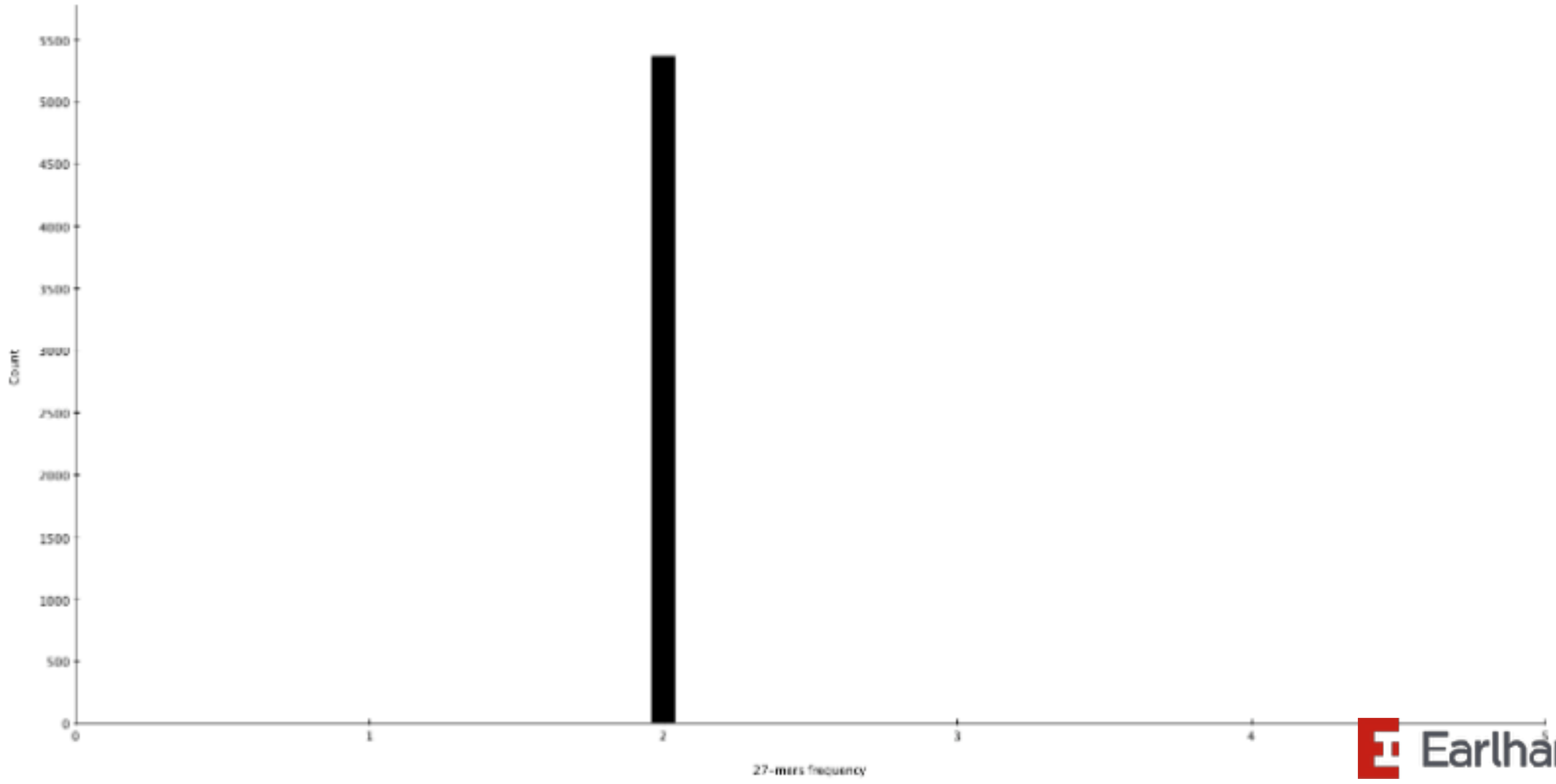
```
$ kat hist -o phiX.hist phiX.fasta
```

```
# Title:27-mer spectra for: phiX.fasta
# XLabel:27-mer frequency
# YLabel:# distinct 27-mers
# Kmer value:27
# Input 1: phiX.fasta
###
1 5360
2 0
3 0
4 0
...
```

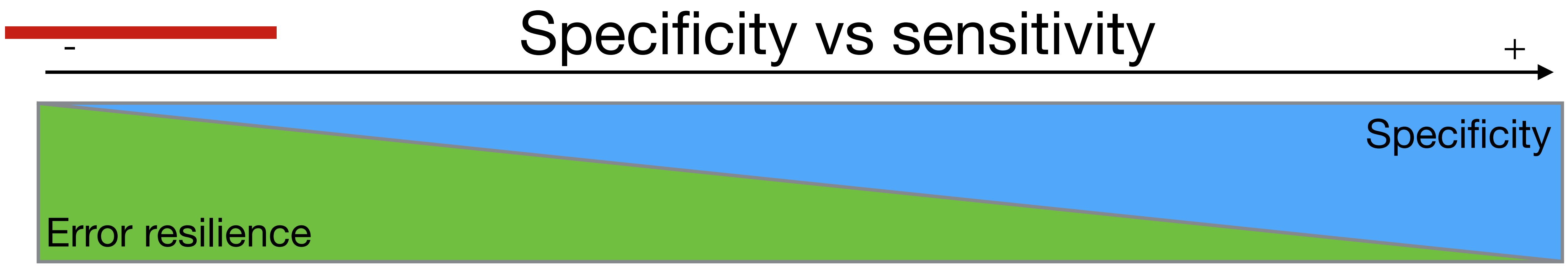


```
$ cat phiX.fasta phiX.fasta > phiX_twice.fasta
$ kat hist -o phiX_twice.hist phiX_twice.fasta
```

```
# Title:27-mer spectra for: phiX_twice.fasta
# XLabel:27-mer frequency
# YLabel:# distinct 27-mers
# Kmer value:27
# Input 1:phiX_twice.fasta
###
1 0
2 5360
3 0
4 0
...
```



Size of K

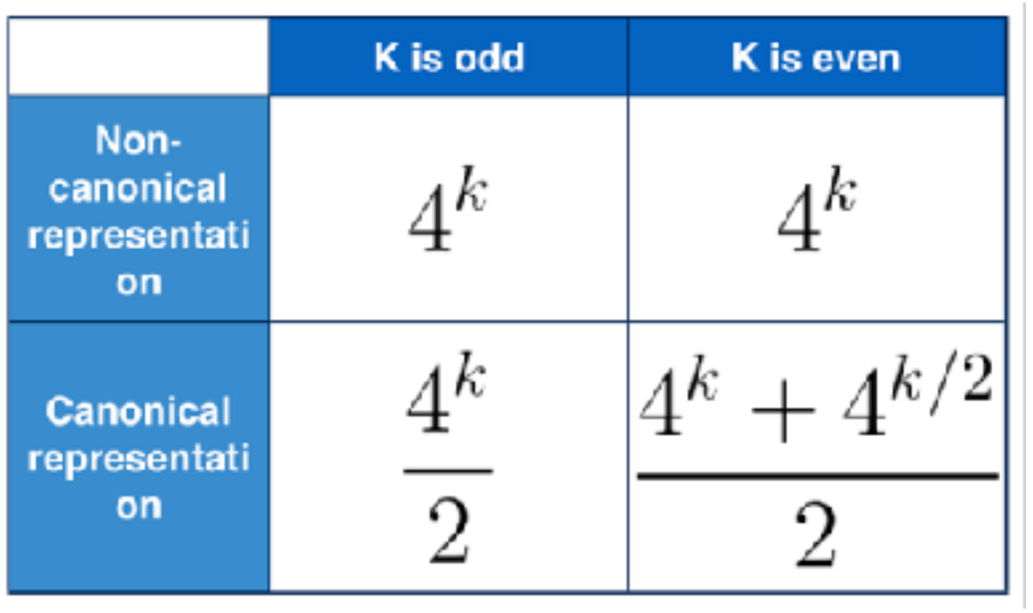


Less proportion of the kmers affected by variation

Ability to detect differences

Alphabet = {A, C, T, G}

Size of the universe ? How many kmers of size K exist?

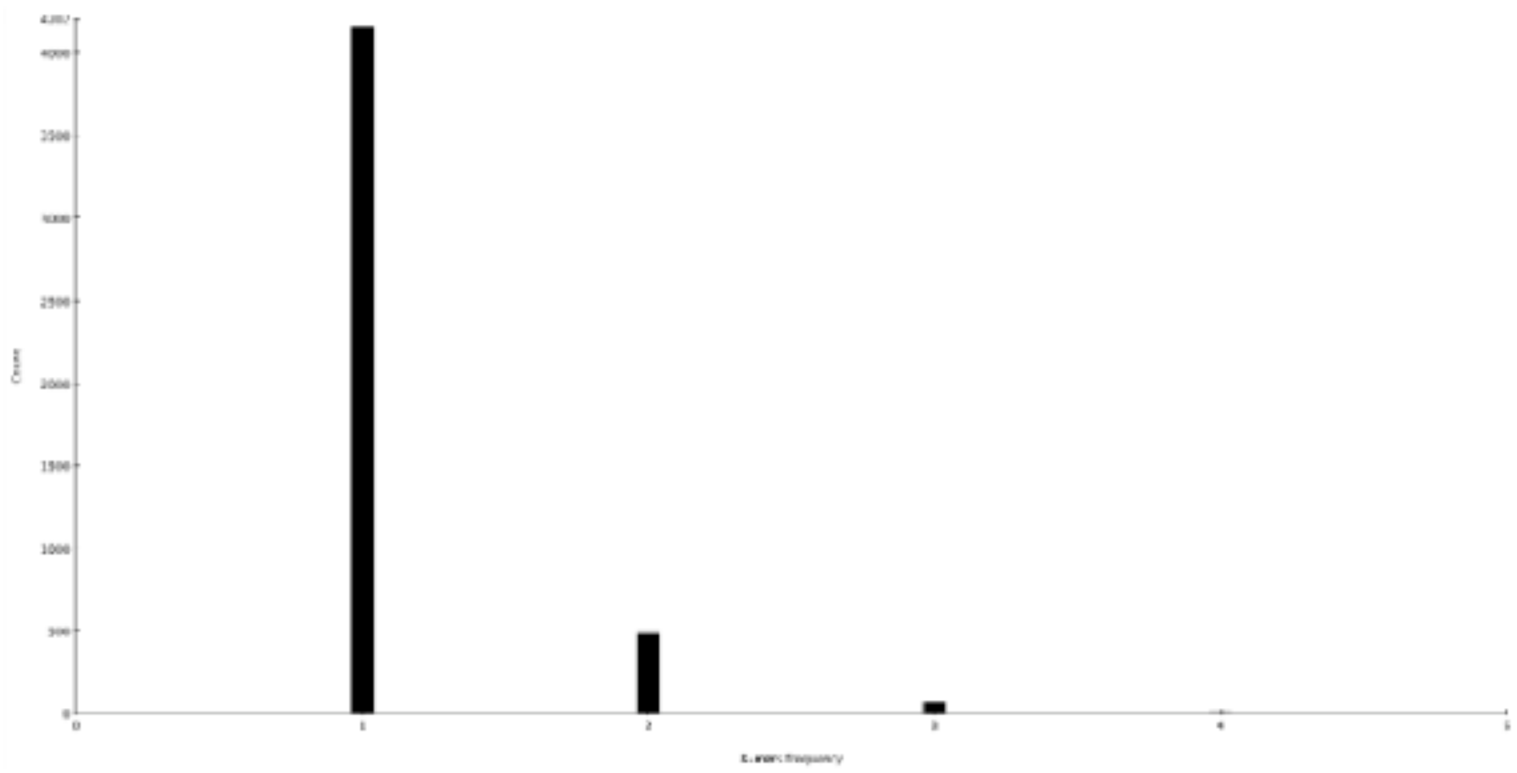


	K is odd	K is even
Non-canonical representation	4^k	4^k
Canonical representation	$\frac{4^k}{2}$	$\frac{4^k + 4^{k/2}}{2}$

Calculate phiX spectras at k size -> 7/8/9/10/15 (groups)

