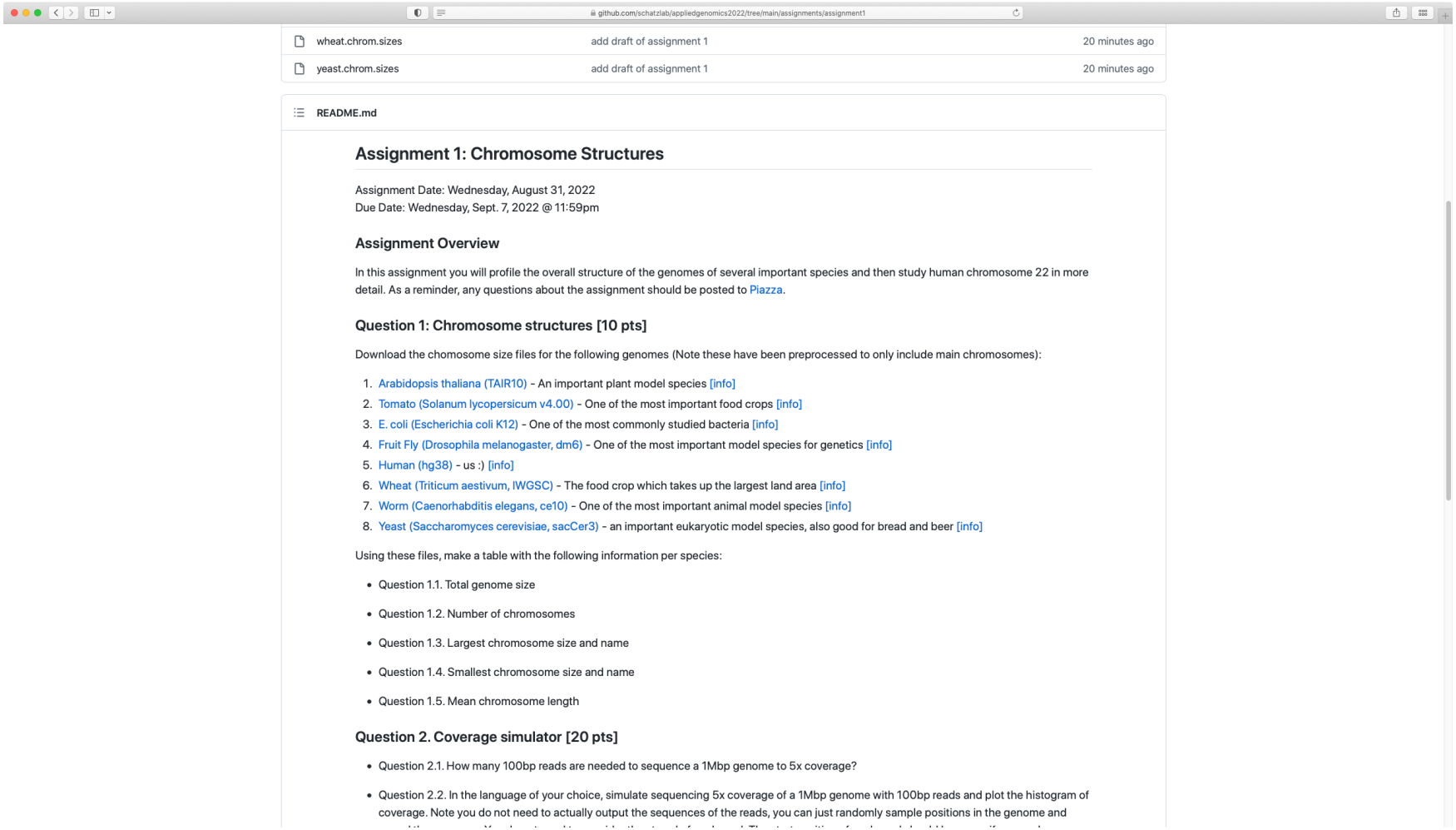
Genome Assembly

Michael Schatz

Sept 12, 2022

Lecture 4: Applied Comparative Genomics

Assignment 1: Chromosome Structures Due Wednesday Sept 7 by 11:59pm



github.com/schatzlab/appliedgenomics2022/tree/main/assignments/assignment1

wheat.chrom.sizes	add draft of assignment 1	20 minutes ago
yeast.chrom.sizes	add draft of assignment 1	20 minutes ago

README.md

Assignment 1: Chromosome Structures

Assignment Date: Wednesday, August 31, 2022
Due Date: Wednesday, Sept. 7, 2022 @ 11:59pm

Assignment Overview

In this assignment you will profile the overall structure of the genomes of several important species and then study human chromosome 22 in more detail. As a reminder, any questions about the assignment should be posted to Piazza.

Question 1: Chromosome structures [10 pts]

Download the chomosome size files for the following genomes (Note these have been preprocessed to only include main chromosomes):

1. Arabidopsis thaliana (TAIR10) - An important plant model species [info]
2. Tomato (Solanum lycopersicum v4.00) - One of the most important food crops [info]
3. E. coli (Escherichia coli K12) - One of the most commonly studied bacteria [info]
4. Fruit Fly (Drosophila melanogaster, dm6) - One of the most important model species for genetics [info]
5. Human (hg38) - us :) [info]
6. Wheat (Triticum aestivum, IWGSC) - The food crop which takes up the largest land area [info]
7. Worm (Caenorhabditis elegans, ce10) - One of the most important animal model species [info]
8. Yeast (Saccharomyces cerevisiae, sacCer3) - an important eukaryotic model species, also good for bread and beer [info]

Using these files, make a table with the following information per species:

- Question 1.1. Total genome size
- Question 1.2. Number of chromosomes
- Question 1.3. Largest chromosome size and name
- Question 1.4. Smallest chromosome size and name
- Question 1.5. Mean chromosome length

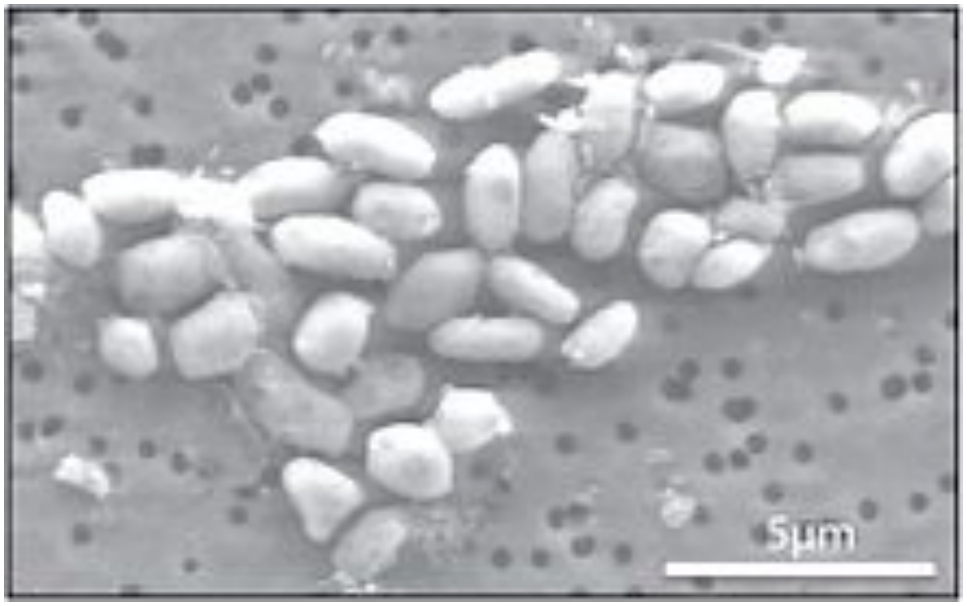
Question 2. Coverage simulator [20 pts]

- Question 2.1. How many 100bp reads are needed to sequence a 1Mbp genome to 5x coverage?
- Question 2.2. In the language of your choice, simulate sequencing 5x coverage of a 1Mbp genome with 100bp reads and plot the histogram of coverage. Note you do not need to actually output the sequences of the reads, you can just randomly sample positions in the genome and

https://github.com/schatzlab/appliedgenomics2022/tree/main/assignments/assignment1

Check Piazza for questions!

Halomonas sp. GFAJ-1



Library 1: Fragment

Avg Read length: 100bp

Insert length: 180bp

Library 2: Short jump

Avg Read length: 50bp

Insert length: 2000bp

A Bacterium That Can Grow by Using Arsenic Instead of Phosphorus
Wolfe-Simon et al (2010) *Science. 332(6034)1163-1166.*

Digital Information Storage

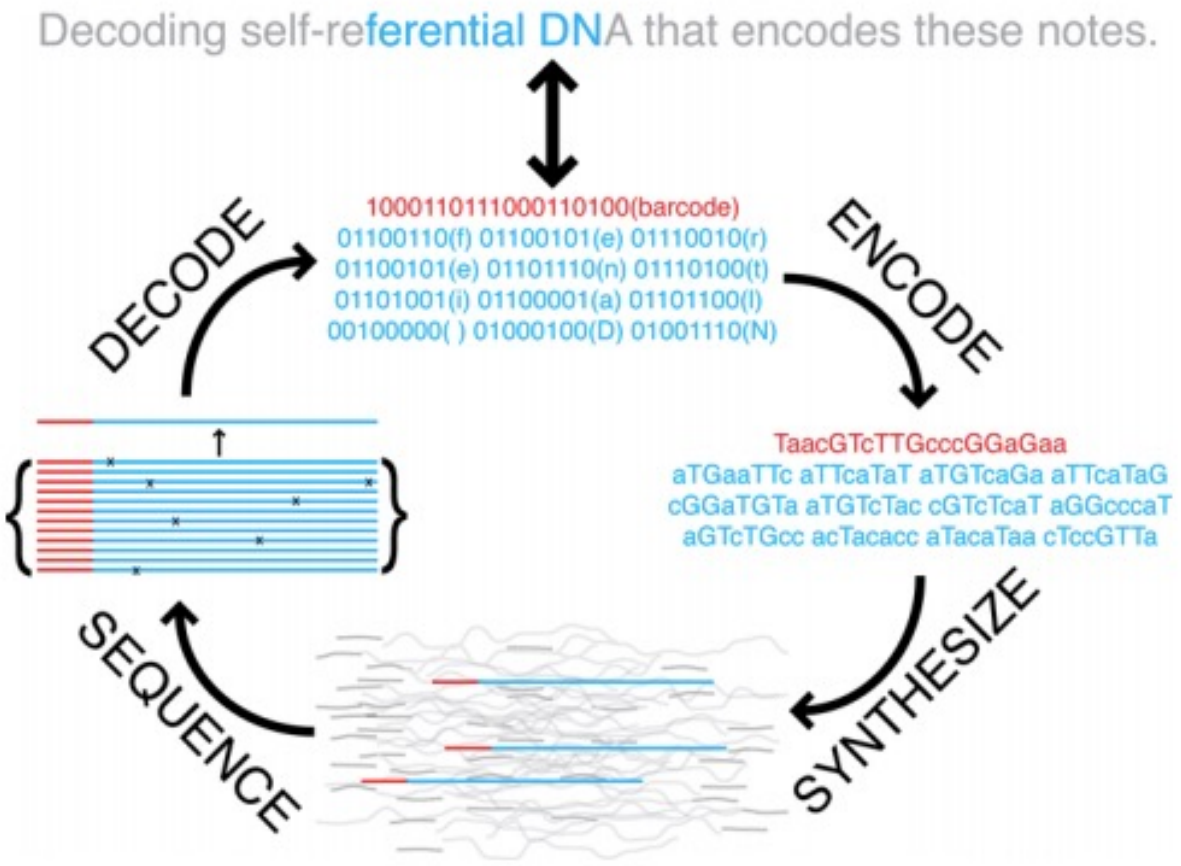


Fig. S1. Schematic of DNA information storage.

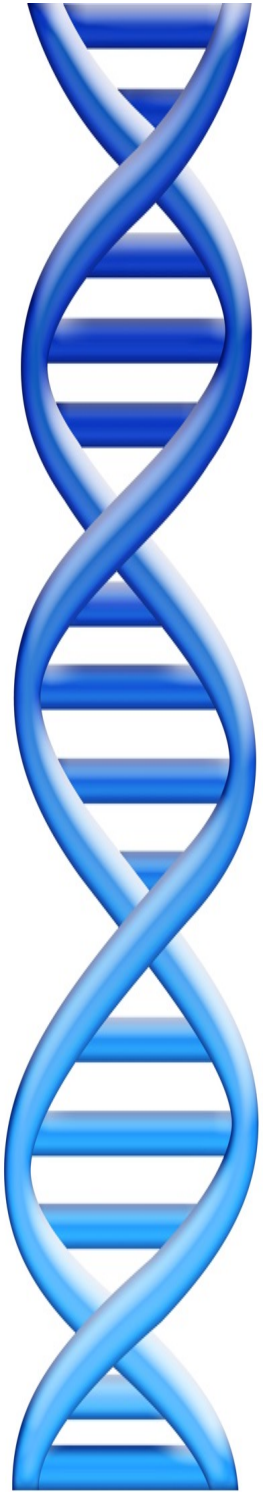
Encoding/decoding algorithm implemented in dna-encode.pl from David Dooling.

Next-generation Digital Information Storage in DNA
Church et al (2010) *Science*. 337(6102)1628

Assignment 2: Genome Assembly
Due Monday Sept 19 @ 11:59pm

1. ***Setup Conda/Docker/Ubuntu***
2. ***Initialize Tools***
3. ***Download Reference Genome & Reads***
4. ***Decode the secret message***
    1. *Estimate coverage, check read quality*
    2. *Check kmer distribution*
    3. *Assemble the reads with spades*
    4. *Align to reference with MUMmer*
    5. *Extract foreign sequence*
    6. *dna-encode.pl -d*

https://github.com/schatzlab/appliedgenomics2022/blob/master/assignments/assignment2/README.md



Outline

1. ***Recap & Assembly theory***
   - Assembly by analogy

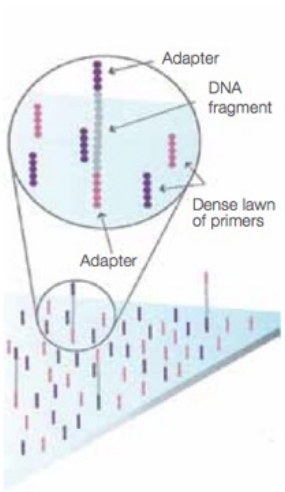
2. ***Practical Issues***
   - Coverage, read length, errors, and repeats

