Assembly

Meghan Forcellati

2nd year Ph.D. student, RGGS

mforcellati@amnh.org

Announcements & Key

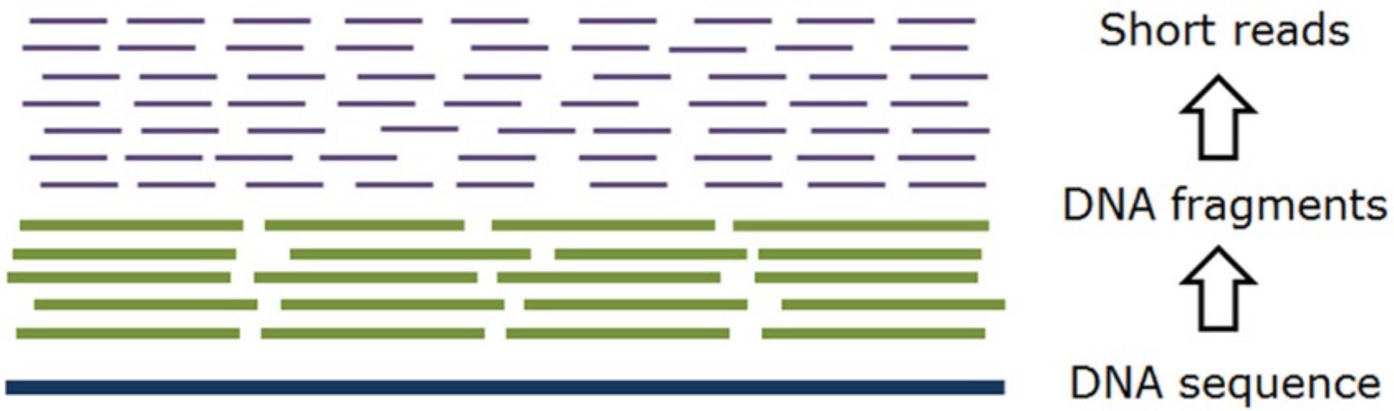
- Highlighted text = Available on Huxley as a module!
- `Code is in this font.` Two useful commands for you all: `module avail`, `module load`
- Exercises at the end of this lecture for you to practice, staggered by level of involvement.
- My background is in working with short read data and I'd be happy to help you, or contact someone else who can.
- Reminder: You have a 15 minute presentation **and annotated bibliography** assignment due **November 14.**
- Reminder: You also have a 350 word proposed research question due **November 14.**
- Session 9 will take place November 1, 9 A.M.
- We need to reschedule Session 13, TBD (week following the holiday).

Most important conceptual points you should try to learn/review after this lecture:

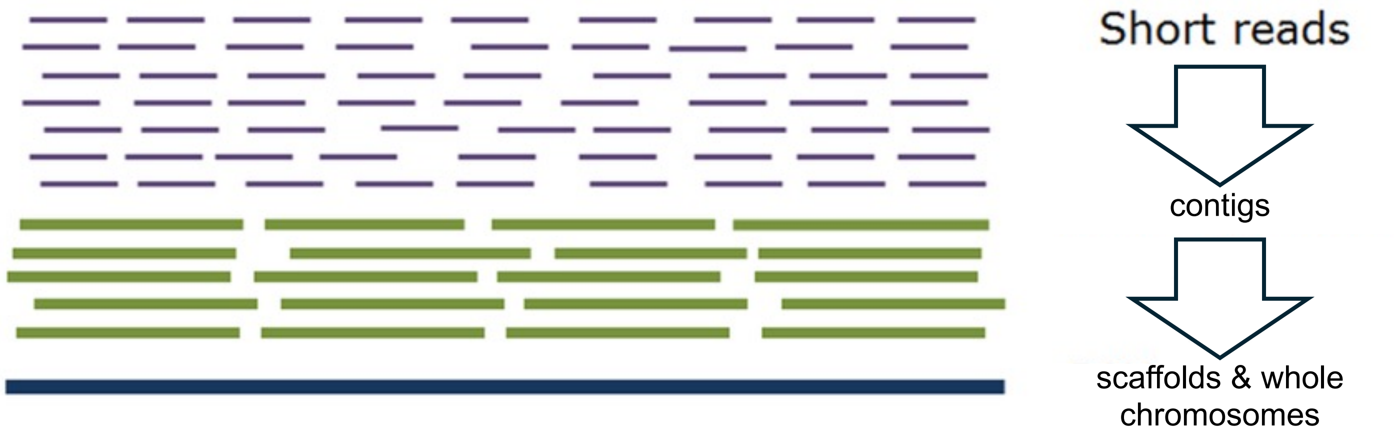
- What is an assembly?
- How do sequencing technologies affect the assembly process?
- What is the difference between *de novo* and reference-guided assembly? Why would you choose one or another for your own research?
- Roughly explain how 1-2 of the algorithms underlying *de novo* assembly work in your own words.
- Why is assembly challenging?
- What are 2-3 ways we could quality-check an assembly?
- What are a few (2-3) assembly programs and 1-2 genome projects?

What is assembly?

Lab work



Assembly



contigs

scaffolds & whole chromosomes

Image: Chu *et al.*, 2013. PLoS ONE 8(3):e59484. Also see Sohn & Nam, 2018. *Briefings in Bioinformatics* 19(1):23-40.