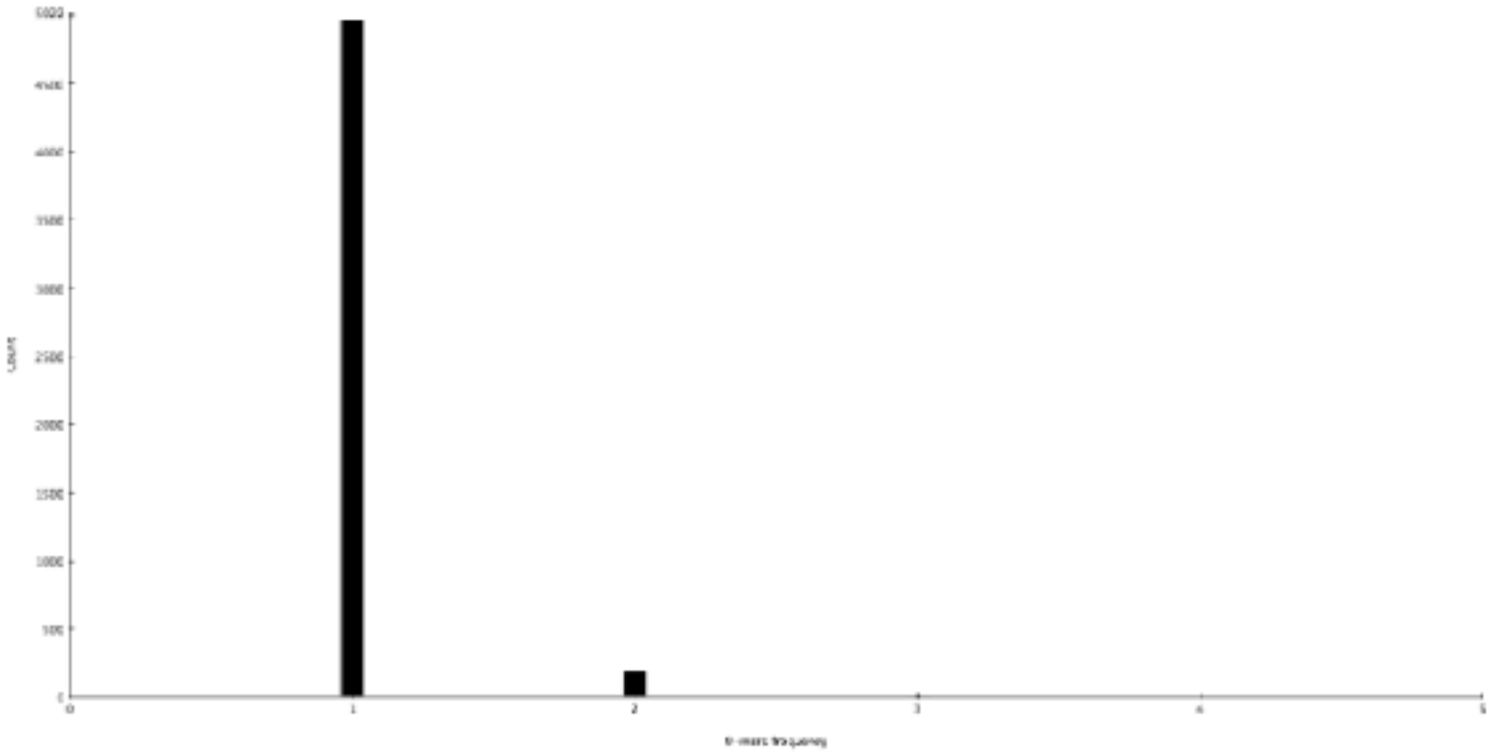
```
$ kat hist -o phiX_8mer.hist -m 8 phiX.fasta
```

```
# Title:8-mer spectra for: phiX.fasta
# XLabel:8-mer frequency
# YLabel:# distinct 8-mers
# Kmer value:8
# Input 1:phiX.fasta
###
1 4159
2 491
3 67
4 8
5 1
6 0
7 0
8 0
9 0
```



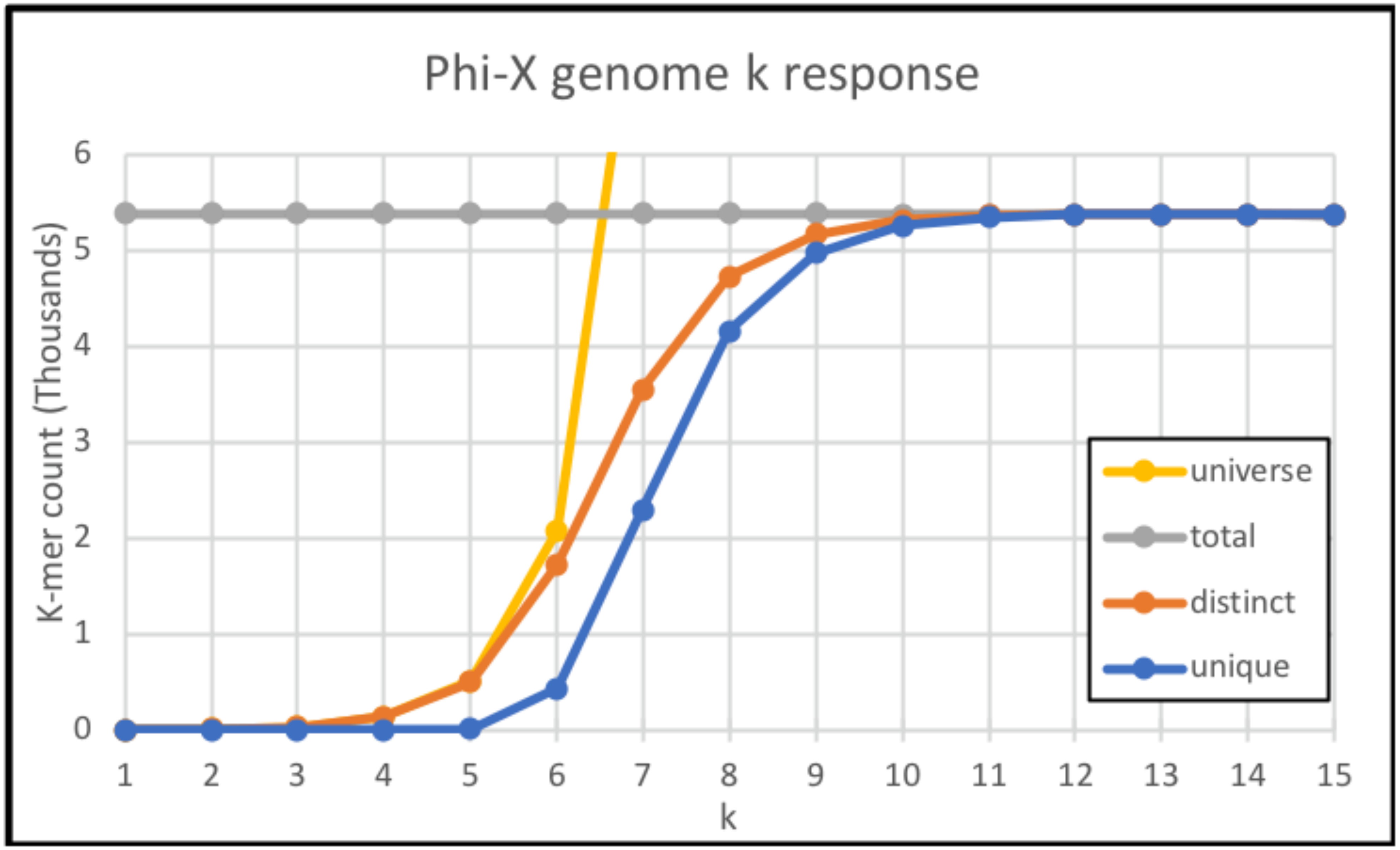
```
kat hist -o phiX_9mer.hist -m 9 phiX.fasta
```

```
# Title:9-mer spectra for: phiX.fasta
# XLabel:9-mer frequency
# YLabel:# distinct 9-mers
# Kmer value:9
# Input 1:phiX.fasta
###
1 4972
2 189
3 8
4 1
5 0
6 0
7 0
8 0
9 0
...
```



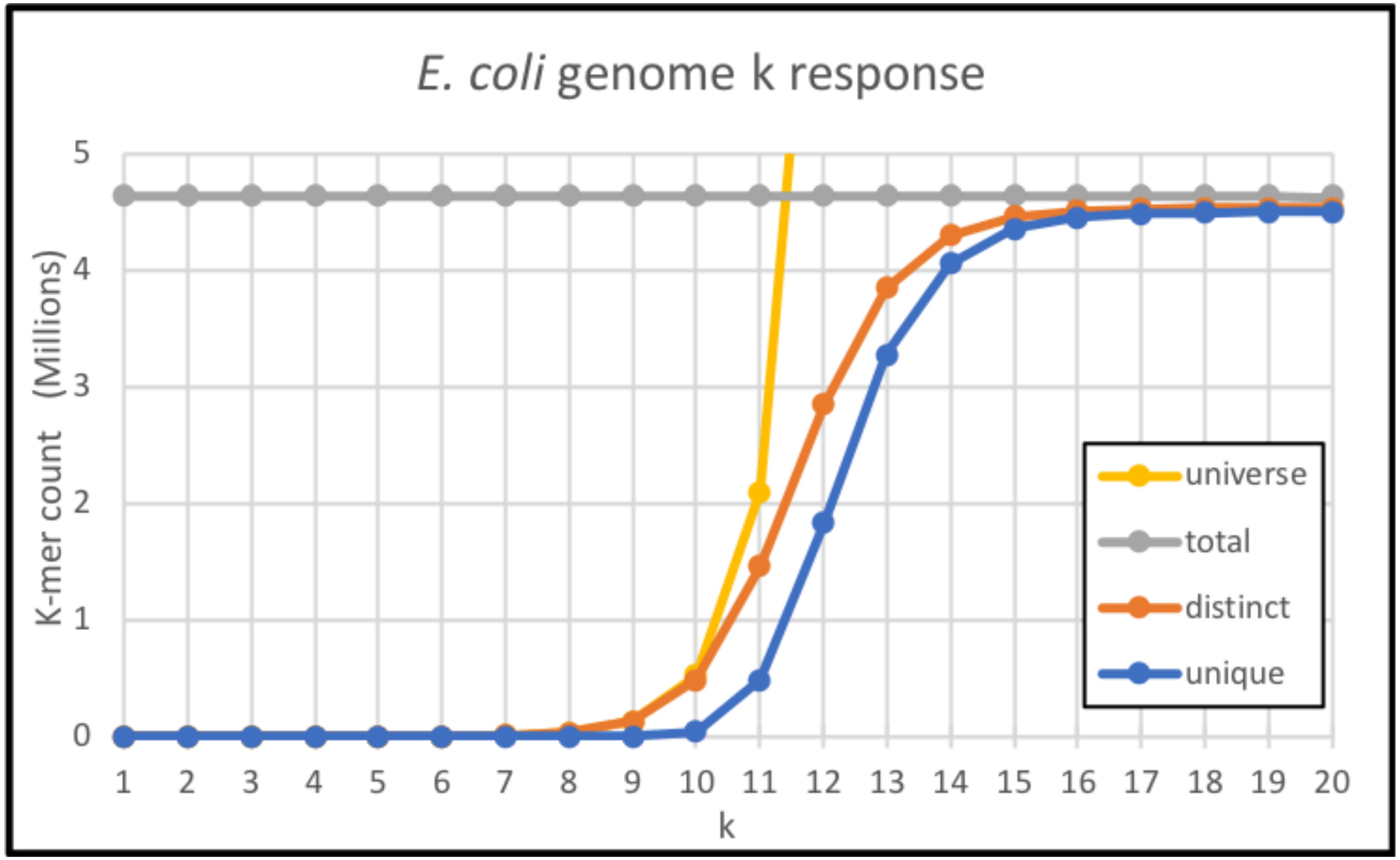
Earlham Institute

K response



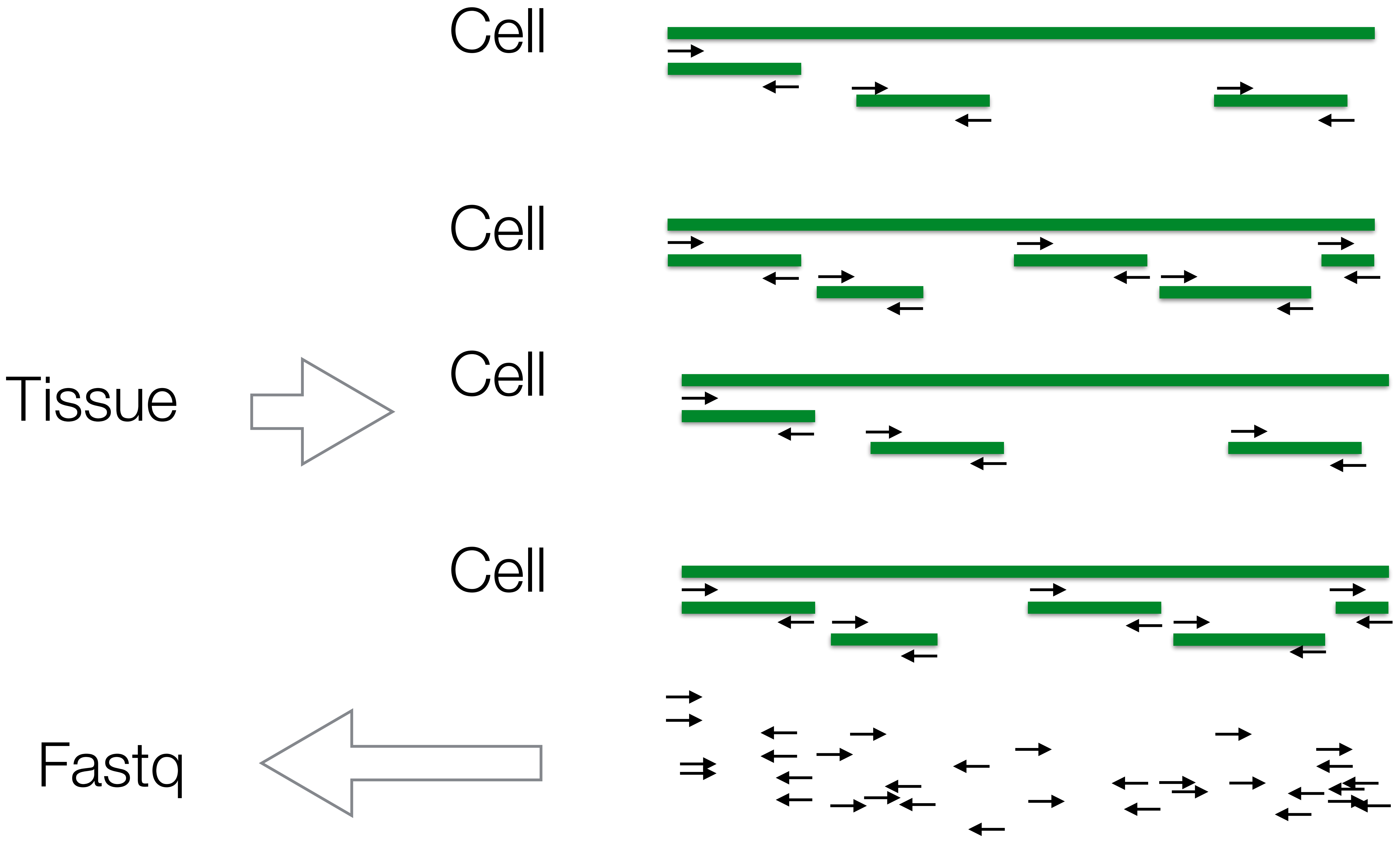
Phi-X genome k response
K-mer count (Thousands)
0
1
2
3
4
5
6
1
2
3
4
5
6
7
8
9
10
11
12
13
14
15
k
universe
total
distinct
unique

