The Idury-Waterman Algorithm
A Brief Overview
Introduction

The Input

The Sequence Graph

Sequence Graph Reduction

Compensating for Errors
- Singletons
- Forks
- Crosses

Summary
What is the Idury-Waterman Algorithm?

- An algorithm for DNA sequence assembly
- Decomposes reads into their spectrum of k-mers
- Creates a de Bruijn graph out of the k-mers
- Performs a modified Euler traversal on the graph to infer the assembly
DNA sequence $L$ bases long

Sequence of one strand $a = a_1, a_2, \ldots, a_L$

Generated at random are $N$ shorter fragments $f_1, f_2, \ldots, f_N$ that we assume are uniformly sampled

- $N << L$
- The sequence of $f_i$ can be determined experimentally, $i = 1, 2, \ldots, N$
Complications

- The sequence of $f_i$ is only approximately known
- The orientation of $f_i$ is unknown
- Exact or approximate repetitive regions
- Long repeats are not captured by a single fragment
To compensate, we

- Introduce the reverse compliment of each read
- Redundantly sample \( a \) at an average depth \( c = \frac{NI}{L} \)
  - A parameter known as coverage
  - Let \( X_t \) be the depth of coverage at base \( a_t \)

\[
\mathbb{P}(X_t = k) = \frac{e^{-c}c^k}{k!}
\]
The algorithm takes as input a set of fragments $f_1, \cdots, f_N$ and their reverse complement, with an average coverage of $c$. 
The spectrum of $a$, denoted $S(a)$, is the set of $k$-tuples $w$ such that

$$S(a) = \{w : w = a_ia_{i+1} \cdots a_{i+k-1}, 1 \leq i \leq L - k + 1\}$$

e.g. The spectrum of 4-mers of ACTGACTG is 
\{ACTG, CTGA, TGAC, GACT\}
The Sequence Graph

Define a graph on the spectrum of $a$ as follows:

- $G = (V, E)$
- $V$ = the vertex set, the set of $(k-1)$-tuples from the spectrum
- $E$ = the edge set of $k$-tuples
- Draw an edge from $u$ to $v$ if the suffix of $u$ is the prefix of $v$, $u_{2:k} = v_{1:k-1}$
- Label each edge with the fragment and position of the fragment it came from

When $k$-tuples do not repeat, we can reconstruct the sequence with a straightforward traversal of the graph.
We can deal with two types of repetitive $k$-tuples:

- $k$-tuples that occur at multiple positions within a single fragment:
  - The fragment and position information allows us to distinguish what paths to take first.

- $k$-tuples that repeat because they correspond to a repetitive region in different fragments:
  - The corresponding edge contains significantly more fragment positions than its nonrepetitive neighbors.
A Sequence Graph

The Sequence Graph for ATGTGCCGCA

- AT
  - $ATG_{1,1}$
  - $TGT_{1,2}$

- TG
  - $GT_{1,1}$
  - $GTG_{1,3}$

- CC
  - $GCC_{1,5}$
  - $CCG_{1,6}$
  - $CGC_{1,7}$

- CG
  - $GCA_{1,8}$

- GA
  - $GCA_{1,8}$
Outline

1. Introduction
2. The Input
3. The Sequence Graph
4. Sequence Graph Reduction
   - Compensating for Errors
   - Singletons
   - Forks
   - Crosses
Sequencing errors create spurious edges in the graph. To deal with these, we first assume the following:

- Errors are uniformly distributed over all fragments, and over the entire length of each fragment
- The error rate is small
- For any position $i$ of $a$, the number of fragments covering the region $i \cdots i + k - 2$ is a Poisson random variable
- There are no repeats of length $k$ or greater in $a$
- The only sequencing errors are substitution errors
Reducing the Graph

Under these assumptions,

- Sequencing errors are rare, and create edges that split off of the correct branch of the tour.
- Tuples containing an error only appear once
- We can recognize and get rid of these errors by reducing the graph
- The reduced graph will be easier to perform an Euler tour on
Outline

1. Introduction
2. The Input
3. The Sequence Graph
4. Sequence Graph Reduction
   - Compensating for Errors
   - Singletons
   - Forks
   - Crosses
Singletons

- Nodes of indegree 1 and outdegree 1
- Compose over 90% of the sequence graph
Collapsing Singletons

The collapse of a chain of singletons into a superedge.
Why Collapse Singletons?

Collapsing singletons offers two main advantages:

- It reduces the size of the graph almost 10-fold
- Super edges can be included in the tour with very high confidence.
Outline

1. Introduction
2. The Input
3. The Sequence Graph
4. Sequence Graph Reduction
   - Compensating for Errors
   - Singletons
   - Forks
   - Crosses
Nodes of indegree 1 and outdegree > 1
Comprise over 70% of nonsingleton nodes
Eliminating Forks

- If we placed $E_1$ on the tour, and it does not correspond to a repetitive region, then we can only leave through one edge.
- Only one of them can be the correct edge, so the other(s) must be extraneous.
- Favor the heaviest super edge for the next candidate edge.
Outline

1. Introduction
2. The Input
3. The Sequence Graph
4. Sequence Graph Reduction
   - Compensating for Errors
   - Singletons
   - Forks
   - Crosses

Compensating for Errors
Singletons
Forks
Crosses
Crosses are nodes of indegree > 1 and outdegree > 1
If we determine that $E_1$ naturally continues onto $E_2$ through fragment/position data or other means, we can merge them.

This reduces the indegree and outdegree of the cross by 1, and does not affect tour through the node.
The input is a set of reads sampled with coverage $c$ and their reverse complement.

The spectrum graph of the reads is found and reduced.

A modified Euler tour infers the assembled sequence.