REBECCA B. DIKOW

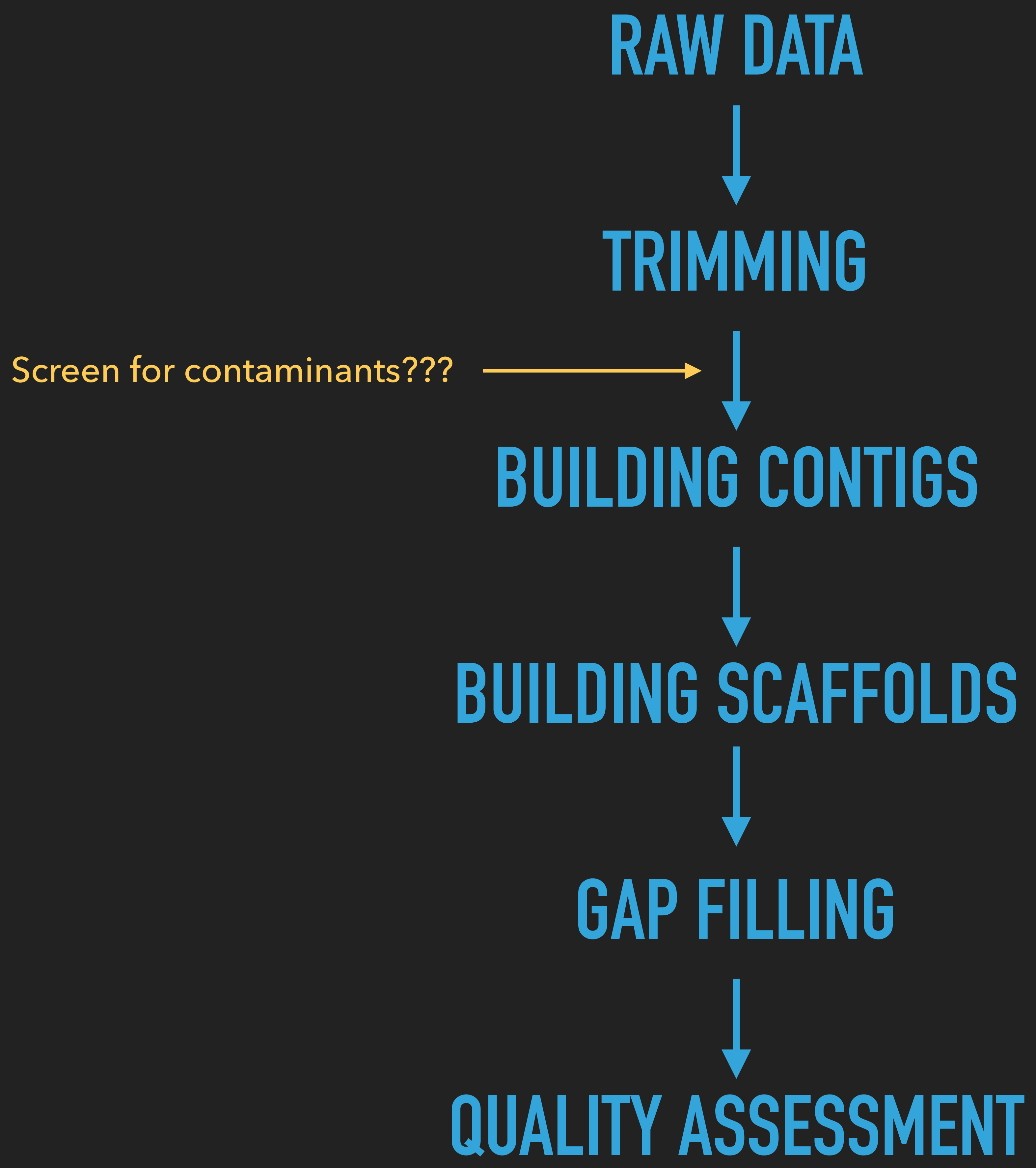
GENOME ASSEMBLY

LONGER COURSE: COMING SOON

- Email me: DikowR@si.edu with topics for which you would like more in-depth coverage.
- 6-week course to start in June (tentatively)

WHY IS ASSEMBLY SUCH A BIG CHALLENGE???

- Mike Schatz (Johns Hopkins) teaches an example using the Charles Dickens novel *A Tale of Two Cities*
- Available on his website: http://schatzlab.cshl.edu/teaching/2014/



RAW DATA

TRIMMING

Screen for contaminants???