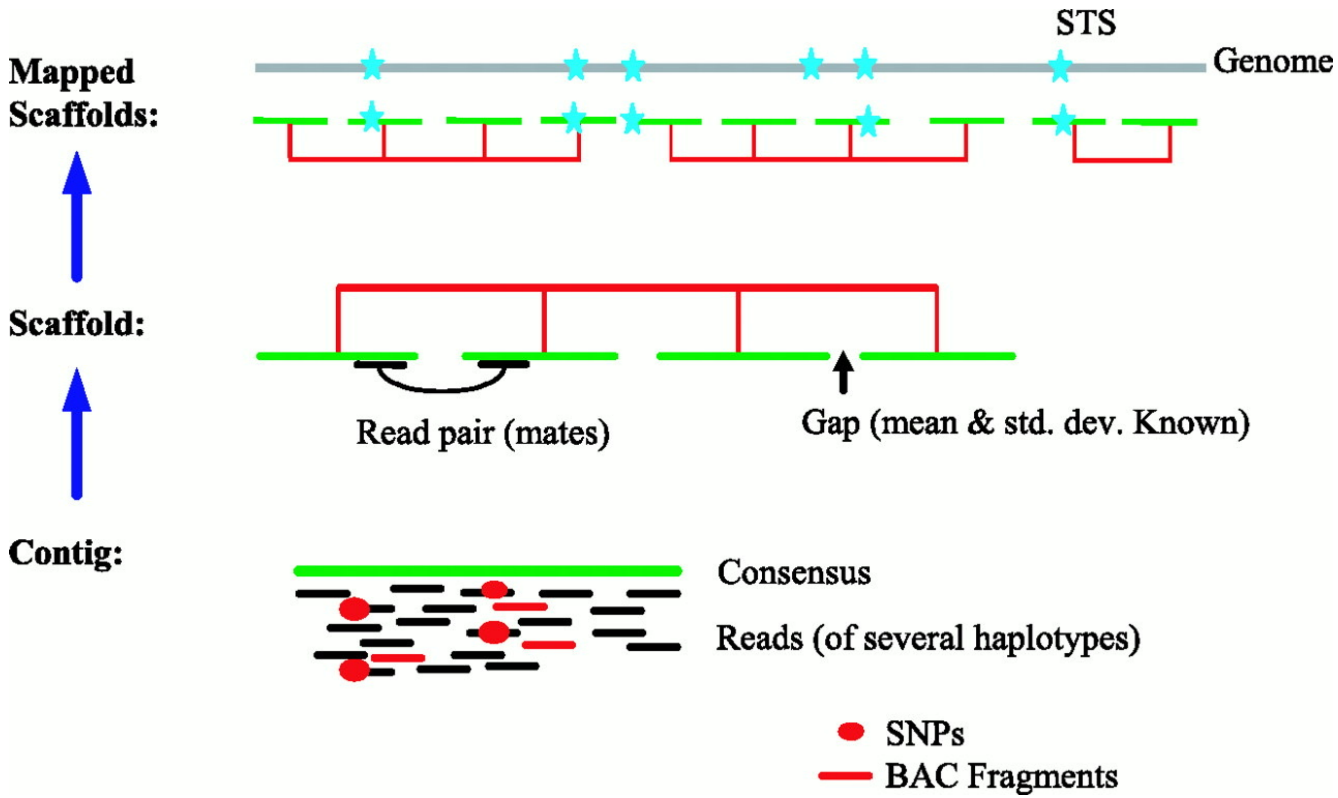
Formal Definition

“**Assembly:** a set of chromosomes, unlocalized and unplaced (random) sequences and alternate loci used to represent an organism's genome. Most current assemblies are a haploid representation of an organism's genome, although some loci may be represented more than once (see Alternate locus, above). This representation may be obtained from a single individual (e.g. chimp or mouse) or multiple individuals (e.g. human reference assembly). Except in the case of organisms which have been bred to homozygosity, the haploid assembly does not typically represent a single haplotype, but rather a mixture of haplotypes. As sequencing technology evolves, it is anticipated that diploid sequences representing an individual's genome will become available.”

-Genome Reference Consortium (NCBI)

How did we get the first genomes?

- *C. elegans*, mouse, human – 500 to 1,000 bp Sanger sequencing reads, 1000s of clones, 200-300 kb inserts, genetic maps. Took 13 years!



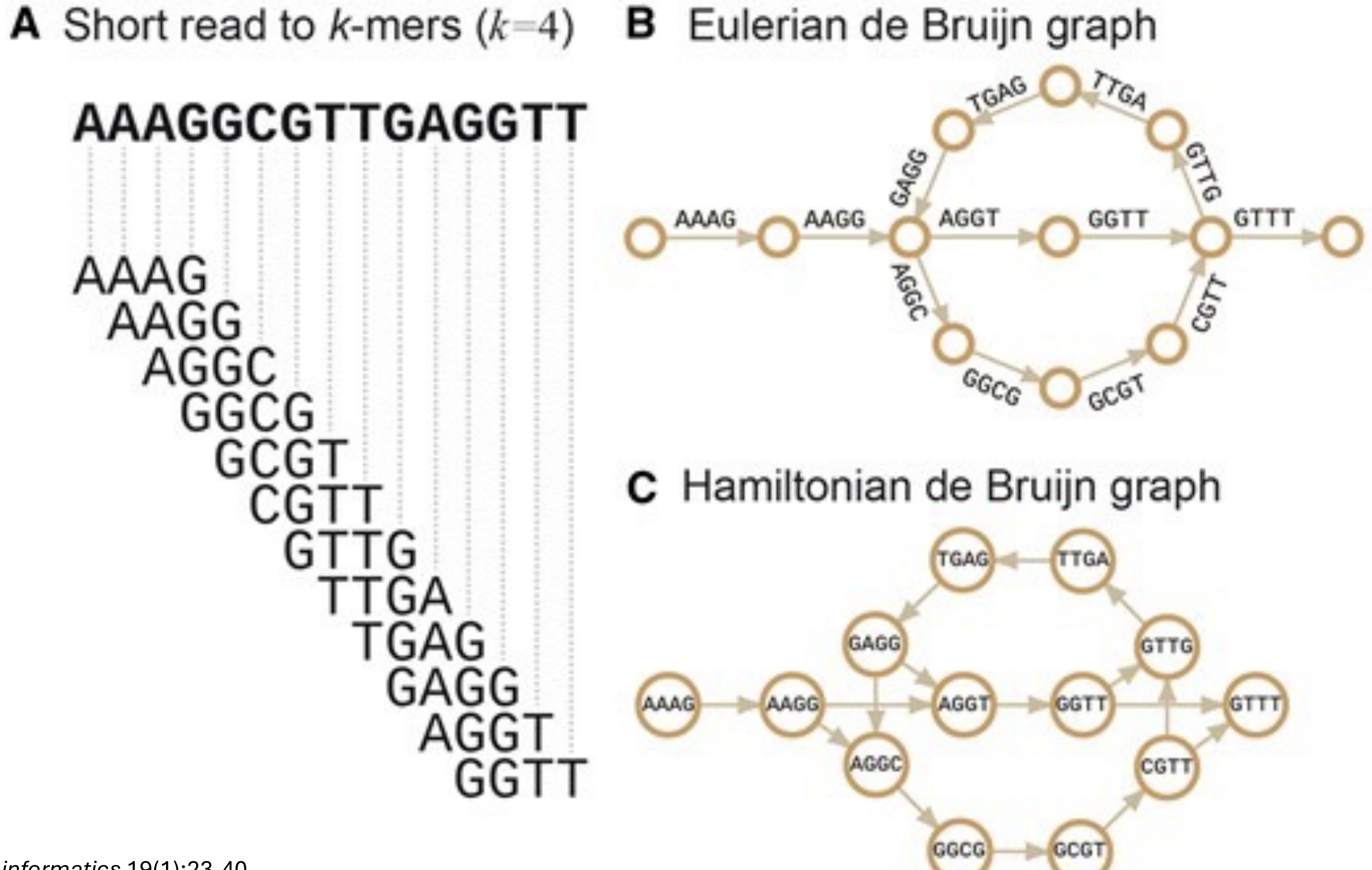
International Human Genome Sequencing Consortium, 2001. *Nature* 409:860-921; Venter *et al.*, 2001. *Science* 291(5507):1304-1351

