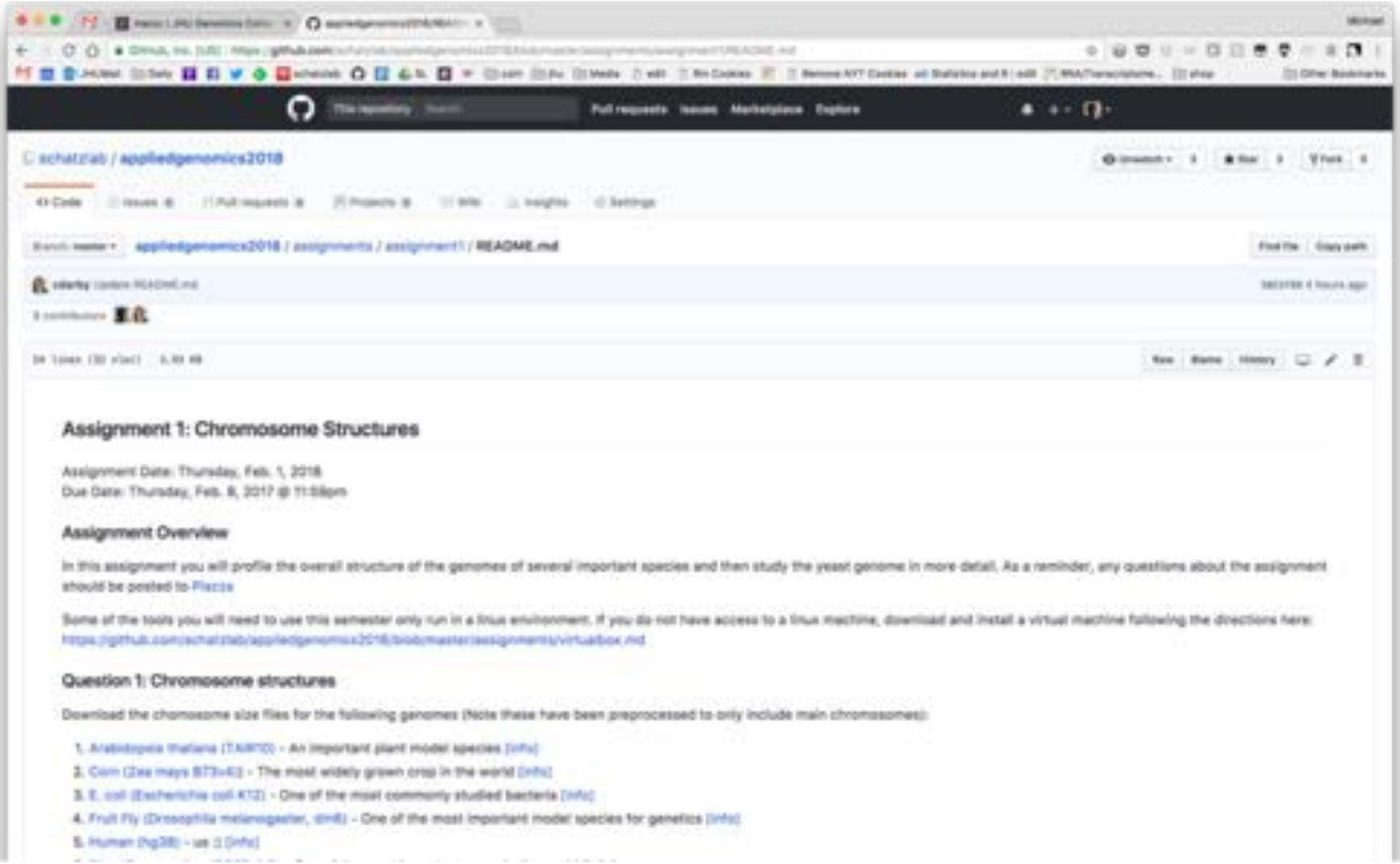
Genome Assembly and Alignment

Michael Schatz

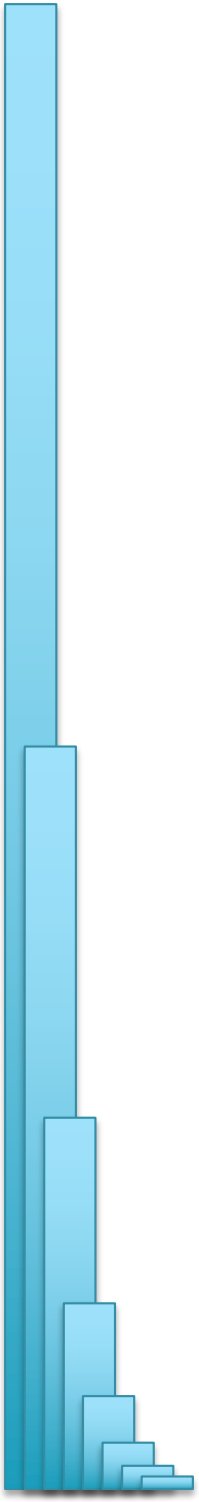
Feb 8, 2018

Lecture 4: Applied Comparative Genomics

Assignment 1: Chromosome Structures
Due Feb 8 @ 11:59pm



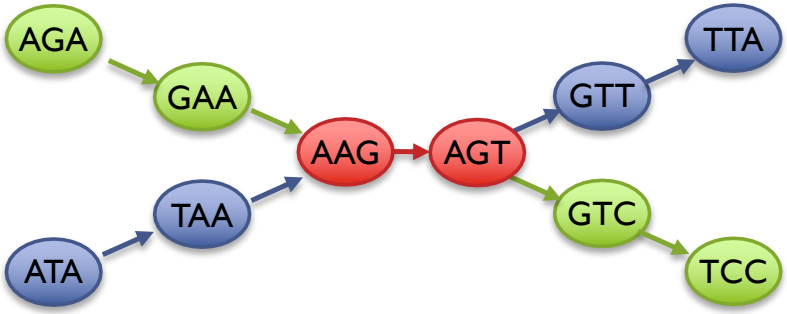
https://github.com/schatzlab/appliedgenomics2018



Part 1: Recap

Two Paradigms for Assembly

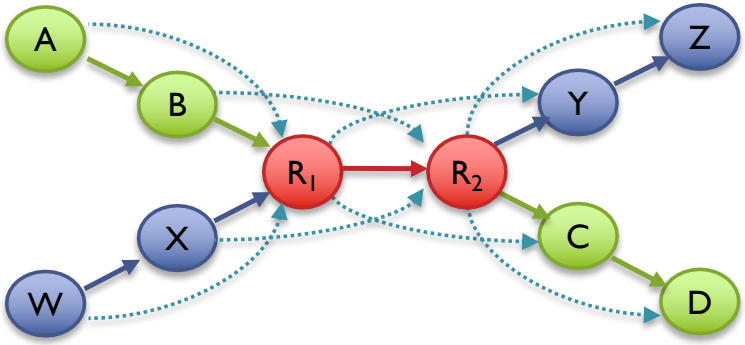
de Bruijn Graph



Short read assemblers

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

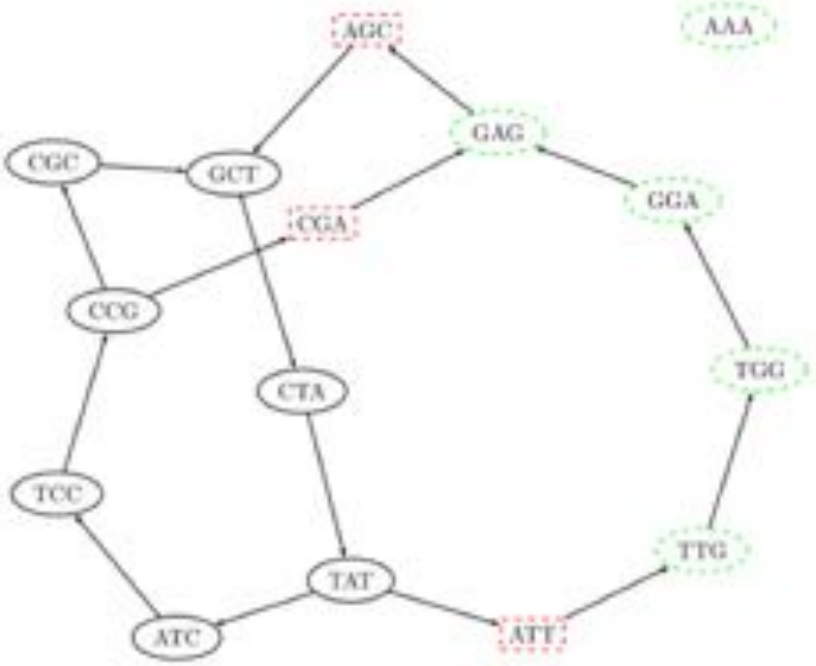
Overlap Graph



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.