3. ***Whole Genome Alignment***
   - MUMmer recommended

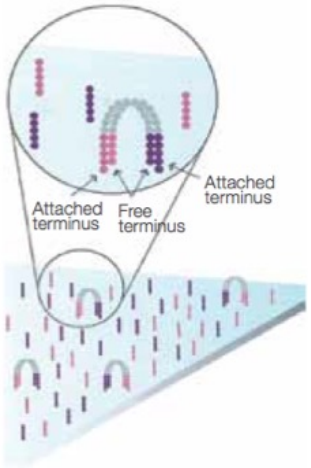
Second Generation Sequencing

Illumina HiSeq 2000
Sequencing by Synthesis

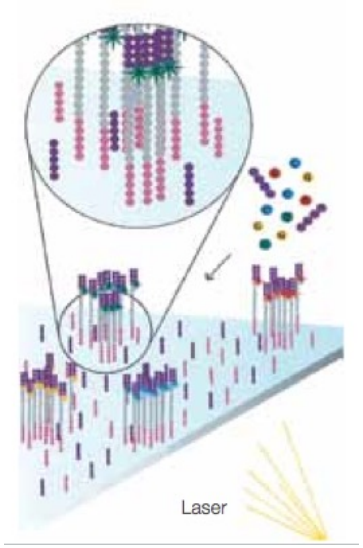
>60Gbp / day



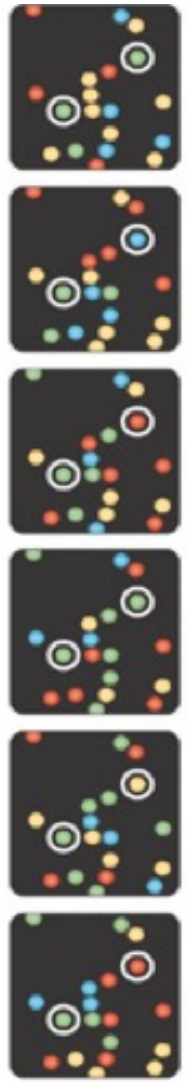
1. Attach



2. Amplify

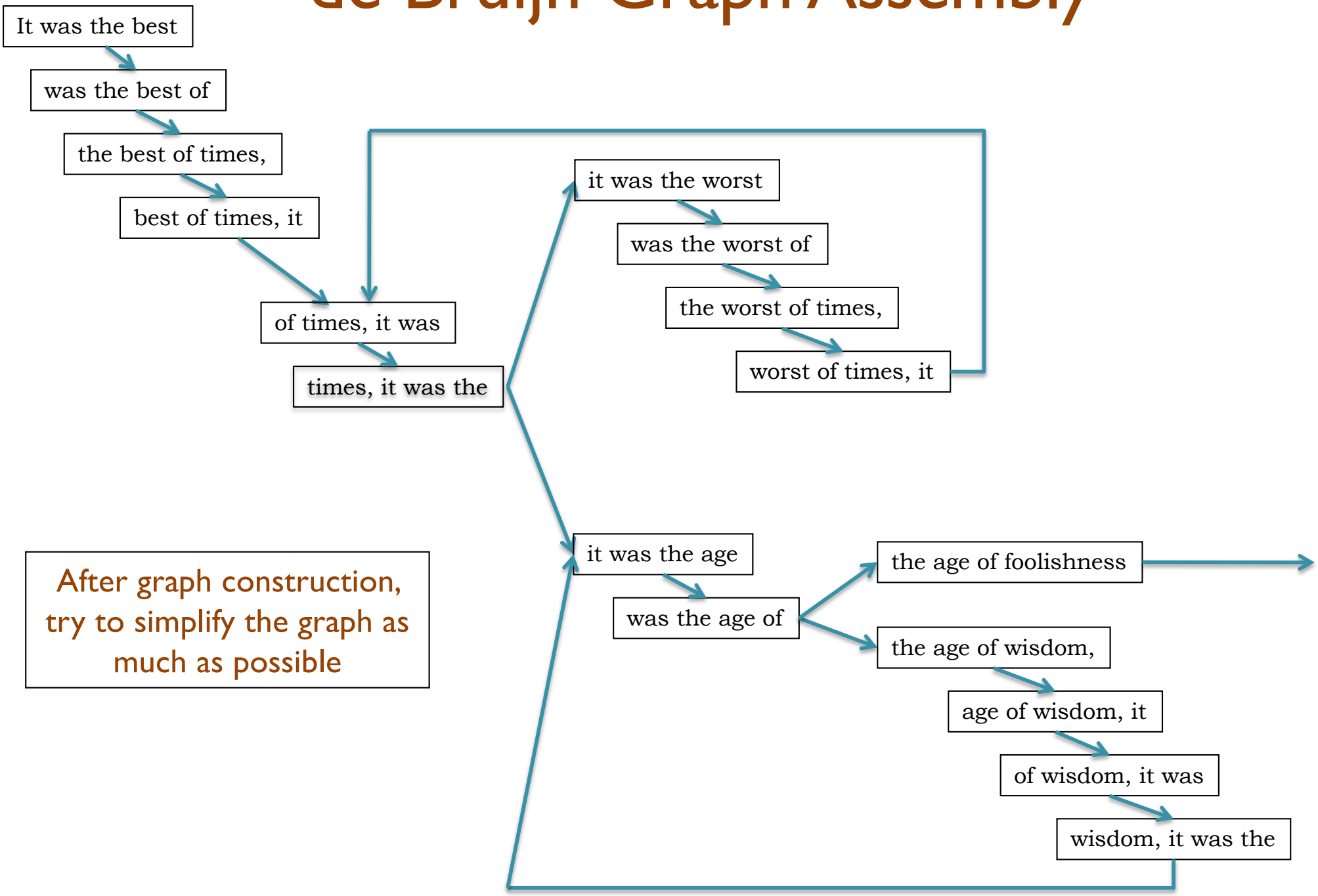


3. Image



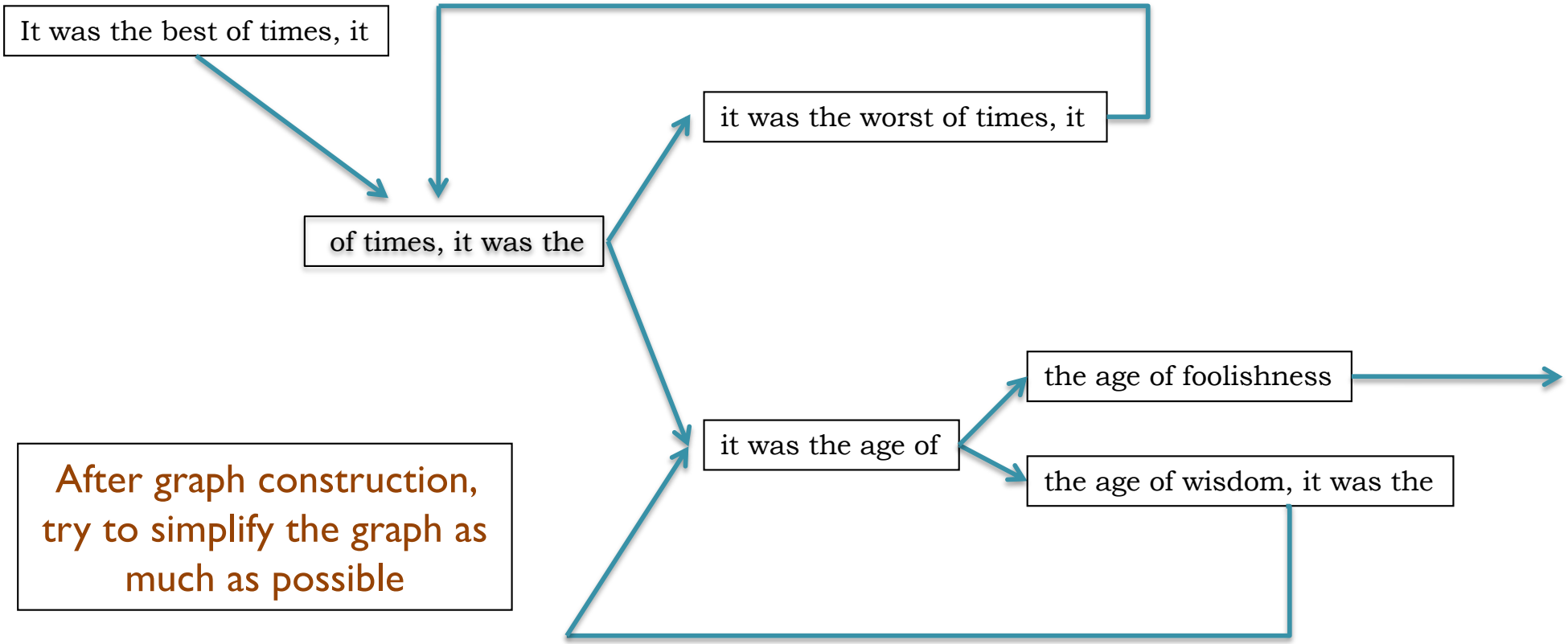
Metzker (2010) Nature Reviews Genetics 11:31-46
https://www.youtube.com/watch?v=fCd6B5HRaZ8

de Bruijn Graph Assembly



After graph construction, try to simplify the graph as much as possible

de Bruijn Graph Assembly



After graph construction, try to simplify the graph as much as possible

The full tale

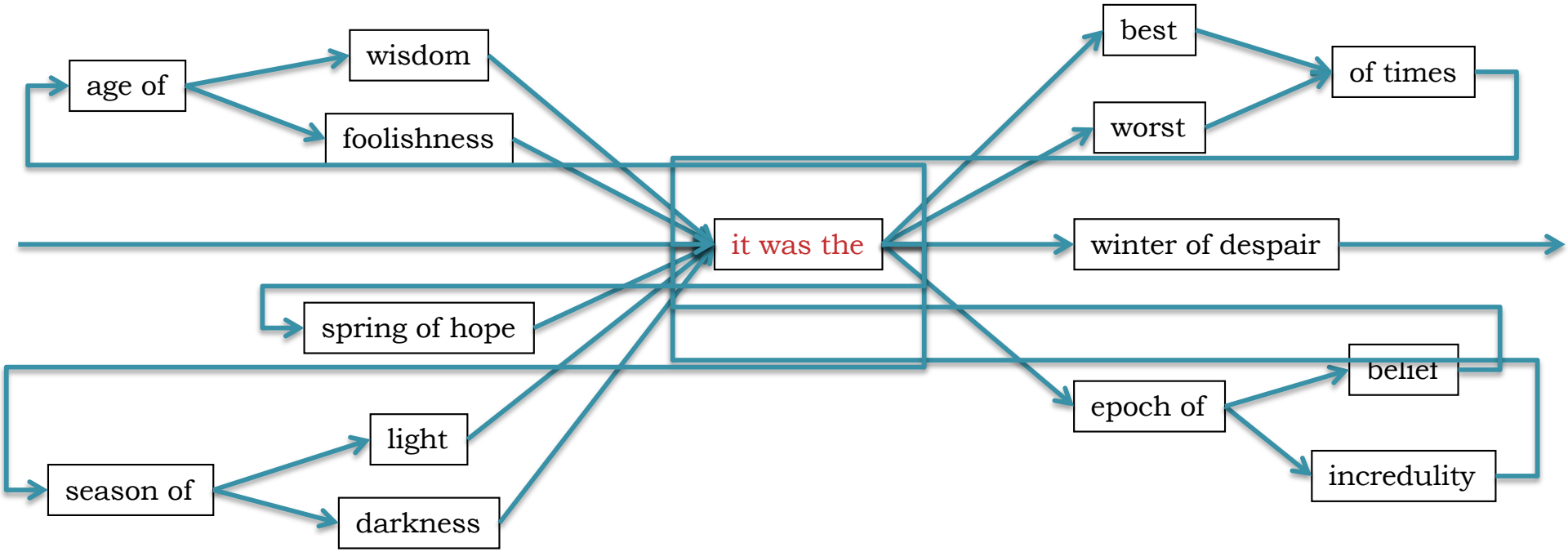
… it was the best of times it was the worst of times …

… it was the age of wisdom it was the age of foolishness …

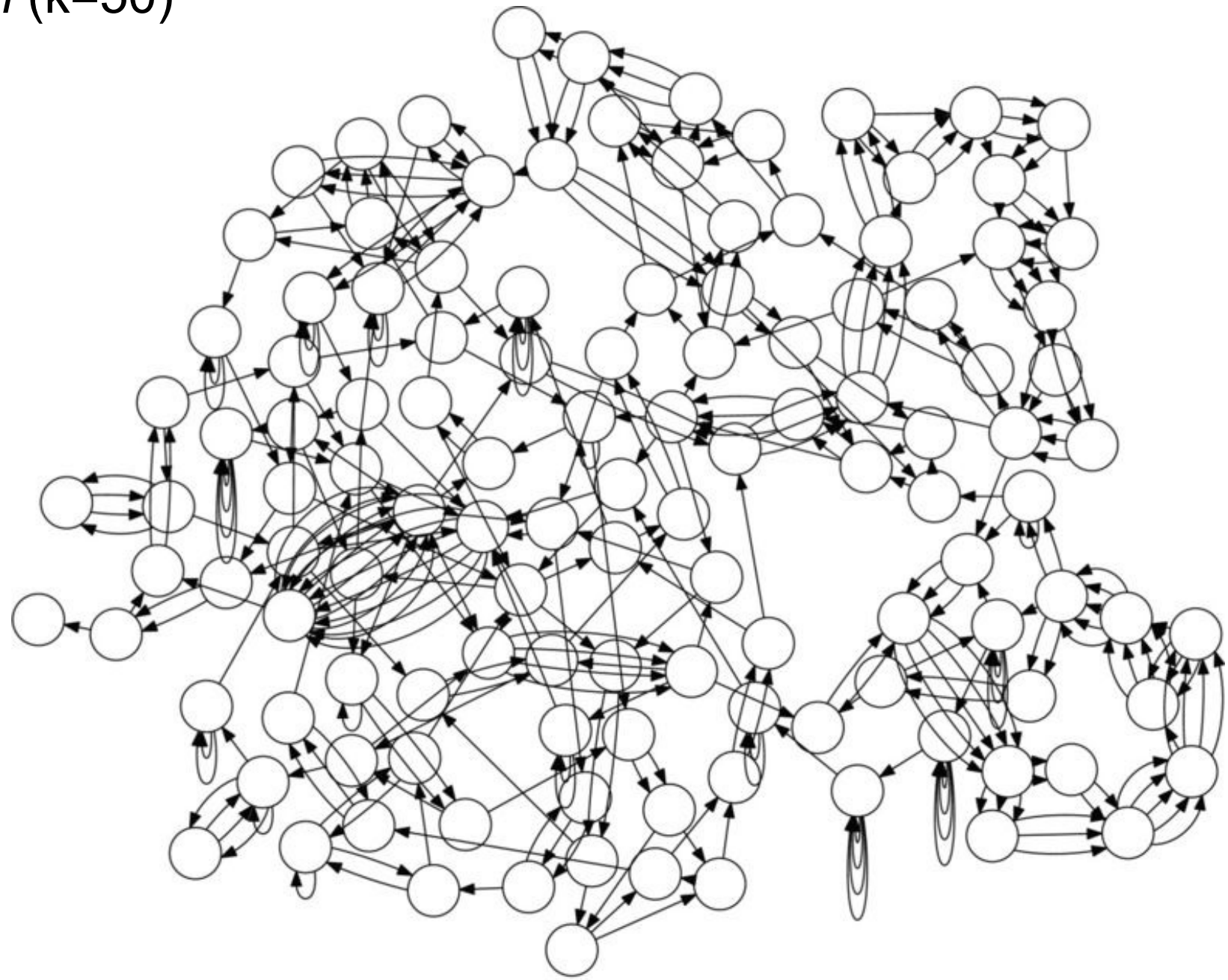
… it was the epoch of belief it was the epoch of incredulity …

… it was the season of light it was the season of darkness …

… it was the spring of hope it was the winder of despair …

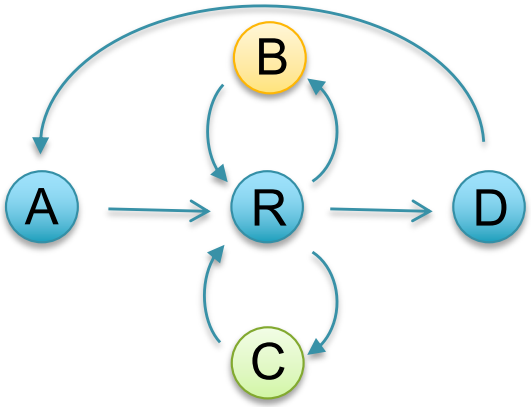


E. coli (k=50)



Reducing assembly complexity of microbial genomes with single-molecule sequencing
Koren et al (2013) Genome Biology. **14**:R101 https://doi.org/10.1186/gb-2013-14-9-r101

Counting Eulerian Cycles



ARBRCRD
or
ARCRBRD

Generally an exponential number of compatible sequences

– Value computed by application of the BEST theorem (Hutchinson, 1975)

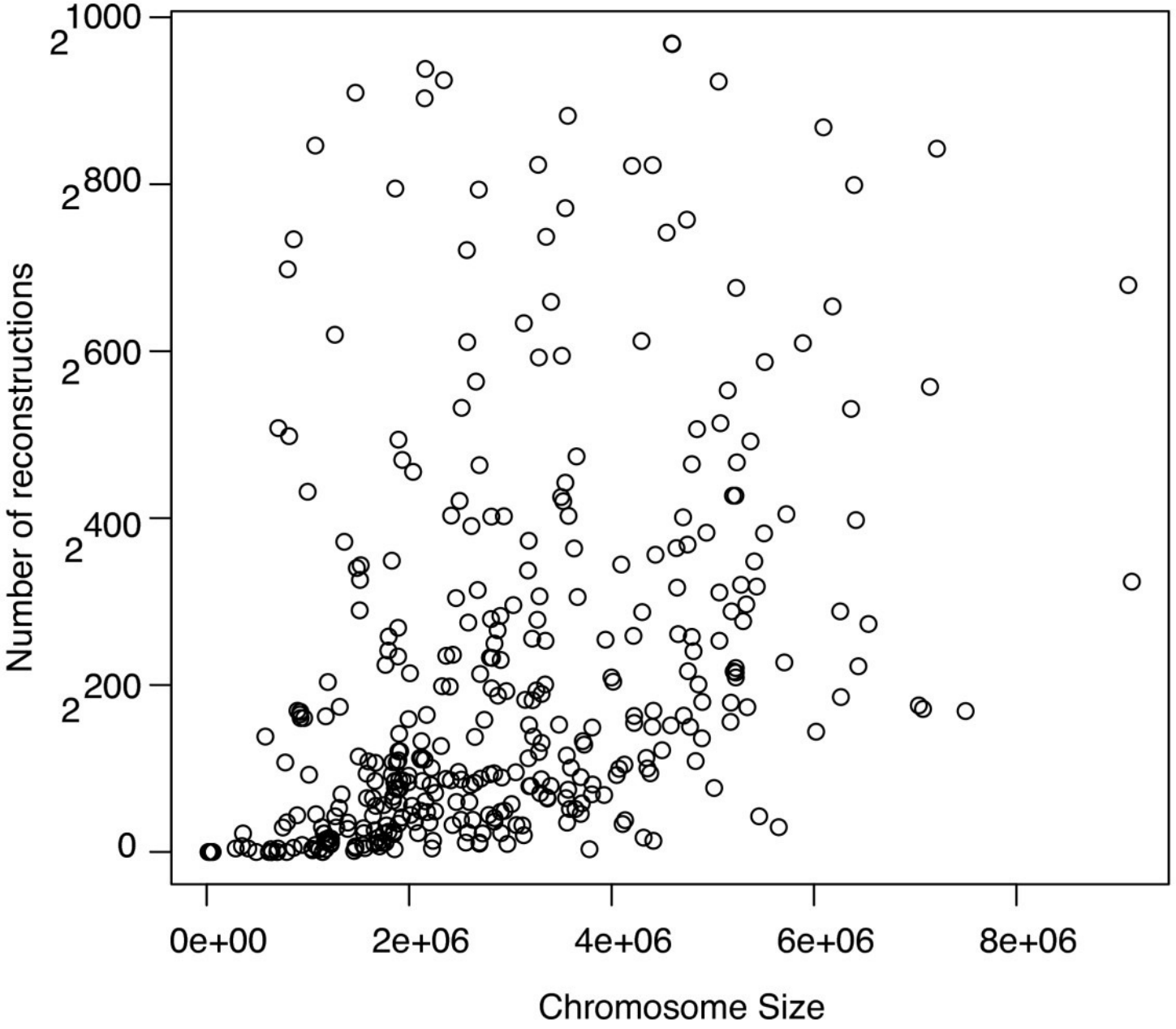
$$\mathcal{W}(G,t) = (\det L)\left\{\prod_{u\in V}(r_u-1)!\right\}\left\{\prod_{(u,v)\in E} a_{uv}!\right\}^{-1}$$

L = *n x n* matrix with r_u-a_{uu} along the diagonal and -a_{uv} in entry uv

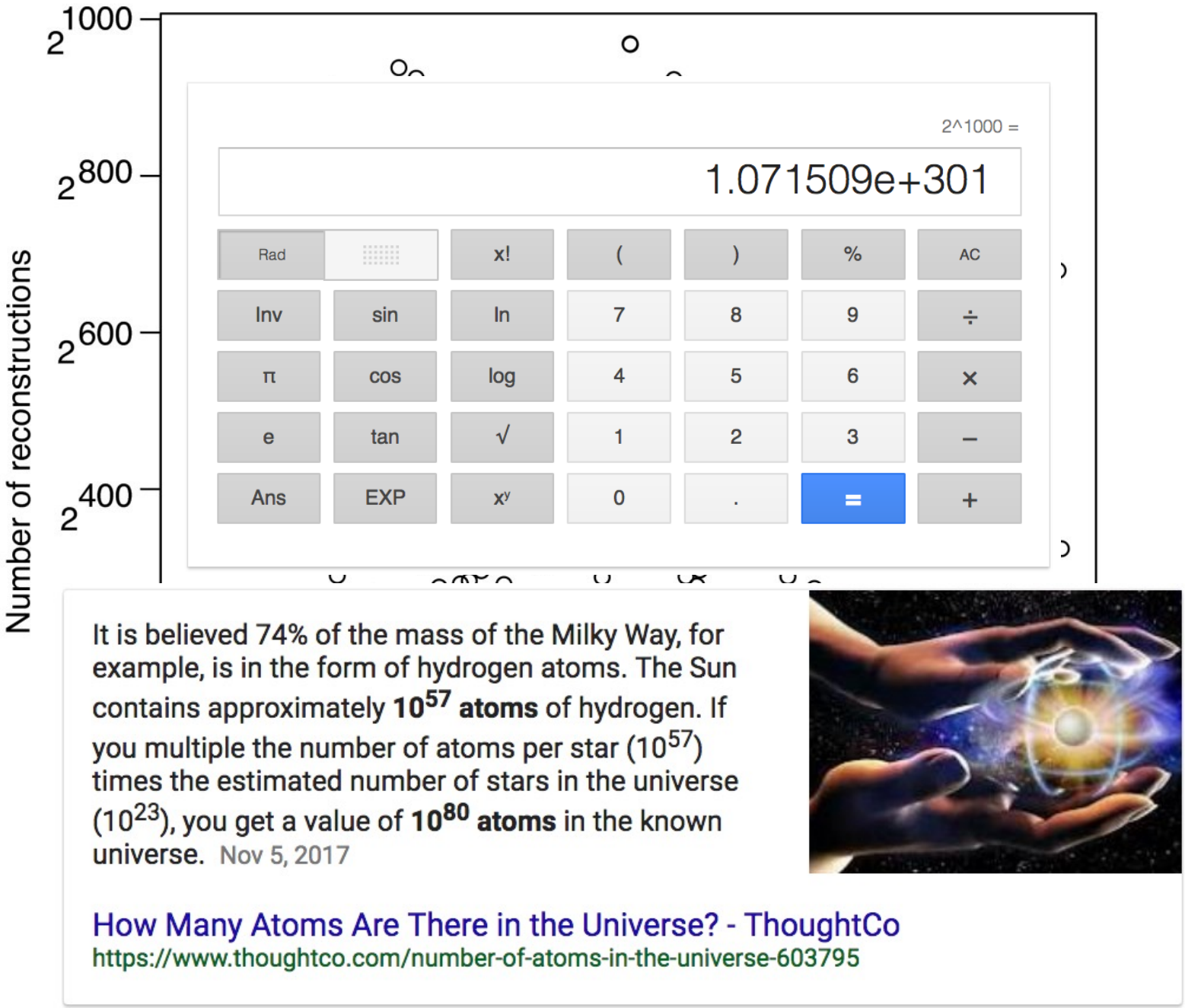
$r_u = d^+(u)+1$ if $u=t$, or $d^+(u)$ otherwise