K response



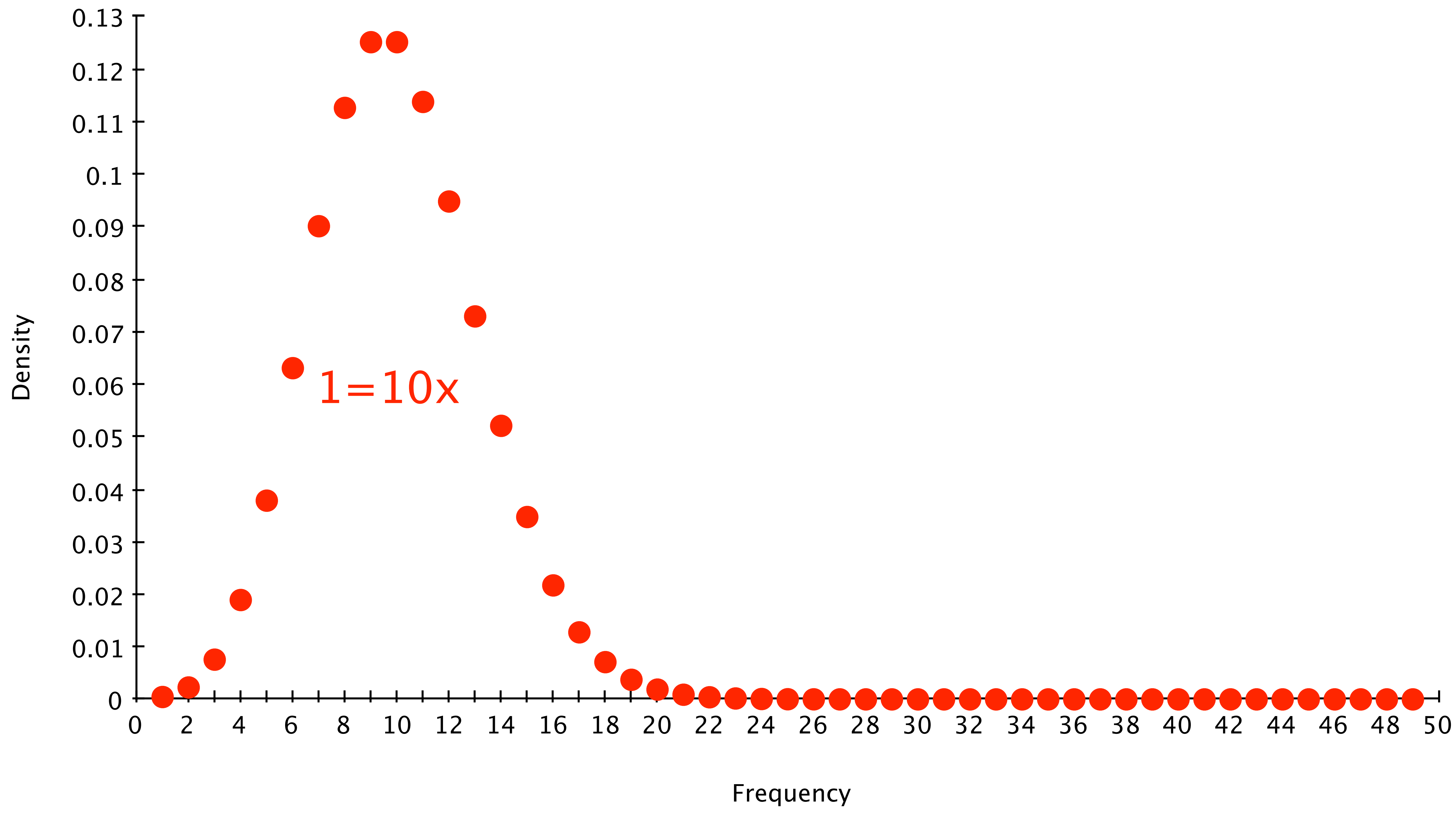
E. coli genome k response
K-mer count (Millions)
0
1
2
3
4
5
1 2 3 4 5 6 7 8 9 10 11 12 13 14 15 16 17 18 19 20
k
universe
total
distinct
unique