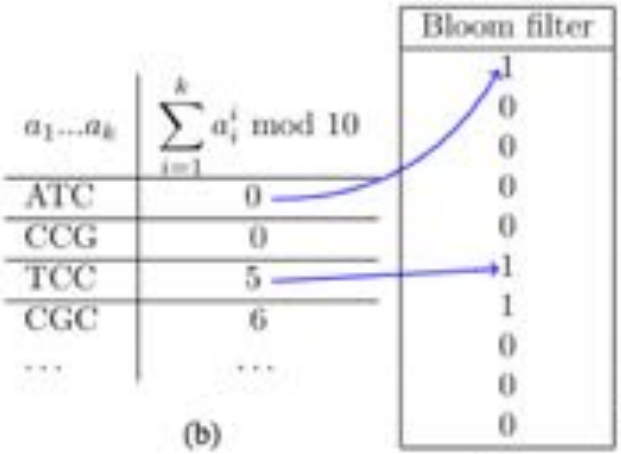


(a)

Table 2 de novo human genome (NA18507) assemblies

Method	Minia	C. & B.	ABySS	SOAPdenovo
Value of *k* chosen	27	27	27	25
Number of contigs (M)	3.49	7.69	4.35	-
Longest contig (kbp)	18.6	22.0	15.9	-
Contig N50 (bp)	1156	250	870	886
Sum (Gbp)	2.09	1.72	2.10	2.08
Nb of nodes/cores	1/1	1/8	21/168	1/16
Time (wall-clock, h)	23	50	15	33
Memory (sum of nodes, GB)	**5.7**	**32**	**336**	**140**

de novo human genome (NA18507) assemblies reported by our assembler (Minia), Conway and Bromage assembler [9], ABySS [8], and SOAPdenovo [7]. Contigs shorter than 100 bp were discarded. Assemblies were made without any pairing information.



(b)

(c)

Nodes self-information:

$$\lceil log_2 \binom{4^3}{7} \rceil = 30 \text{ bits}$$

Structure size:

$$\underbrace{10}_{\text{Bloom}} + \underbrace{3 \cdot 6}_{\text{False positives}} = 28 \text{ bits}$$

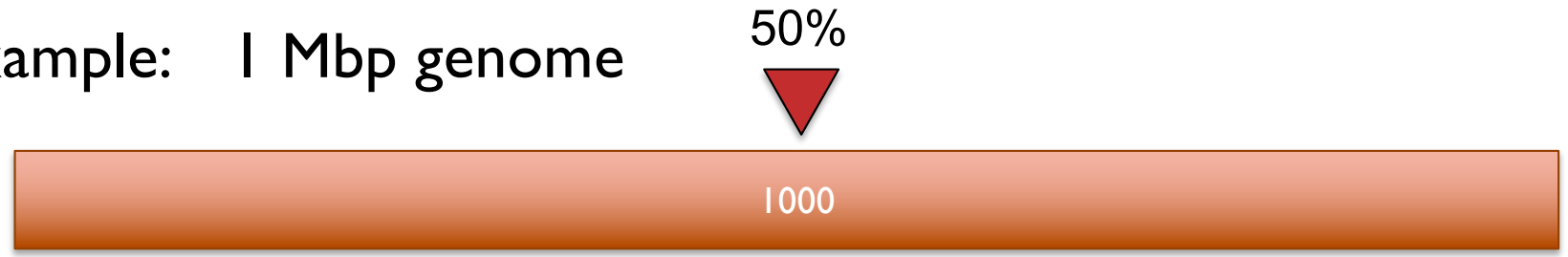
(d)

Space-efficient and exact de Bruijn graph representation based on a Bloom filter
Chikhi and Rizk (2013) *Algorithms for Molecular Biology*. 8:22