Assembly Algorithms

- 2 main classes
  - Overlap Layout Consensus
    - Time-complexity is higher O(n+a), $O(N^2)$.
    - Better at dealing with repeat elements.
    - <u>(It is called this because these are literally the steps you follow, so it is easy to remember.)</u>
    - **Common for long reads or PCR.**
  - De Bruijn graphs
    - Time-complexity is lower.
    - Worse at dealing with repeat elements.
    - **Common for short reads.**

Loosely based on teaching materials by Ben Langmead, John Hopkins, & David Grifford, Lecture 6: Genome Assembly, Foundations of Computational Systems Biology, MIT.

Graphs are used to represent sequence data for assembly algorithms.



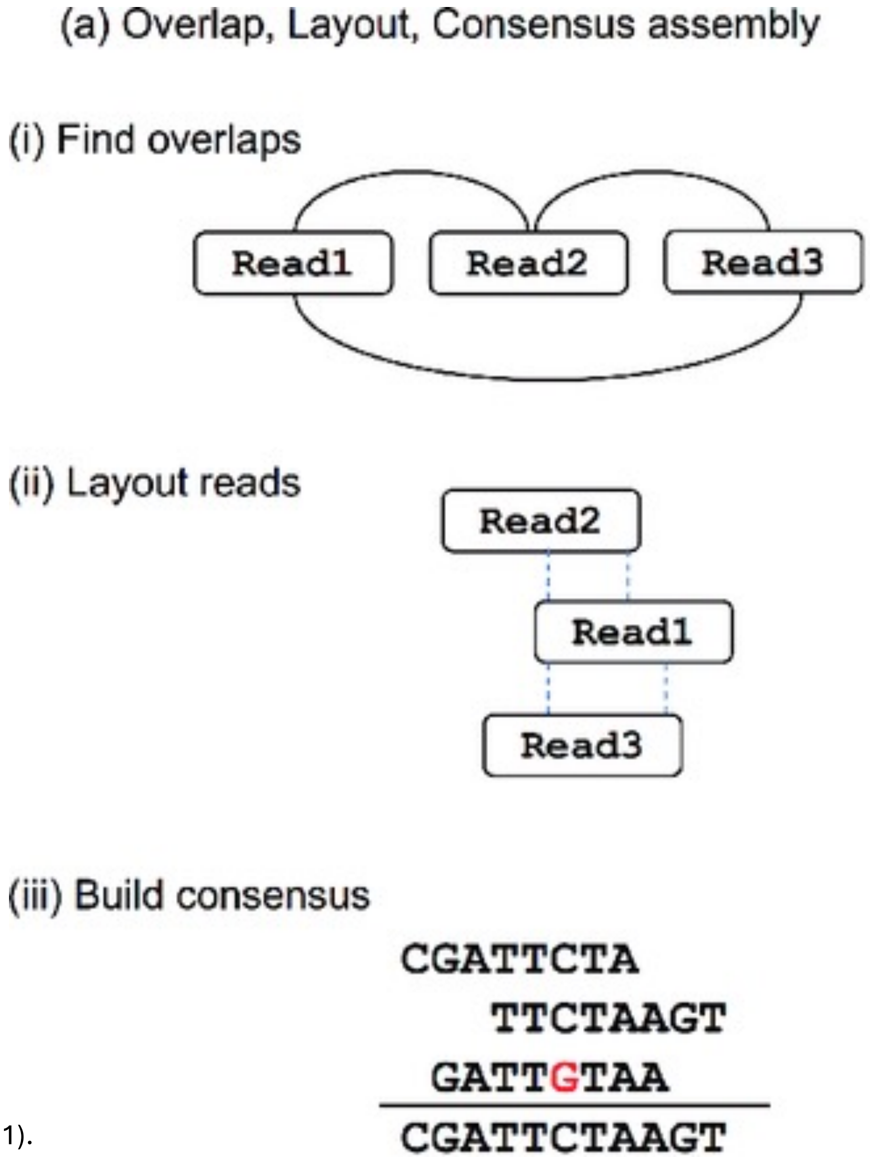
Sohn & Nam, 2018. *Briefings in Bioinformatics* 19(1):23-40.