a_{uv} = multiplicity of edge from *u* to *v*

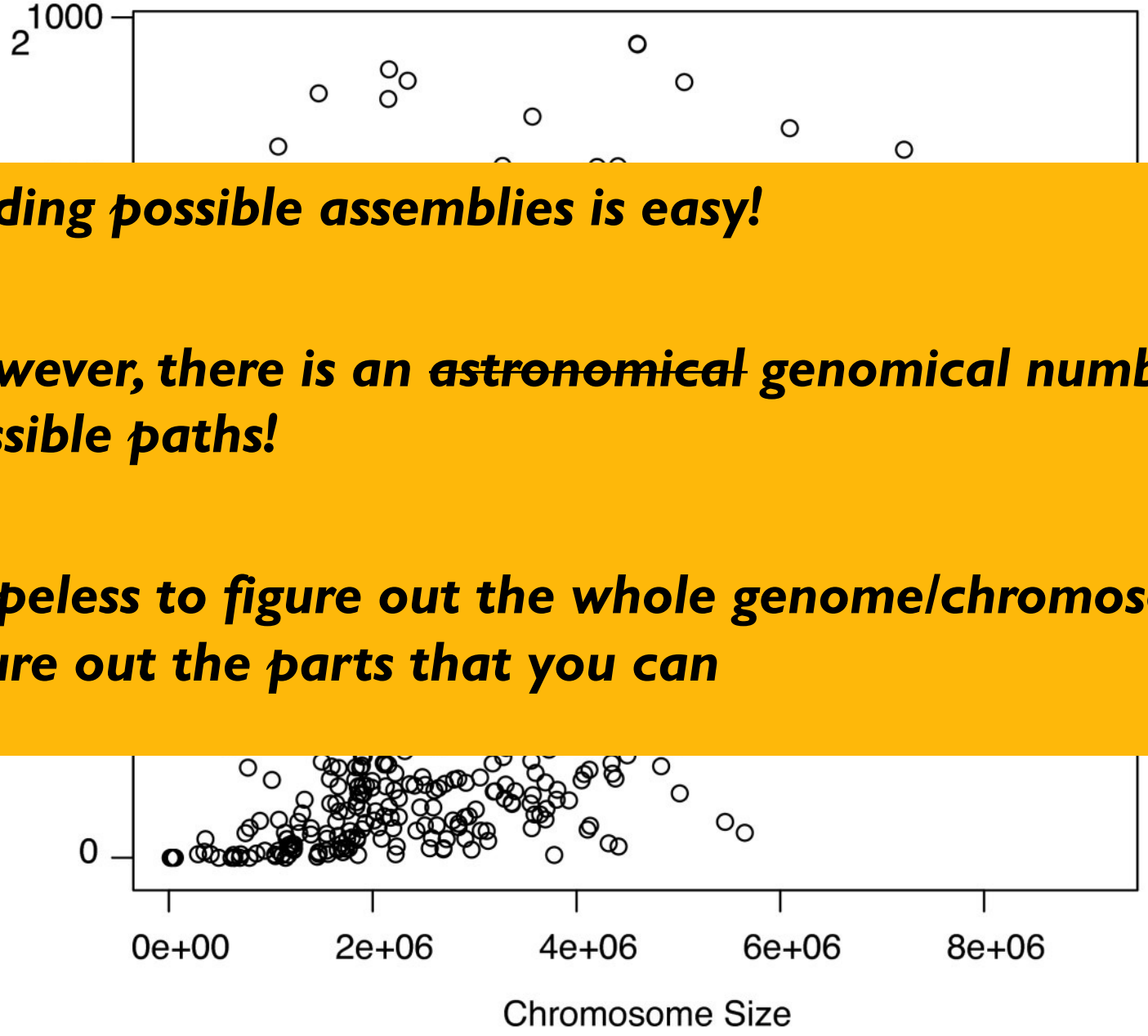
Assembly Complexity of Prokaryotic Genomes using Short Reads.
Kingsford C, Schatz MC, Pop M (2010) *BMC Bioinformatics*.



Assembly Complexity of Prokaryotic Genomes using Short Reads.
Kingsford C, Schatz MC, Pop M (2010) *BMC Bioinformatics*.



Assembly Complexity of Prokaryotic Genomes using Short Reads.
Kingsford C, Schatz MC, Pop M (2010) *BMC Bioinformatics*.



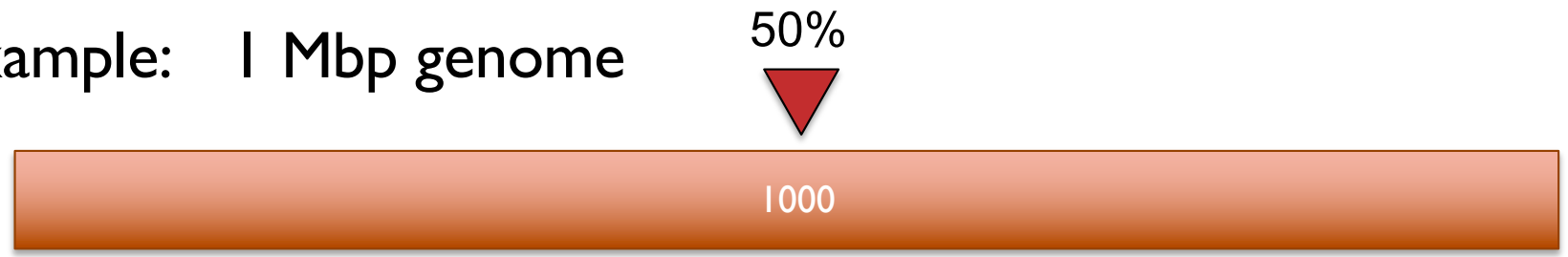
- ***Finding possible assemblies is easy!***
- ***However, there is an ~~astronomical~~ genomical number of possible paths!***
- ***Hopeless to figure out the whole genome/chromosome, figure out the parts that you can***

Assembly Complexity of Prokaryotic Genomes using Short Reads.
Kingsford C, Schatz MC, Pop M (2010) *BMC Bioinformatics*.

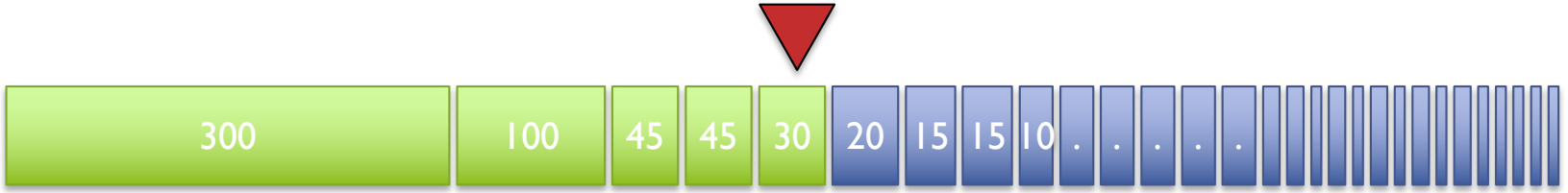
Contig N50

Def: 50% of the genome is in contigs as large as the N50 value

Example: 1 Mbp genome

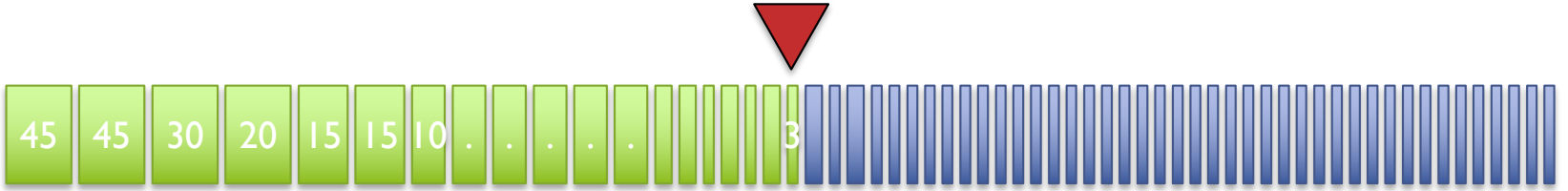


A



N50 size = 30 kbp

B



N50 size = 3 kbp

Contig N50

Def: 50% of the genome is in contigs as large as the N50 value

Better N50s improves the analysis in every dimension

- Better resolution of genes and flanking regulatory regions
- Better resolution of transposons and other complex sequences
- Better resolution of chromosome organization
- Better sequence for all downstream analysis

Just be careful of N50 inflation!

- *A very very very bad assembler in 1 line of bash:*
- *cat *.reads.fa > genome.fa*

N50 size = 3 kbp

Pop Quiz 1

Assemble these reads using a de Bruijn graph approach (k=3):

```
ATTA
GATT
TACA
TTAC
```

Pop Quiz 1

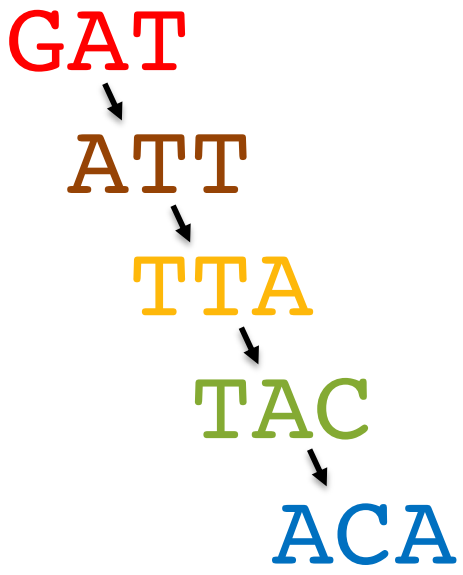
Assemble these reads using a de Bruijn graph approach (k=3):

```
ATTA: ATT -> TTA
GATT: GAT -> ATT
TACA: TAC -> ACA
TTAC: TTA -> TAC
```

Pop Quiz 1

Assemble these reads using a de Bruijn graph approach (k=3):

```
ATTA:  ATT -> TTA
GATT:  GAT -> ATT
TACA:  TAC -> ACA
TTAC:  TTA -> TAC
```



GATTACA

Pop Quiz 2

Assemble these reads using a de Bruijn graph approach (k=3):

```
ACGA
ACGT
ATAC
CGAC
CGTA
GACG
GTAT
TACG
```

Pop Quiz 2

Assemble these reads using a de Bruijn graph approach (k=3):

~~ACGA~~

ACGT

ATAC

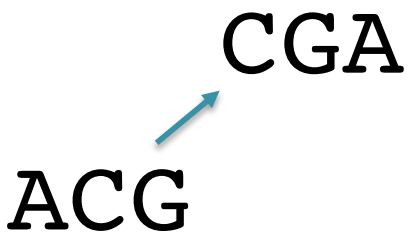
CGAC

CGTA

GACG

GTAT

TACG



Pop Quiz 2

Assemble these reads using a de Bruijn graph approach (k=3):

~~ACGA~~

~~ACGT~~

ATAC

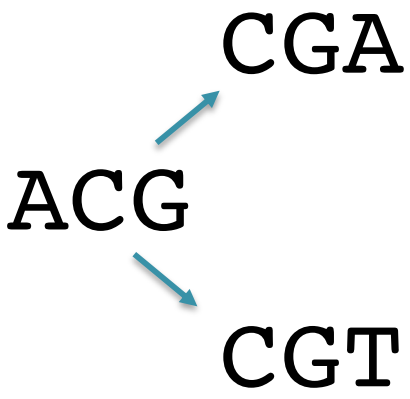
CGAC

CGTA

GACG

GTAT

TACG



Pop Quiz 2

Assemble these reads using a de Bruijn graph approach (k=3):

~~ACGA~~

~~ACGT~~

ATAC

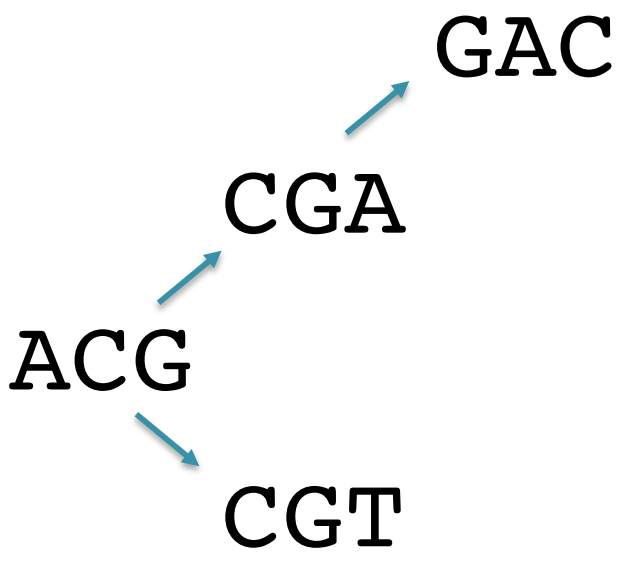
~~CGAC~~

CGTA

GACG

GTAT

TACG



Pop Quiz 2

Assemble these reads using a de Bruijn graph approach (k=3):

~~ACGA~~

~~ACGT~~

ATAC

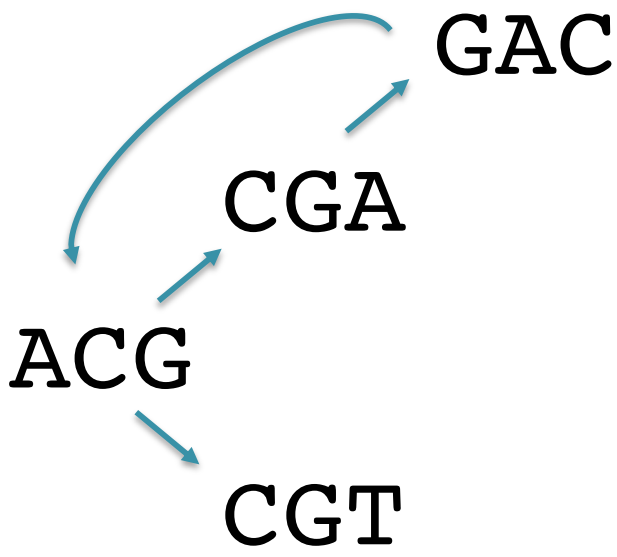
~~CGAC~~

CGTA

~~GACG~~

GTAT

TACG



Pop Quiz 2

Assemble these reads using a de Bruijn graph approach (k=3):

~~ACGA~~

~~ACGT~~

ATAC

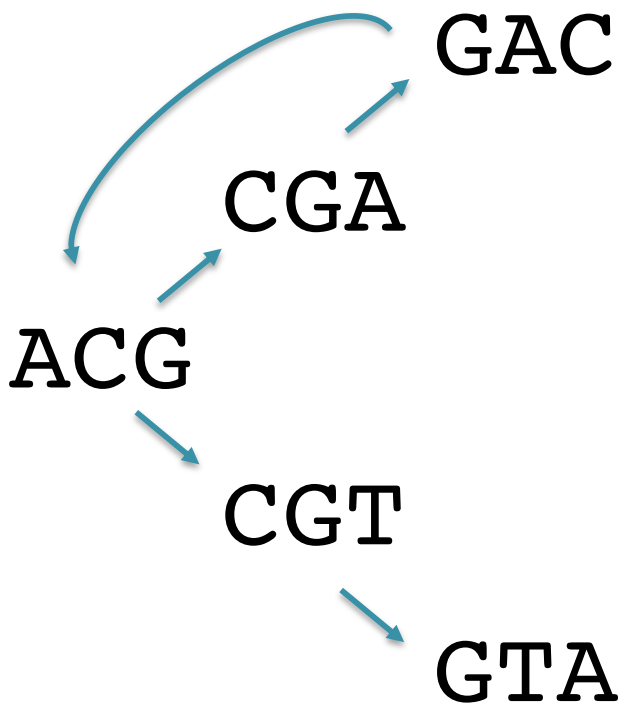
~~CGAC~~

~~CGTA~~

~~GACG~~

GTAT

TACG



Pop Quiz 2

Assemble these reads using a de Bruijn graph approach (k=3):

~~ACGA~~

~~ACGT~~

ATAC

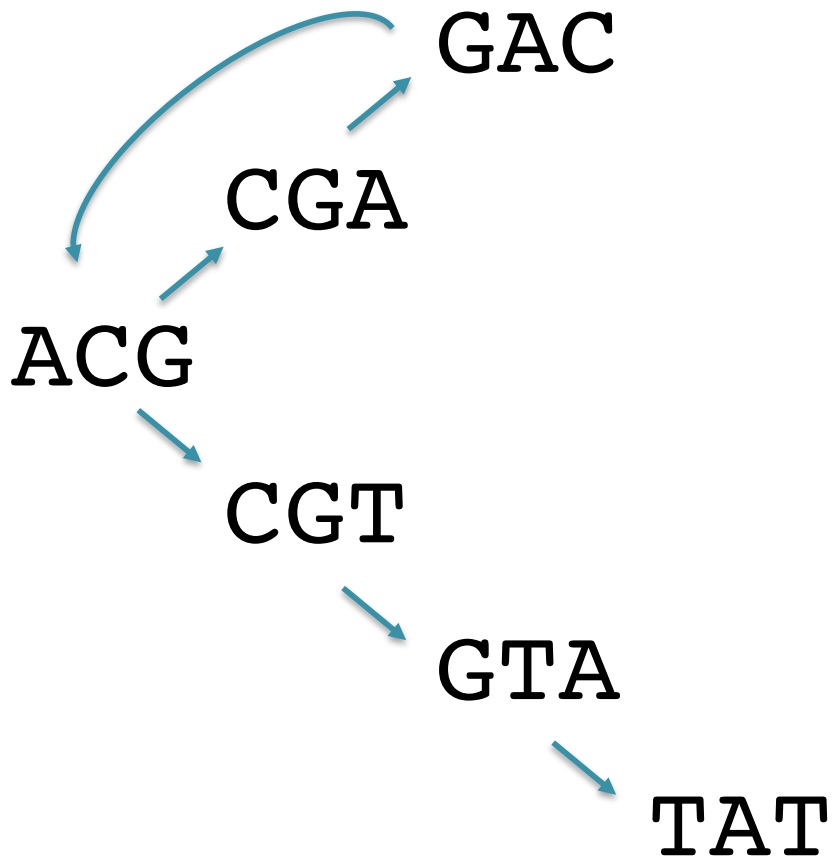
~~CGAC~~

~~CGTA~~

~~GACG~~

~~GTAT~~

TACG



Pop Quiz 2

Assemble these reads using a de Bruijn graph approach (k=3):