BUILDING CONTIGS

BUILDING SCAFFOLDS

GAP FILLING

QUALITY ASSESSMENT

TERMINOLOGY!

READ

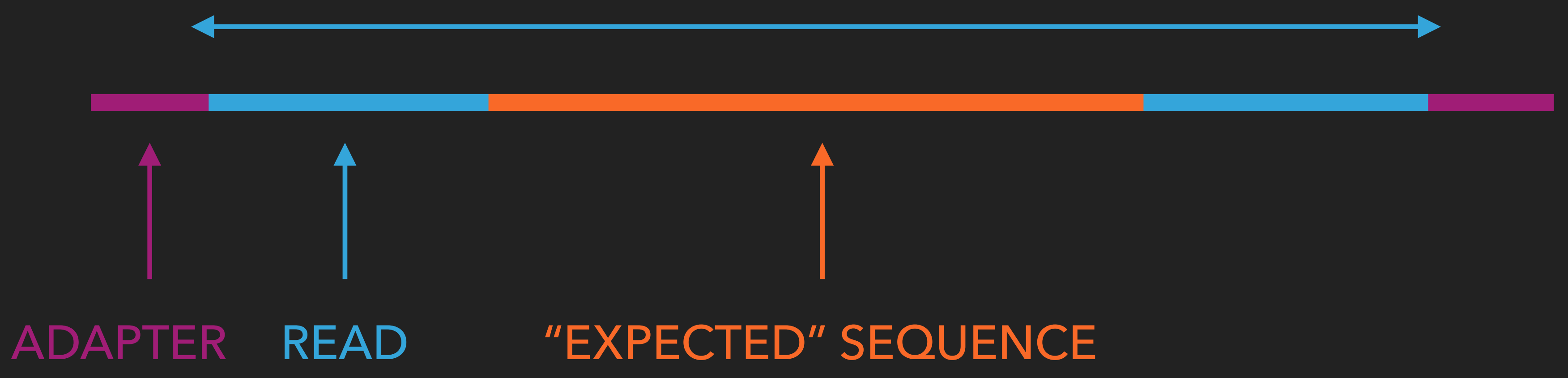
- Raw fragment of DNA as it has been “read” by the sequencer
- These are “real” except for any base-calling errors
  - Note: different sequencing platforms produce different kinds of errors with different frequencies

LIBRARY

- Set of DNA fragments of a particular size, attached to adapters
    - e.g. a 250bp library, or a 3Kbp library

INSERT SIZE

- Length of your fragment with adapters excluded



*This is an Illumina paired-end HiSeq example

PAIRED-END

- Sequencing from both ends of a particular fragment.

MATE-PAIR

- A kind of library that allows you to have large insert sizes (up to 40 Kbp for Illumina sequencing).

KMER

- A short substring of a particular length (k)
- Before contigs can be built, de Bruijn graph assemblers count occurrences of all such substrings
- kmer distribution can give us an estimate of genome size, and repeats
- JELLYFISH is the best known kmer counting program

CONTIG

- Definition from Celera website:
    - A contig consists of a set of reads, a layout that includes all the reads and leaves no gaps, a multiple sequence alignment of the reads, and a consensus sequence. In practice contigs consist of one or more unitigs. Note the consensus may contain (small) gaps spanned by reads even though the layout includes no (0X) gaps.

UNITIG

- Definition from Celera website:
    - A high-confidence contig seed. The end of a unitig is, by definition, a place where the overlap data shows multiple, mutually contradictory, paths. Unitigs are supposed to end at repeats.

SCAFFOLD

- Definition from Celera website:
    - A linear ordering of contigs joined by mate pairs. A scaffold defines the order and orientation (DNA strand) for each component contig. There are two ways to measure scaffold length. "Scaffold bases" is sum of contig lengths. "Scaffold span" is that plus the sum of gap lengths. Celera Assembler uses complex criteria to build scaffolds, but some generalizations apply. Every gap in a scaffold was spanned by at least two mate pairs. A gap with negative length means the sequence data and mate data disagree. Usually, negative gaps are small (20bp) and induced by low-quality sequence at the end of a read. In the FASTA representation of a scaffold, negative gaps are represented by a fixed number (20) of N's.

N50

- The contig length such that using equal or longer contigs produces half the bases of the genome.