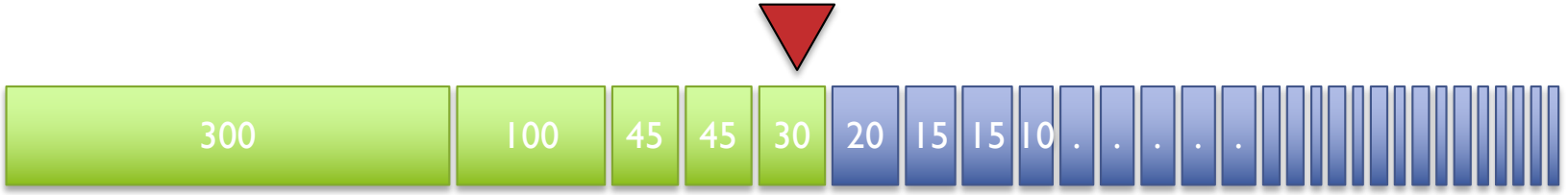
Contig N50

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

Example: 1 Mbp genome

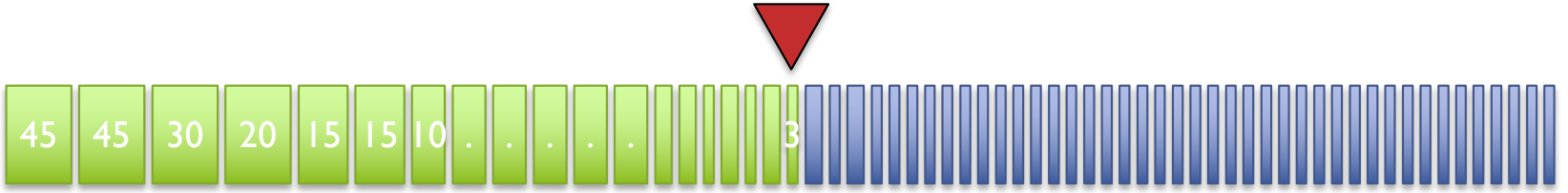


A



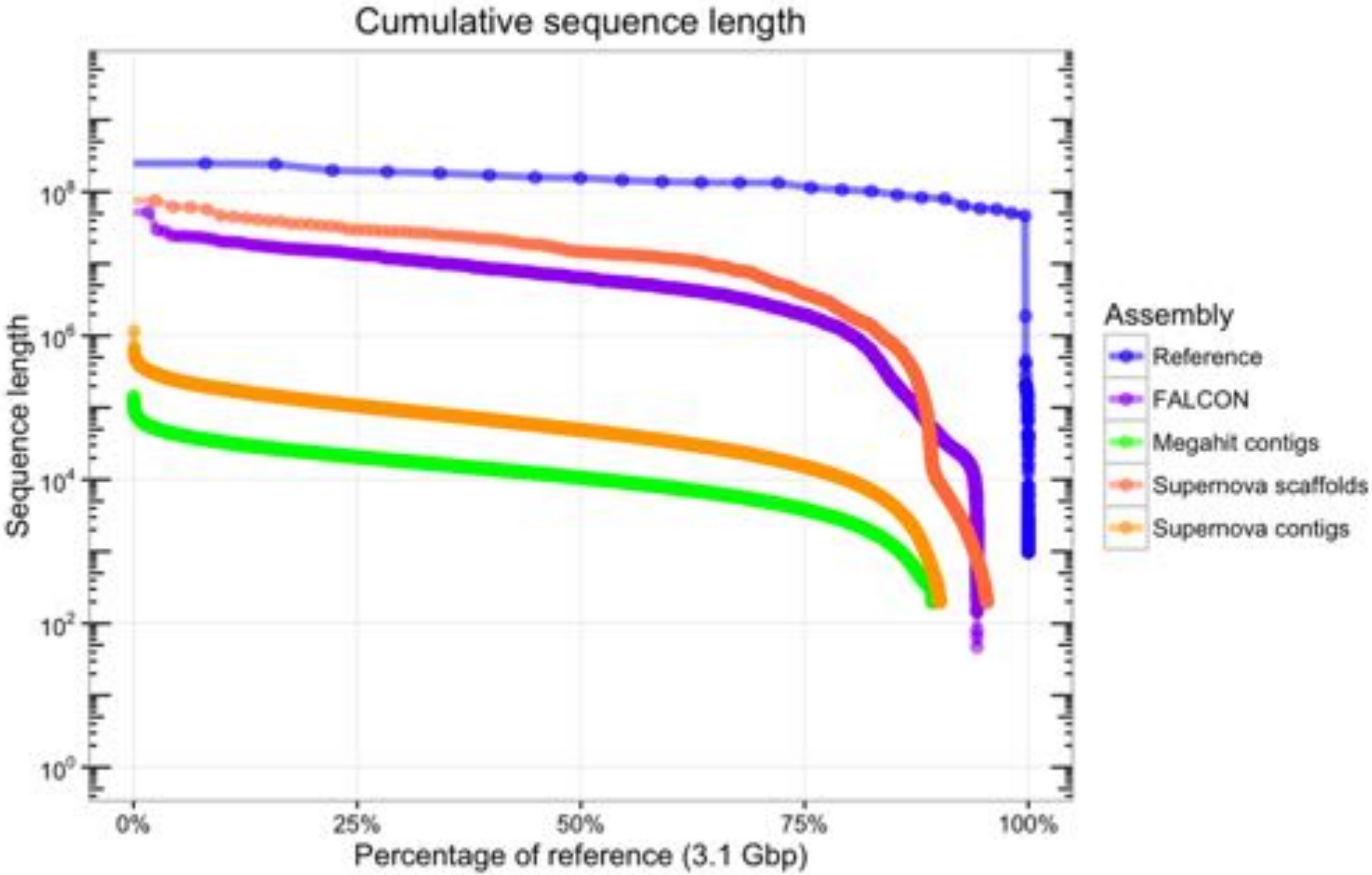
N50 size = 30 kbp

B

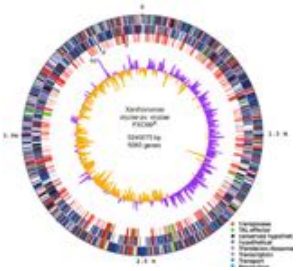


N50 size = 3 kbp

Contig Nchart



Assembly Summary

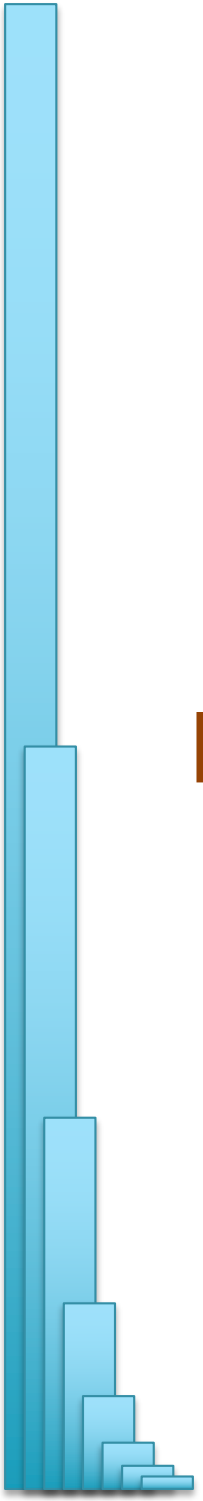


Assembly quality depends on

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



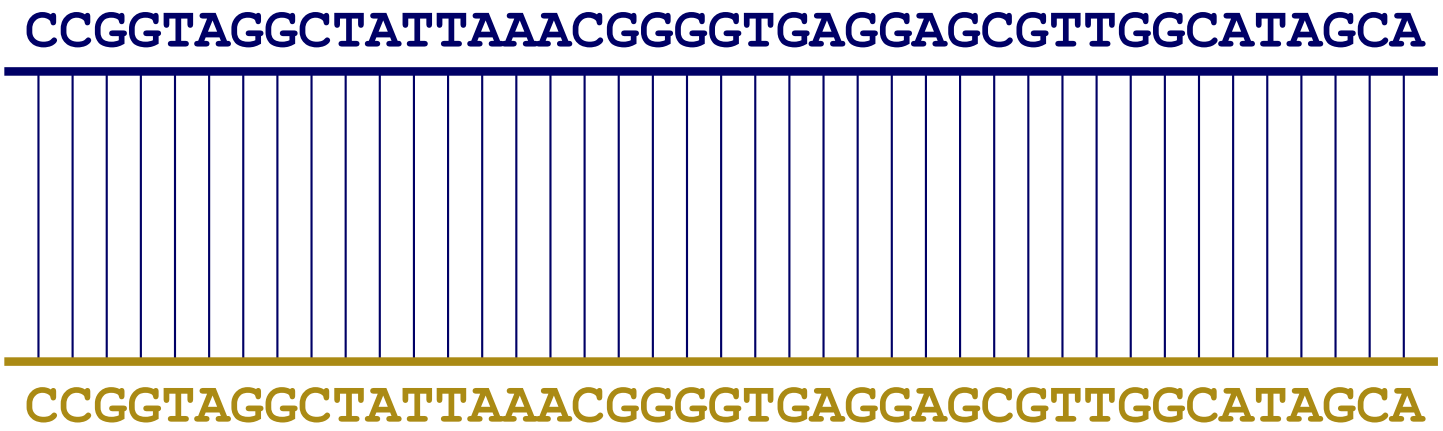
Part 2: Whole Genome Alignment & HW2

Whole Genome Alignment with MUMmer

Slides Courtesy of Adam M. Phillippy
NHGRI

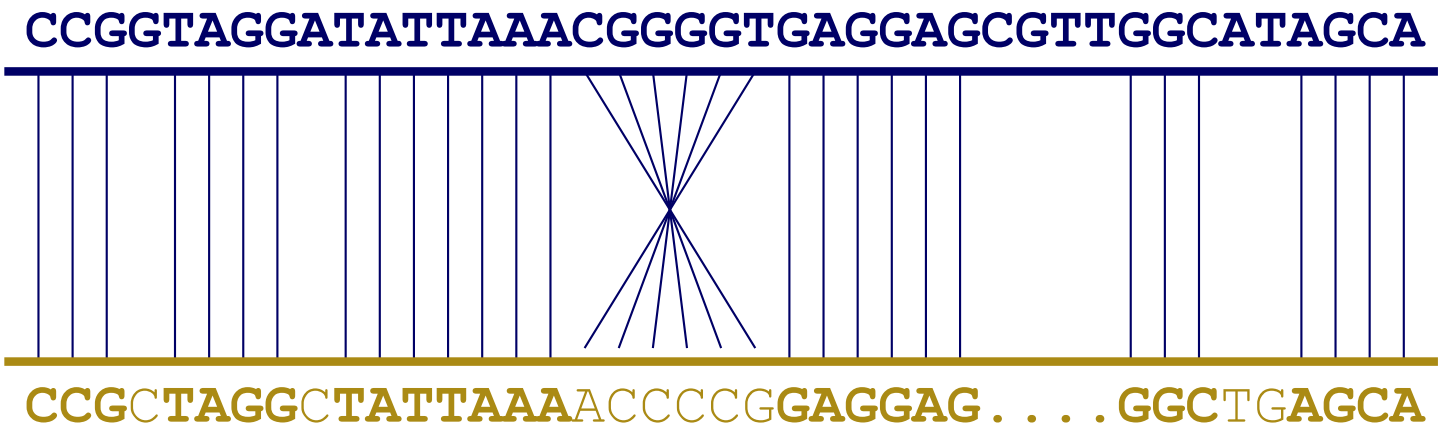
Goal of WGA

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



Not so fast...

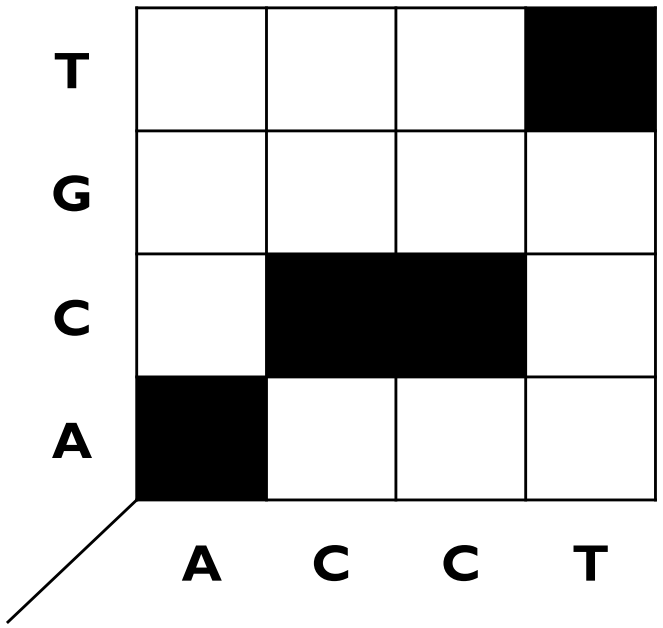
- Genome *A* may have insertions, deletions, translocations, inversions, duplications or SNPs with respect to *B* (sometimes all of the above)



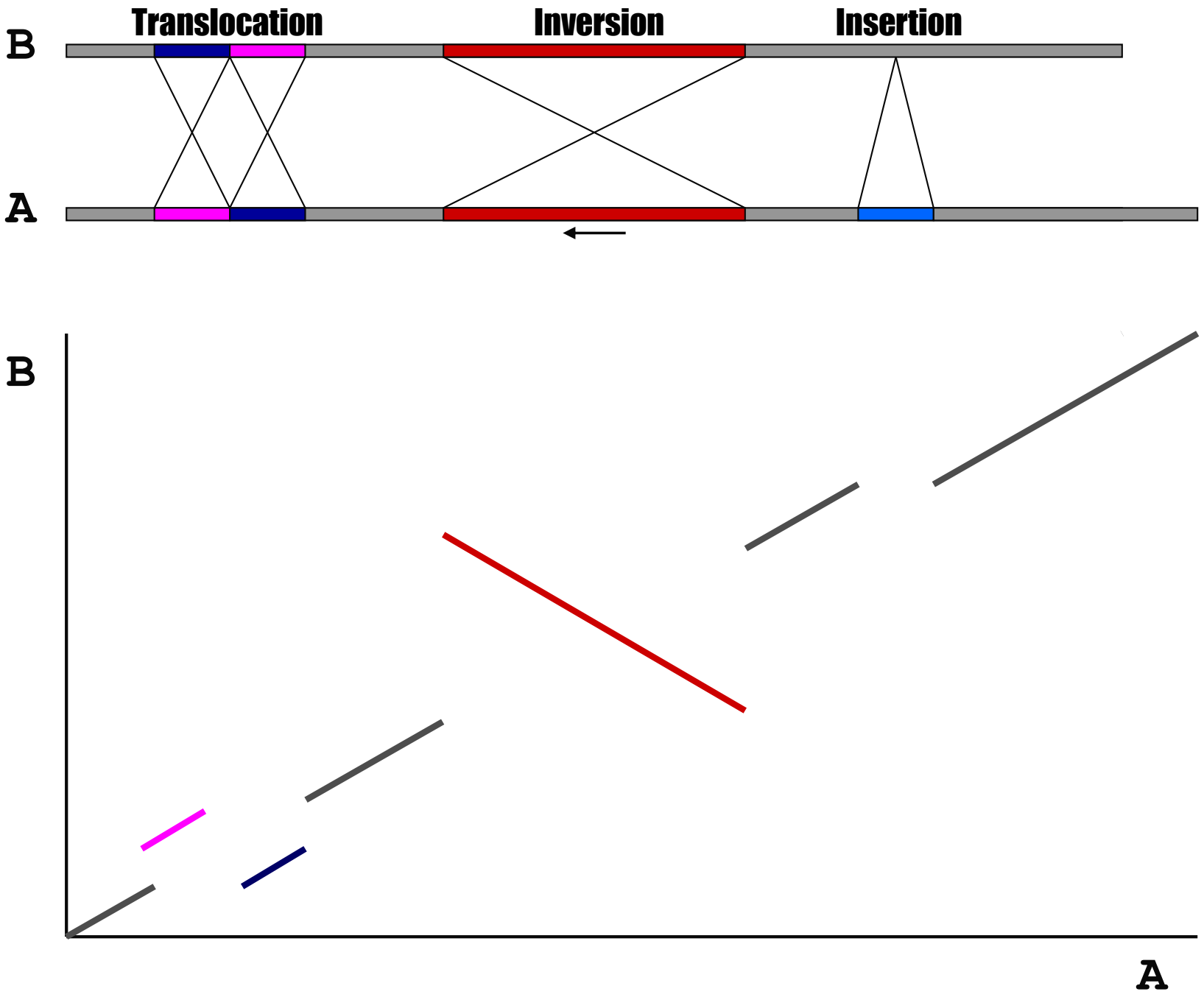
WGA visualization

- How can we visualize *whole* genome alignments?

