Outline: Jan 26

- Continue: overlap graph assembly
- Begin: de Bruijn graph assembly

Notes:
- Lab 1 due Wednesday
- Assembly reading posted: spend 1.5 hours max
Steps of Overlap Graph Assembly (also called “overlap-layout-consensus”)

1) Compute **overlaps between all pairs of reads**. With \( n = \) number of reads and \( L = \) length of reads, this is naively \( O(n^2L^2) \). We will learn better ways of “aligning” sequences next week.

2) Construct a **graph with reads as the nodes** and **directed, weighted edges** between reads with \( \geq T \) overlap.
Activity example: $L = 10, T = 5$

ATATATA\textcolor{red}{CTGGCGTATCGCAGTAAACGCGCCG}

R1: ACTGGCGTAT
R2: TGGCGTATCG
R3: GGGCGTATCGC
R4: CGTGATCGCAG
R5: TATCGCAGTA
R6: CGCAGTAAAC

Li et al, Briefings in Functional Genomics (2012)
Issues with overlap graphs
Bubbles

sequencing error

Velvet Assembler, Wikipedia
Repeats

read 1:   TAACTGTTCGCATCATCATCAT
read 2:   CATCATCATCATCATCATCATCATGCT
Repeats

read 1:   TAACTGTTCGCATCATCATCATCATCAT
read 2:   CATCATCATCATCATCATCATCATGATGCT

TAAC\textcolor{red}{T}GTTCGCATCATCATCATCATCATCATCATCATGATGCT

2 CATs
Repeats

read 1:   TAACTGTTCGCATCATCATCATCAT
read 2:   CATCATCATCATCATCATCATGCATGCT

TAACGTGTTCGCATCATCATCAT
CATCATCATCATCATCATCATGCATGCT

3 CATs
Repeats

read 1: TAACTGTTTCGCATCATCATCATCATCAT
read 2: CATCATCATCATCATCATCATGCATGCT

TAACTGTTTCGCATCATCATCATCATCATCATGCATGCT

4 CATs
What would the graph look like for these reads?
What would the graph look like for these reads?

Back to overlap graph algorithm
Steps of Overlap Graph Assembly
(also called “overlap-layout-consensus”)

1) Compute **overlaps between all pairs of reads**. With \( n = \) number of reads and \( L = \) length of reads, this is naively \( O(n^2L^2) \). We will learn better ways of “aligning” sequences next week.

2) Construct a **graph with reads as the nodes** and directed, weighted edges between reads with \( \geq T \) overlap.

3) “Layout” the graph and try to “group” stretches of the graph into “**contigs**” (short for contiguous), these are (hopefully) long portions of the original genome.

4) Find a “consensus” **sequence** for each contig.
Activity example: \( L = 10, T = 5 \)

ATATAT\[ACTGGCGTATCGCAGTAAAC\]CGCCG

R1: ACTGGCGTAT
R2: TGGCGTATCG
R3: GGCATATCGC
R4: CGTATCGCAG
R5: TATCGCAGTA
R6: CGCAGTAAAC

Li et al, Briefings in Functional Genomics (2012)
Activity example: $L = 10, \ T = 5$

$\text{ATATATACGTCGTCGCAACTAG}$

R1: ACTGGCGTAT
R2: TGGCGTATCG
R3: GGCATATCGC
R4: CGTATCGCAG
R5: TATCGCAGTA
R6: CGCAGTAAAC

First simplification: remove edges that can be (transitively) inferred from other edges

Li et al, Briefings in Functional Genomics (2012)
Overlap graph is big and messy. Contigs don’t “pop out” at us.

Below: part of the overlap graph for

to_every_thing_turn_turn_turn_turn_there_is_a_season

$L = 7, T = 4$
In this example: green edges can be inferred from blue
Layout: remove transitivity-inferrable edges

Before:
Layout: remove transitively-inferrable edges

After removing edges that skip one node

Slide: adapted from Ben Langmead, John Hopkins
After removing edges that skip one or two nodes
Emit *contigs* corresponding to the non-branching stretches

Contig 1

to_every_thing_turn_

Contig 2

turn_there_is_a_season

Unresolvable repeat
In practice, layout step also has to deal with spurious subgraphs, e.g. because of sequencing error.

Mismatch could be due to sequencing error or repeat. Since the path through b ends abruptly we might conclude it’s an error and prune b.
Consensus

```
TAGATTACACAGATTACTGA TTGATGGCGTAA CTA
TAGATTACACAGATTACTGACTTGATGCGTAAACTA
TAG TTACACAGATTATGGACTTCCATGGCGTAA CTA
TAGATTACACAGATTACTGACTTGGATGCGTAA CTA
TAGATTACACAGATTACTGACTTGGATGCGTAA CTA
```

Take reads that make up a contig and line them up

```
TAGATTACACAGATTACTGACTTGGATGCGTAA CTA
```

Take consensus, i.e. majority vote
Issues with overlap graph assembly

- Next-generation sequencing produces 100’s of millions (or even billions) of reads
- With one node per read this is computationally intractable for large genomes
- What if the nodes in our graph were not reads?
De Bruijn Graph (DBG) Assembly
DBG: de Bruijn Graph assembly

$k$-mer: substring of length $k$

$S: \text{GCGGATT} \overline{\text{CATCG}}$

$4$-mer: $\overline{\text{ATT}}$

all $3$-mers:

$\text{GCG}$
$\text{GCG}$
$\text{GCA}$
$\text{GAT}$

...$
\text{TCG}$

$\# \text{ total } k$-mers:
Examples: Ben Langmead, John Hopkins
de Bruijn graph: directed multigraph.

\[ V = \{ a, b, c \} \]
\[ E = \{ (a, b), (a, b), (a, c), (c, b) \} \]
indegree: # incoming edges
outdegree: # outgoing edges

Node is balanced if: indegree = outdegree

semi-balanced: \(|\text{indegree} - \text{outdegree}| = 1\)

Goal: visit each edge exactly once

Eulerian path
Theorem: There exists an Eulerian path if and only if all nodes are balanced except for 2 that are semi-balanced.

\[ k = 3, \text{ what is the } \text{de Bruijn graph.} \]