Overlap-layout consensus algorithm

1. Overlap of reads

2. Layout all the information on a graph.

3. Generate a **<u>C</u>**onsensus (summary)



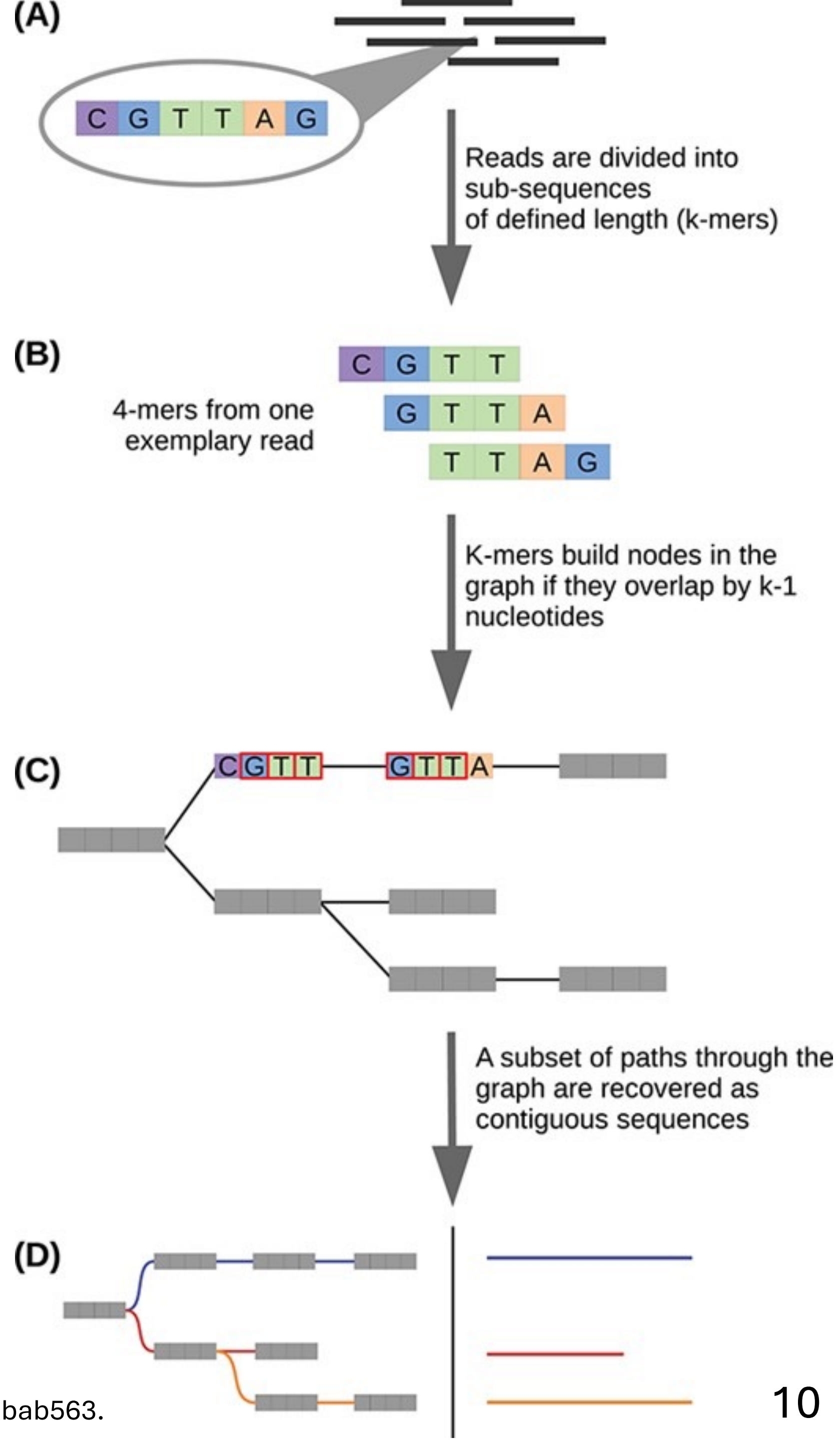
Staden, 1979. *Necleic Acids Res.*, 6:2601-10; Li *et al.*, 2012. 11(1):25-37. Ayling *et al.*, 2019, *Briefings in Bioinformatics* 21(D1).

De Bruijn Graph (DBG)

1. Chop reads into k-mers

2. Form a DBG with k-mers

3. Infer genome from DBG by grouping & filtering.

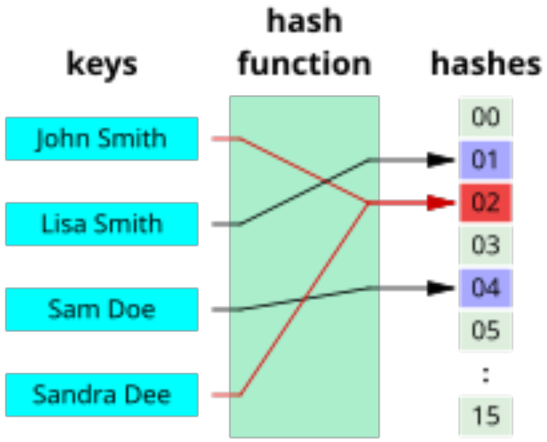


(A)
C G T T A G
Reads are divided into sub-sequences of defined length (k-mers)
(B)
4-mers from one exemplary read
C G T T
G T T A
T T A G
K-mers build nodes in the graph if they overlap by k-1 nucleotides
(C)
C G T T
G T T A
A subset of paths through the graph are recovered as contiguous sequences
(D)

Idury & Waterman, 1995. *J Comput Biol.* 2:291-306; Li *et al.*, 2012. 11(1):25-37; Raghavan *et al.*, 2022. *Briefings in Bioinformatics* 23(3):bbab563.