NG50

- From a set of sorted scaffold lengths, at what contig or scaffold length do we see a sum length that is greater than half of the genome size?

L50

- Bradman: the number of sequences evaluated at the point when the sum length exceeds 50% of the assembly size is sometimes referred to as the L50 number. Admittedly, this is somewhat confusing: N50 describes a sequence length whereas L50 describes a number of sequences

FINISHED GENOME

- Assembled to chromosome:
  - Lots of Bacteria and Archaea
  - Arguably no Eukaryotes
    (even the human genome has gaps)

OLC ASSEMBLERS VS. DE BRUIJN GRAPH ASSEMBLERS

- Take home message:
  - OLC developed first
  - De Bruijn graph method developed to deal with repetitive bacterial genome
  - De Bruijn methods work better with short fragment data, which took over for a while
  - OLCs are back now, with the infusion of PacBio

OLC

- Overlap, Layout, Consensus
- e.g. Celera (Myers, 2000)
    - Celera most known (and still used) OLC assembler: others are TIGR, Arachne, Newbler, Phrap, PCAP
    - Canu is the fork of Celera assembler still being developed to deal with PacBio and Nanopore data: https://github.com/marbl/canu

DE BRUIJN GRAPH

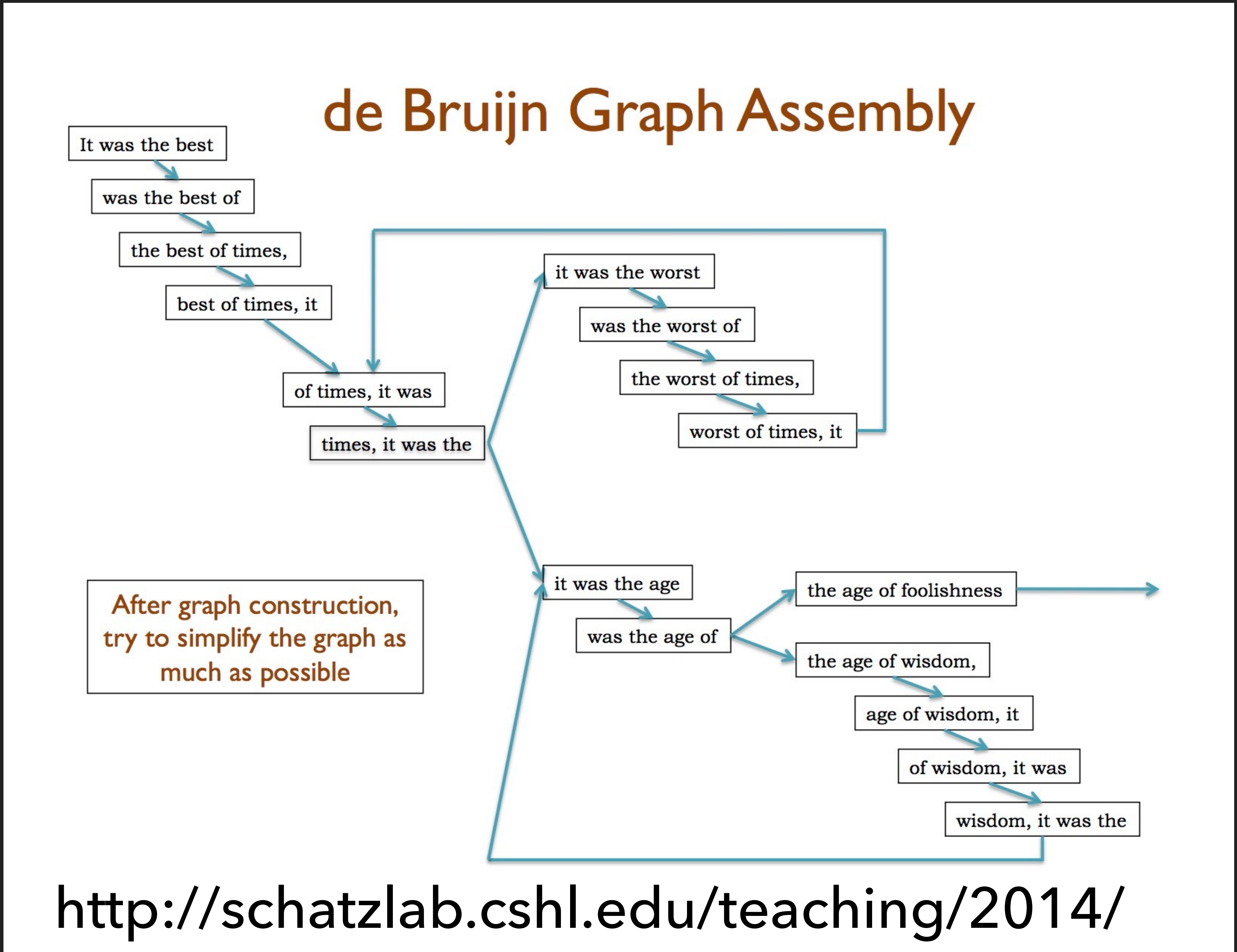
- De Bruijn graph: Pevzner, 2001 (Euler)
    - Most of the assemblers you know use De Bruijn graphs.