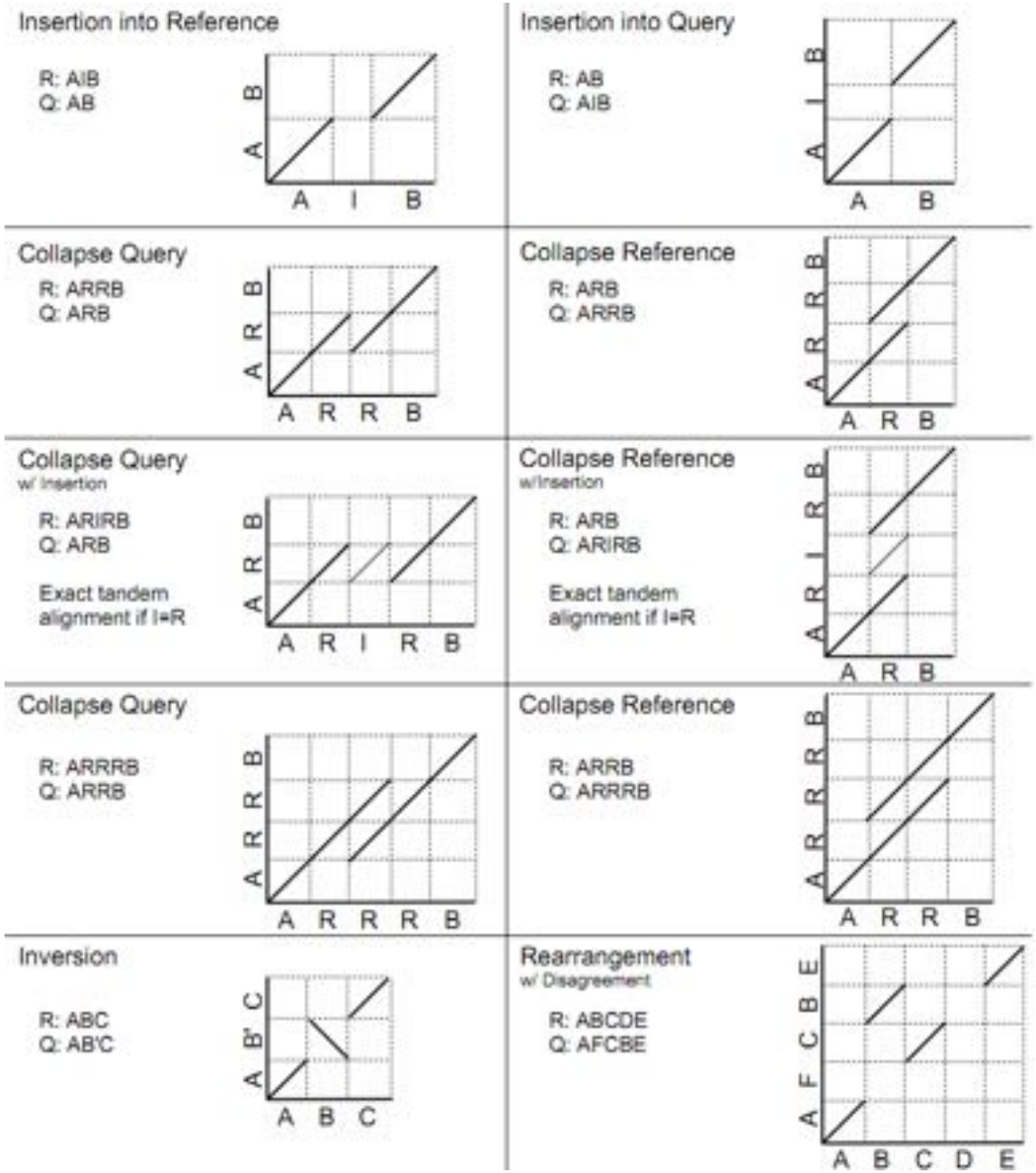
- With an alignment dot plot
  - *N* x *M* matrix
    - Let *i* = position in genome *A*
    - Let *j* = position in genome *B*
    - Fill cell *(i,j)* if A_i shows similarity to B_j