Read sampling

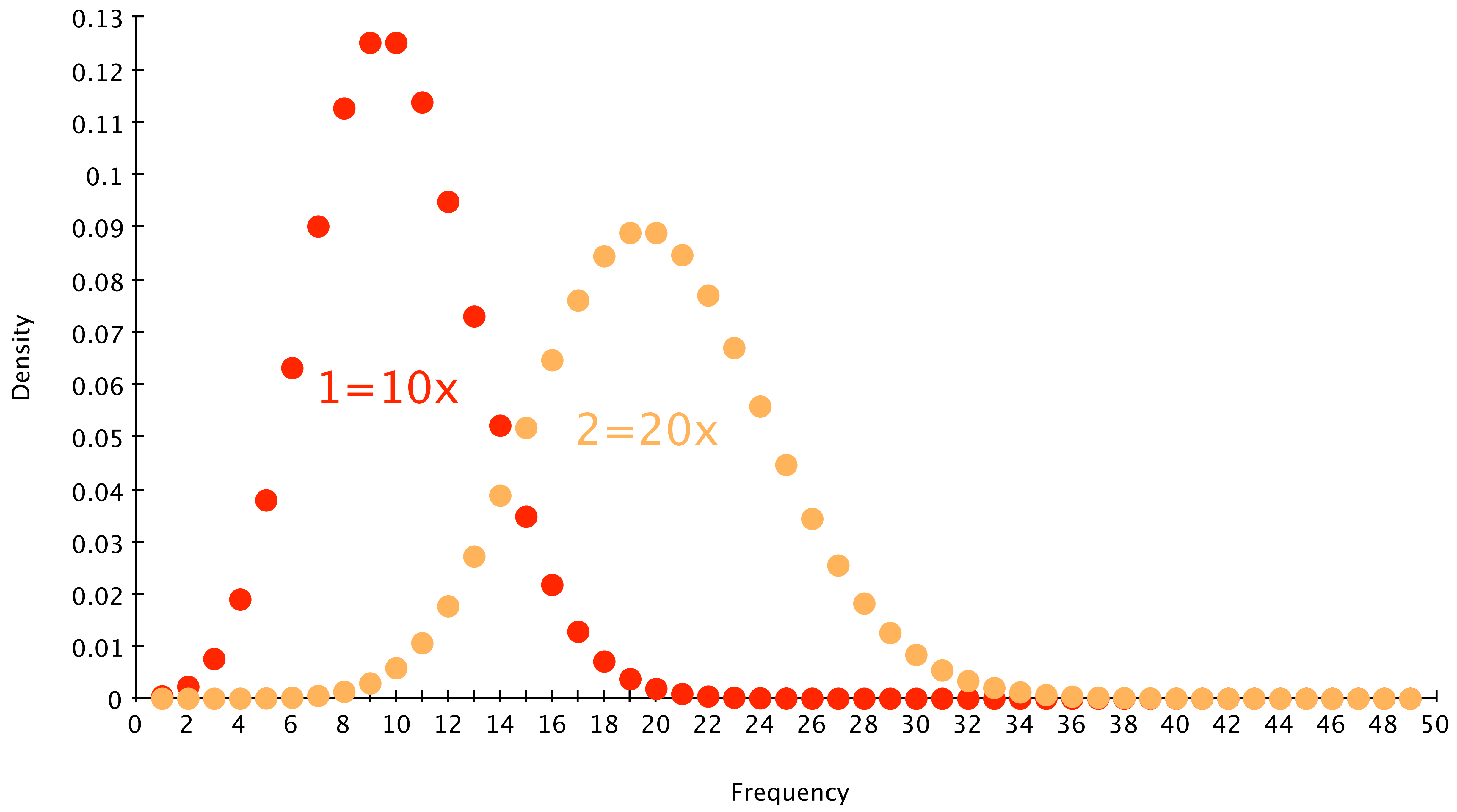


It is somehow similar to “catting” genome files

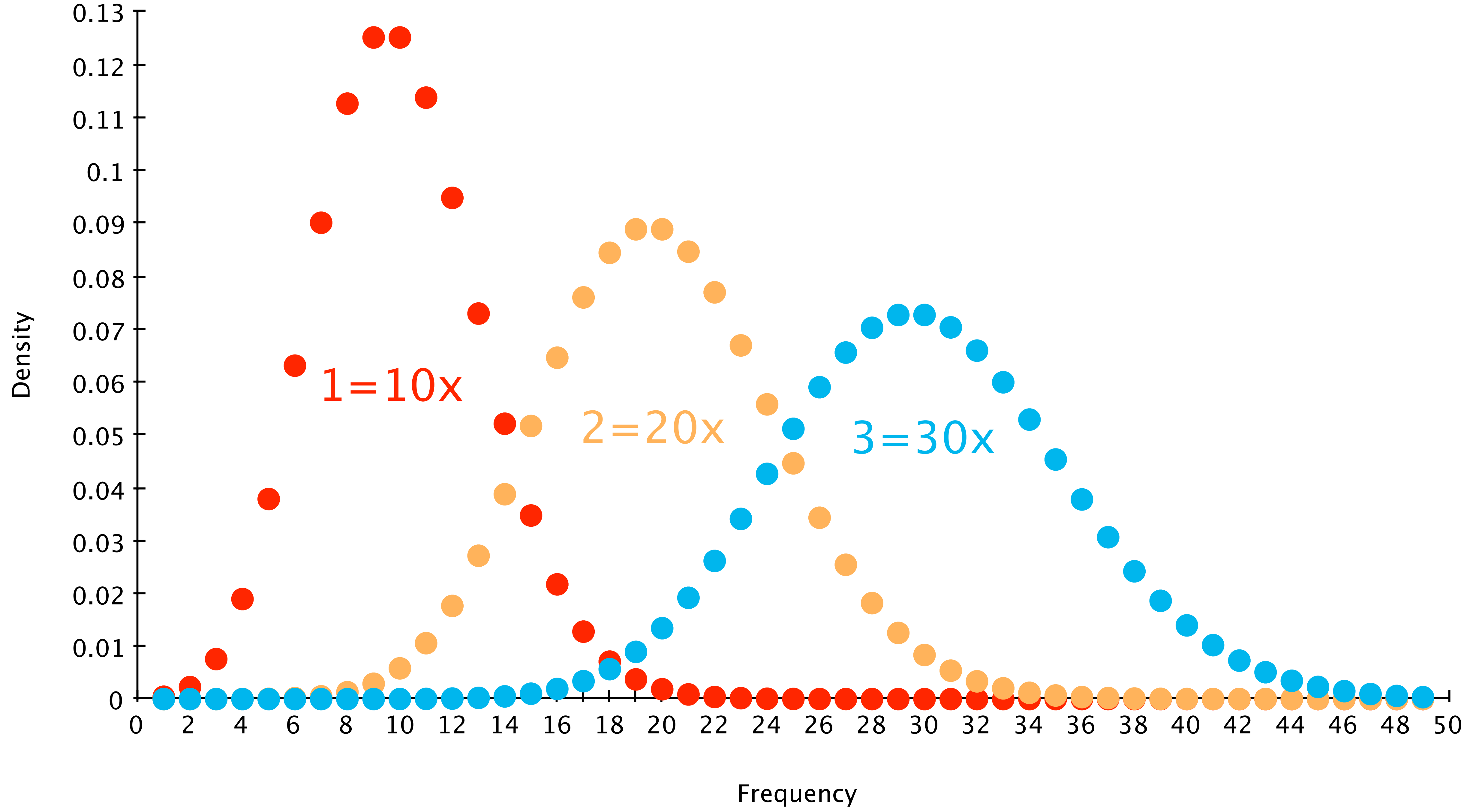
1 Chromosome genome



Earlham Institute



Earlham Institute



Earlham Institute

Assembly QC

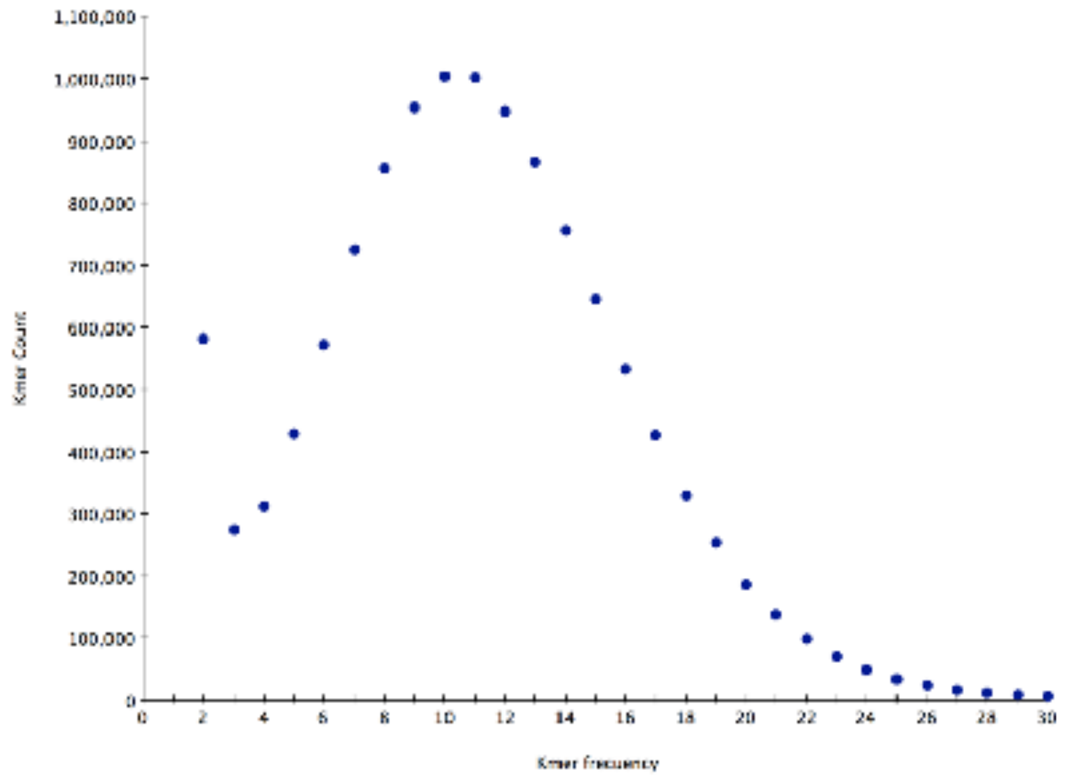
The right *motifs,*

the correct number of times,

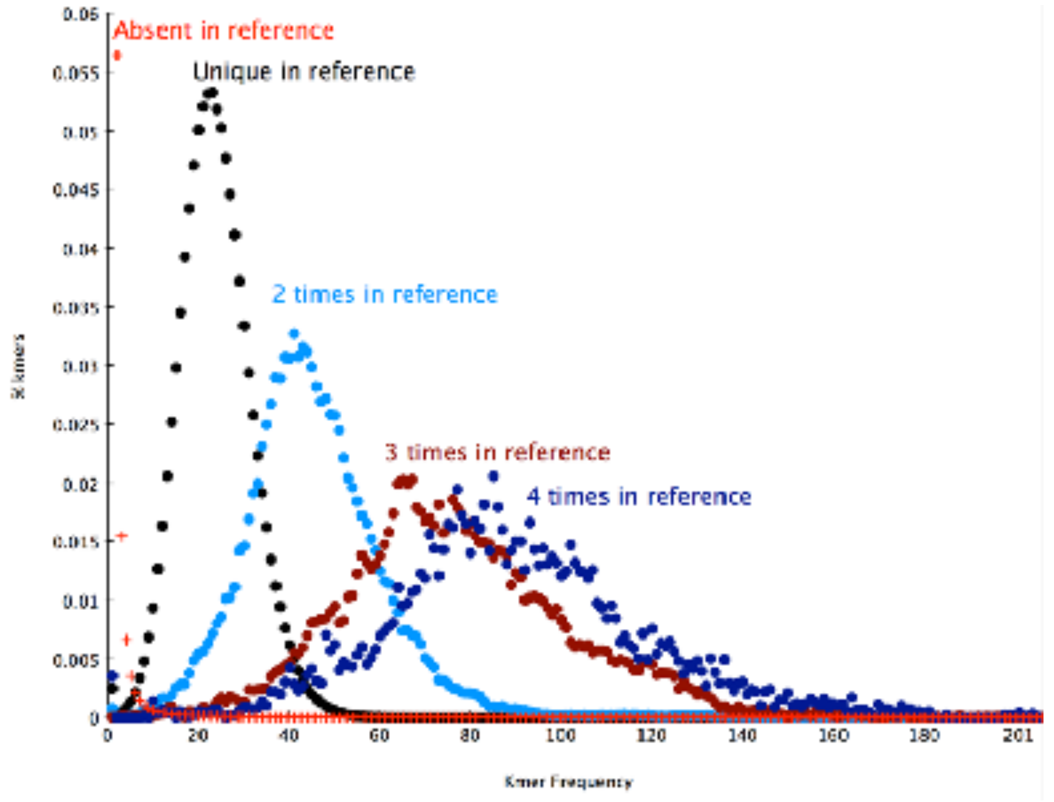
in correct order and position.

Any ideas??

Earlham Institute



Kmer Count
Kmer Frequency



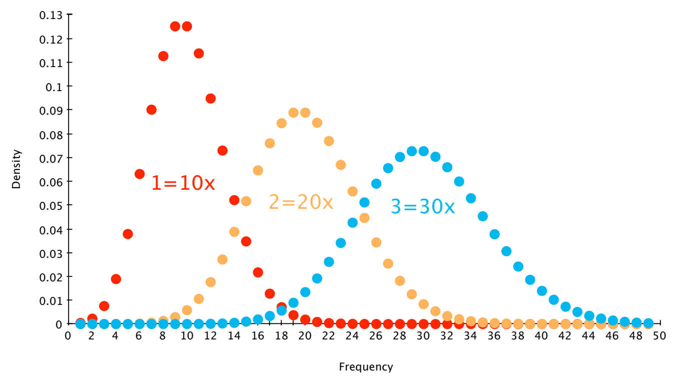
Absent in reference
Unique in reference
2 times in reference
3 times in reference
4 times in reference
% kmers
Kmer Frequency

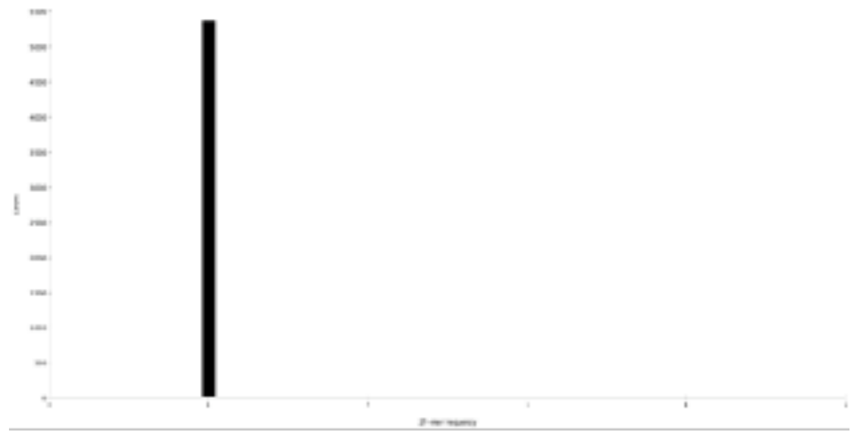
Assembly QC

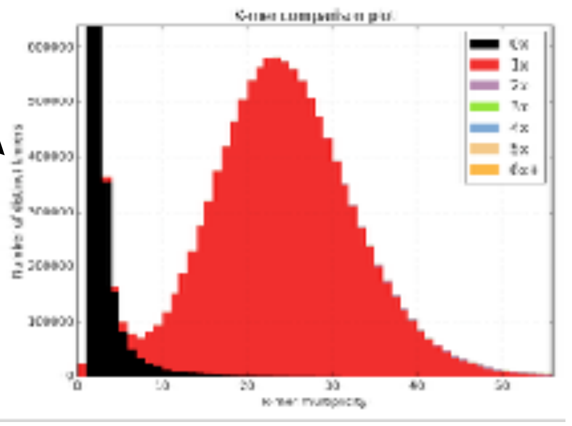
Compare the copy number in the genome and the expected frequency in the reads

How many times a kmer is expected according to the reads

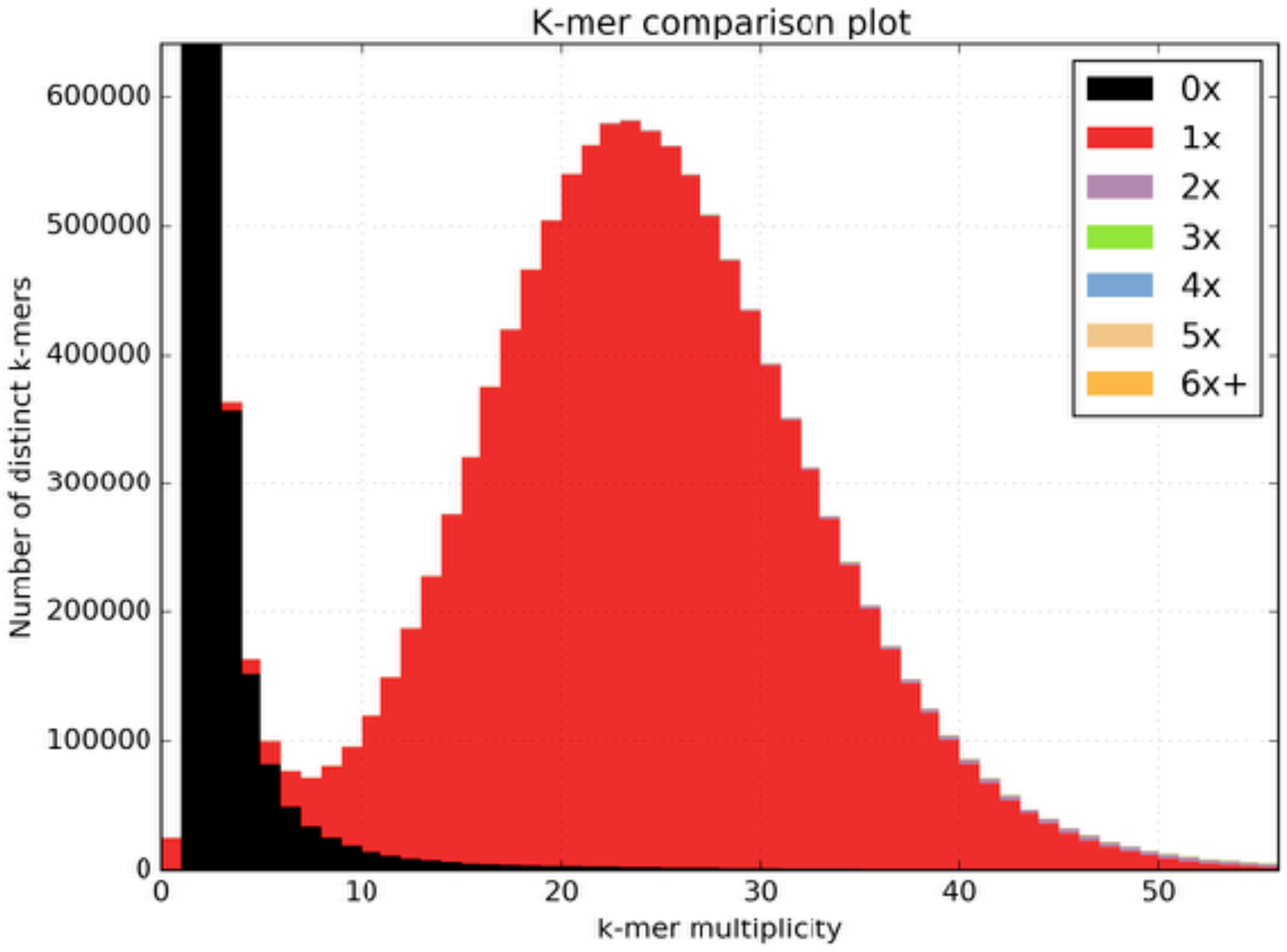
How many times the same kmer is counted in the assembly







Earlham Institute


K-mer comparison plot
Number of distinct k-mers
k-mer multiplicity
0
100000
200000
300000
400000
500000
600000
0
10
20
30
40
50
0x
1x
2x
3x
4x
5x
6x+