~~ACGA~~

~~ACGT~~

ATAC

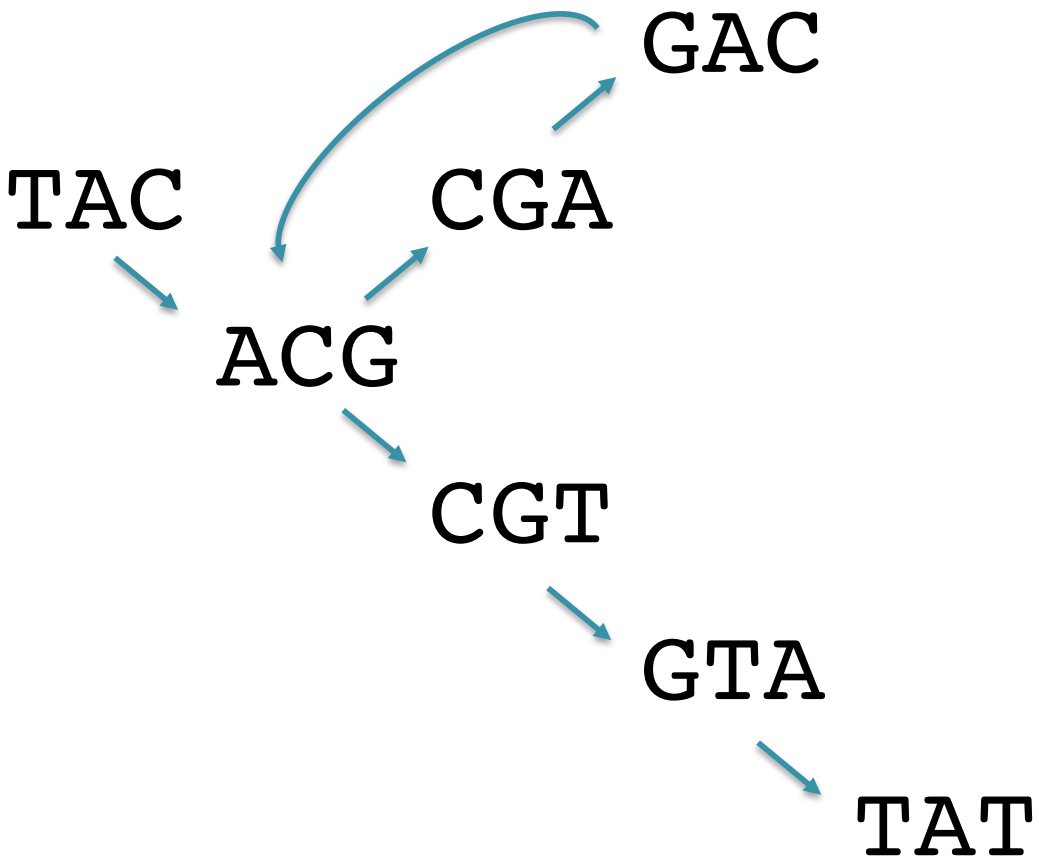
~~CGAC~~

~~CGTA~~

~~GACG~~

~~GTAT~~

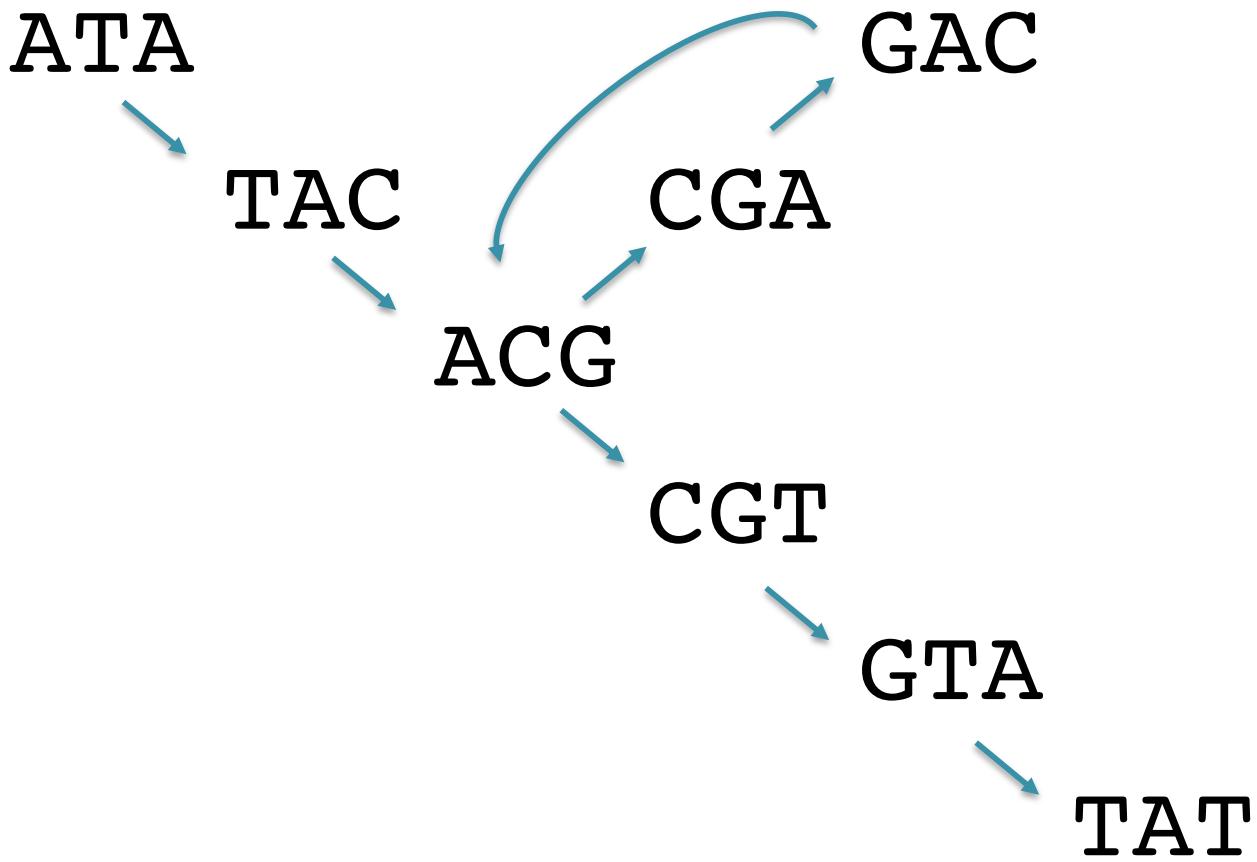
~~TACG~~



Pop Quiz 2

Assemble these reads using a de Bruijn graph approach (k=3):

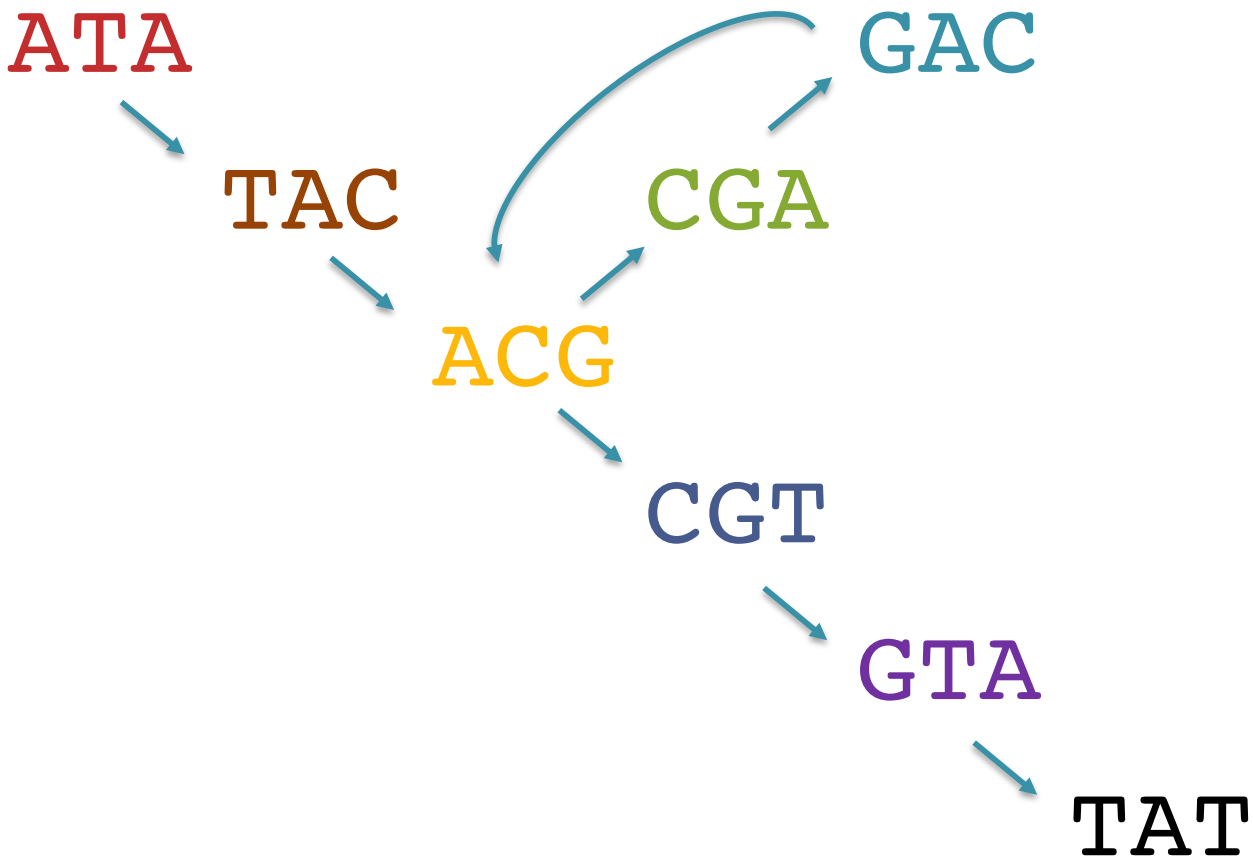
~~ACGA~~
~~ACGT~~
~~ATAC~~
~~CGAC~~
~~CGTA~~
~~GACG~~
~~GTAT~~
~~TACG~~



Pop Quiz 2

Assemble these reads using a de Bruijn graph approach (k=3):

~~ACGA~~
~~ACGT~~
~~ATAC~~
~~CGAC~~
~~CGTA~~
~~GACG~~
~~GTAT~~
~~TACG~~

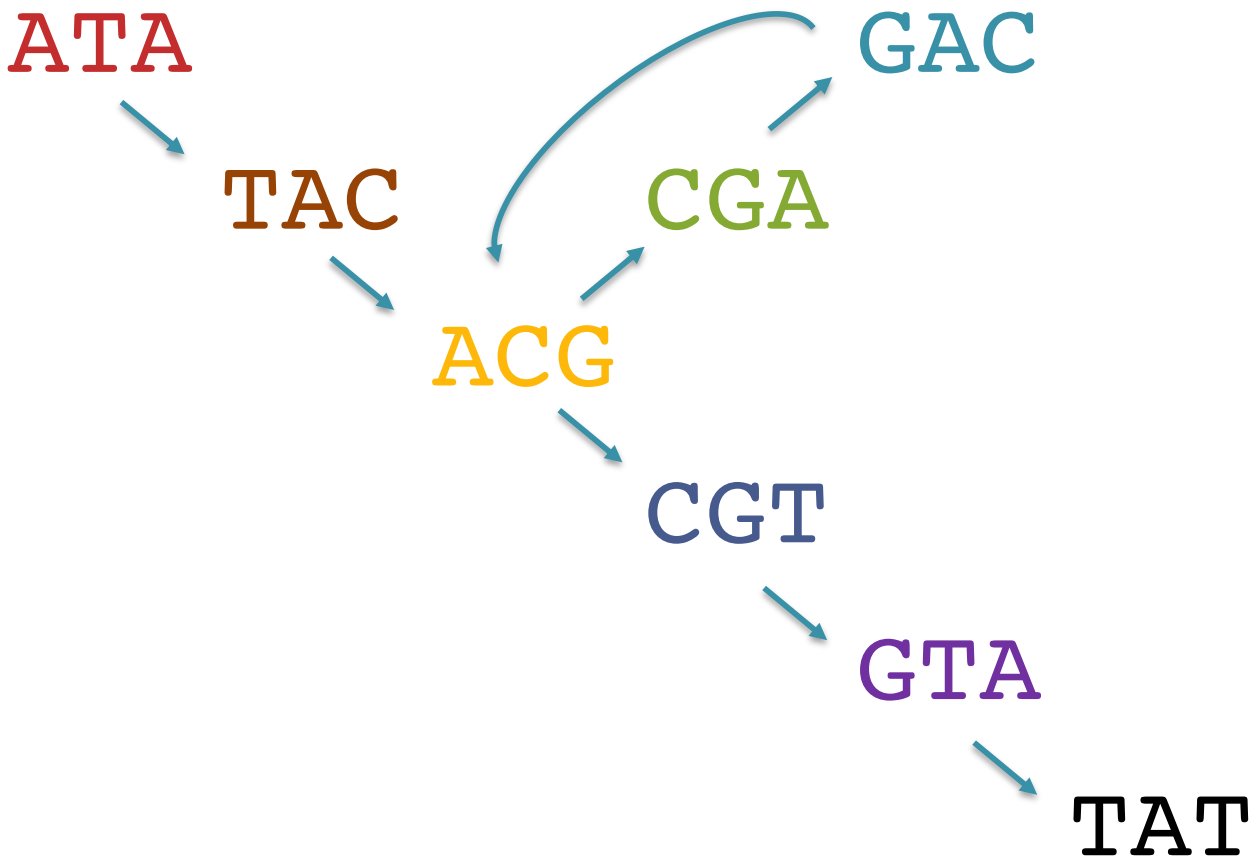


ATACGACGTAT

Pop Quiz 2

Assemble these reads using a de Bruijn graph approach (k=3):

~~ACGA~~
~~ACGT~~
~~ATAC~~
~~CGAC~~
~~CGTA~~
~~GACG~~
~~GTAT~~
~~TACG~~



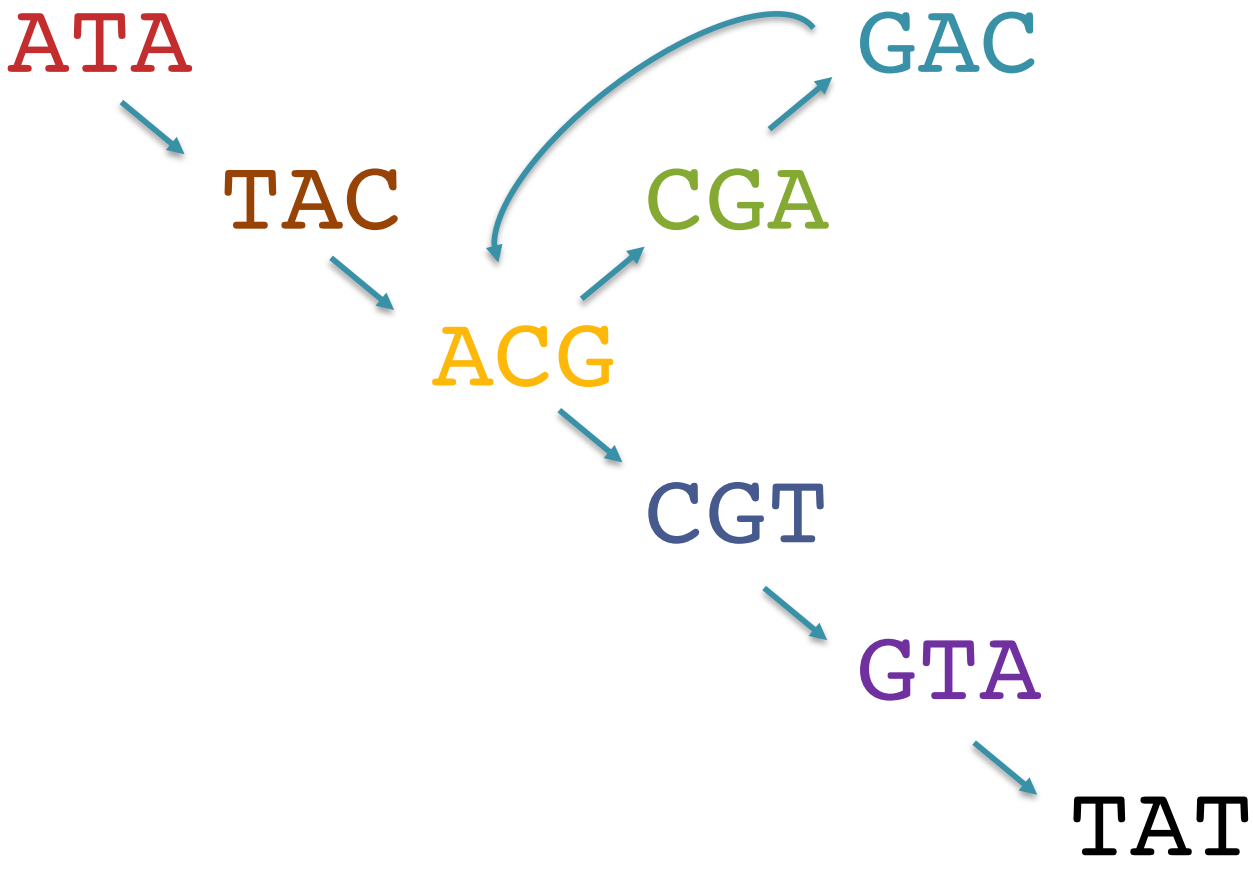
Whats another possible genome?

ATACGACGTAT

Pop Quiz 2

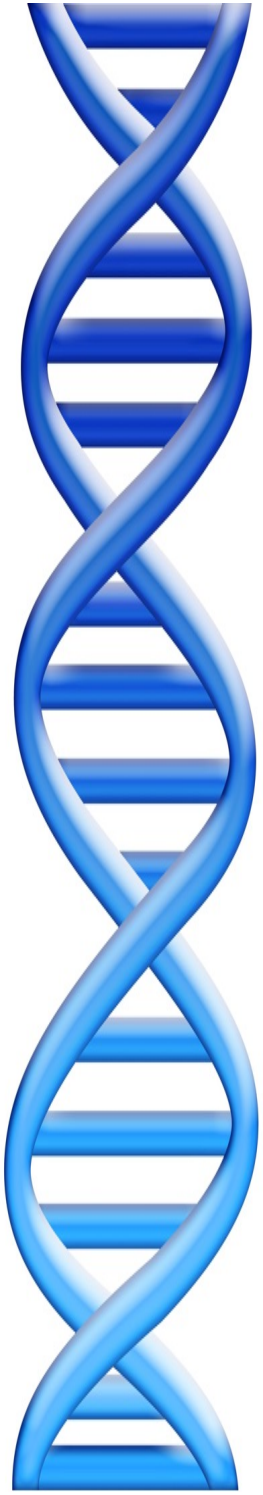
Assemble these reads using a de Bruijn graph approach (k=3):

~~ACGA~~
~~ACGT~~
~~ATAC~~
~~CGAC~~
~~CGTA~~
~~GACG~~
~~GTAT~~
~~TACG~~



Should we add the edge TAT -> ATA?