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

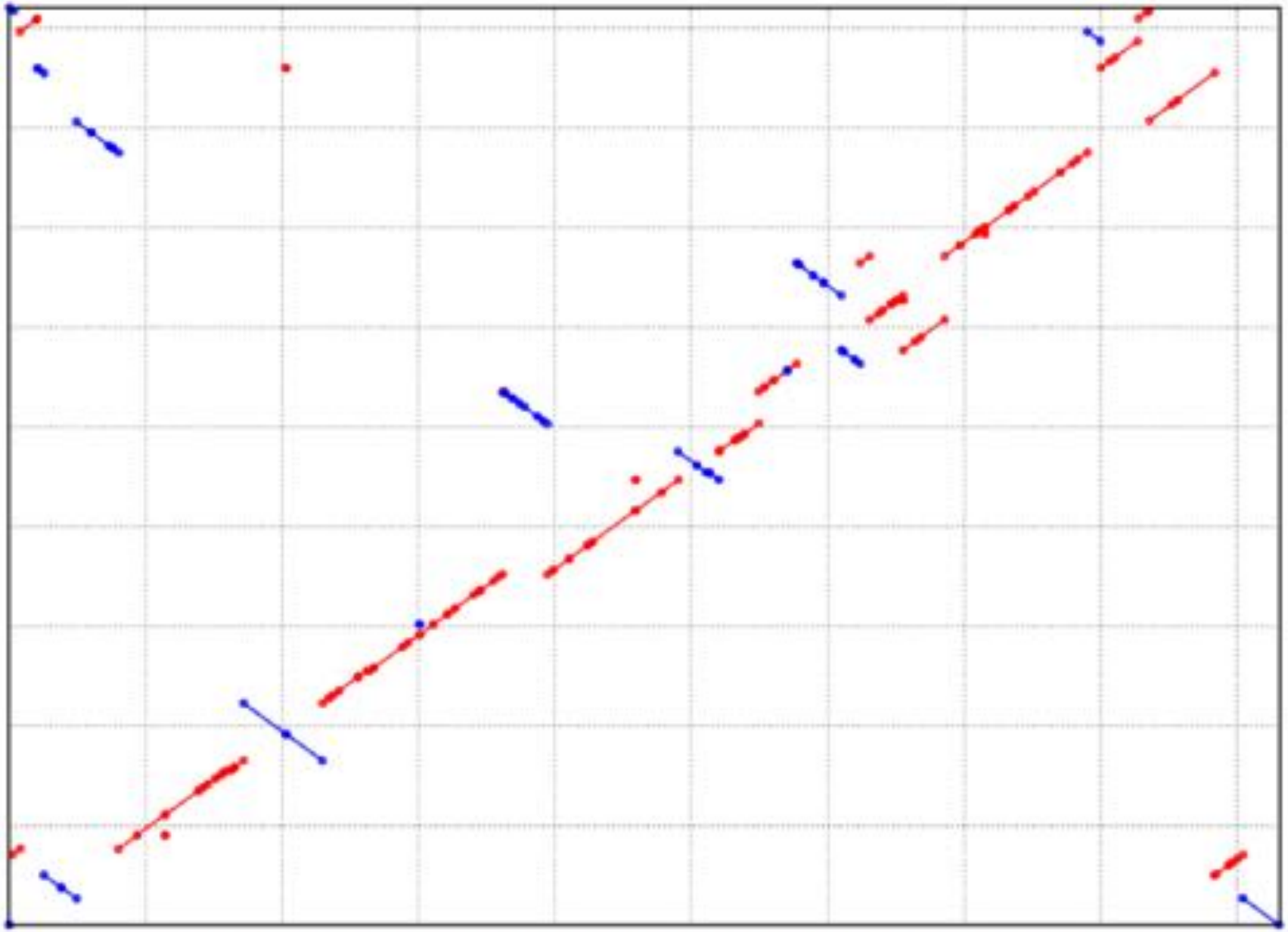

Translocation
Inversion
Insertion
B
A
B
A

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

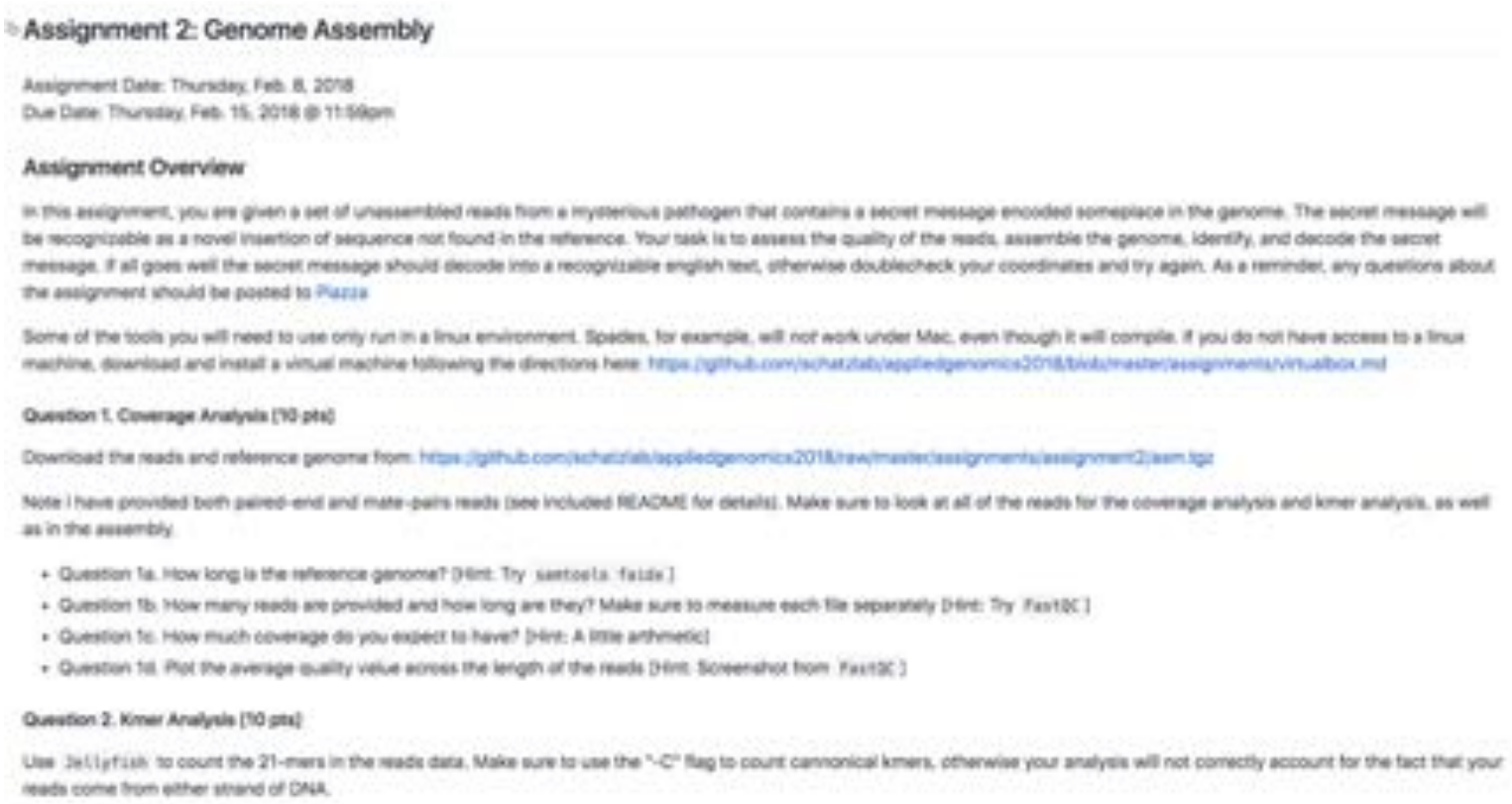
http://mummer.sf.net/manual/AlignmentTypes.pdf



Alignment of 2 strains of *Y. pestis*

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

Assignment 2: Genome Assembly
Due Feb 15 @ 11:59pm



Assignment 2: Genome Assembly

Assignment Date: Thursday, Feb. 8, 2018
Due Date: Thursday, Feb. 15, 2018 @ 11:59pm

Assignment Overview

In this assignment, you are given a set of unassembled reads from a mysterious pathogen that contains a secret message encoded someplace in the genome. The secret message will be recognizable as a novel insertion of sequence not found in the reference. Your task is to assess the quality of the reads, assemble the genome, identify, and decode the secret message. If all goes well the secret message should decode into a recognizable english text, otherwise doublecheck your coordinates and try again. As a reminder, any questions about the assignment should be posted to Piazza

Some of the tools you will need to use only run in a linux environment. Spades, for example, will not work under Mac, even though it will compile. If you do not have access to a linux machine, download and install a virtual machine following the directions here: https://github.com/schatzlab/appliedgenomics2018/blob/master/assignments/virtualbox.md

Question 1. Coverage Analysis [10 pts]

Download the reads and reference genome from: https://github.com/schatzlab/appliedgenomics2018/raw/master/assignments/assignment2/asm.tgz

Note I have provided both paired-end and mate-pairs reads (see included README for details). Make sure to look at all of the reads for the coverage analysis and kmer analysis, as well as in the assembly.

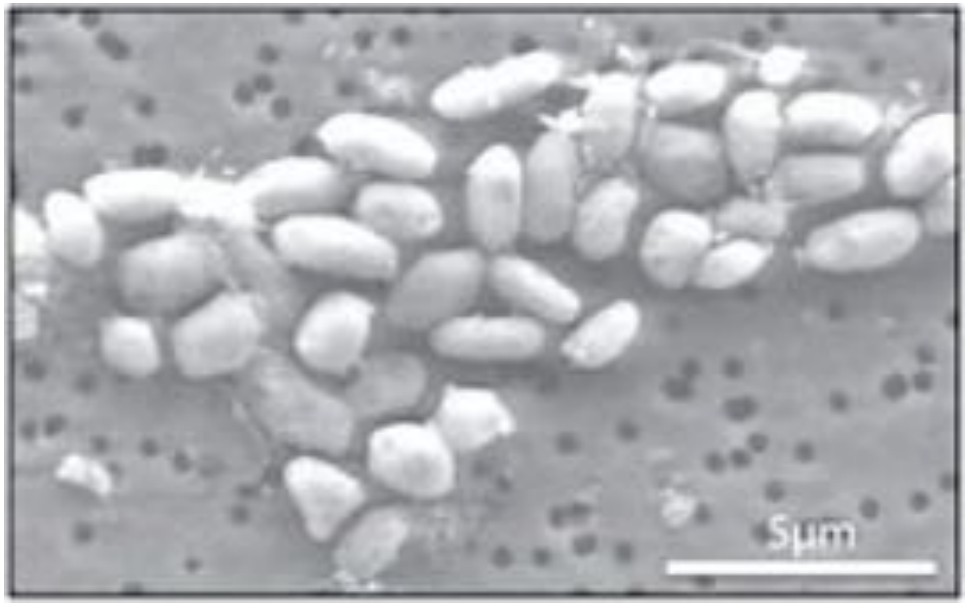
- Question 1a. How long is the reference genome? [Hint: Try `samtools faidx`]
- Question 1b. How many reads are provided and how long are they? Make sure to measure each file separately [Hint: Try `FastQC`]
- Question 1c. How much coverage do you expect to have? [Hint: A little arithmetic]
- Question 1d. Plot the average quality value across the length of the reads [Hint: Screenshot from `FastQC`]

Question 2. Kmer Analysis [10 pts]

Use `Jellyfish` to count the 21-mers in the reads data. Make sure to use the "-C" flag to count cannonical kmers, otherwise your analysis will not correctly account for the fact that your reads come from either strand of DNA.

https://github.com/schatzlab/appliedgenomics2018

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

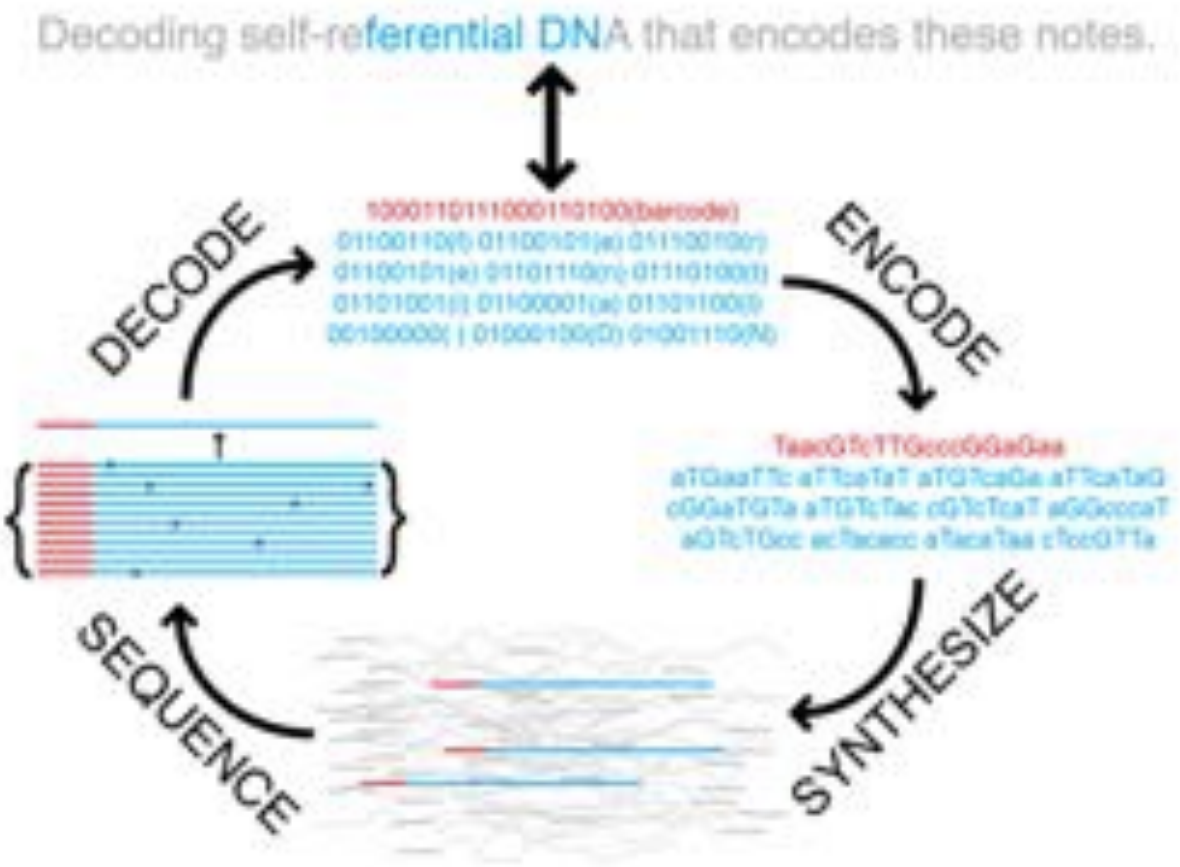


Fig. S1. Schematic of DNA information storage.