OLC VS. DE BRUIJN

- With short reads, overlap consensus assembly suffers from two main problems:
  - short read length means the overlaps must be calculated over a large proportion of the read to retain accuracy
  - the huge number of reads increases the number of links, so that the contig path is difficult to compute.
- The de Bruijn graph approach circumvents the problems of overlap consensus assembly. Rather than using the reads 'as is' and trying to link them, the k-mers (all subsequences of length k within the reads) are computed and the reads are represented as a path through the k-mers. Such a paradigm handles redundancy better than the overlap consensus approach and makes the computation of paths more tractable.

MacLean *et al.* 2009, Nature Reviews Microbiology

DE BRUIJN GRAPH



de Bruijn Graph Assembly

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

http://schatzlab.cshl.edu/teaching/2014/

STRING GRAPH (OLC)

- Myers, 2005

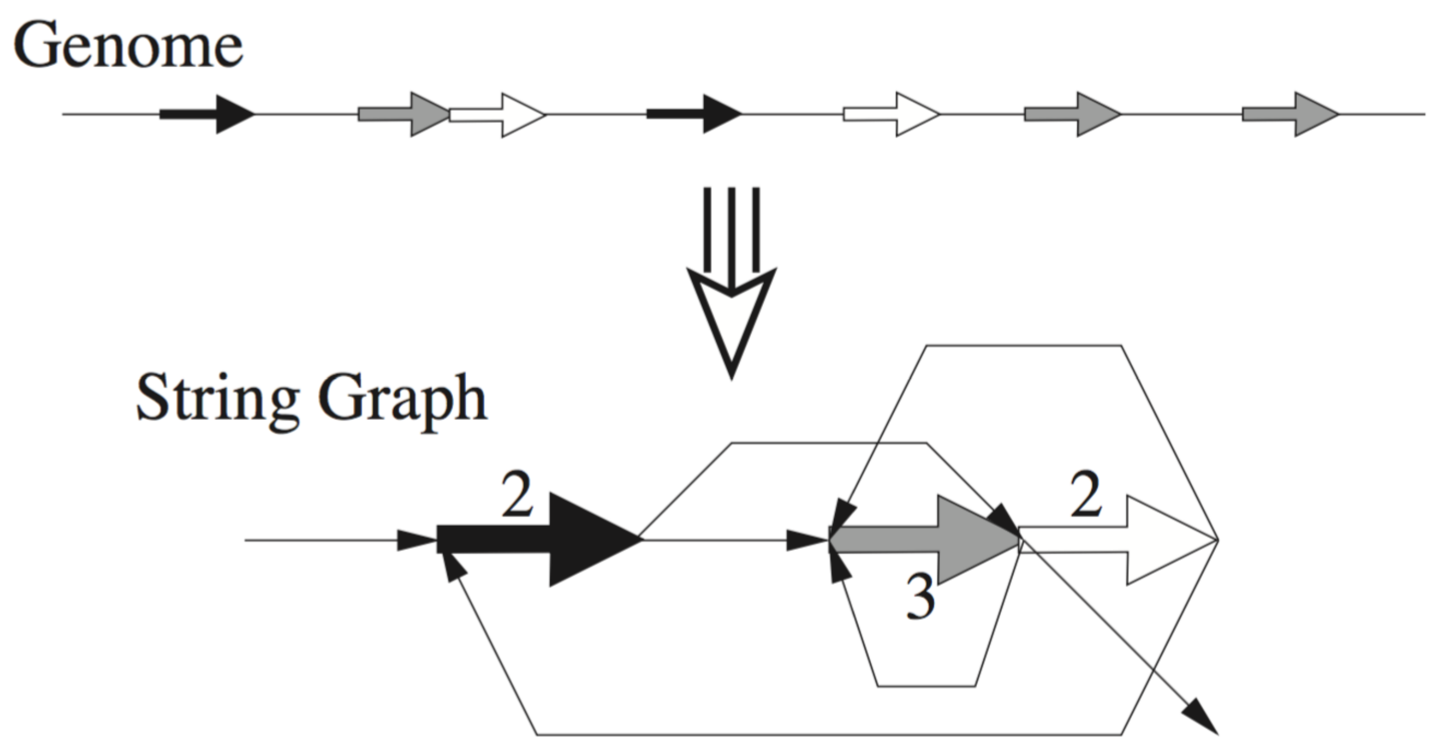


Fig. 1. A genome and its string graph. The thick arrows of the same shade represent identical repetitive sequences. The numbers in the string graph give the number of copies of each repeat inferable by counting entry and exits into the collapsed segment.

STRING GRAPH (OLC)