Alignment - Other useful algorithms to know

- Hashing – Map data of arbitrary size to fixed-size values.
    - (ex: SHRiMP, Maq, RMAP, ZOOM, ABySS, Meraculous)
- Burrows-Wheeler Transform (BWT) – Efficiently (O(n)) and reversibly transform strings to run faster and use less memory.
    - (ex: BowTie, BWA)



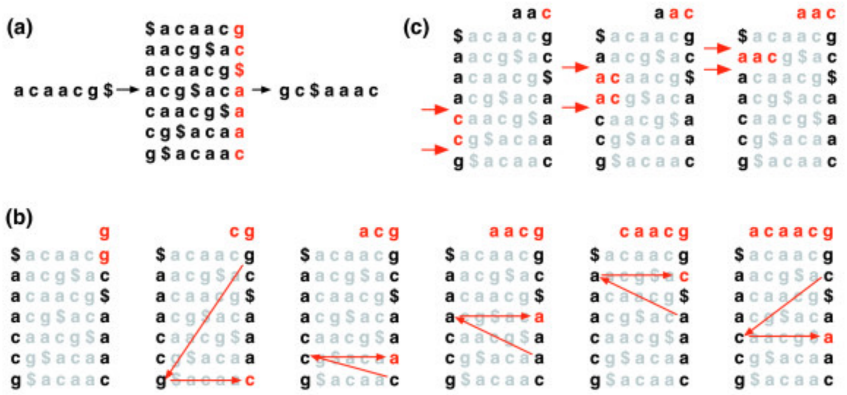
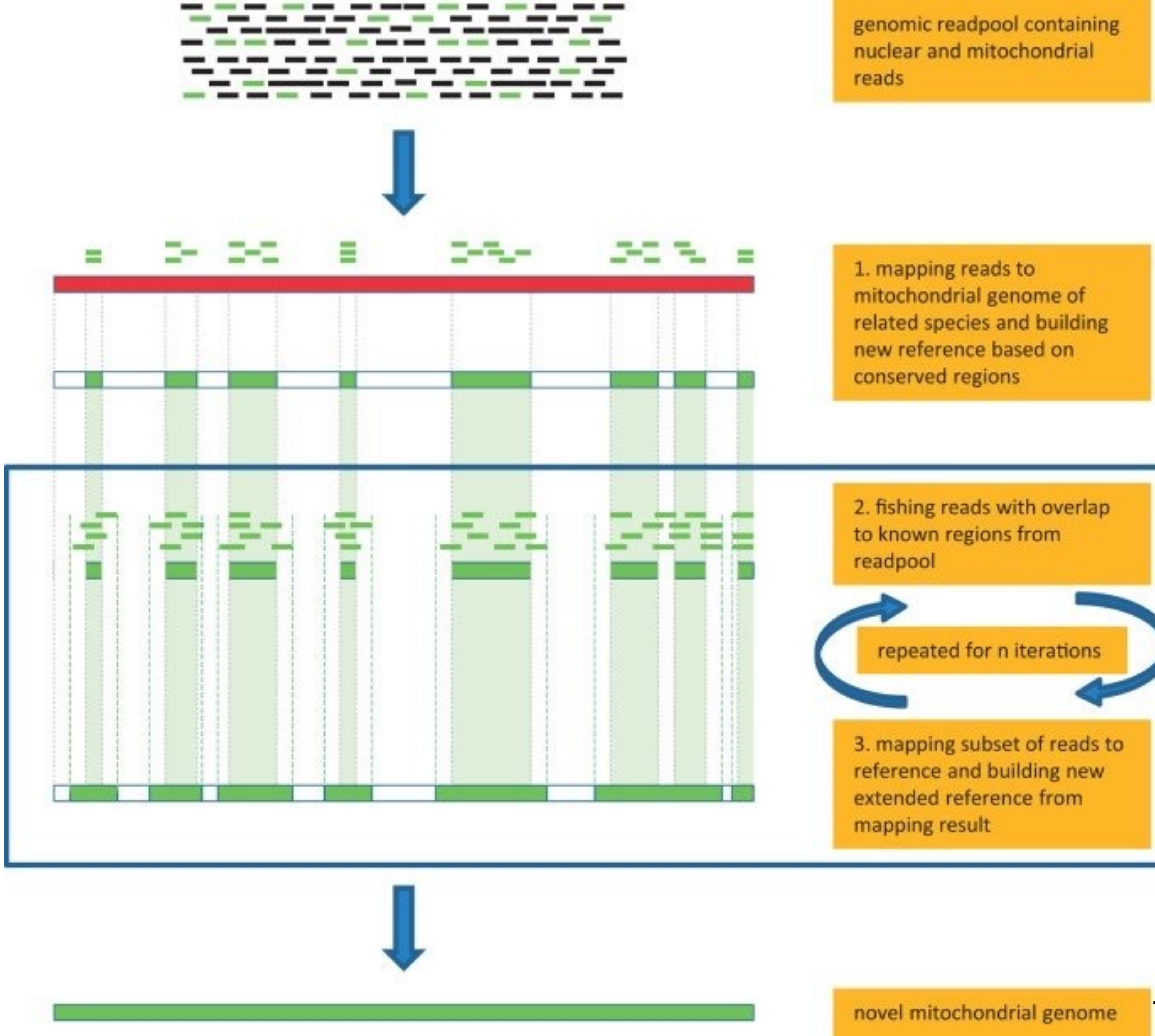


Image: Jorge Stolfi; Langmead *et al.*, 2009. *Genome Biol* 10(3):R25

Iterative mapping



genomic readpool containing nuclear and mitochondrial reads

1. mapping reads to mitochondrial genome of related species and building new reference based on conserved regions