Ecoli spectra and spectra-cn

~1.6Mbp

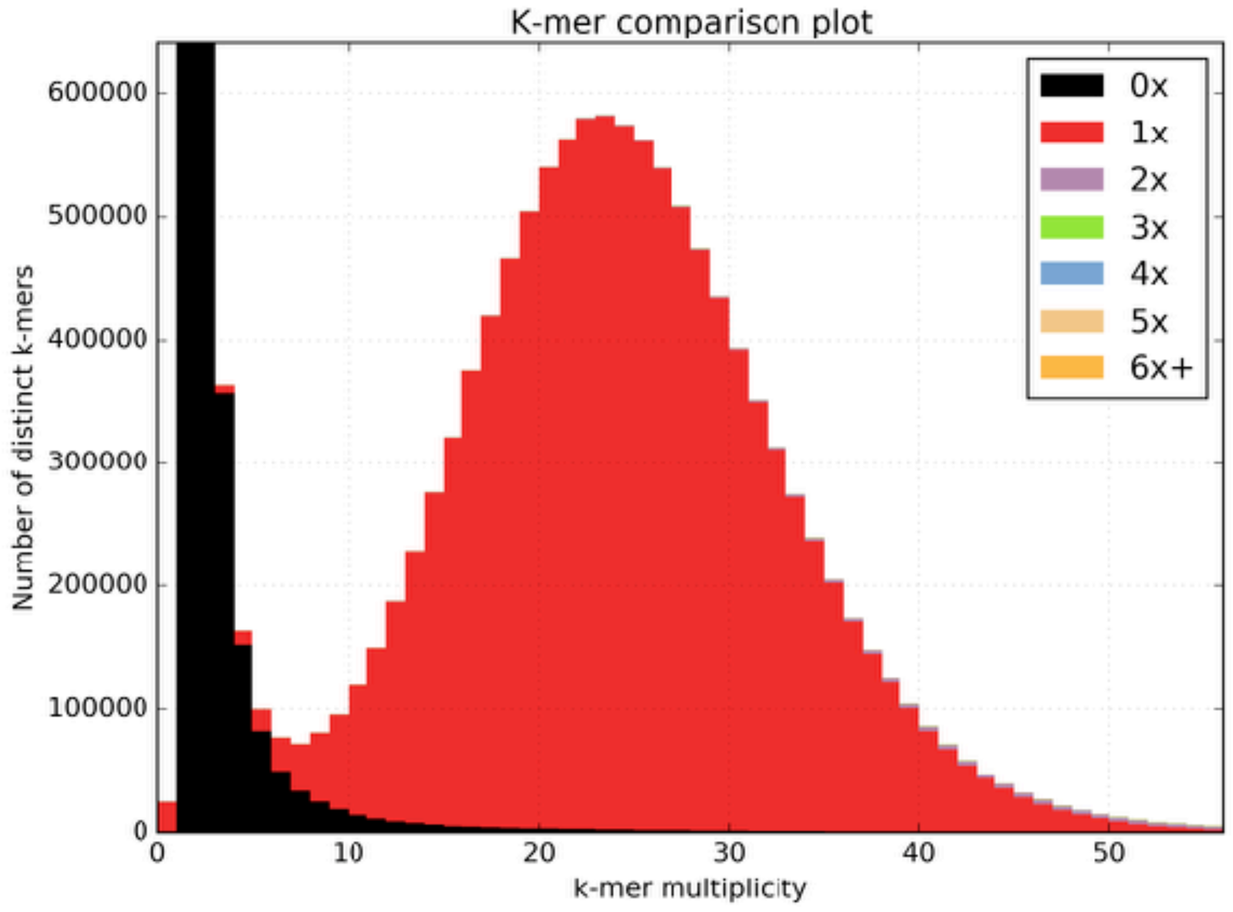
kat hist --help

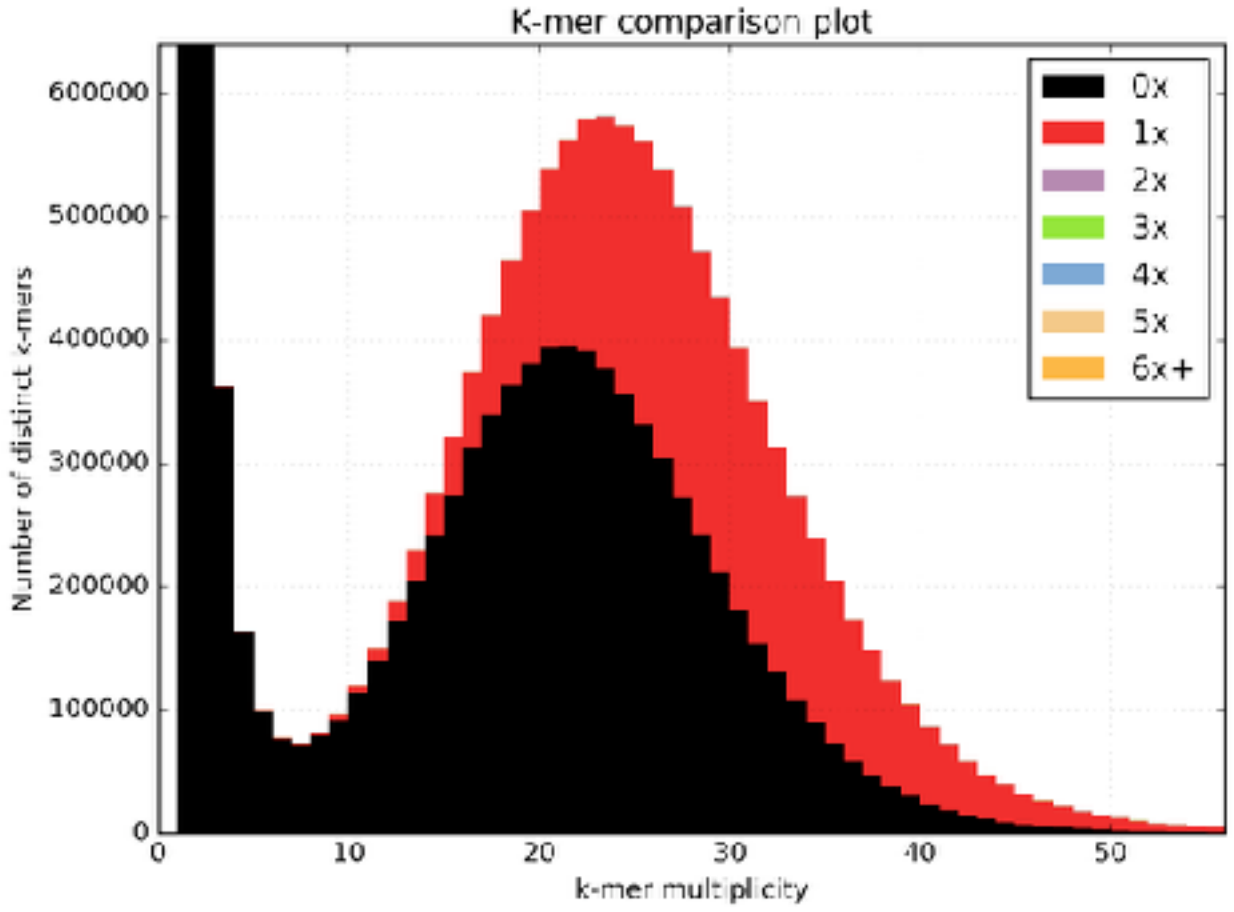
Use the spectra to check the genome size (excel)