- Myers, 2005
  - *Campylobacter jejuni*

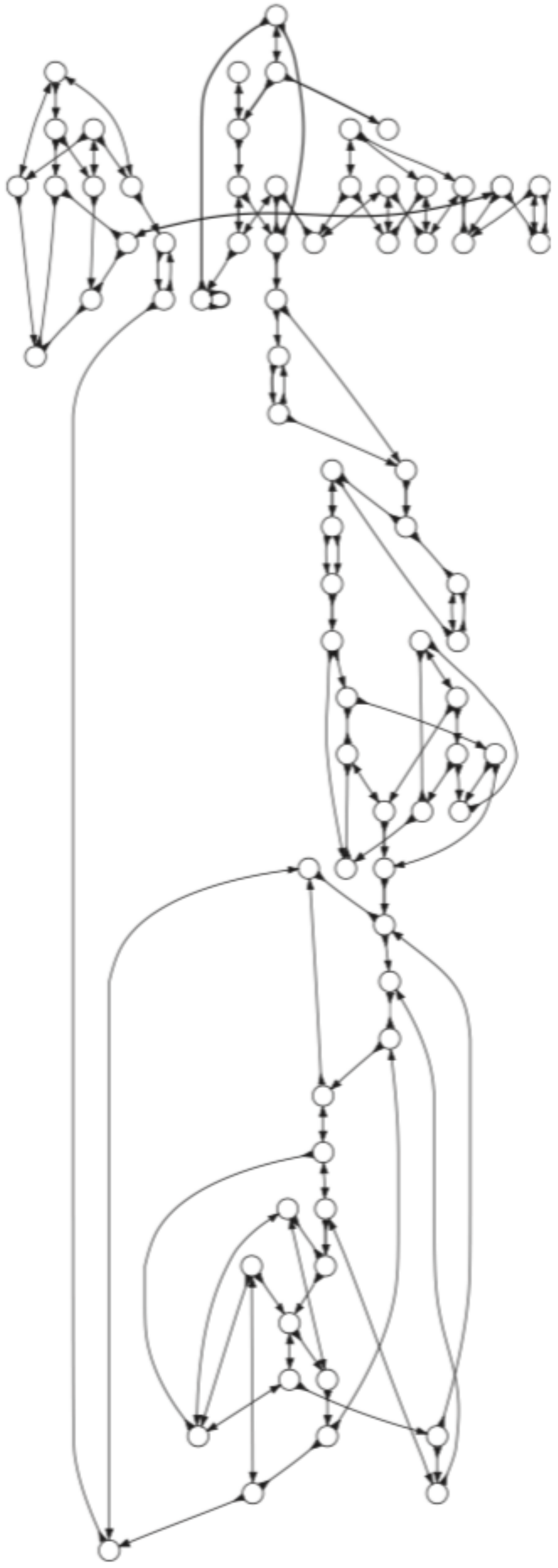
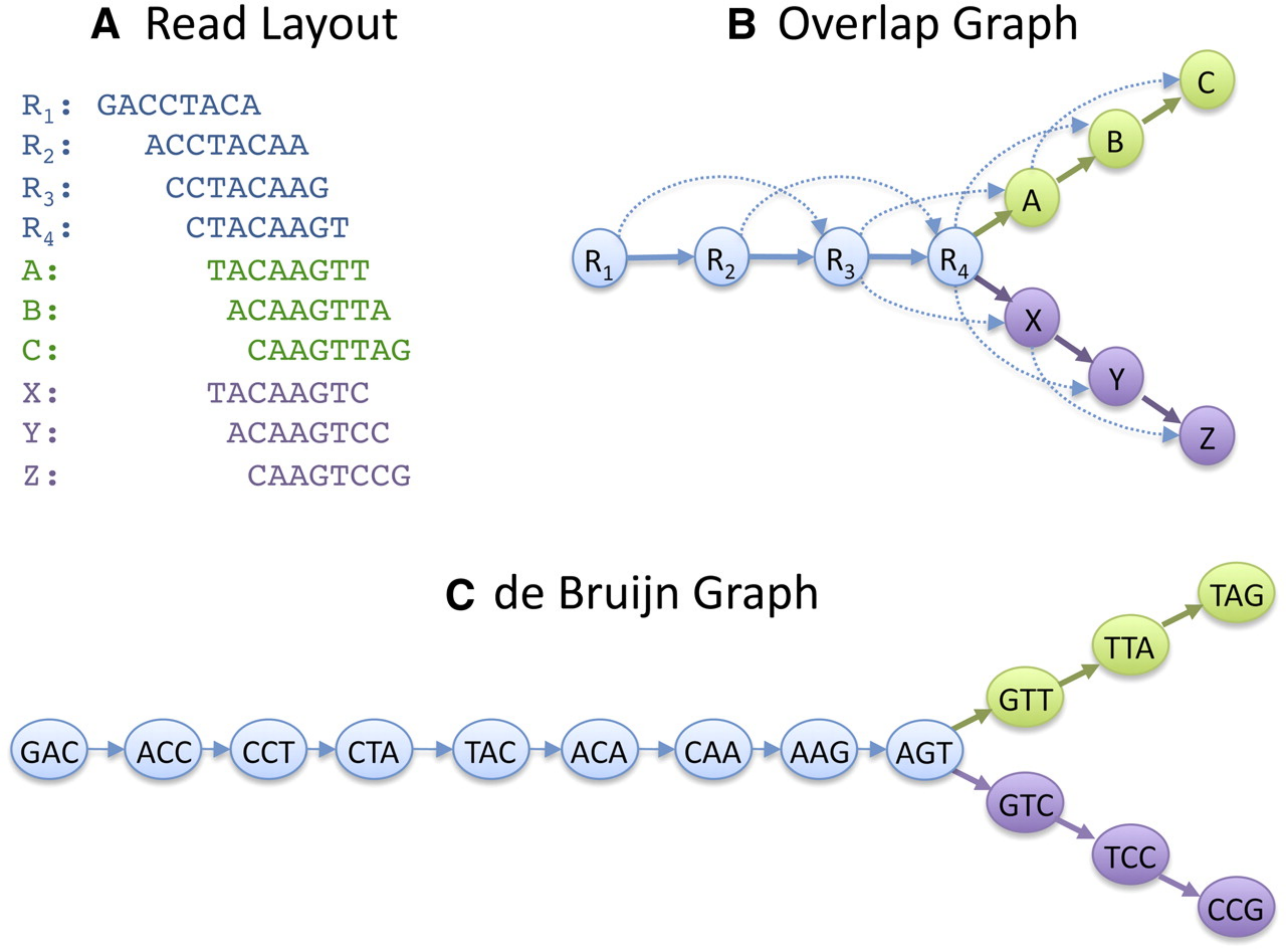


Fig. 3. The bidirected string graph of *C.jejuni* (prior to traversal analysis and compression).



(A), we can build an overlap graph (B) in which each read is a node, and overlaps >5 bp are indicated by directed edges. Transitive overlaps, which are implied by other longer overlaps, are shown as dotted edges. In a de Bruin graph (C), a node is created for every k-mer in all the reads; here the k-mer size is 3. Edges are drawn between every pair of successive k-mers in a read, where the k-mers overlap by k − 1 bases. In both approaches, repeat sequences create a fork in the graph.

Genome Res. 2010. doi:10.1101/gr.101360.109

WHICH ASSEMBLER DO I NEED?