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

Mission Impossible

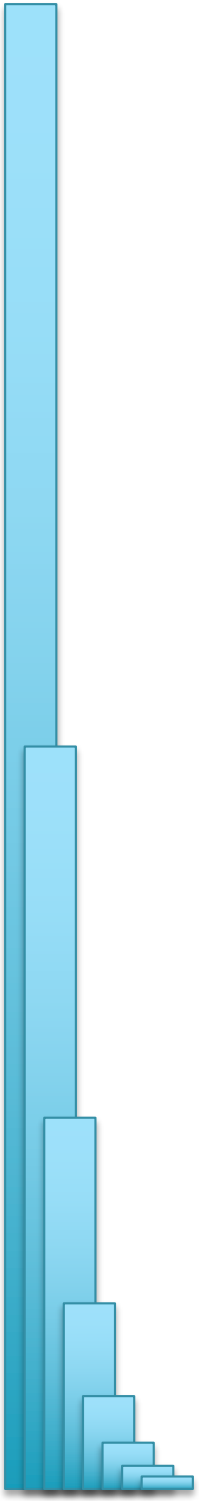
1. ***Setup VirtualBox***
2. ***Initialize Tools***
3. ***Download Reference Genome & Reads***
4. ***Decode the secret message***
    1. *Estimate coverage, check read quality*
    2. *Check kmer distribution*
    3. *Assemble the reads with spades*
    4. *Align to reference with MUMmer*
    5. *Extract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

https://github.com/schatzlab/appliedgenomics2018/blob/master/assignments/assignment2/README.md

Find and decode

```
nucmer -maxmatch ref.fasta \
   default/ASSEMBLIES/test/final.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 default/ASSEMBLIES/test/final.contigs.fasta
   Index the fasta file

samtools faidx default/ASSEMBLIES/test/final.contigs.fasta \
    contig_XXX:YYY-ZZZ | ./dna-encode -d
```

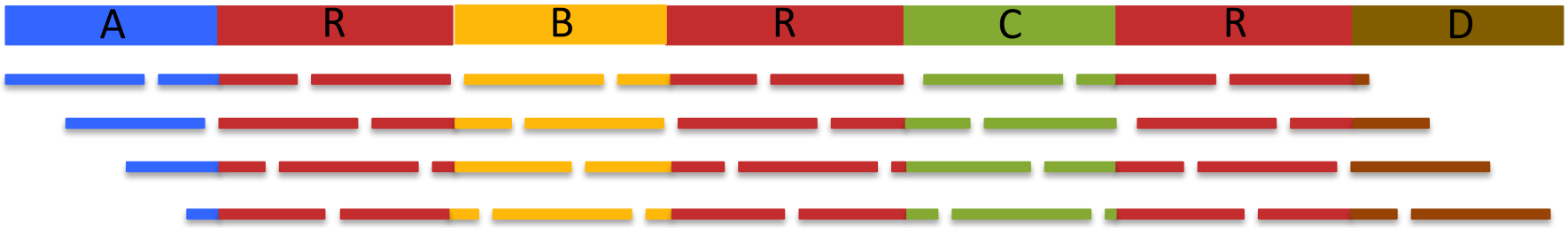
See manual at http://mummer.sourceforge.net/manual