kat comp --help

Use kat plot spectra-cn to display the complete spectra

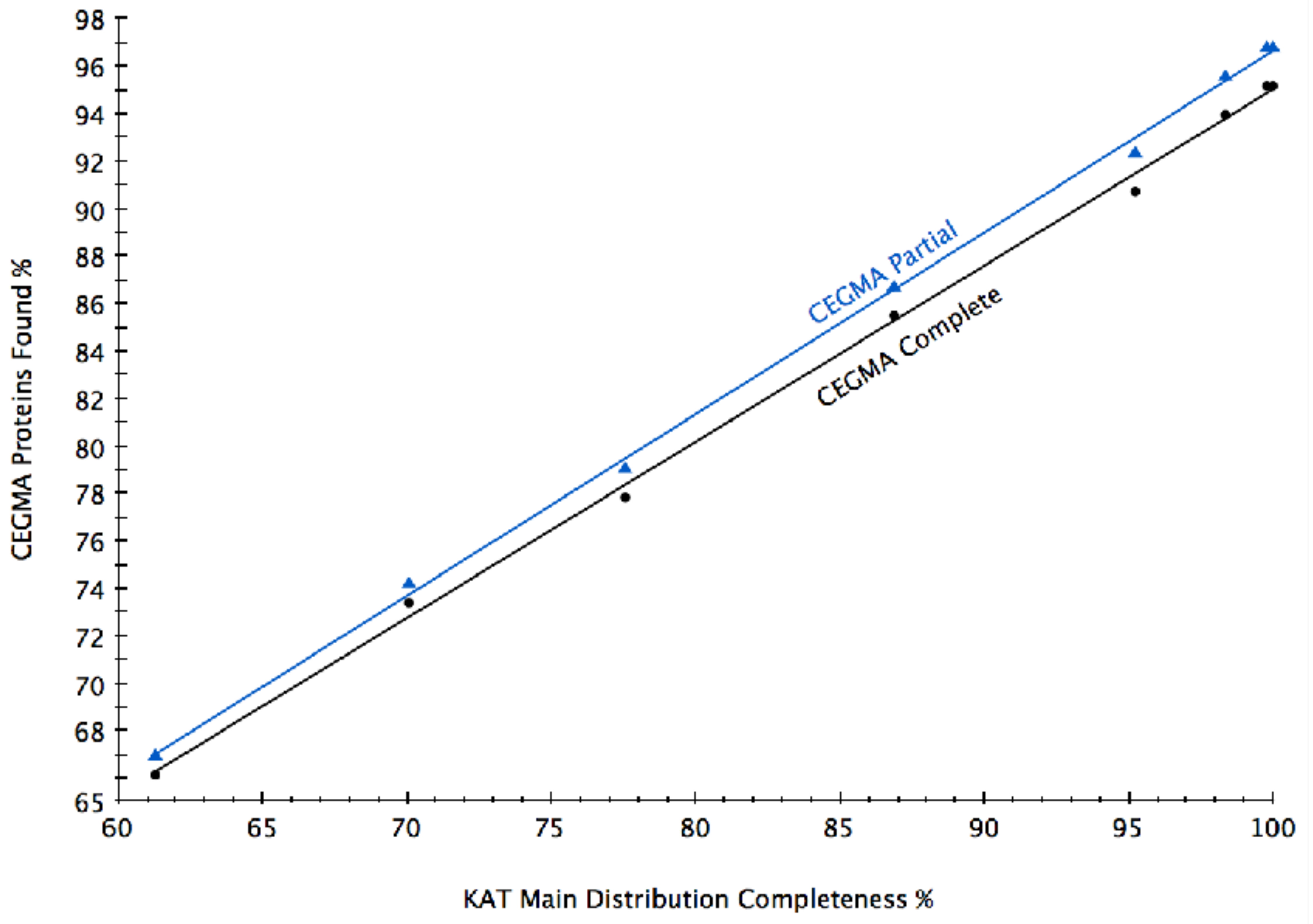
Content check





Earlham Institute

KAT vs. CEGMA

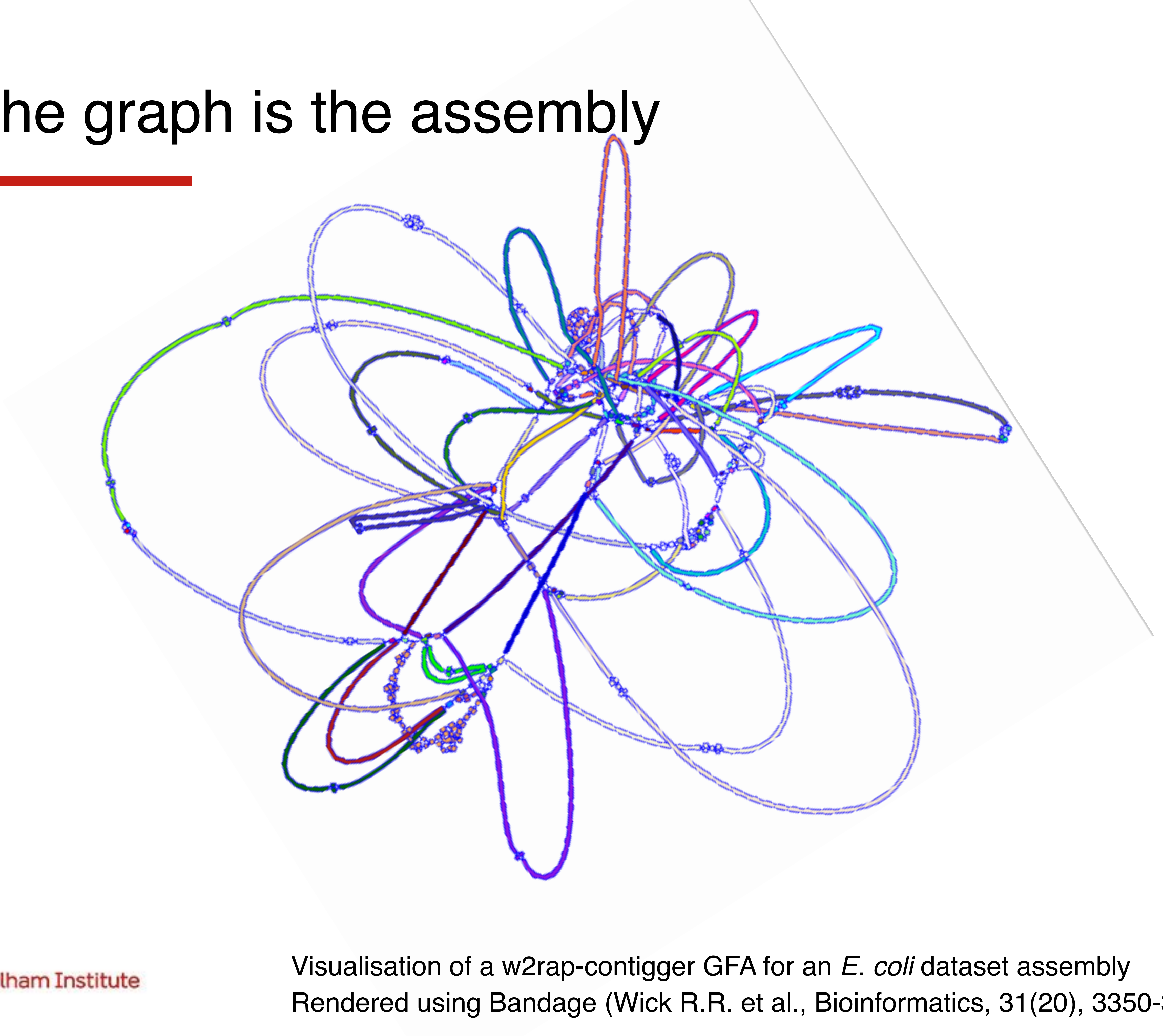


C. fraxinea - “Ash Dieback” - NORNEX

Guess the spectra-cn game

Figures in shared dir

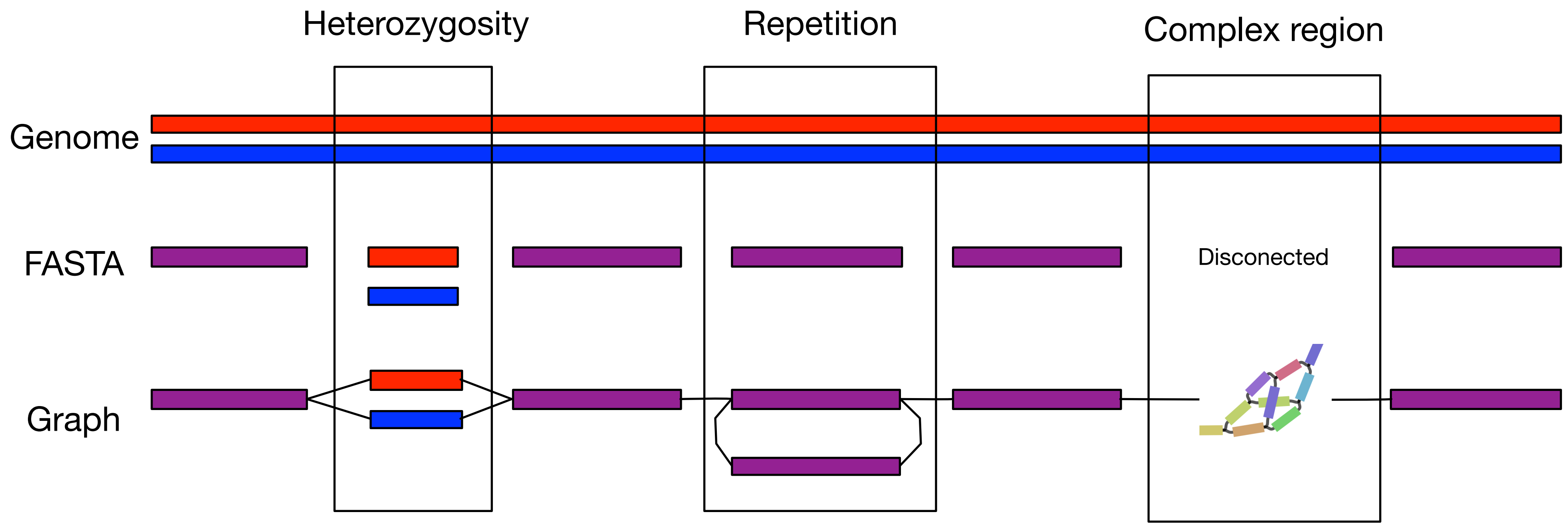
The graph is the assembly

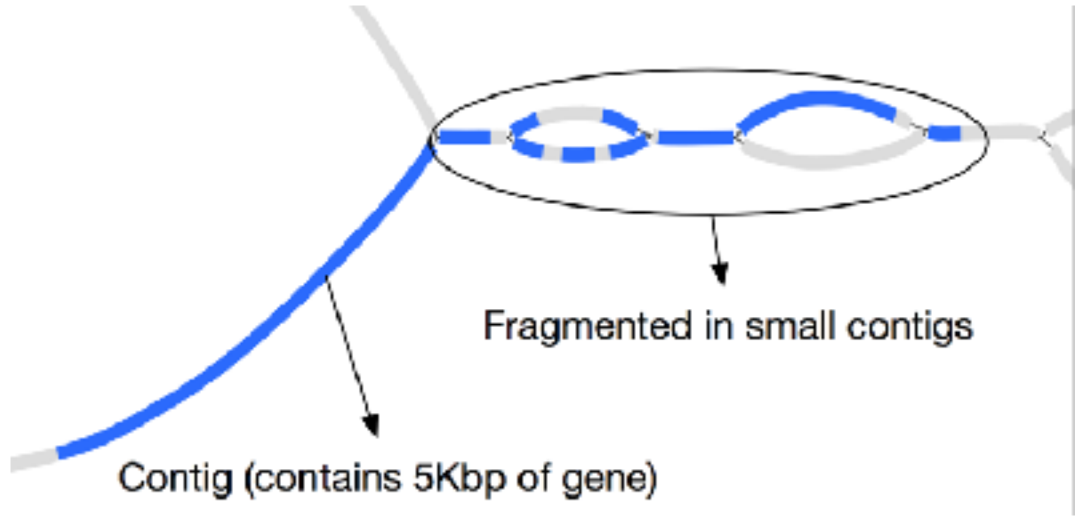


Earlham Institute

Visualisation of a w2rap-contigger GFA for an *E. coli* dataset assembly
Rendered using Bandage (Wick R.R. et al., Bioinformatics, 31(20), 3350-3352)

Signatures in the spectra-cn





Example: *A. thaliana* assembly

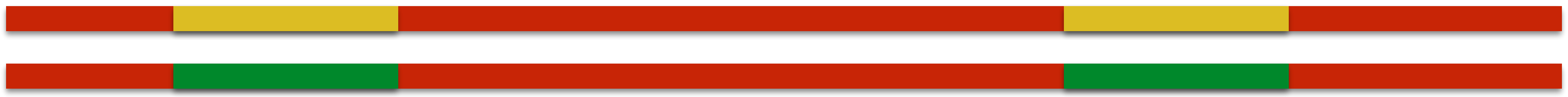
Total TAIR CDSs: 27,416

- In contigs: 27,079 (98.77%)
- Not in contigs: 337 (1.23%)
  - Paths found in paths: 175 (51.93%)
  - Most of the rest have paths, just more complex

Earlham Institute

Heterozygosity

Random heterozygous genome 1



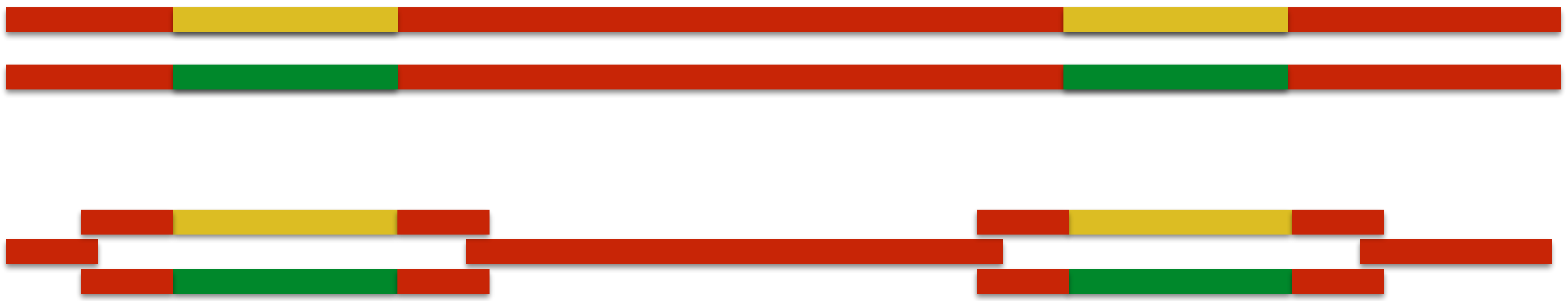
What structure is this going to produce in the graph.

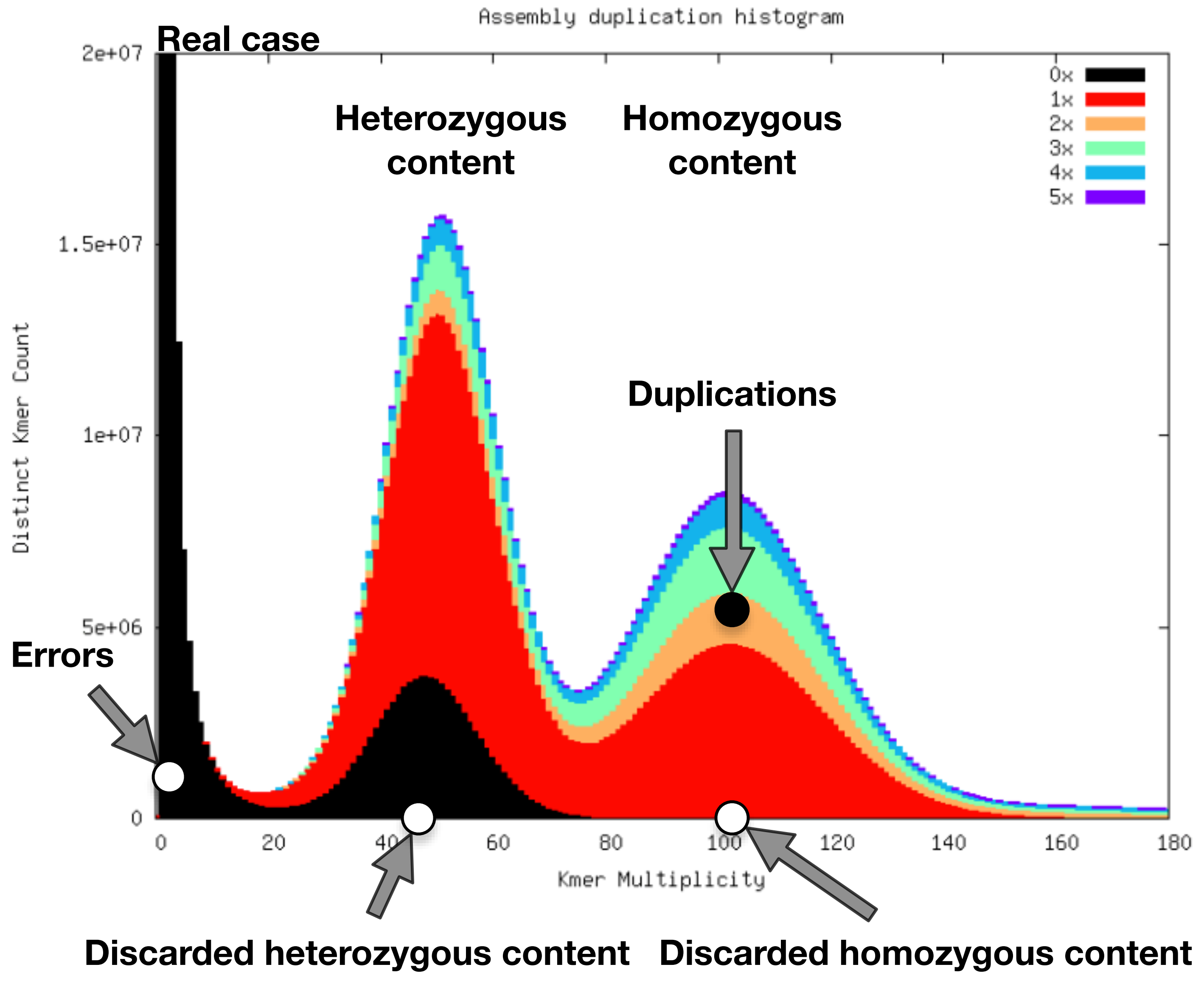
How the heterozygosity is going to appear up in the spectra-cn

Is the structure of the graph going to affect the spectra-cn?