- Depends on:
    - Data (sequencing platform, libraries)
    - Genome size
    - Compute resources at your disposal

ILLUMINA PAIRED-END ONLY

- DISCOVAR (but only if 2X250)
- SPAdes (but only small genomes: bacteria, archaea, fungi, protists)
- ABySS
- MIRA
- Meraculous
- Velvet

ILLUMINA PAIRED END + MATE-PAIR

- ALLPATHS-LG
- SOAP
- MaSuRCA
- Meraculous
- Platanus

MITOCHONDRIAL OR CHLOROPLAST GENOMES

- MITObim (uses MIRA)
- Velvet
- SPAdes
- ABySS

HIGHLY HETEROZYGOUS GENOMES

- DISCOVAR
- Platanus
- Haplomerger (not actually an assembler, but tries to merge your contigs split apart due to heterozygosity)

PACBIO ONLY

- HGAP (smaller genomes) and Quiver
    - Quiver: calls consensus from multiple PacBio reads
- Canu
    - MHAP + Celera
- FALCON (larger genomes + diploid) and Quiver
    - overlapping done with Daligner
- PBJelly 2 to gap fill (using PacBio reads to fill in gaps in scaffolds. Has been shown to work with genomes >1 Gb.)

ILLUMINA + PACBIO

- The reason the Illumina data are good to have is because PacBio data can have problems with indels.
- Two major ways to do this:
  - correct the PacBio with Illumina reads, then assemble the error corrected PacBio reads: MaSuRCA (can use fairly low coverage PacBio ~10X)
  - assemble the PacBio reads then polish with Illumina: Falcon, Canu (need higher than 30X PacBio)

HYBRID ASSEMBLY

- MaSuRCA
  - lots of specialized error correction plus Celera or SOAP
- SPAdes (not for large genomes)

OTHER COMBINATIONS, LEGACY DATA

- MIRA (no PacBio yet, but 454, Sanger, Illumina, Ion Torrent)
- SPAdes (but not for large genomes)

STAND-ALONE SCAFFOLDING

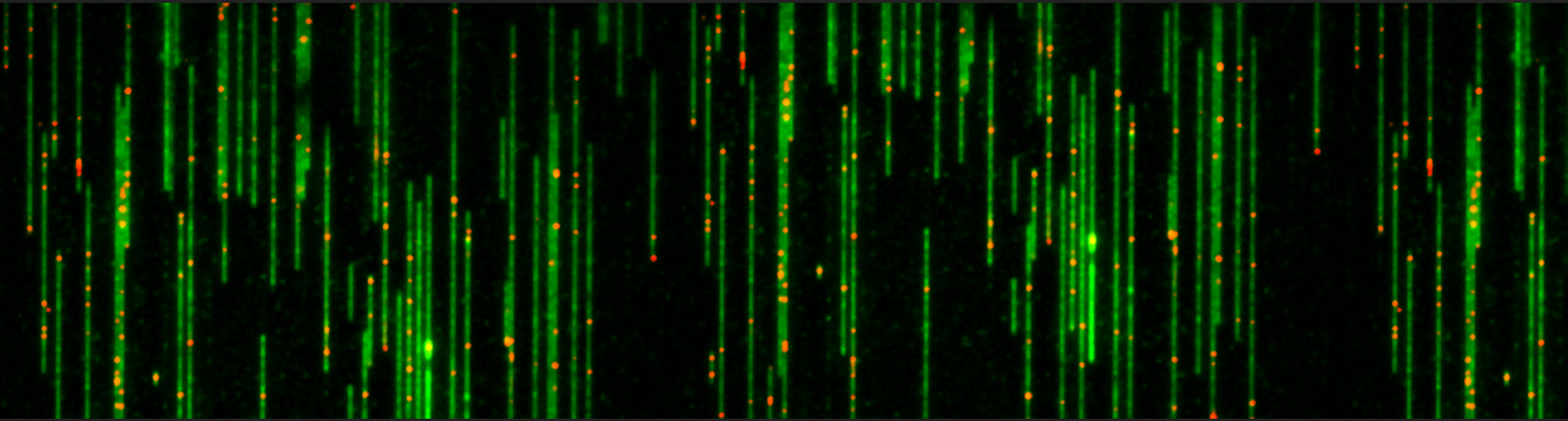
- SSPACE
- Bambus

GAP FILLING

- SSPACE
- PBJelly 2

OPTICAL MAPPING

- BioNano:



VISUALIZATION

- JBROWSE
- UCSC genome browser

QUALITY ASSESSMENT

- BUSCO
- QUAST
- KRAKEN