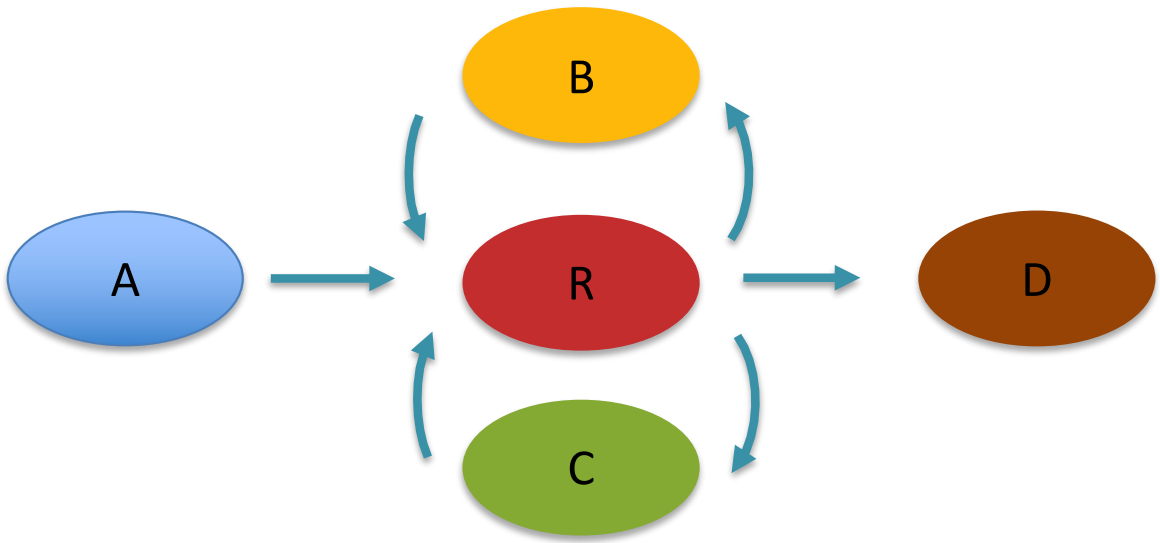


Part 3: Long Range Sequencing and Assembly

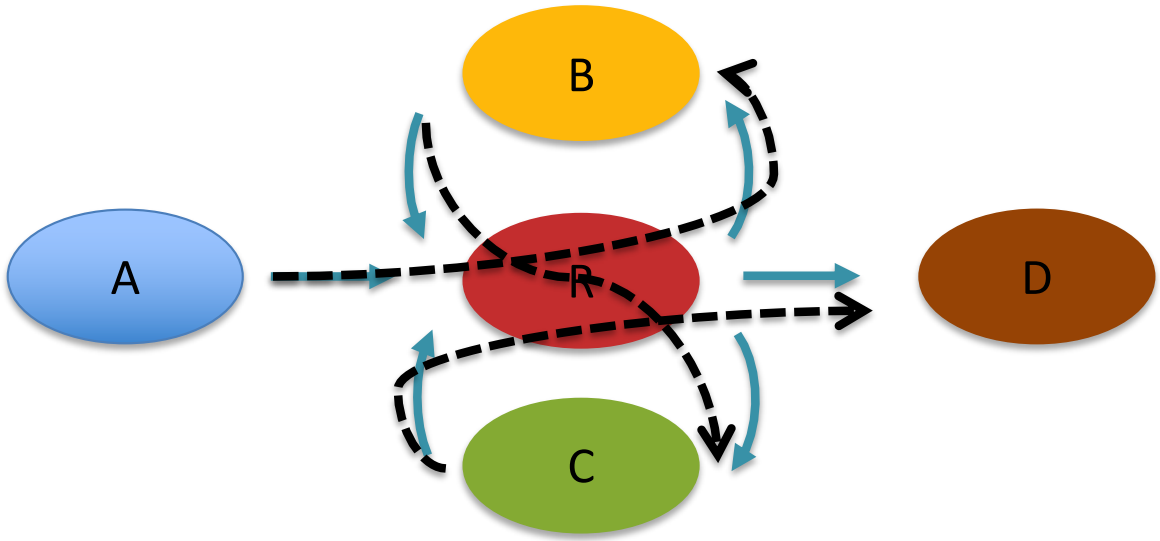
Genomics Arsenal in the year 2018

Assembly Complexity

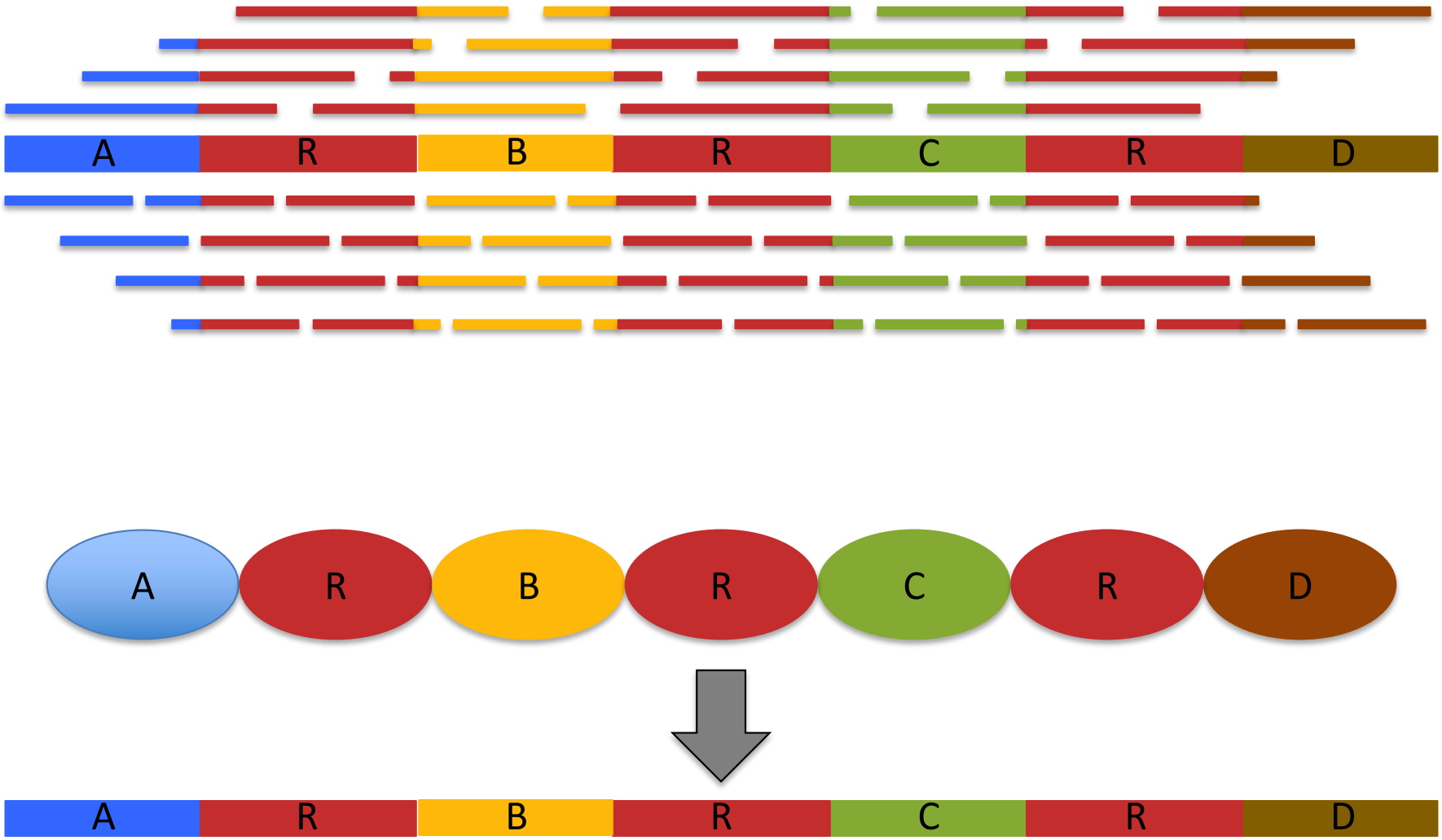




Assembly Complexity



Assembly Complexity

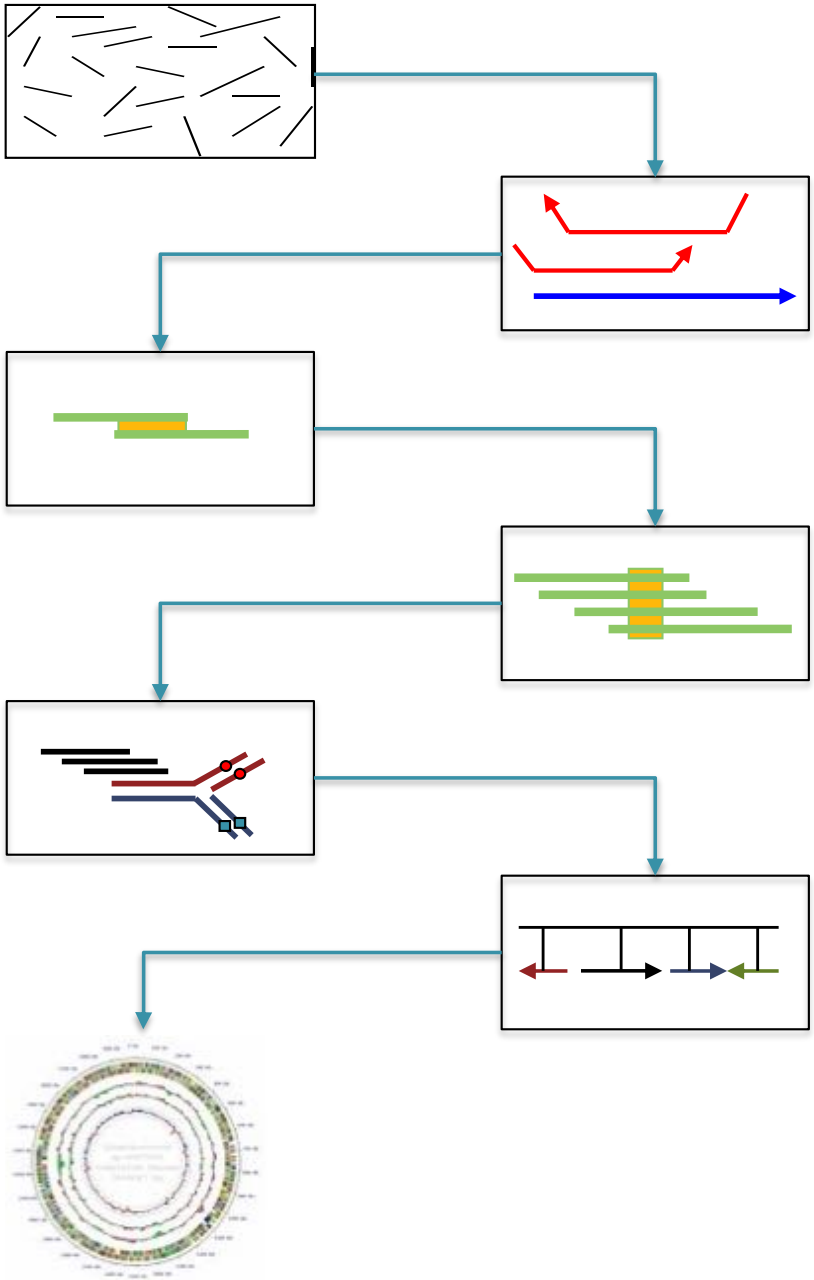


The advantages of SMRT sequencing
Roberts, RJ, Carneiro, MO, Schatz, MC (2013) *Genome Biology*. 14:405

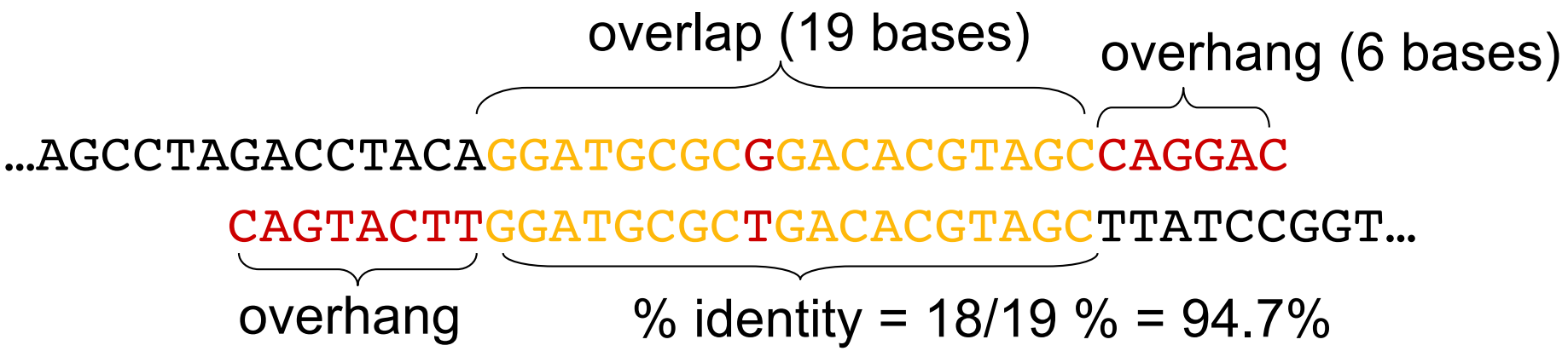
Celera Assembler aka CANU

https://github.com/marbl/canu

1. Pre-overlap
   - Consistency checks
2. Trimming
   - Quality trimming & partial overlaps
3. Compute Overlaps
   - Find high quality overlaps
4. Error Correction
   - Evaluate difference in context of overlapping reads
1. Unitigging
   - Merge consistent reads
1. Scaffolding
   - Bundle mates, Order & Orient
1. Finalize Data
   - Build final consensus sequences



Overlap between two sequences



overlap - region of similarity between regions
overhang - un-aligned ends of the sequences

The assembler screens merges based on:

- length of overlap
- % identity in overlap region
- maximum overhang size.

[How do we compute the overlap?]

Very fast overlapping

nature biotechnology

Assembling large genomes with single-molecule sequencing and locality-sensitive hashing

Konstantin Berlin[1–3,6], Sergey Koren[4,6], Chen-Shan Chin[5], James P Drake[5], Jane M Landolin[5] & Adam M Phillippy[4]

Long-read, single-molecule real-time (SMRT) sequencing is routinely used to finish microbial genomes, but available assembly methods have not scaled well to larger genomes. We introduce the MinHash Alignment Process (MHAP) for overlapping noisy, long reads using probabilistic, locality-sensitive hashing. Integrating MHAP with the Celera Assembler enabled reference-grade de novo assemblies of Saccharomyces cerevisiae, Arabidopsis thaliana, Drosophila melanogaster and a human hydatidiform mole cell line (CHM1) from SMRT sequencing. The resulting assemblies are highly continuous, include fully resolved chromosome arms and close persistent gaps in these reference genomes. Our assembly of D. melanogaster revealed previously unknown heterochromatic and telomeric transition sequences, and we assembled low-complexity sequences from CHM1 that fill gaps in the human GRCh38 reference. Using MHAP and the Celera Assembler, single-molecule sequencing can produce de novo near-complete eukaryotic assemblies that are 99.99% accurate when compared with available reference genomes.