kat comp hetero_1 vs reference

kat comp hetero_1 vs abyss assembly

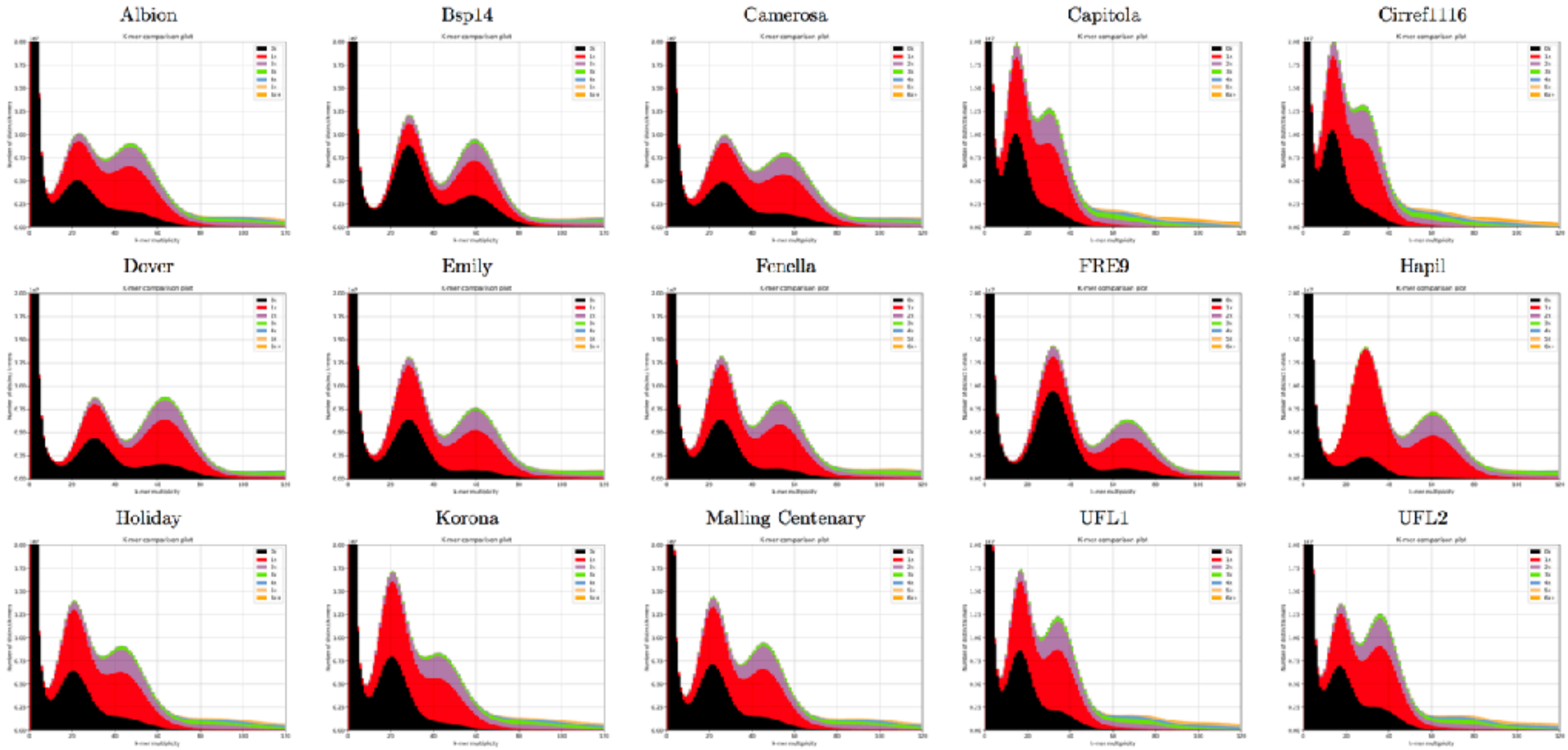




Assembly duplication histogram
Real case
Heterozygous content
Homozygous content
Duplications
Errors
Discarded heterozygous content
Discarded homozygous content
Distinct Kmer Count
Kmer Multiplicity
0x
1x
2x
3x
4x
5x
2e+07
1.5e+07
1e+07
5e+06
0
0
20
40
60
80
100
120
140
160
180

Kmer sources does not have to be from the same sample,

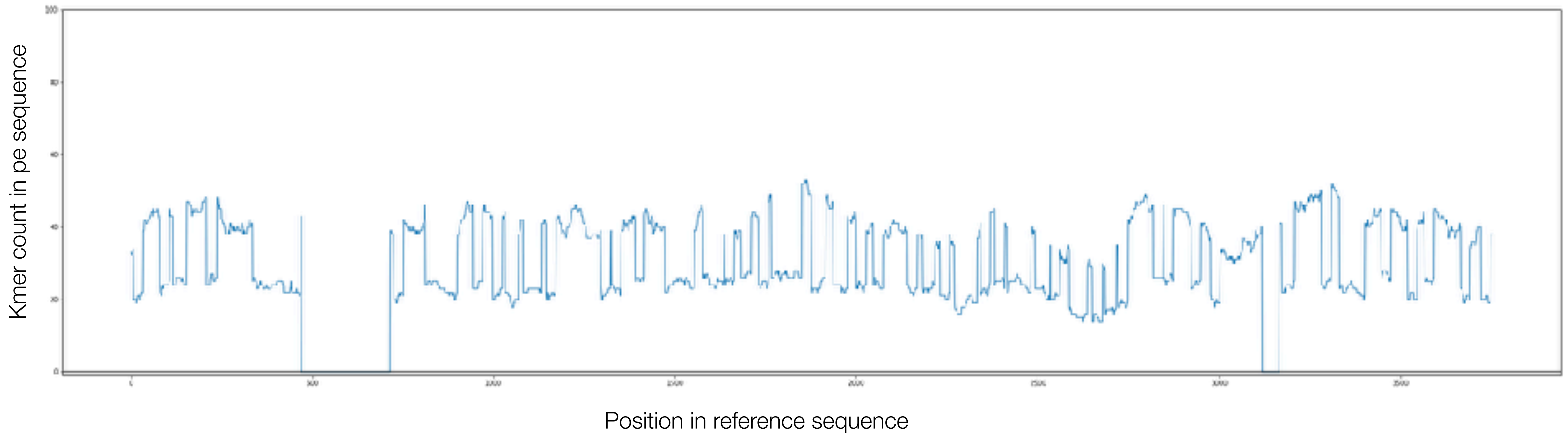
15 Strawberry cultivars (raw reads) vs Red Gauntlet (reference)



Unpublished Work - Octoploid Strawberry - In collaboration with Richard Harrison’s lab at NIAR EMR

Kat sect

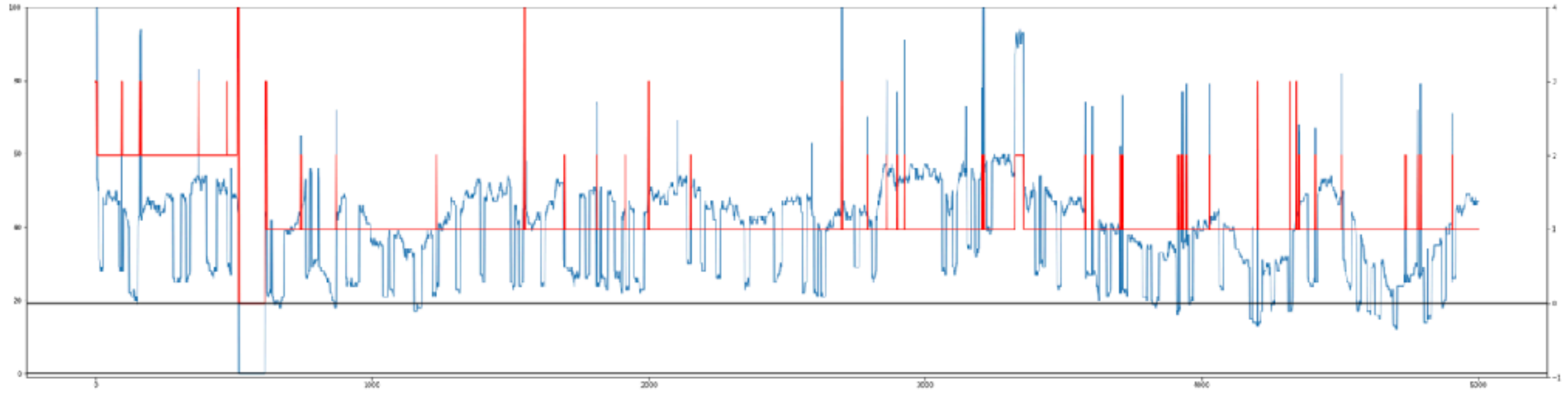
Kat sect is a tool to project kmers from a kmer set on top of a sequence



Hetero_1 example

kat sect pe vs scaffolds

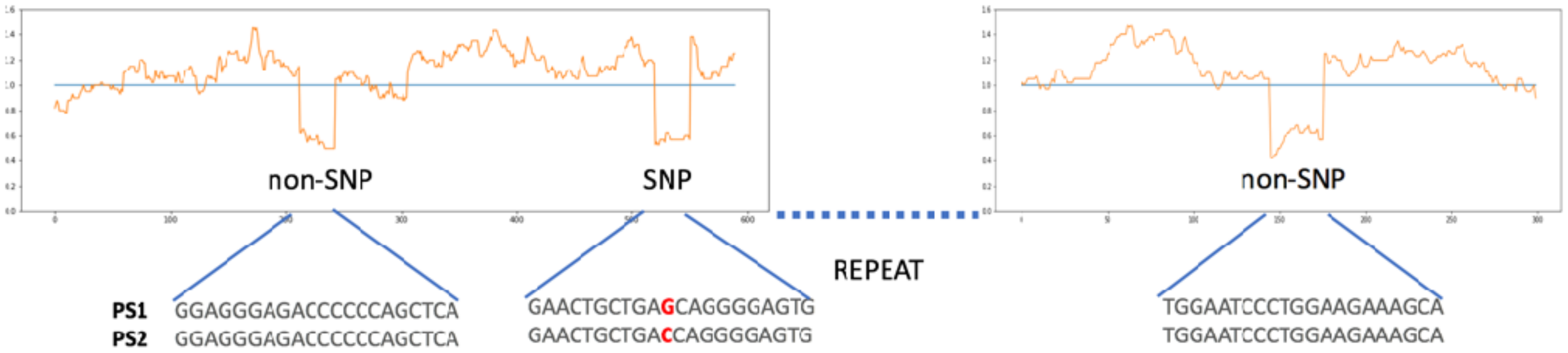
kat sect scaffolds vs scaffolds

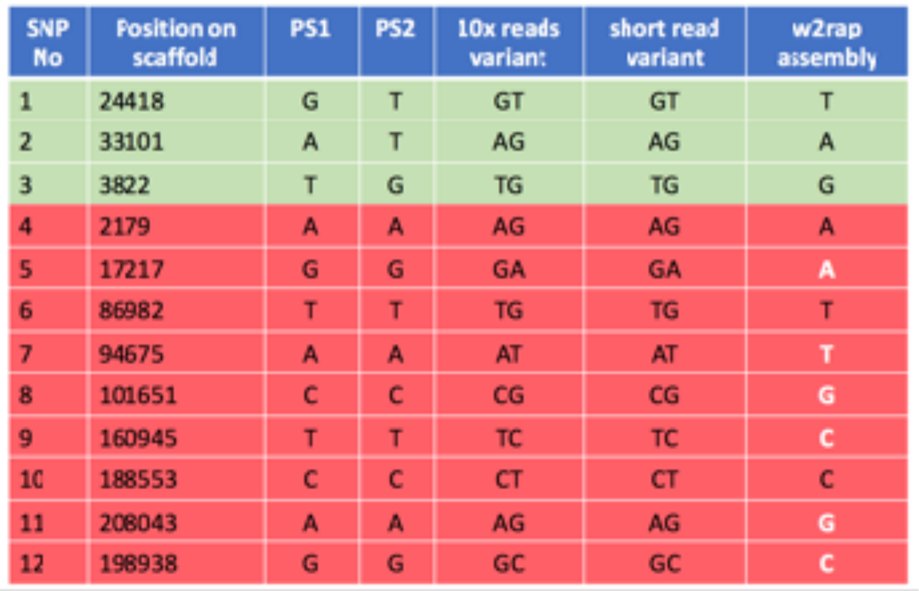


Blue pe vs scaffolds
Red scaffolds vs self

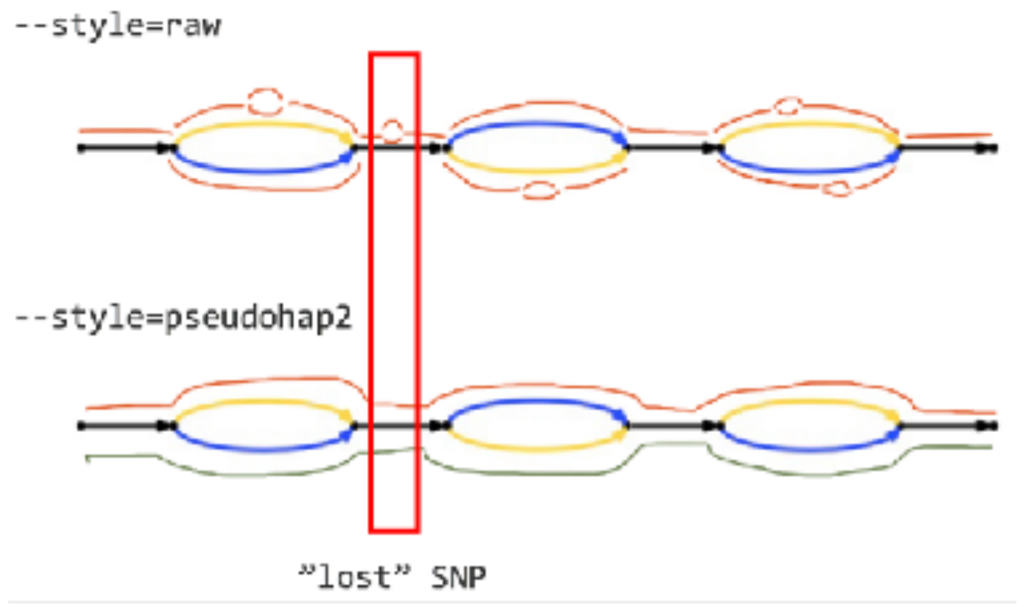
The 10x “lost SNPs tale”