ATACGACGTAT



Outline

1. ***Assembly theory***
   - Assembly by analogy

2. ***Practical Issues***
   - Coverage, read length, errors, and repeats

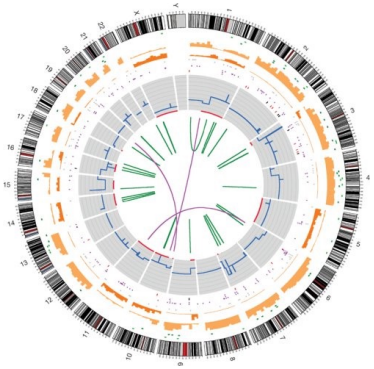
3. ***Whole Genome Alignment***
   - MUMmer recommended

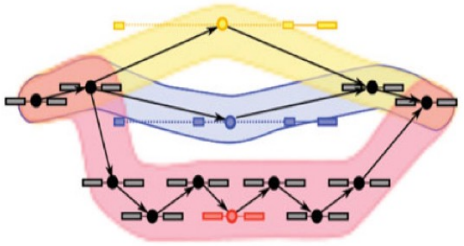
Assembly Applications

- Novel genomes

- Metagenomes

- Sequencing assays
  - Structural variations
  - Transcript assembly
  - …





Why are genomes hard to assemble?

1. ***Biological***:
   - (Very) High ploidy, heterozygosity, repeat content

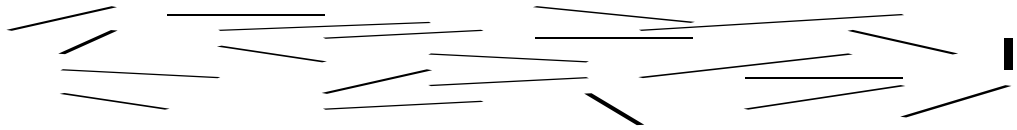
2. ***Sequencing***:
   - (Very) large genomes, imperfect sequencing

3. ***Computational***:
   - (Very) Large genomes, complex structure

4. ***Accuracy***:
   - (Very) Hard to assess correctness

Assembling a Genome

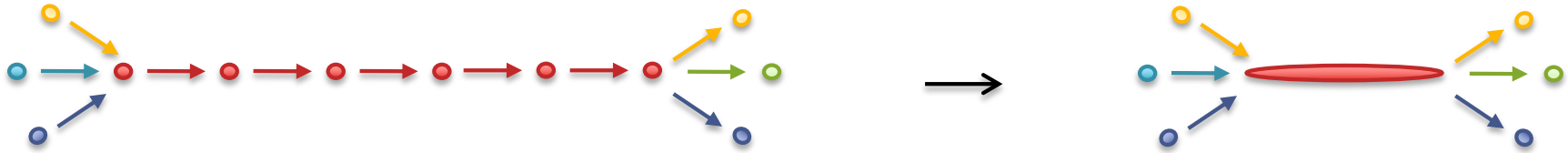
1. Shear & Sequence DNA



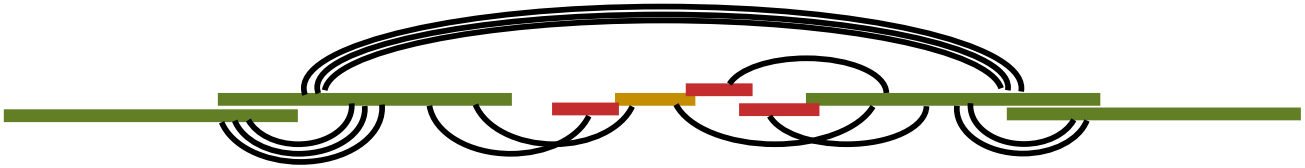
2. Construct assembly graph from reads (de Bruijn / overlap graph)

...AGCCTAGGGATGCGCGACACGT
GGATGCGCGACACGTCGCATATCCGGTTTGGTCAACCTCGGACGGAC
CAACCTCGGACGGACCTCAGCGAA...

3. Simplify assembly graph

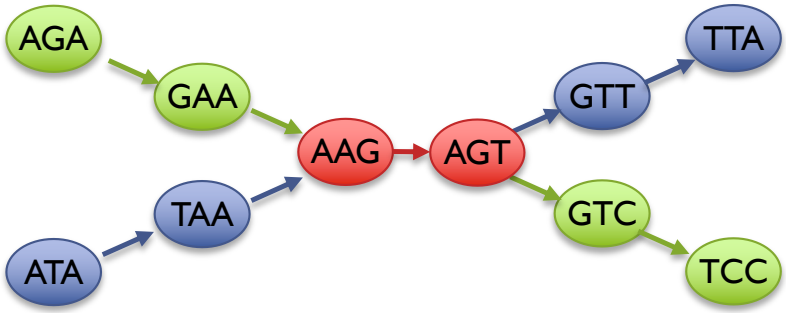


4. Detangle graph with long reads, mates, and other links



Two Paradigms for Assembly

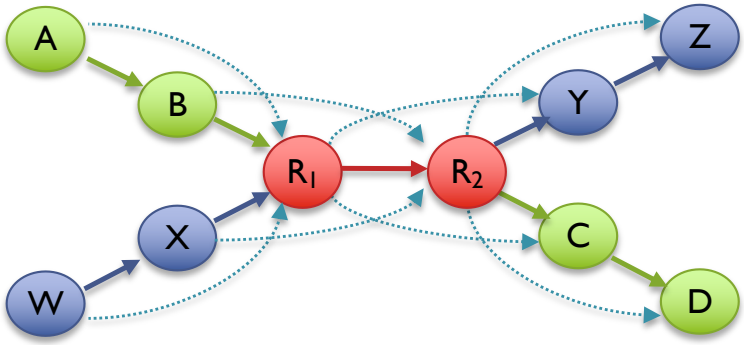
de Bruijn Graph



Short read assemblers

- Repeats depends on word length
- Read coherency, placements lost
- Robust to high coverage

Overlap Graph



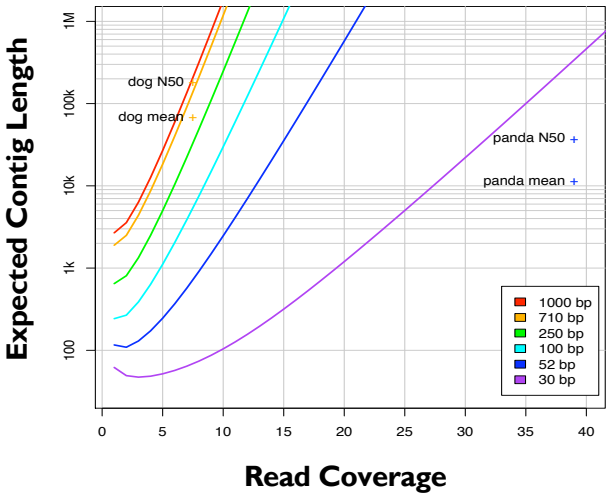
Long read assemblers

- Repeats depends on read length
- Read coherency, placements kept
- Tangled by high coverage

Assembly of Large Genomes using Second Generation Sequencing
Schatz MC, Delcher AL, Salzberg SL (2010) *Genome Research*. 20:1165-1173.

Ingredients for a good assembly

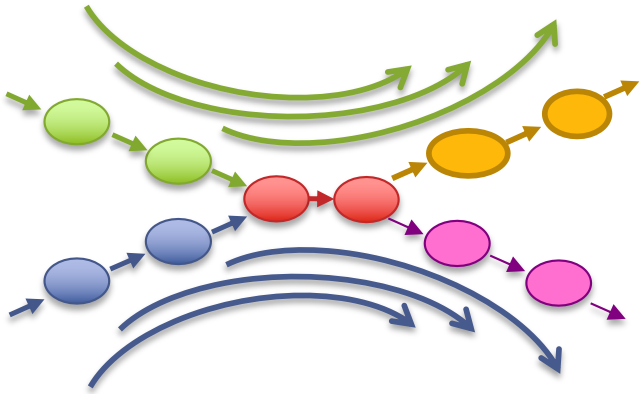
Coverage



High coverage is required

- Oversample the genome to ensure every base is sequenced with long overlaps between reads
- Biased coverage will also fragment assembly

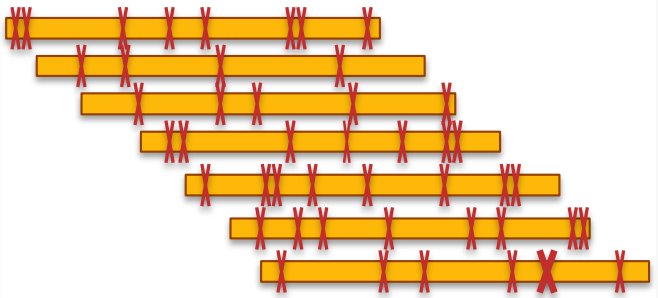
Read Length



Reads & mates must be longer than the repeats

- Short reads will have ***false overlaps*** forming hairball assembly graphs
- With long enough reads, assemble entire chromosomes into contigs

Quality



Errors obscure overlaps

- Reads are assembled by finding kmers shared in pair of reads
- High error rate requires very short seeds, increasing complexity and forming assembly hairballs

Current challenges in *de novo* plant genome sequencing and assembly
Schatz MC, Witkowski, McCombie, WR (2012) *Genome Biology*. 12:243

Coverage Statistics

$$\text{sequencing\_coverage} = \frac{\text{total\_bases\_sequenced}}{\text{genome\_size}}$$

$$\text{genome\_size} = \frac{\text{total\_bases\_sequenced}}{\text{sequencing\_coverage}}$$

$$\text{genome\_size} = \frac{100\text{Gb}}{50\text{x}} = 2\text{Gb}$$

But how can you figure out
the coverage without a genome?

K-mer counting

Kmer-ize

```
Read 1: GATTACA  => GAT,ATT,TTA,TAC,ACA
Read 2: TACAGAG  => TAC,ACA,CAG,AGA,GAG
Read 3: TTACAGA  => TTA,TAC,ACA,CAG,AGA
```

list

```
GAT   ACA   ACA:3
ATT   ACA
TTA   ACA
TAC   AGA   AGA:2
ACA   AGA
TAC   ATT   ATT:1
ACA   CAG   CAG:2
CAG   CAG
AGA   GAG   GAG:1
GAG   GAT   GAT:1
TTA   TAC   TAC:3
TAC   TAC
ACA   TAC
CAG   TTA   TTA:2
AGA   TTA
```

sort count

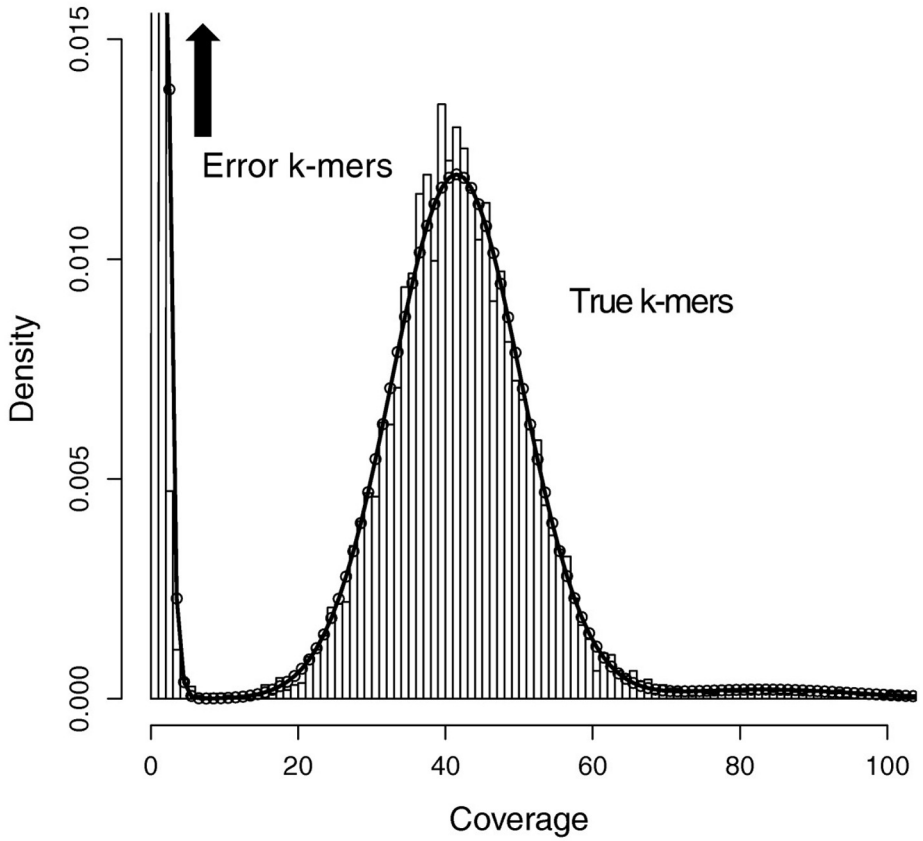
tally

```
3 kmers occur 1x
3 kmers occur 2x
2 kmers occur 3x
```

From read k-mers alone, can learn something about how frequently different sequences occur (aka coverage)

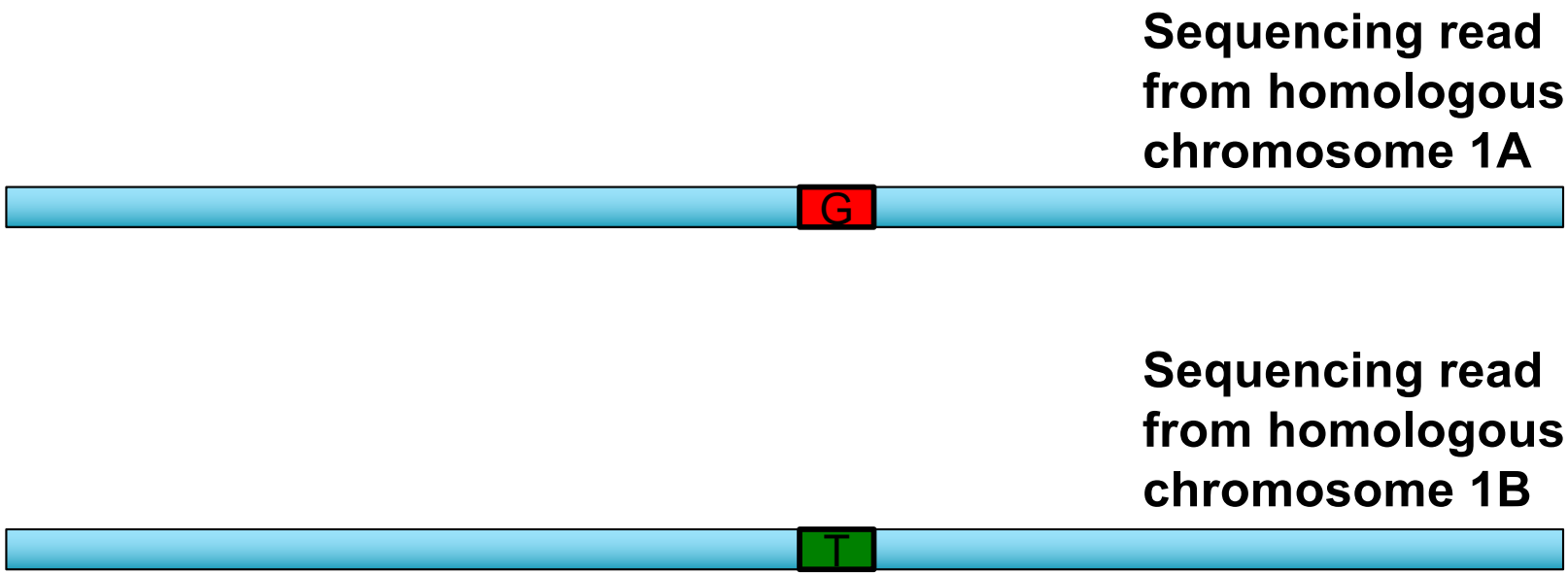
Fast to compute even over huge datasets

K-mer counting in real genomes

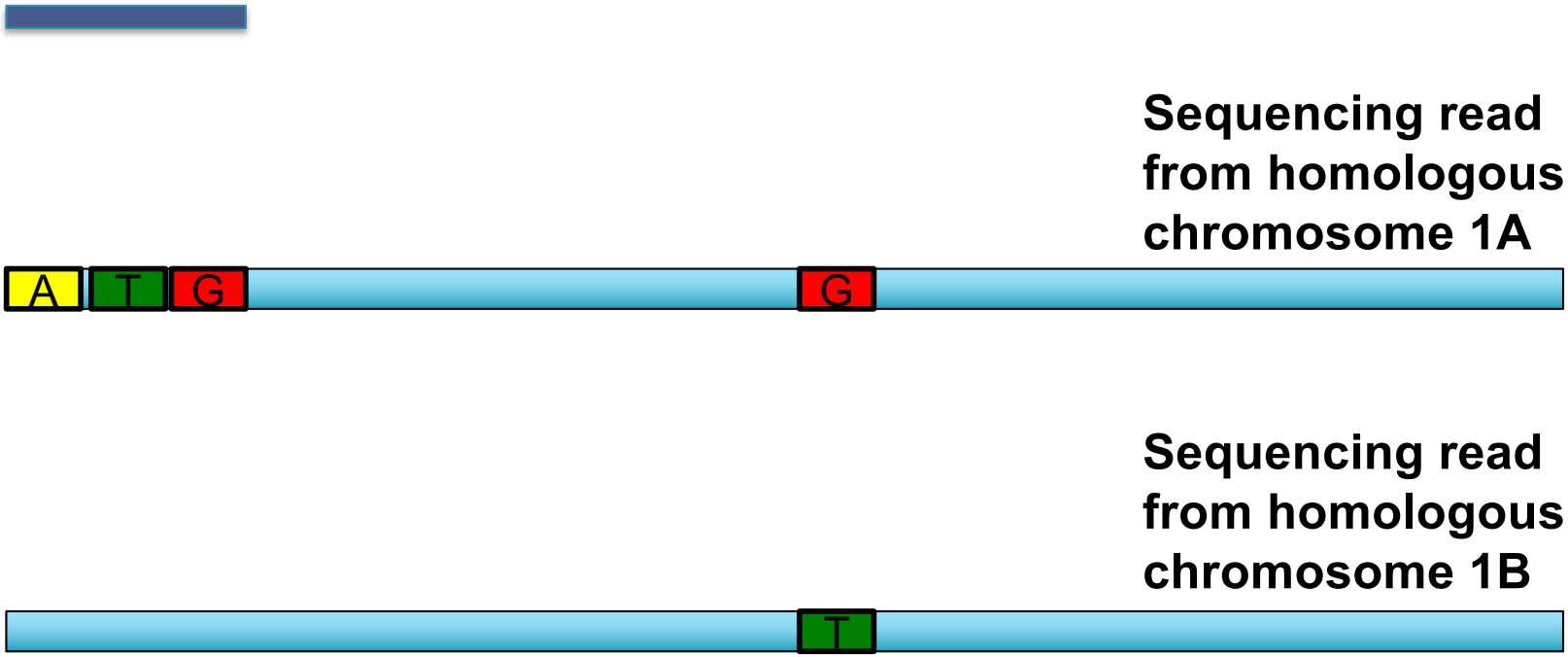


- The tally of k-mer counts in real genomes reveals the coverage distribution.
- Here we sequenced 120Gb of reads from a female human (haploid human genome size is 3Gb), and indeed we see a clear peak centered at 40x coverage
- There are also many kmers that only occur <5 times. These are from errors in the reads
- There are also kmers that occur many times (>>70 times). These are repeats in the genome