2. fishing reads with overlap to known regions from readpool

repeated for n iterations

3. mapping subset of reads to reference and building new extended reference from mapping result

novel mitochondrial genome

Hahn *et al.*, 2013. *Nucleic Acids Research* 41(13):e129

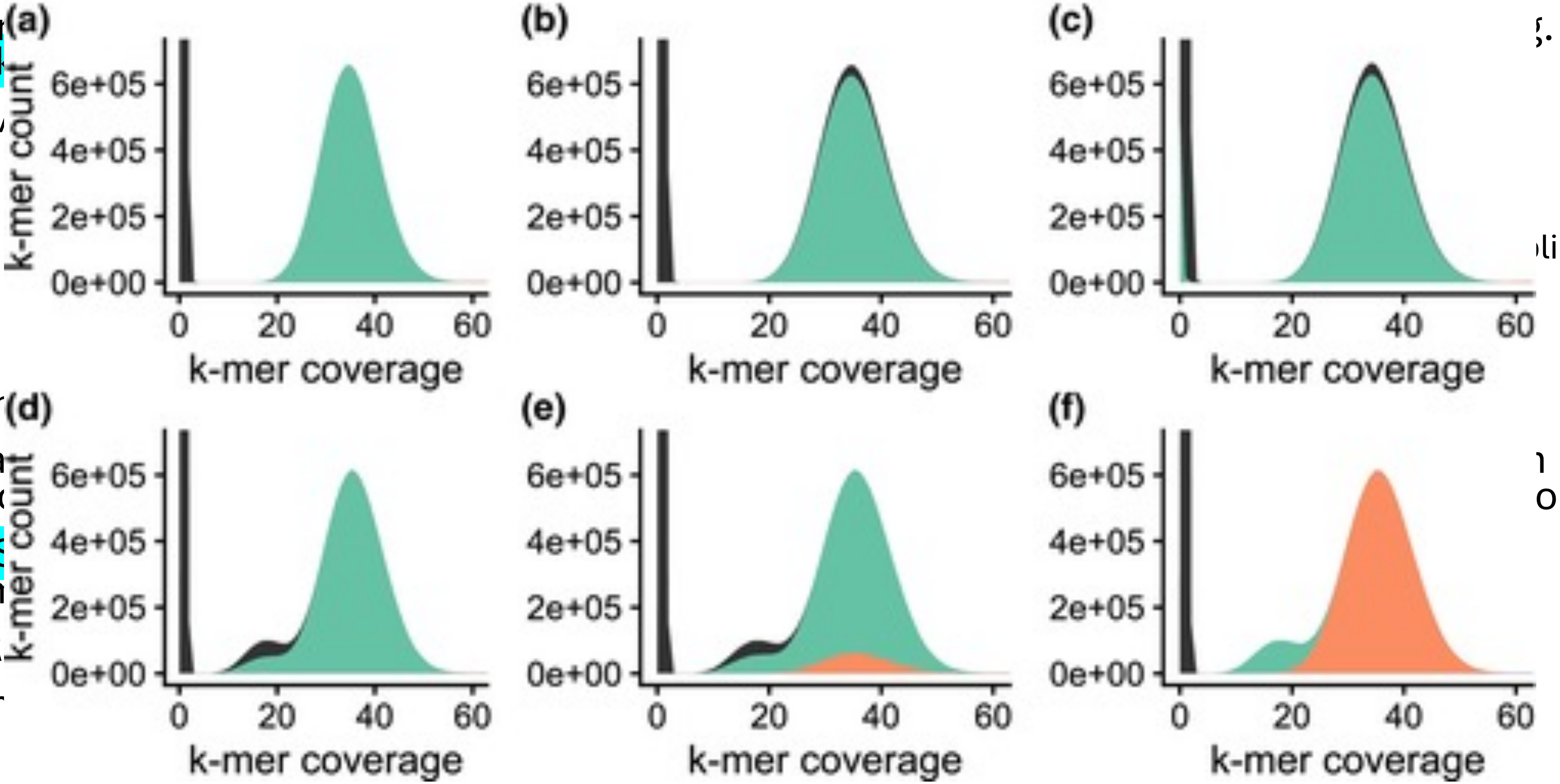
Quality-Checking Assemblies

- A priori: trimming adapters, size-selection, multiqc/fastqc, exploring contamination (e.g. kraken2 or fastq screen).
- Coverage – Percentages of bases covered by the assembled contigs. **LN/G**
  - Extent - % of the genome covered.
  - Depth – % of a particular region is "covered"
  - "More" is not always better – too much data can swamp assembler. If so, consider randomly sampling reads.
  - Coverage formula: https://www.illumina.com/documents/products/technotes/technote_coverage_calculation.pdf
- Number of scaffolds, lengths, ungapped lengths
- Scaffold/contig N50 (Mb) – The scaffold/contig length such that 50% of the genome lie in blocks of this size or larger. N80, N60, etc. are for 80%, 60%, etc. PacBio says >1 Mb is good.
- BUSCO (Benchmarking Universal Single-Copy Ortholog) Scores – Measures genome completeness in terms of **gene content.** Want >95% according to PacBio.
- LTR Assembly Index measures completeness of repetitive content of assembly.
- K-mer coverage:

Simao *et al.*, 2015. *Bioinformatics* 31(19):3210-3212; Ou *et al.*, 2018; *Nucleic Acids Research* 46(21):e126; for coverage see Lander & Waterman, 1988. *Genomics* 2(3):231-239; Whibley *et al.*, 2020. *Molec Ecol Resour* 21(3):641-652.

Quality-Checking Assemblies

- A pr … g.
  krak
- Cov
  -
  -
  - … ling
  -
- Nur
- Sca … n
  bloc … ood.
- BUS
  con
- LTR
- K-m



Simao *et al.*, 2015. *Bioinformatics* 31(19):3210-3212; Ou *et al.*, 2018; *Nucleic Acids Research* 46(21):e126; for coverage see Lander & Waterman, 1988. *Genomics* 2(3):231-239; Whibley *et al.*, 2020. *Molec Ecol Resour* 21(3):641-652.