HGP: 21796





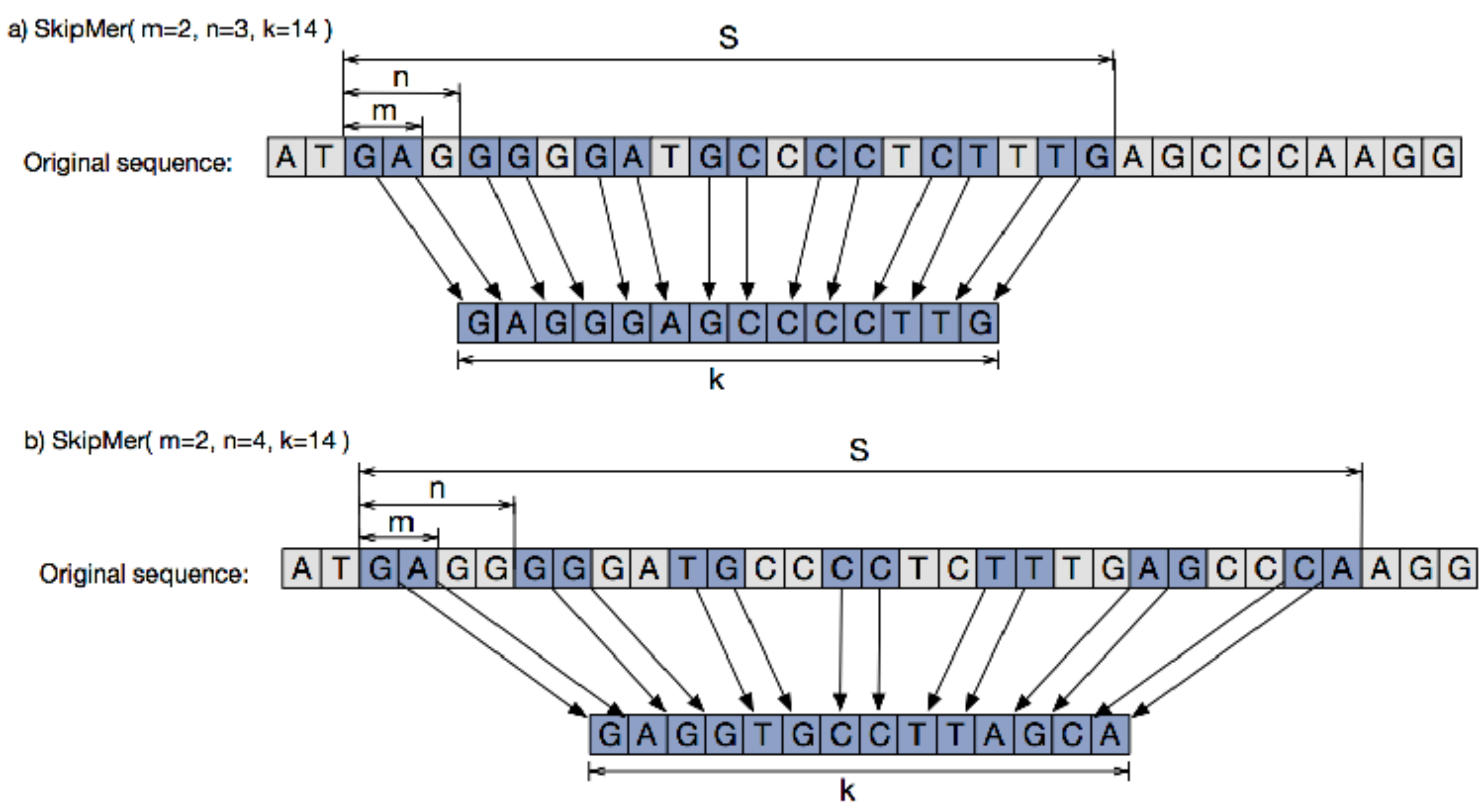
SNP No	Position on scaffold	PS1	PS2	10x reads varian:	short read variant	w2rap assembly
1	24418	G	T	GT	GT	T
2	33101	A	T	AG	AG	A
3	3822	T	G	TG	TG	G
4	2179	A	A	AG	AG	A
5	17217	G	G	GA	GA	A
6	86982	T	T	TG	TG	T
7	94675	A	A	AT	AT	T
8	101651	C	C	CG	CG	G
9	160945	T	T	TC	TC	C
10	188553	C	C	CT	CT	C
11	208043	A	A	AG	AG	G
12	198938	G	G	GC	GC	C



Earlham Institute

Courtesy of Graham Etherington

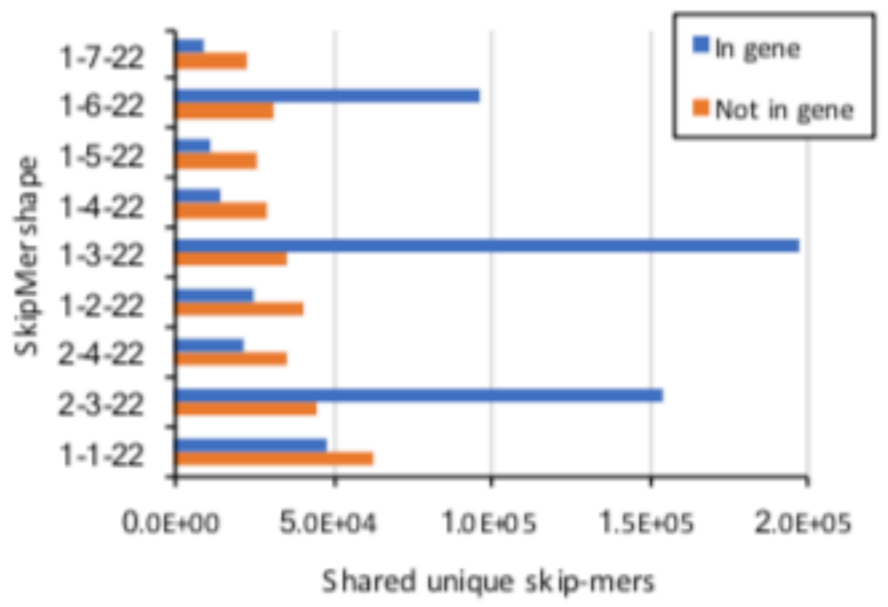
Skip-mers: higher entropy, higher sensitivity



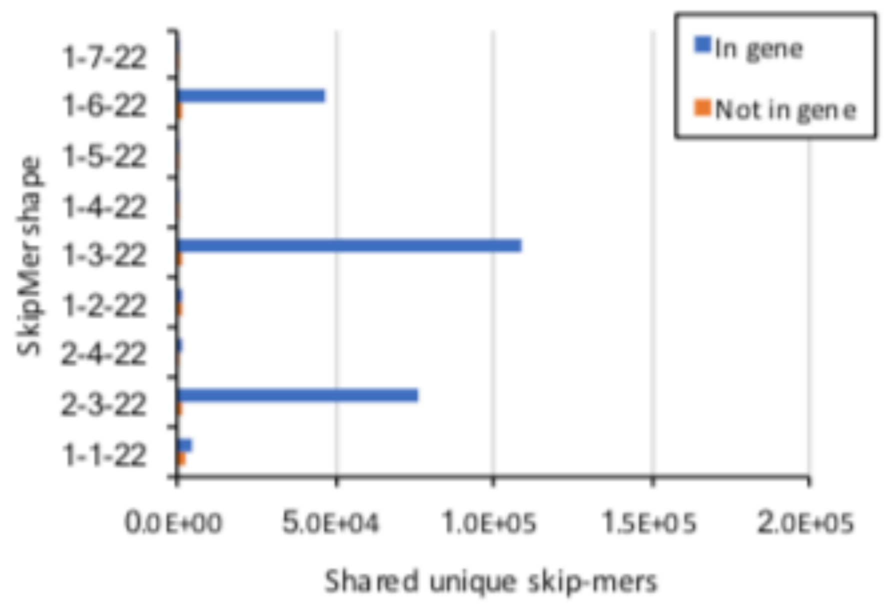
a) SkipMer(m=2, n=3, k=14)
S
n
m
Original sequence:
A T G A G G G G G A T G C C C C T C T T T G A G C C C A A G G
G A G G G A G C C C C T T G
k
b) SkipMer(m=2, n=4, k=14)
S
n
m
Original sequence:
A T G A G G G G G A T G C C C C T C T T T G A G C C C A A G G
G A G G T G C C T T A G C A
k

Figure 1. Different *SkipMer*(m,n,k) cycles defined over the same sequence region, resulting in different combinations of bases. The shape of the underlying cyclic q-gram is defined by the variables m (used bases per cycle), n (cycle length), and k (total number of bases).

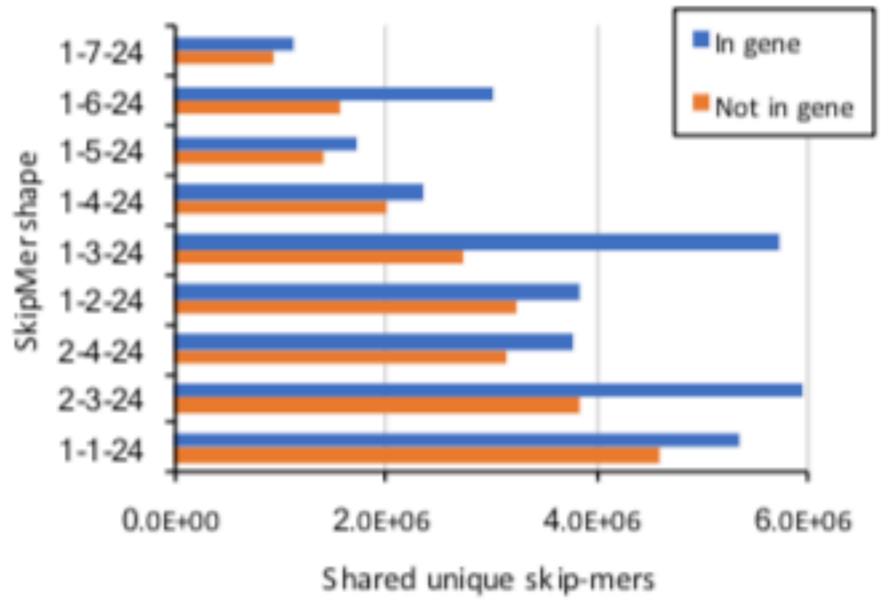
Earlham Institute https://github.com/bioinfologics/skm-tools



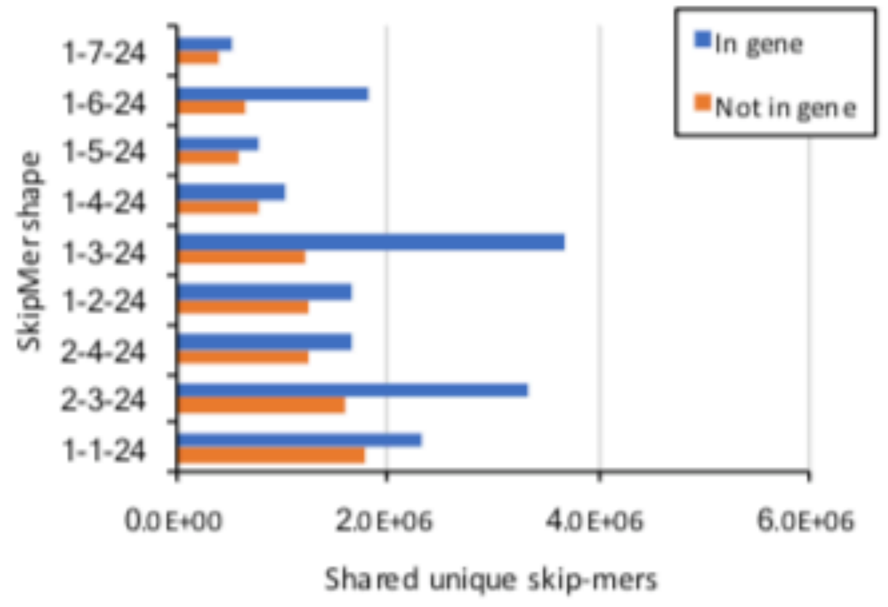
(a) 2-way: *Arabidopsis thaliana* ∧ *Orzya sativa*



(b) 3-way: *Arabidopsis thaliana* ∧ *Orzya sativa* ∧ *Brachypodium distachion*



(c) 2-way: *Homo sapiens* ∧ *Mus musculus*



(d) 3-way: *Homo sapiens* ∧ *Mus musculus* ∧ *Canis familiaris*

Figure 3. Effect of different combinations of m and n, while keeping k constant, for 2-way and 3-way skip-mer intersections. Only unique skip-mers are considered and skip-mers originating from sequence annotated with gene features on the first genome are classified as *"In gene"*. The skip-mer shapes are sorted along the vertical axis according to total skip-mer span (S), with the largest span on top.