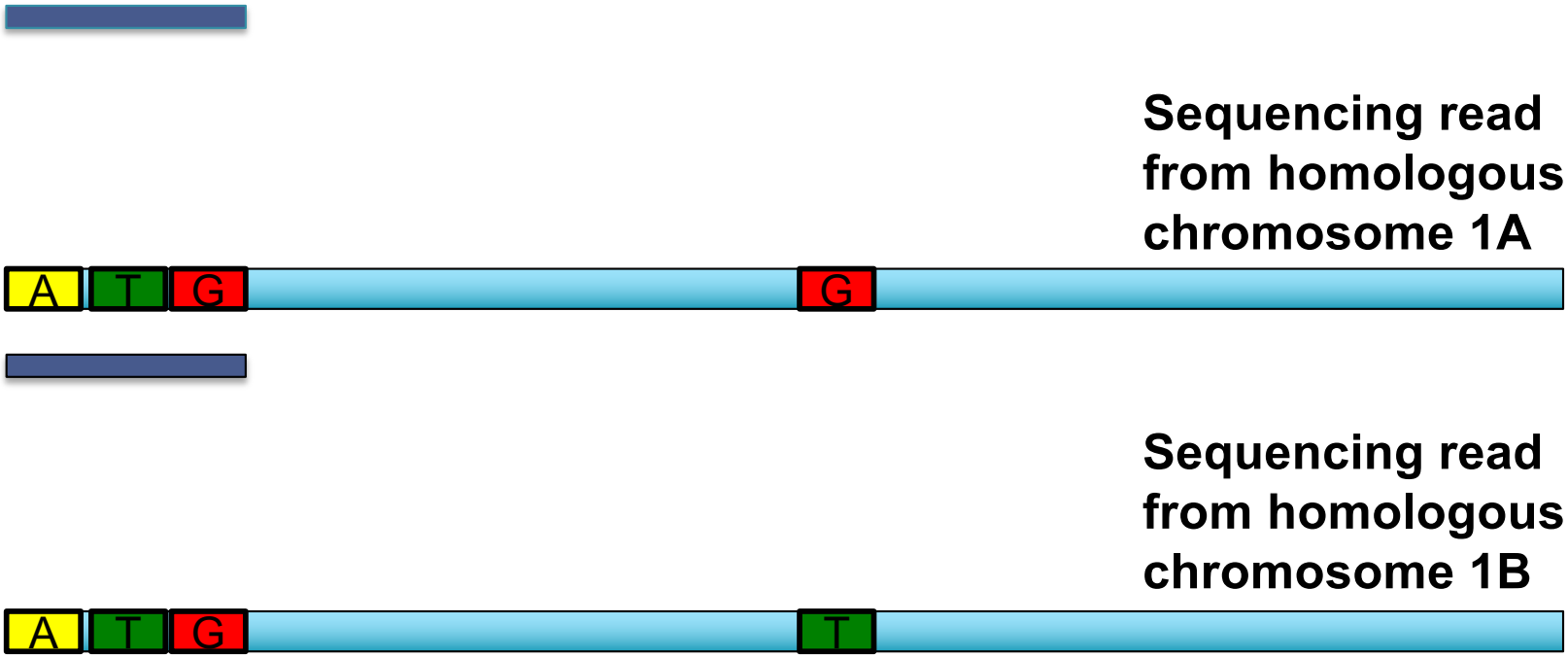
K-mer counting in heterozygous genomes



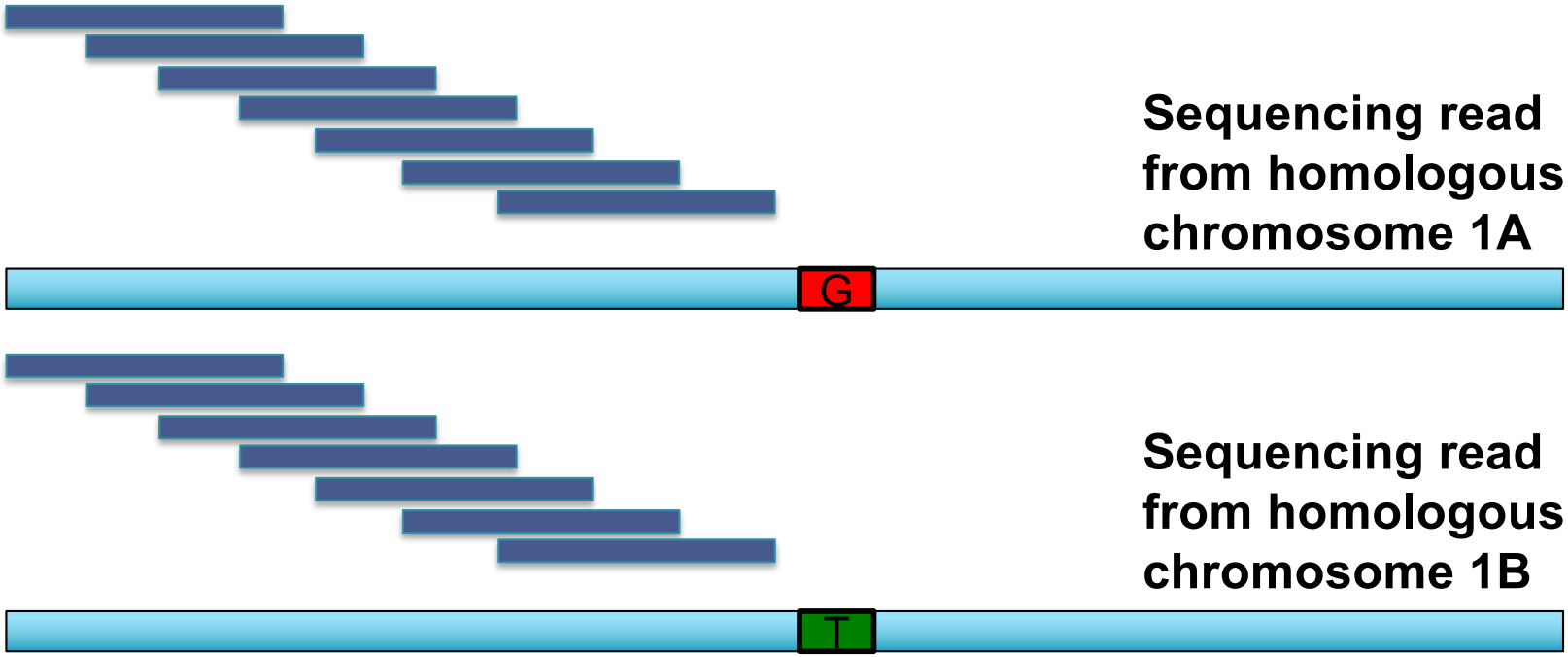
K-mer counting in heterozygous genomes



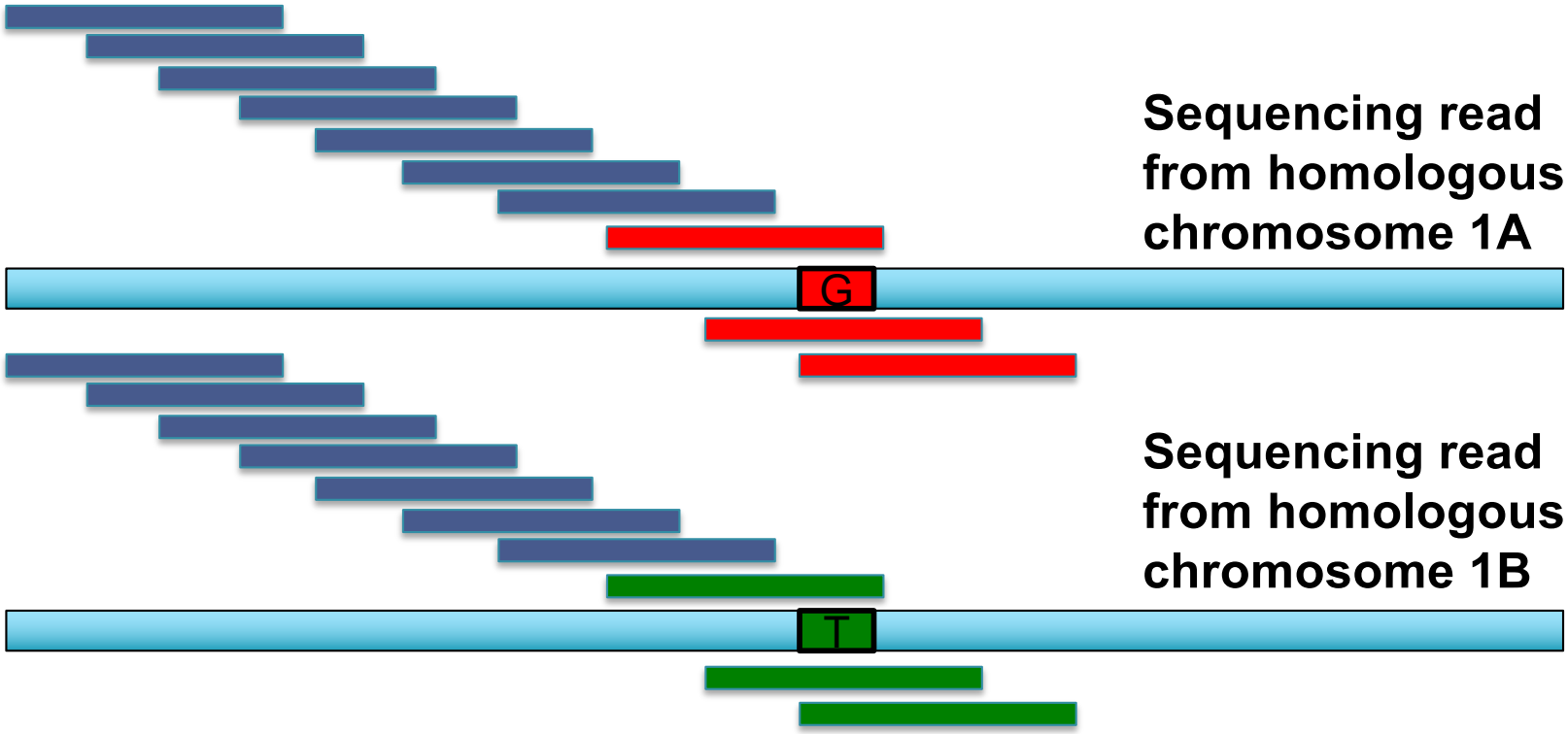
K-mer counting in heterozygous genomes



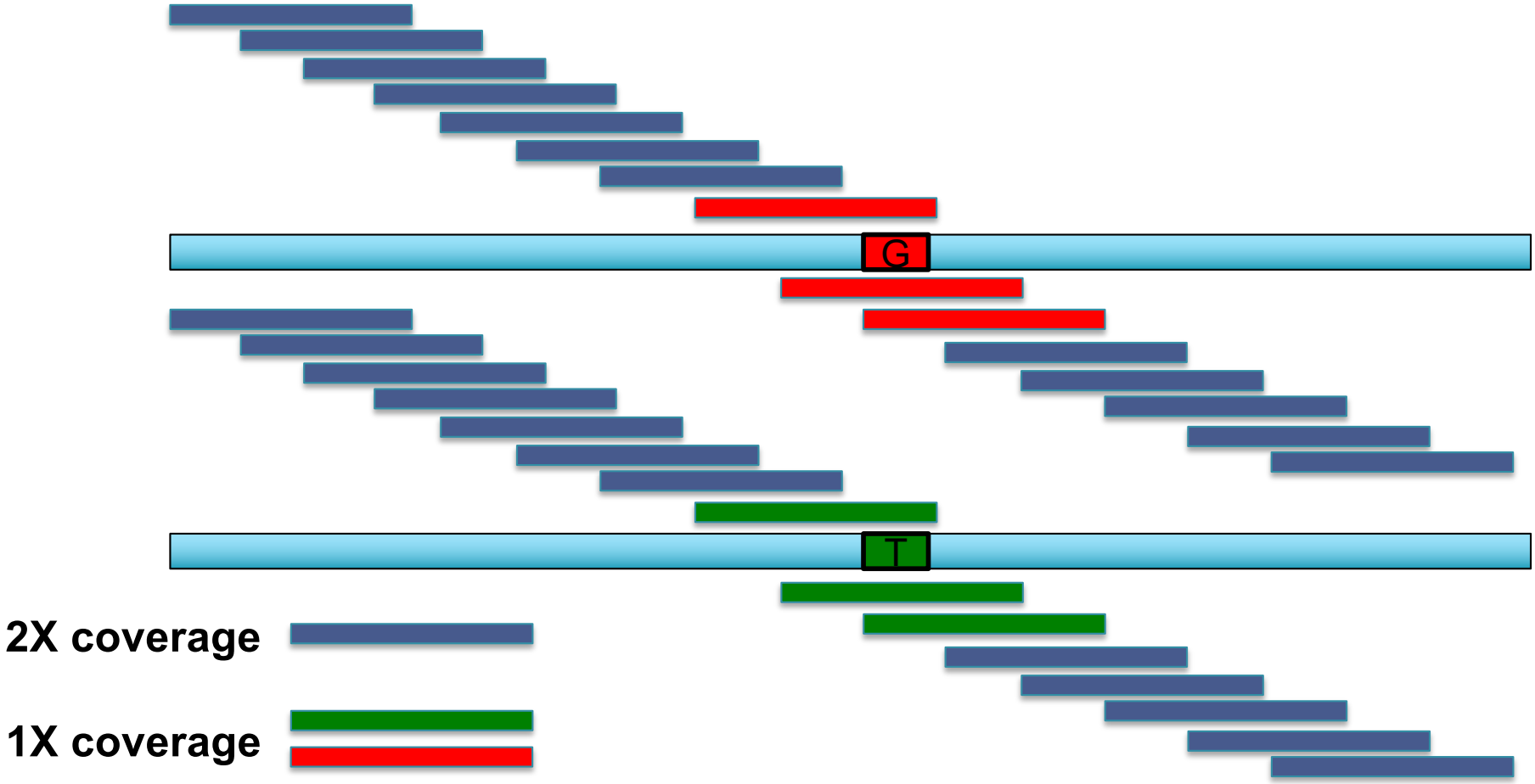
K-mer counting in heterozygous genomes



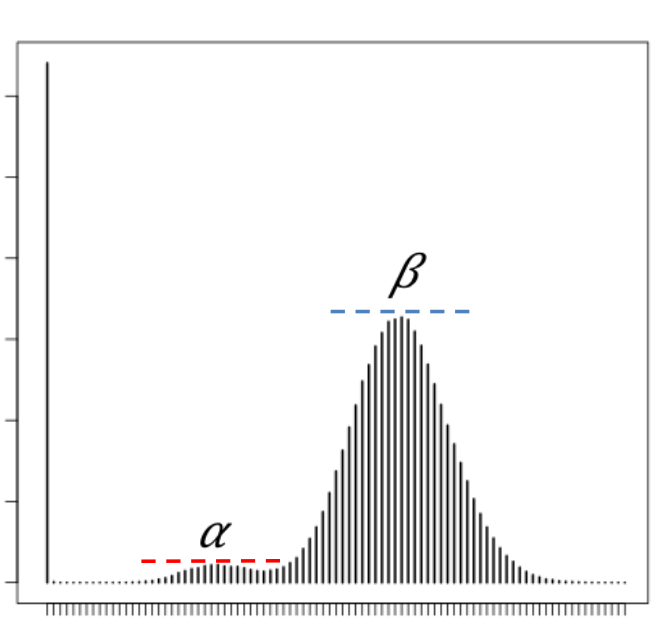
K-mer counting in heterozygous genomes



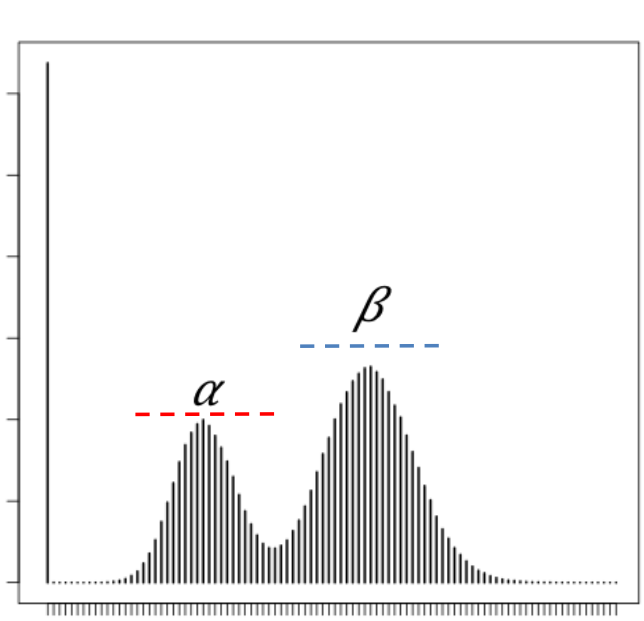
K-mer counting in heterozygous genomes



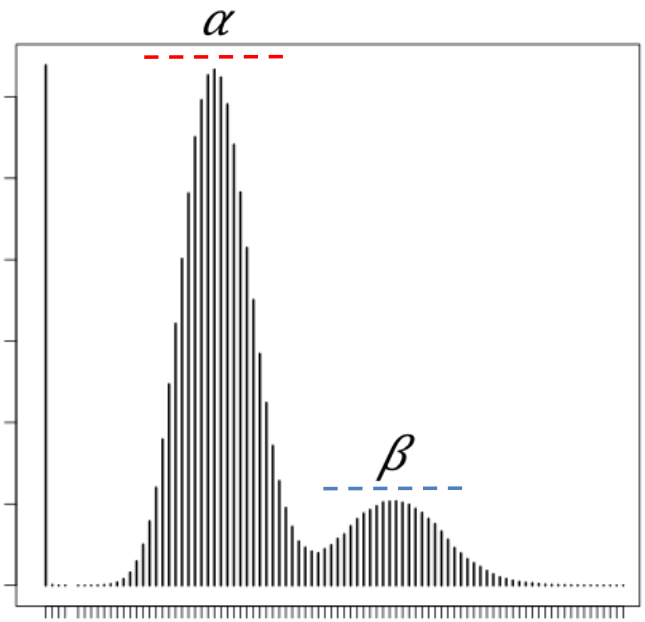
Heterozygous Kmer Profiles



0.1% heterozygosity



1% heterozygosity



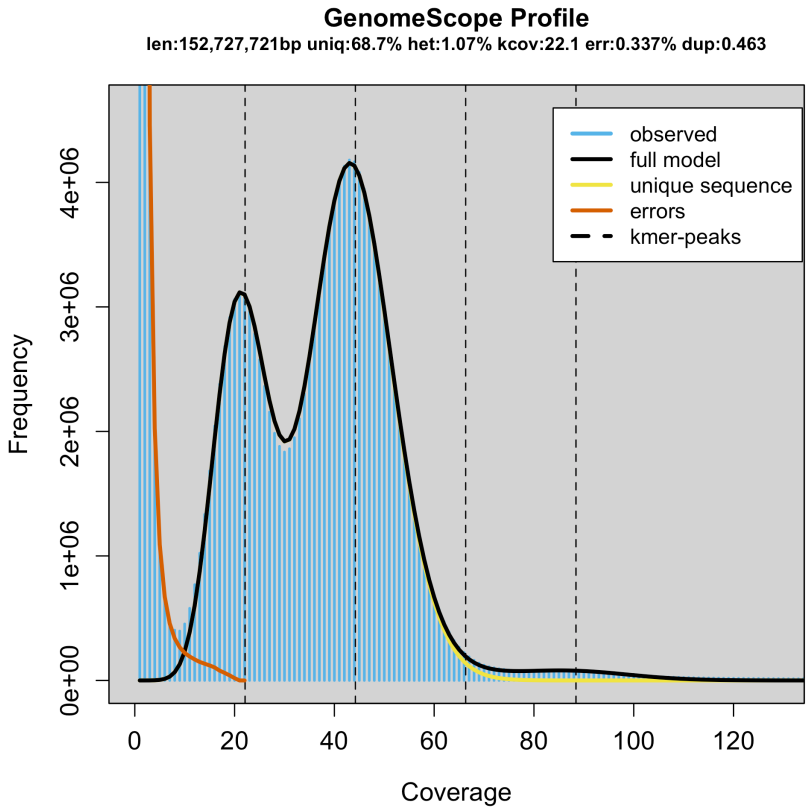
5% heterozygosity

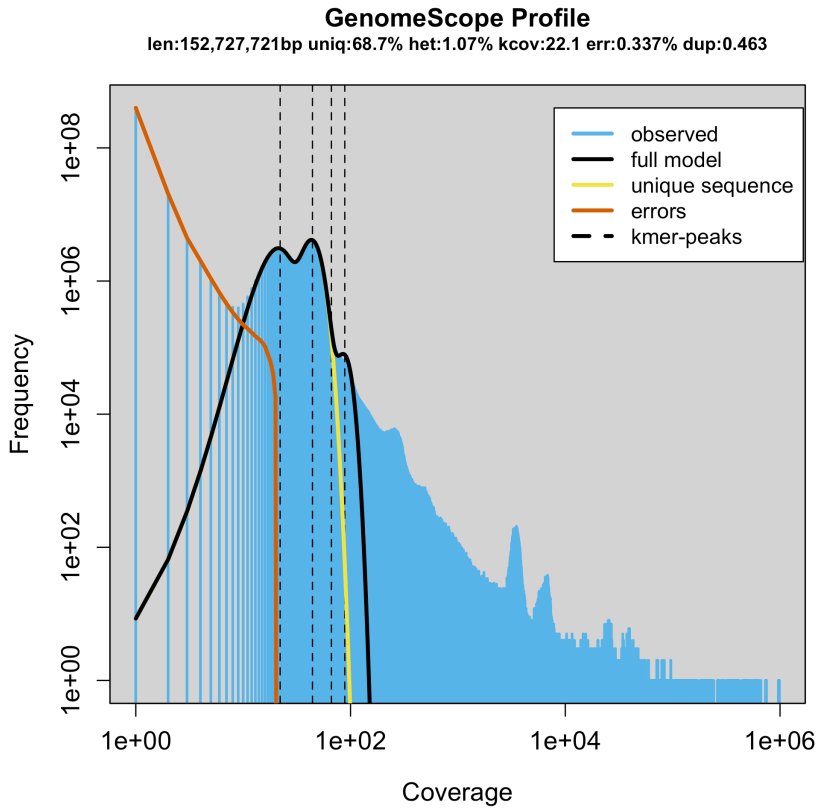
- ***Heterozygosity creates a characteristic "double-peak" in the Kmer profile***
  - Second peak at twice k-mer coverage as the first: heterozygous kmers average 50x coverage, homozygous kmers average 100x coverage

- ***Relative heights of the peaks is directly proportional to the heterozygosity rate***
  - The peaks are balanced at around 1.25% because each heterozygous SNP creates 2*k heterozygous kmers (typically k = 21)

GenomeScope: Fast genome analysis from short reads

http://genomescope.org



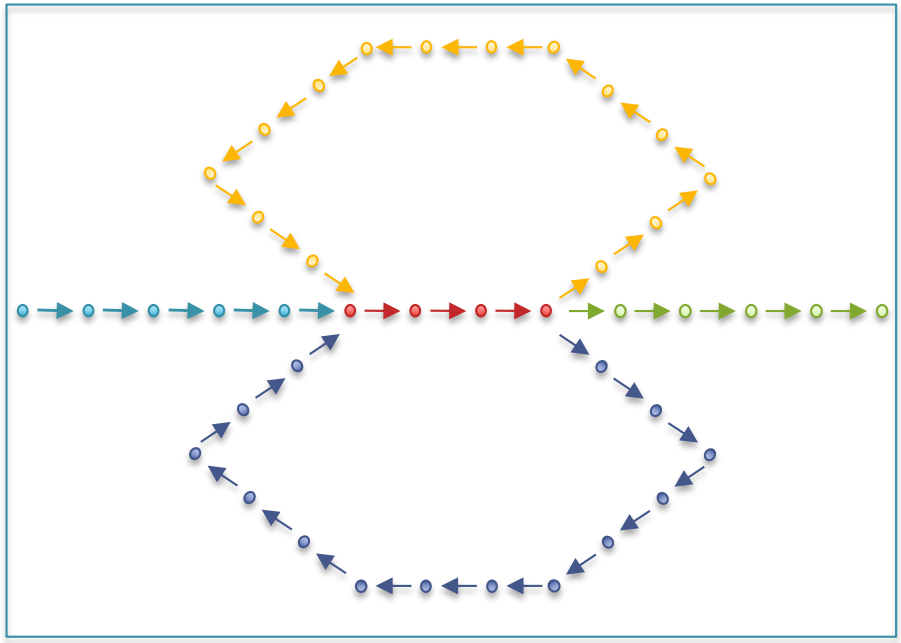


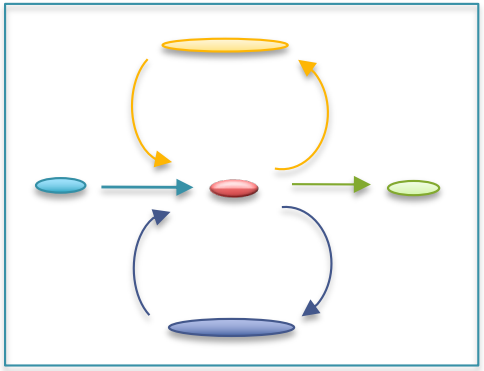
- Theoretical model agrees well with published results:
  - Rate of heterozygosity is higher than reported by other approaches but likely correct.
  - Genome size of plants inflated by organelle sequences (exclude very high freq. kmers)

Vurture, GW*, Sedlazeck FJ*, et al. (2017) *Bioinformatics*
Ranallo-Benavidez, TR. *et al.* (2020) *Nature Communication*

Unitigging / Unipathing

- After simplification and correction, compress graph down to its non-branching initial contigs
  - Aka "unitigs", "unipaths"





Why do contigs end?

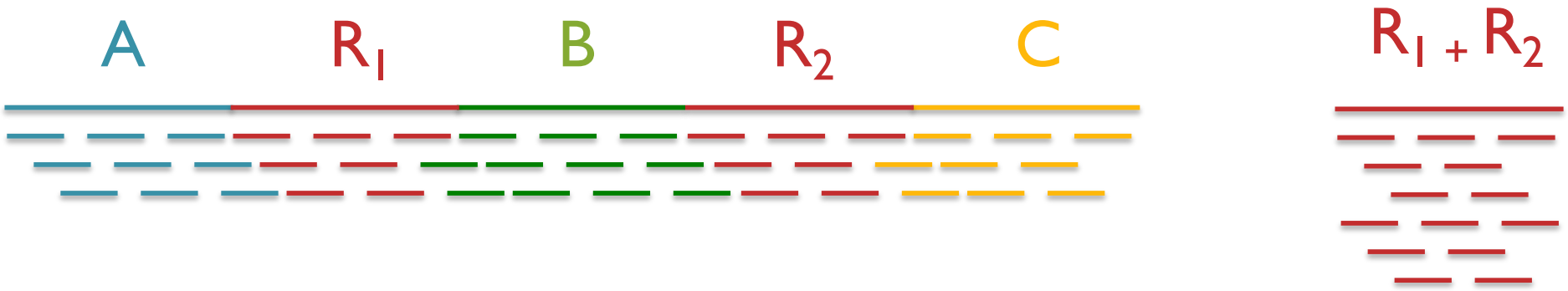
(1) End of chromosome! ☺, (2) lack of coverage, (3) errors,
(4) heterozygosity and (5) repeats

Repetitive regions

Repeat Type	Definition / Example	Prevalence
Low-complexity DNA / Microsatellites	$(b_1 b_2 \ldots b_k)^N$ where $1 \leq k \leq 6$ CACACACACACACACACACA	2%
SINEs (Short Interspersed Nuclear Elements)	*Alu* sequence (~280 bp) Mariner elements (~80 bp)	13%
LINEs (Long Interspersed Nuclear Elements)	~500 – 5,000 bp	21%
LTR (long terminal repeat) retrotransposons	Ty1-copia, Ty3-gypsy, Pao-BEL (~100 – 5,000 bp)	8%
Other DNA transposons		3%
Gene families & segmental duplications		4%

- Over 50% of mammalian genomes are repetitive
  - Large plant genomes tend to be even worse