Helpful cheatsheets by Illumina for short read assembly & PacBio for Long Read:

- https://www.illumina.com/Documents/products/technotes/technote_denovo_assembly_ecoli.pdf
- https://www.pacb.com/blog/beyond-contiguity/#:~:text=The%20aim%20is%20to%20have,score%20above%2095%25%20considered%20good.&text=Correctness%2C%20the%20third%20and%20final,is%20more%20challenging%20to%20measure.)

Problems with Assemblies

- Complexity
  - Lots of data, and lots of steps, makes it take a long time or require lots of memory to assemble.
- Ambiguity
  - We don’t “know” the answer ahead of time, so we can’t check our work.
- Repeats
  - See next slide.

Repeats cause an issue for assembly that can partly be addressed by new chemistry.

Original: ATACTTTGGGGAAAAAAATACGATATATATATATATATATATATATATATATATCGGCTGAC

7 bp read pileup: ATACTTT CGATATA TATATAC TTTGGGG AAAAAAA TACAATA CGATATA
ATCGGCT GACTATA TCGGCTG ACGGGGA AAAAAAT AGGCTGG GAAAATA
TCGGCTC GATATAT ATATATA TATATAA TACGATA TATCGGC ATATATA TCGGCTG

Longer read pileup: TACGATATATATATATATATATATATATATATCGG
ATACTTTGGGGAAAAAAATACGATATATATATA
TATATATATATATATATATATATATATATCGGCTGAC
etc.

Assembly Programs

Data type --> Programs.

<u>Short read</u>

Plastid → MITObim , GetOrganelle

Genome → ALLPATHS-LG, SOAPdenovo, SparseAssembler, SGA, MaSuRCA, Meraculous, JR-Assembler, Velvet, SPAdes, ABySS

<u>Long read</u>

Genome → Canu, wtdbg2

Polishing → Arrow, Nanopolish Pilon, Racon

<u>Transcriptome</u>

Transcriptome → Trinity, SOAPdenovo-Trans, Oases, Trans-ABySS, IDBA-Tran, inGAP-CDG, RNA-Bloom, rnaSPAdes

Amarasinghe *et al.*, 2020. *Genome Biology* 21(30)

Some examples of assembly papers here at the museum, in our own classroom!

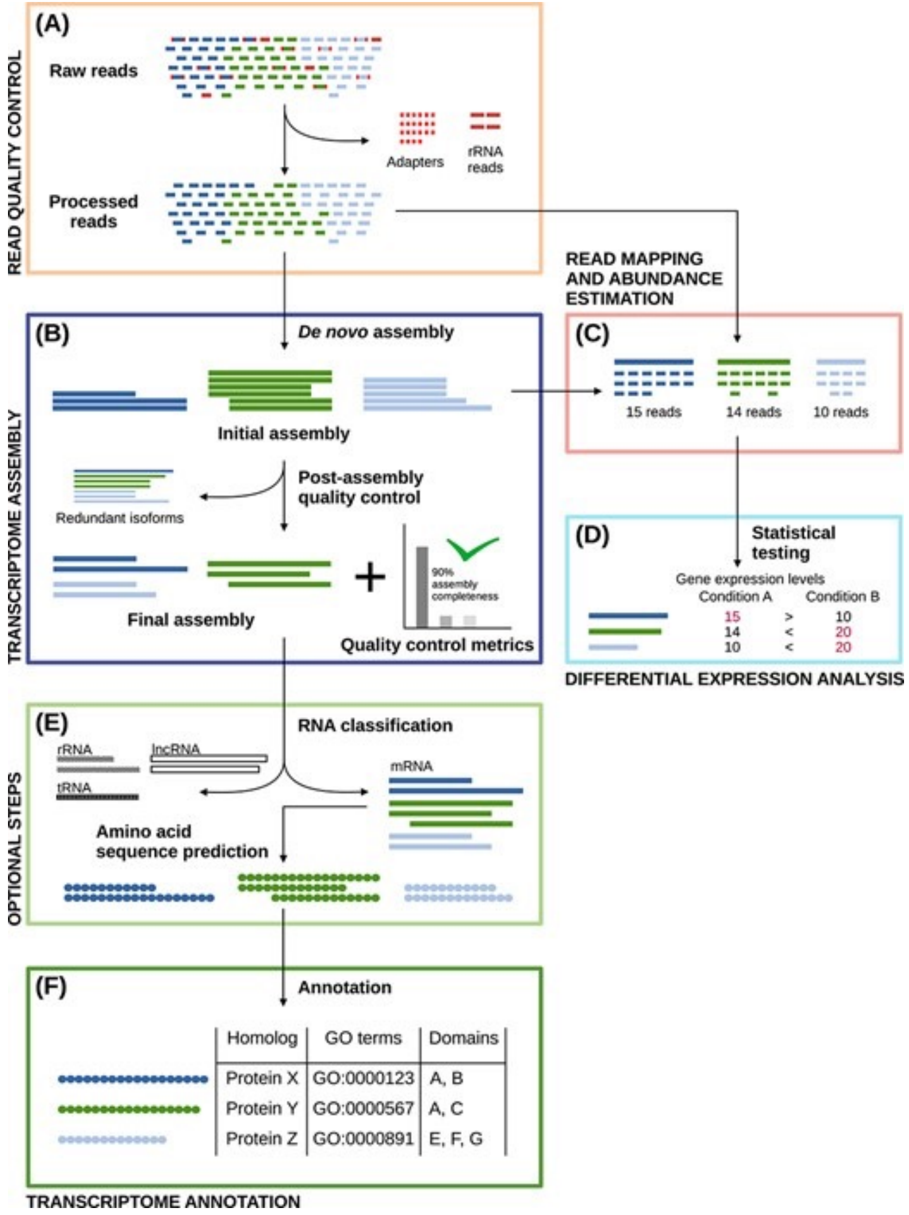
- Amanda has a great paper on long read assembly of a luna moth: Markee *et al.*, 2024. *Genome Biol. Evol.* 16(7):1-8
- Lina has a great paper on assembly from pair-end, 150 bp shotgun sequence reads: Raubold *et al.*, 2024. *Biodiversity Genomes:* https://doi.org/10.56179/001c.118546.
- Violet has a great paper on long read assembly of hyena genomes: Shao *et al.*, 2022. *Mol. Biol. Evol.* 39(3):msac011.