Genome assembly is the process of reconstructing a genome from a collection of short sequencing reads and is an integral step in any genome project[1–3]. Unlike resequencing projects, de novo assembly is performed without the aid of a reference genome, rather, the genome is reconstructed from scratch. An accurate reconstruction is crucial, as both the continuity and base accuracy of an assembly can affect the results of all downstream analyses[4]. However, repetitive sequences make assembly difficult when the repeat length exceeds the read length[5]. Most high-throughput sequencing methods generate sequencing reads of only a few hundred base pairs, which is shorter than most common repeats. Although short reads are sufficient for many analyses, they are not sufficient for resolving major repeat families in either microbial or eukaryotic genomes, leading to fractured and incomplete assemblies[6].

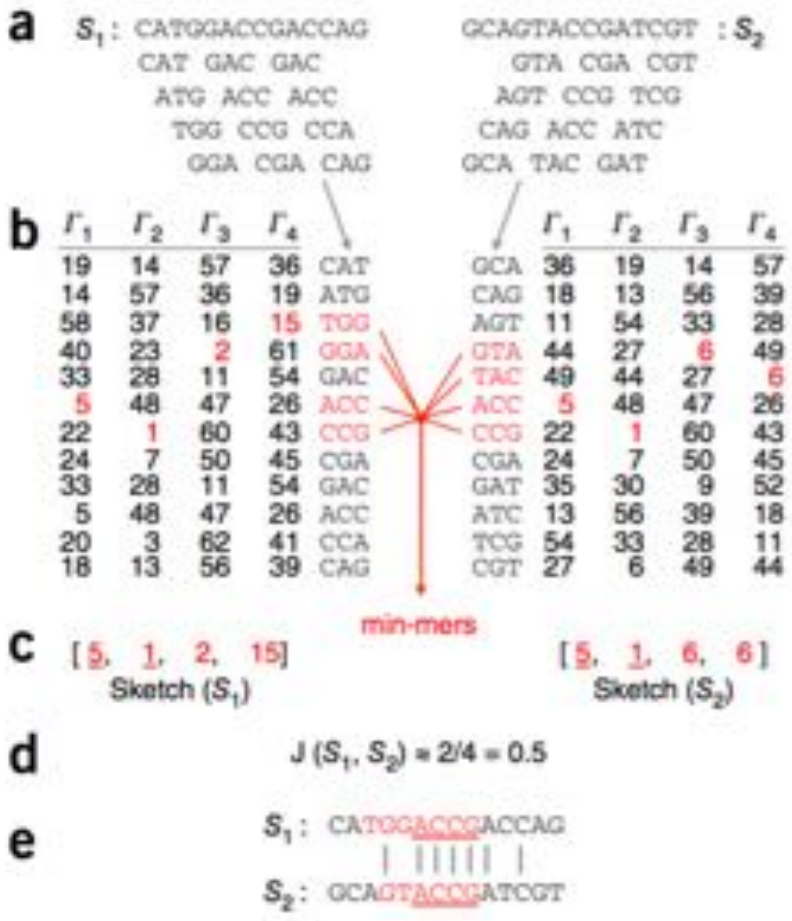
Recent advances in single-molecule sequencing technologies have promised reads hundreds of fold longer than second-generation methods[6,7]. Most notably, Pacific Biosciences' SMRT sequencing was the first commercially available long-read technology[7]. Using a DNA polymerase anchored in a zero-mode waveguide, PacBio SMRT sequencing has delivered usable reads of up to 54 kbp[8]. Preliminary results from Oxford Nanopore suggest that >10 kbp read lengths are possible using MinION nanopore sequencing[9]. Such long-read lengths dramatically simplify genome assembly by resolving repetitive structures in the assembly graph[10–12]. However, the long reads generated by single-molecule sequencing currently suffer from low accuracy (81–87% PacBio[13], 78–80% MinION[9]), and new algorithms have been needed to compensate for mistakes in the raw sequences[10,11,14]. Although SMRT sequencing may be error prone, it exhibits less sequencing bias than previous technologies[14,15], and theoretical research has shown that random error can be overcome algorithmically[16]. Thus, by oversampling the genome at sufficient coverage (e.g., 30× of PacBio P5C3), SMRT sequencing can be used to produce highly accurate and continuous assemblies[11,13,14], including automatically finished genomes for most bacteria and archaea[11].

Early assemblies of noisy, long reads have been successful, but have suffered from a substantial computational cost. For example, an initial assembly of *D. melanogaster* from SMRT reads required more than 600,000 CPU hours, when tools available at the time were used—the equivalent of more than 32 days running on a thousand-core computer cluster[17]. Even small bacterial genomes previously required a day to assemble using the HGAP[11] or PBcR[12] assembly pipelines. The primary bottleneck of long-read assembly has been the sensitive all-versus-all alignment required to determine overlapping read pairs. For the *D. melanogaster* SMRT assembly, this overlapping step consumed 93% of the total runtime. Even if the accuracy of long-read technologies improves, all-pairs overlapping will remain a substantial bottleneck in overlap-layout-consensus assembly[3]. Taken together, the computational cost and the comparatively high sequencing costs have prevented widespread application of SMRT sequencing to genomes larger than 100 Mbp. The steadily increasing throughput of the PacBio instrument has begun to address sequencing costs, but the computational cost of assembling larger genomes has remained beyond the reach of most investigators.

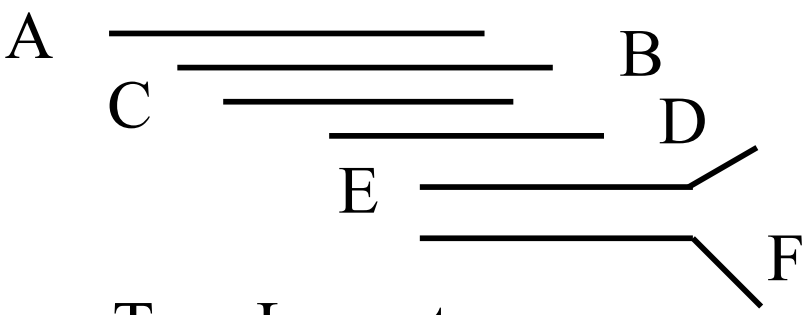
To solve the computational problem of long-read assembly, we present a probabilistic algorithm for efficiently detecting overlaps between noisy long reads. MHAP uses a dimensionality reduction technique termed MinHash[18] to create a more compact representation of sequencing reads. Originally developed to determine the similarity of web pages[19], MinHash reduces a text or string to a small set of fingerprints, called a sketch. MinHash sketches have been successfully

[1]Department of Chemistry and Biochemistry, University of Maryland, College Park, Maryland, USA. [2]Institute for Advanced Computer Studies, University of Maryland, College Park, Maryland, USA. [3]Computer Labs, Arlington, Virginia, USA. [4]National Biodefense Analysis and Countermeasures Center, Frederick, Maryland, USA. [5]Pacific Biosciences of California, Inc., Menlo Park, California, USA. [6]These authors contributed equally to this work. Correspondence should be addressed to K.B. (konstantin@cbcb.umd.edu).

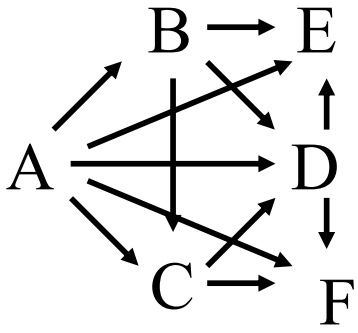
Received 14 August 2014; accepted 8 April 2015; published online 25 May 2015; doi:10.1038/nbt.3238



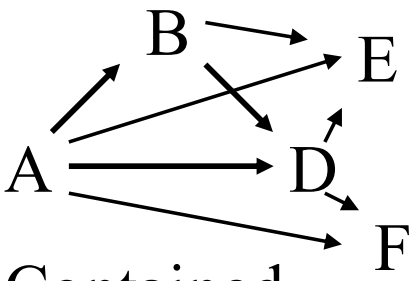
Unitigging: Pruning the Overlap



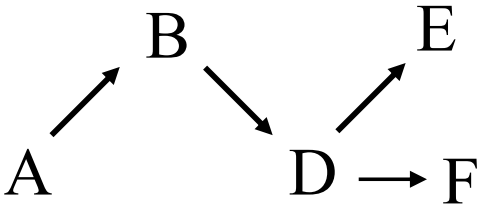
True Layout



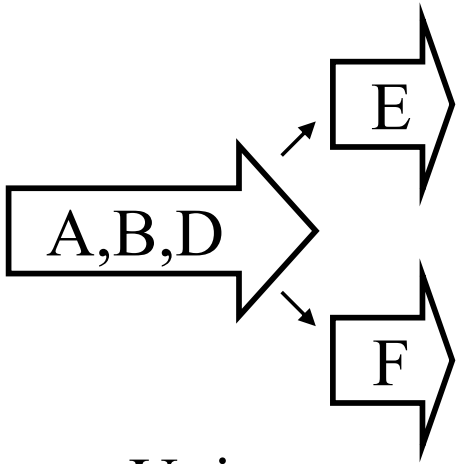
Original Overlap Graph



Contained Read Removal



Transitive Edge Removal



Unique Join Collapsing

Theorem: SCS of unitigs = SCS of reads


Welcome to Applied Comparative Genomics
https://github.com/schatzlab/appliedgenomics2018

Questions?