  - Wheat: 16 Gbp; Pine: 24 Gbp

Repeats and Coverage Statistics



- If *n* reads are a uniform random sample of the genome of length *G*, we expect $k=n\Delta/G$ reads to start in a region of length Δ.
  - If we see many more reads than k (if the arrival rate is > A) , it is likely to be a collapsed repeat

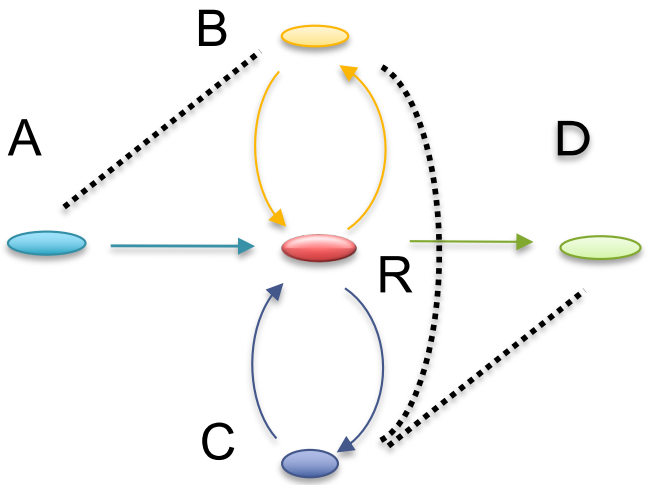
$$\Pr(X-copy)=\binom{n}{k}\left(\frac{X\Delta}{G}\right)^{k}\left(\frac{G-X\Delta}{G}\right)^{n-k}$$

$$A(\Delta,k)=\ln\left(\frac{\Pr(1-copy)}{\Pr(2-copy)}\right)=\ln\left(\frac{\frac{(\Delta n/G)^{k}}{k!}e^{\frac{-\Delta n}{G}}}{\frac{(2\Delta n/G)^{k}}{k!}e^{\frac{-2\Delta n}{G}}}\right)=\frac{n\Delta}{G}-k\ln 2$$

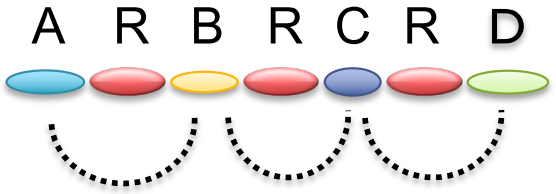
The fragment assembly string graph
Myers, EW (2005) Bioinformatics. 21(suppl 2): ii79-85.

Scaffolding

- Initial contigs (*aka* unipaths, unitigs) terminate at
  - *Coverage gaps*: especially extreme GC
  - *Conflicts*: errors, repeat boundaries



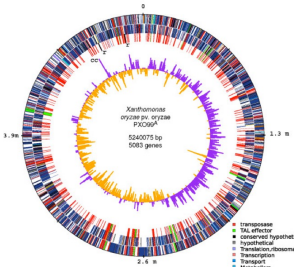
- Use mate-pairs to resolve correct order through assembly graph
  - Place sequence to satisfy the mate constraints
  - Mates through repeat nodes are tangled



- Final scaffold may have internal gaps called sequencing gaps
  - We know the order, orientation, and spacing, but just not the bases. Fill with Ns instead

Why do scaffolds end?

Assembly Summary

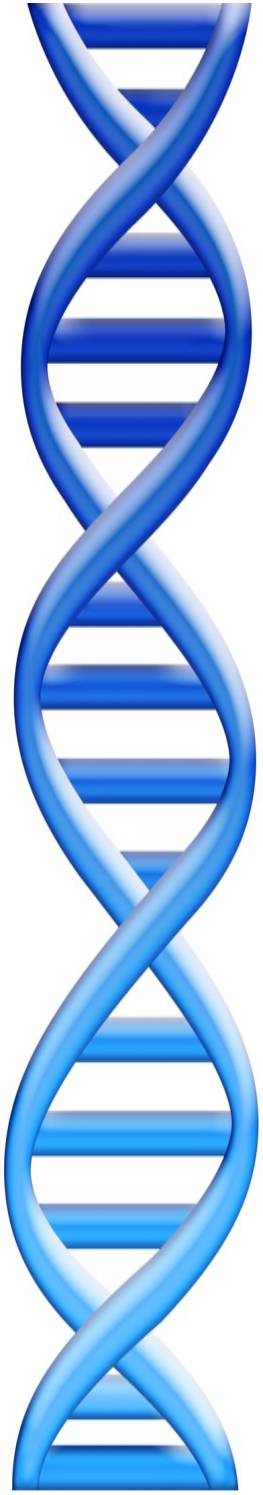


Assembly quality depends on

1. ***Coverage***: low coverage is mathematically hopeless
2. ***Repeat composition***: high repeat content is challenging
3. ***Read length***: longer reads help resolve repeats
4. ***Error rate***: errors reduce coverage, obscure true overlaps

- Assembly is a hierarchical, starting from individual reads, build high confidence contigs/unitigs, incorporate the mates to build scaffolds
  - Extensive error correction is the key to getting the best assembly possible from a given data set

- Watch out for collapsed repeats & other misassemblies
  - Globally/Locally reassemble data from scratch with better parameters & stitch the 2 assemblies together



Outline

1. ***Recap & Assembly theory***
   - Assembly by analogy

2. ***Practical Issues***
   - Coverage, read length, errors, and repeats

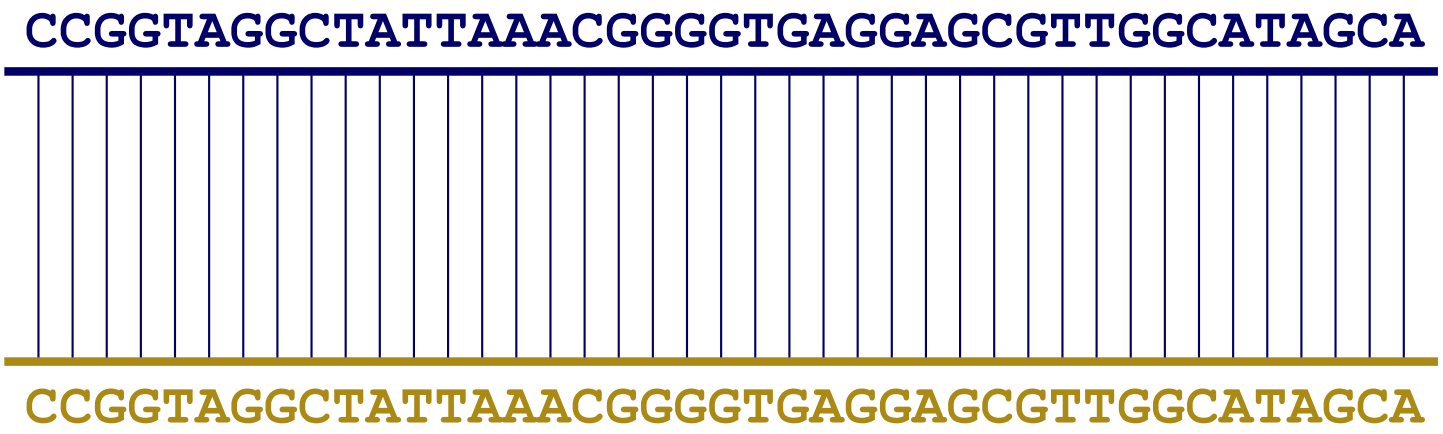
3. **Whole Genome Alignment**
   - MUMmer recommended

Whole Genome Alignment with MUMmer

Slides Courtesy of Adam M. Phillippy
NHGRI

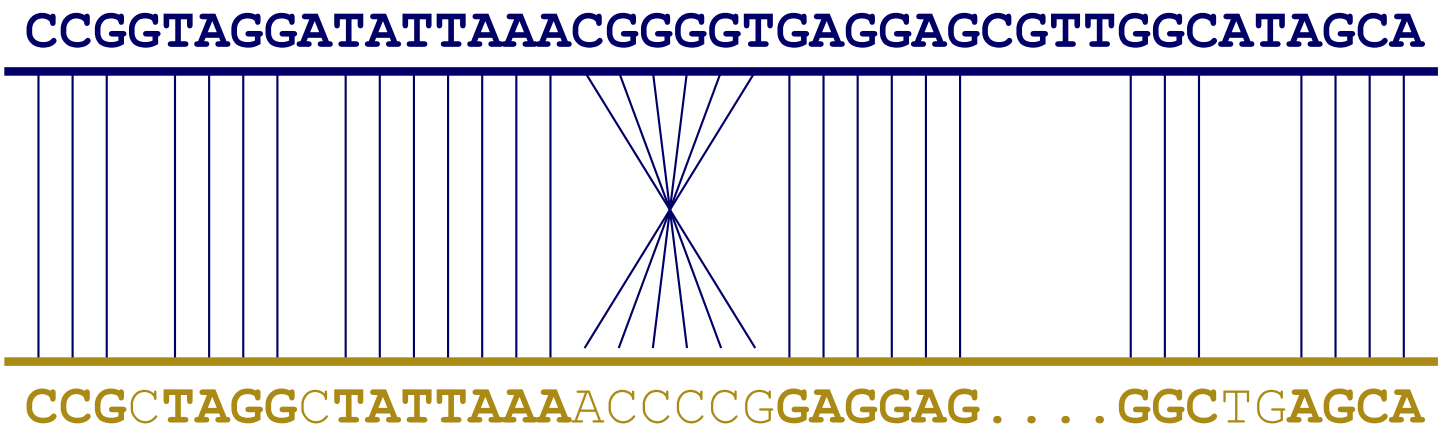
Goal of WGA

- For two genomes, *A* and *B*, find a mapping from each position in *A* to its corresponding position in *B*



Not so fast...