Key Players in Assembly

- Human Genome Project (1990-2003): https://www.genome.gov/human-genome-project
- Vertebrate Genomes Project (~2017-present): https://vertebrategenomesproject.org
- Genome Reference Consortium (~2007-present): https://www.ncbi.nlm.nih.gov/grc

RNA VS DNA

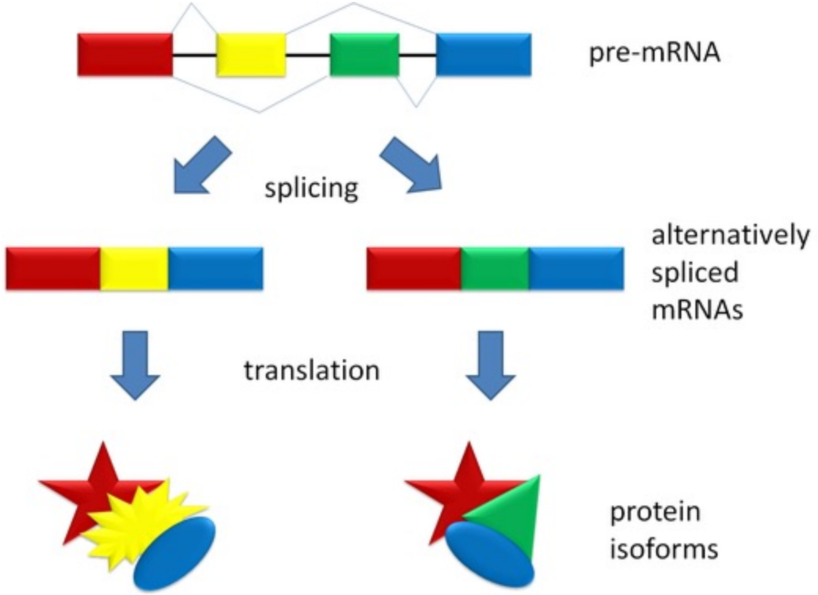
- RNA-seq technology
- Note: You may need to deal with different contaminant filtering.
  - rRNAs (pre-assembly), non-coding RNAs (post-assembly)
- Normalization (*in silico*) to handle disproportionate read abundance:
  - Trinity, khmer, Bignorm, NeatFreq, ORNA



Raghavan *et al.*, 2022. *Briefings in Bioinformatics* 23(3):bbab563.

Functional Annotation & Other QC

- De novo: Augustus, Genscan, SNAP, GlimmerHMM. Next 2 weeks!
- Consider: ExN50 because of transcript isoforms; SeqKit, BUSCO >80%, DOGMA, mapping.



Raghavan *et al.*, 2022. *Briefings in Bioinformatics* 23(3):bbab563. Image: Rosalind.info.

RNA Programs: Take a peek at this spreadsheet!

Table S2: Generally a very comprehensive list of programs you can use from a review on *de novo* transcriptome assembly and annotation.

Raghavan *et al.*, 2022. *Briefings in Bioinformatics* 23(3):bbab563.

Assembly Vocabulary Cheat-Sheet

Assembly: The set of sequences used to represent an organism's genome.

Graph: In math, a structure used to model pairwise relationships between objects, comprising vertices/nodes/points connected by edges/arcs/links/lines.

Consensus: A summary of your reads. Majority rule, for example, is taking the most common sequence between reads which are aligned together. There are other threshholds like 90%, 70%, etc.

K-mer: A string extracted from reads with specified length K.

Time complexity (big O): A theoretical estimate of how long a program should take to run as the amount of data or other relevant parameters change. For example, O(n) means that there's a linear relationship between how many samples (n) you have and how long it takes for the program to run.

Iterative mapping: Starting with mapping reads to a reference-guided scaffold and then iteratively re-assembling based on variant calls/consensuses of your short reads, rather than starting entirely from the raw sequence data.

Reference-guided: assembly which is based on the genome of a close relative as a "model"

de novo: assembly entirely based on your sequence data alone, independent of "models."

Overlap layout consensus: A shared overlap and graphically-based assembly algorithm which is inefficient for lots of reads but better at handling repeated elements.

De Bruijn graph: Short reads are split into k-mers before graphs are built. Efficient for lots of reads but worse at handling repeat elements.

Hamiltonian – The k-mers are nodes, connected by overlapping prefix and suffix (k-1)-mers.

Eulerian - The k-mers are edges, connected by overlapping prefix and suffix (k-1)-mers.

Coverage: Percentages of bases covered by the assembled contigs. **LN/G**

Exercises (Optional)

Easy: Answer the questions in Slide 2 in your own words.

Medium: Look up 2-3 assembly programs which are useful for your own data and compare what types of questions they are better at answering, what data types they are compatible with, and how long it takes for them to run. Explain in your own words how they work.

Hard: Take your short reads or download data from NCBI (whichever is easiest).

A.) Prepare them as is necessary to run for a program of your choice (do not forget to trim adapters – module TrimGalore may help!). You can perform other quality-control such as removing ambiguous scores (q-score filter) or size filters (length of read > 30), such as using `fastp`.

B) Run an assembly program with them on Huxley. If you are having issues, let us know.

Tips: use `module avail` to see if a module for the assembly software exists. If it does, run `module run XXXX`. Otherwise, try installing it yourself on Huxley or email Sajesh or us for help in getting it set up on the server. Also check to see whether a conda or other similar package managing environment exists for these programs.

Hard: Go through some of the tutorials on this website: https://www.langmead-lab.org/teaching.html