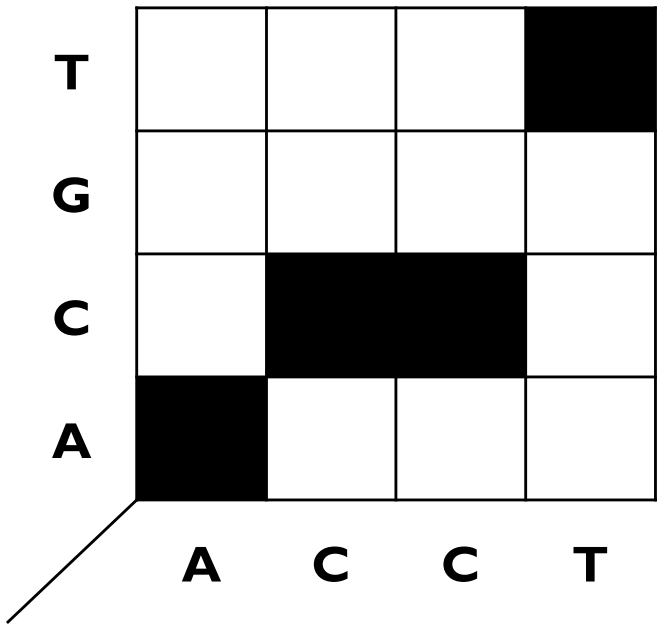
- Genome *A* may have insertions, deletions, translocations, inversions, duplications or SNPs with respect to *B* (sometimes all of the above)



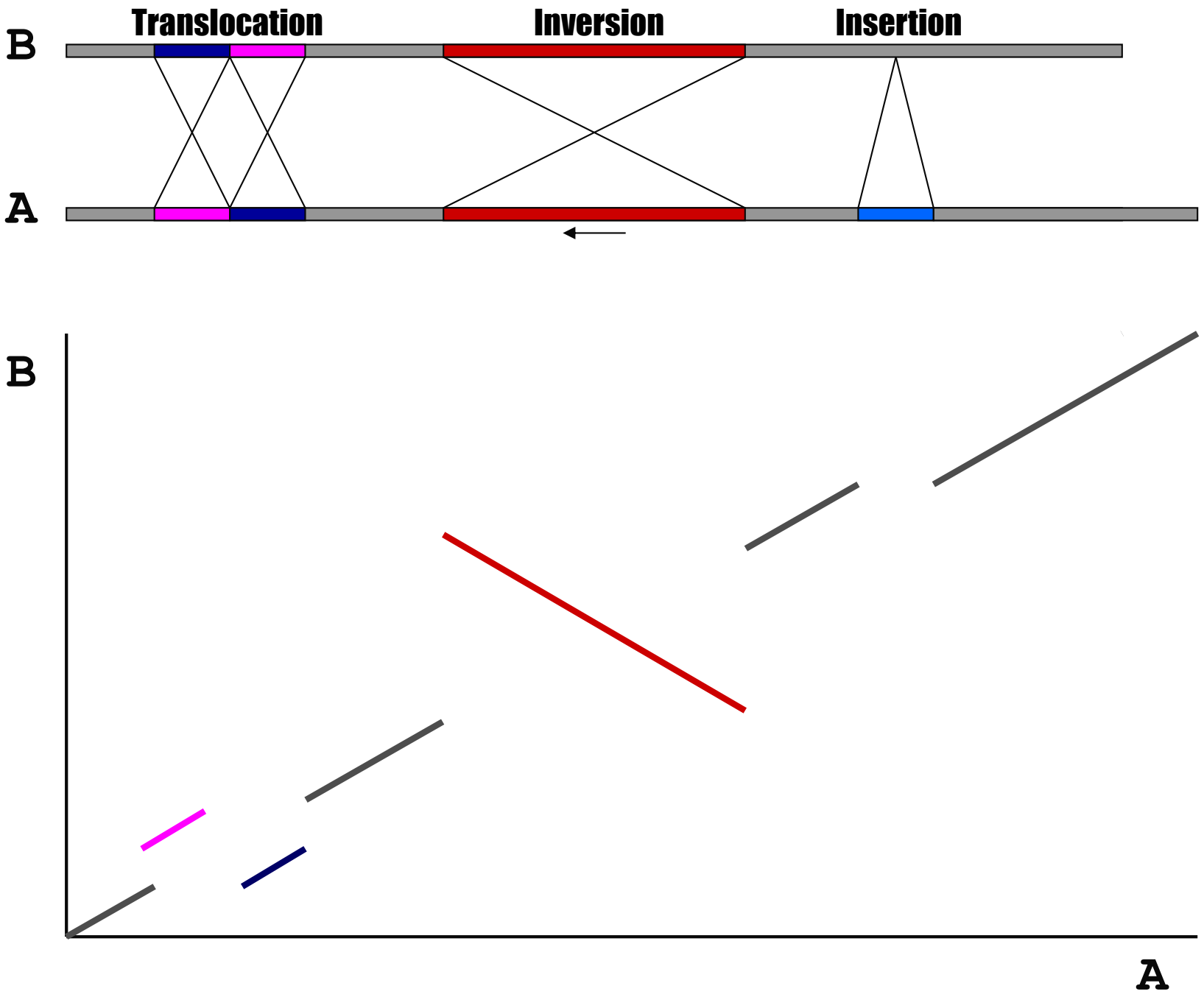
WGA visualization

- How can we visualize *whole* genome alignments?

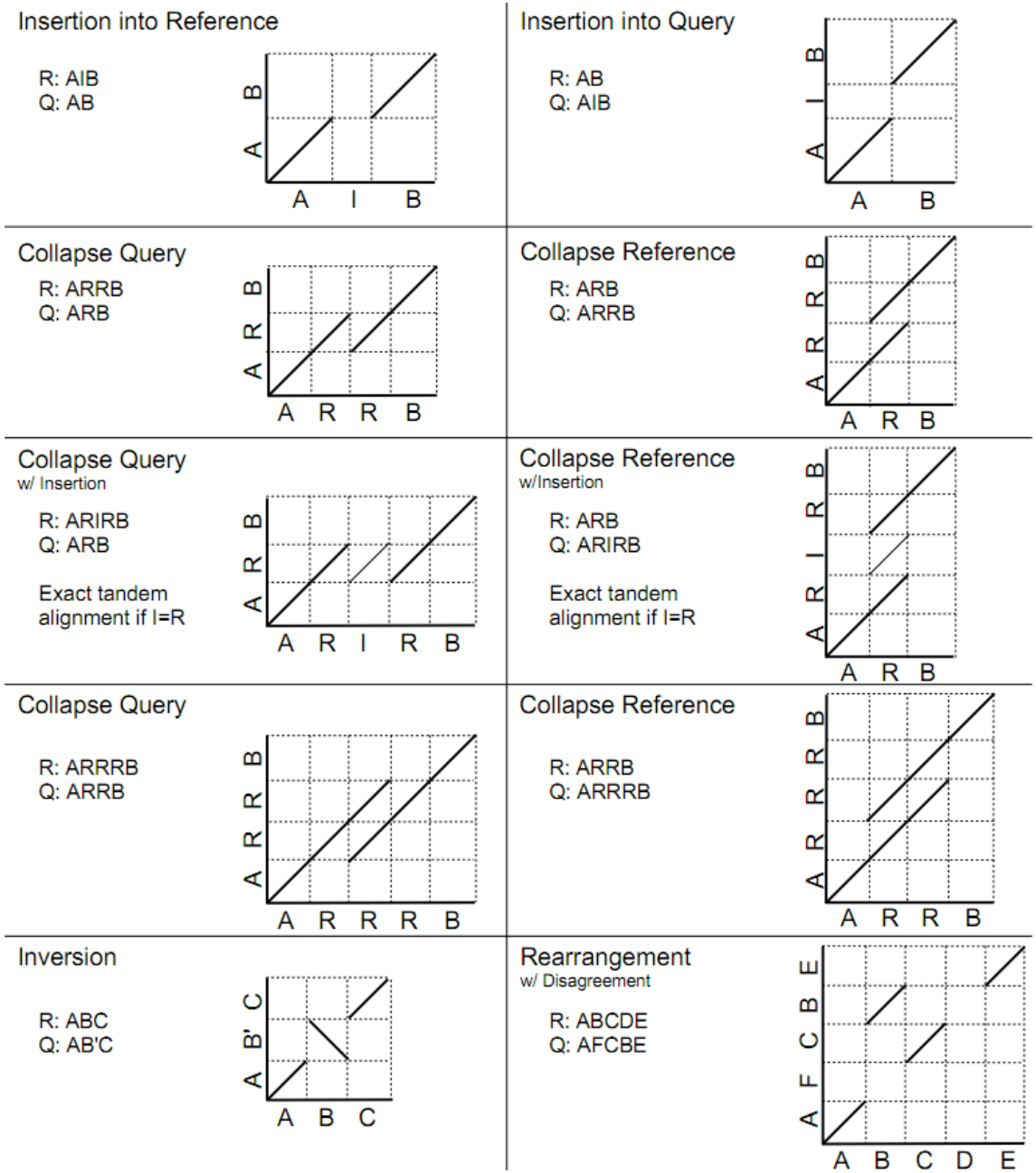
- With an alignment dot plot
  - *N* x *M* matrix
    - Let *i* = position in genome *A*
    - Let *j* = position in genome *B*
    - Fill cell *(i,j)* if A_i shows similarity to B_j



  - A perfect alignment between *A* and *B* would completely fill the positive diagonal

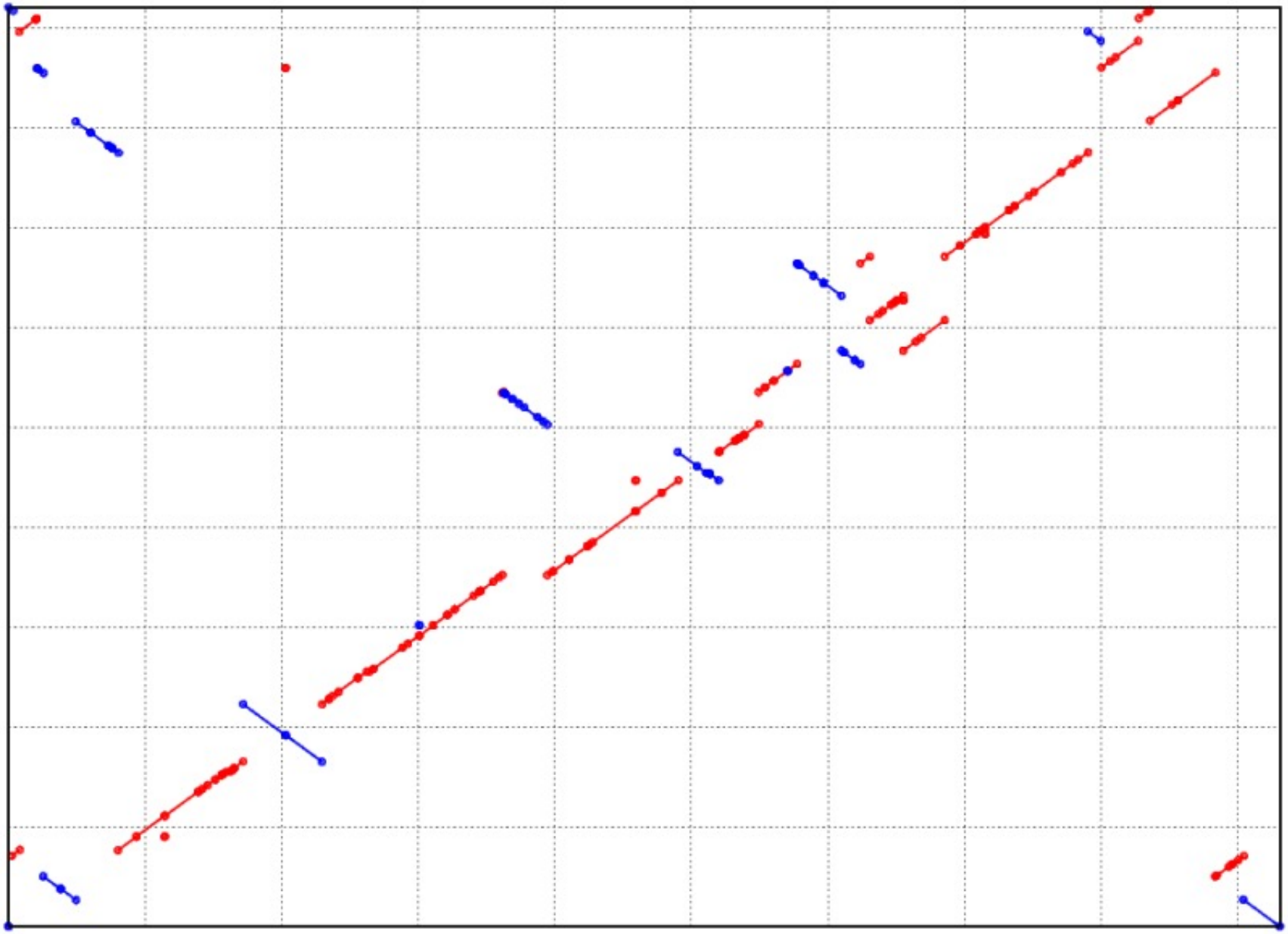

Translocation
Inversion
Insertion
B
A
B
A

SV Types



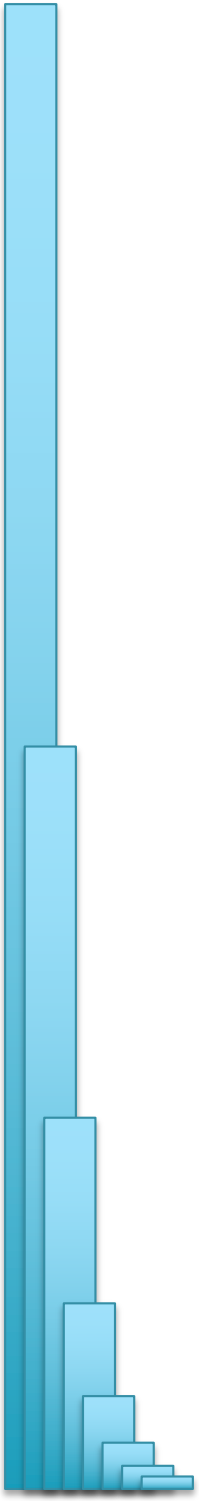
- Different structural variation types / misassemblies will be apparent by their pattern of breakpoints
- Most breakpoints will be at or near repeats
- Things quickly get complicated in real genomes

http://mummer.sf.net/manual/AlignmentTypes.pdf



Alignment of 2 strains of *Y. pestis*

http://mummer.sourceforge.net/manual/



Notes

Find and decode

nucmer –maxmatch ref.fasta contigs.fasta

```
   -maxmatch     Find maximal exact matches (MEMs) without repeat filtering
   -p refctg     Set the output prefix for delta file
```

mummerplot --layout --png out.delta

```
   --layout      Sort the alignments along the diagonal
   --png         Create a png of the results
```

show-coords –rclo out.delta

```
   -r            Sort alignments by reference position
   -c            Show percent coverage
   -l            Show sequence lengths
   -o            Annotate each alignment with BEGIN/END/CONTAINS
```

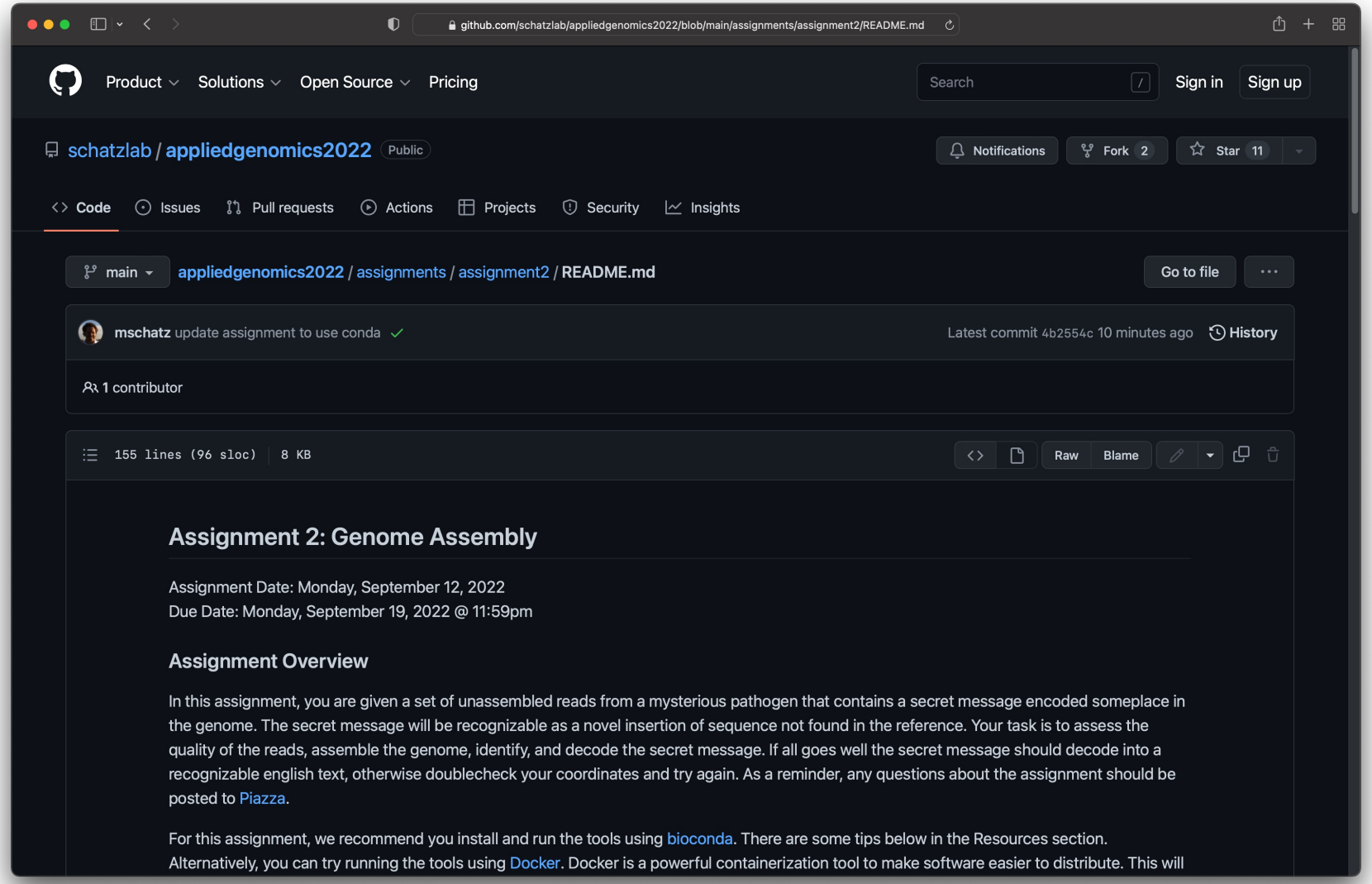
samtools faidx contigs.fasta

```
   Index the fasta file
```

**samtools faidx contigs.fasta **
**   contig_XXX:YYY-ZZZ | ./dna-encode -d**

See manual at http://mummer.sourceforge.net/manual

Assignment 2: Genome Assembly
Due Monday Sept 19 by 11:59pm



https://github.com/schatzlab/appliedgenomics2022/tree/main/assignments/assignment2
Check Piazza for questions!