Outline

1. Assembly by Analogy

2. Genome Assembly
   1. Coverage, read length, repeats, and errors
   2. Genome assemblers & Assemblathon

3. Whole genome alignment
Shredded Book Reconstruction

- Dickens accidentally shreds the first printing of *A Tale of Two Cities*
  - Text printed on 5 long spools

```
It was the best of times, it was the worst of times, it was the age of wisdom, it was the age of foolishness, ...

It was the best of times, it was the worst of times, it was the age of wisdom, it was the age of foolishness, ...

It was the best of times, it was the worst of times, it was the age of wisdom, it was the age of foolishness, ...

It was the best of times, it was the worst of times, it was the age of wisdom, it was the age of foolishness, ...

It was the best of times, it was the worst of times, it was the age of wisdom, it was the age of foolishness, ...
```

- How can he reconstruct the text?
  - 5 copies × 138,656 words / 5 words per fragment = 138k fragments
  - The short fragments from every copy are mixed together
  - Some fragments are identical
**Greedy Reconstruction**

It was the best of times, it was the worst of times, it was the age of wisdom, it was the age of foolishness, it was the best of times, it was the worst of times, it was the age of wisdom, it was the age of foolishness.

The repeated sequence make the correct reconstruction ambiguous

- It was the best of times, it was the [worst/age]

Model the assembly problem as a graph problem.
de Bruijn Graph Construction

- $D_k = (V,E)$
  - $V =$ All length-$k$ subfragments ($k < l$)
  - $E =$ Directed edges between consecutive subfragments
    - Nodes overlap by $k-1$ words

- Locally constructed graph reveals the global sequence structure
  - Overlaps between sequences implicitly computed

Original Fragment  Directed Edge
It was the best of  It was the best  was the best of

- de Bruijn, 1946
- Idury and Waterman, 1995
- Pevzner, Tang, Waterman, 2001
It was the best of times, it was the age of wisdom, it was the worst of times, it was the age of foolishness.

After graph construction, try to simplify the graph as much as possible.
de Bruijn Graph Assembly

It was the best of times, it was the worst of times, it was the age of wisdom, it was the age of foolishness.

After graph construction, try to simplify the graph as much as possible.
Assembly Applications

• Novel genomes

• Metagenomes

• Sequencing assays
  – Structural variations
  – Transcript assembly
  – …

Like Dickens, we must computationally reconstruct a genome from short fragments
Ingredients for a good assembly

**Coverage**

*High coverage is required*
- Oversample the genome to ensure every base is sequenced with long overlaps between reads
- Biased coverage will also fragment assembly

**Read Length**

*Reads & mates must be longer than the repeats*
- Short reads will have false overlaps forming hairball assembly graphs
- With long enough reads, assemble entire chromosomes into contigs

**Quality**

*Errors obscure overlaps*
- Reads are assembled by finding kmers shared in pair of reads
- High error rate requires very short seeds, increasing complexity and forming assembly hairballs

---

Current challenges in de novo plant genome sequencing and assembly
Initial Contigs

• After constructing assembly graph, compress graph down to its non-branching initial contigs
  – Aka “unitigs”, “unipaths”
Repeats and Coverage Statistics

• If $n$ reads are a uniform random sample of the genome of length $G$, we expect $k=n \Delta / G$ reads to start in a region of length $\Delta$.
  – If we see many more reads than $k$ (if the arrival rate is $> A$) , it is likely to be a collapsed repeat
  – Requires an accurate genome size estimate

$$
\Pr(X - \text{copy}) = \binom{n}{k} \left( \frac{X \Delta}{G} \right)^k \left( \frac{G - X \Delta}{G} \right)^{n-k}
$$

$$
A(\Delta, k) = \ln \left( \frac{\Pr(1 - \text{copy})}{\Pr(2 - \text{copy})} \right) = \ln \left( \frac{(\Delta n / G)^k}{(2 \Delta n / G)^k} e^{-\Delta n / G} \right) = \frac{n \Delta}{G} - k \ln 2
$$
Scaffolding

- Initial contigs (aka unipaths, unitigs) terminate at
  - Coverage gaps: especially extreme GC regions
  - Conflicts: sequencing errors, repeat boundaries

- Iteratively resolve longest, ‘most unique’ contigs
  - Both overlap graph and de Bruijn assemblers initially collapse repeats into single copies
  - Uniqueness measured by a statistical test on coverage
**N50 size**

Def: 50% of the genome is in contigs larger than N50

Example: 1 Mbp genome

\[ \text{N50 size} = 30 \text{ kbp} \]
\[ (300k + 100k + 45k + 45k + 30k = 520k \geq 500kbp) \]

Note:
N50 values are only meaningful to compare when base genome size is the same in all cases
## Assembly Algorithms

<table>
<thead>
<tr>
<th>ALLPATHS-LG</th>
<th>SOAPdenovo</th>
<th>Celera Assembler</th>
</tr>
</thead>
<tbody>
<tr>
<td><img src="image1.png" alt="De Bruijn Graph" /></td>
<td><img src="image2.png" alt="De Bruijn Graph" /></td>
<td><img src="image3.png" alt="Overlap Graph" /></td>
</tr>
</tbody>
</table>
| **Broad’s assembler**  
(Gnerre et al. 2011) | **BGI’s assembler**  
(Li et al. 2010) | **JCVI’s assembler**  
(Miller et al. 2008) |
| De bruijn graph  
Short + PacBio (patching) | De bruijn graph  
Short reads | Overlap graph  
Medium + Long reads |
| Easy to run if you have compatible libraries | Most flexible, but requires a lot of tuning | Supports Illumina/454/PacBio  
Hybrid assemblies |

- http://www.broadinstitute.org/software/allpaths-lg/blog/
- http://wgs-assembler.sf.net
Attempt to answer the question: “What makes a good assembly?”

Organizers provided simulated sequence data
  - Simulated 100 base pair Illumina reads from simulated diploid organism

41 submissions from 17 groups

Results demonstrate trade-offs assemblers must make
Assembly Results
Final Rankings

<table>
<thead>
<tr>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>BGI</td>
<td>36</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Broad</td>
<td>37</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>WTSI-S</td>
<td>46</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>CSHL</td>
<td>52</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>BCCGSC</td>
<td>53</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>DOEJGI</td>
<td>56</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>RHUL</td>
<td>58</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>WTSI-P</td>
<td>64</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>EBI</td>
<td>64</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>CRACS</td>
<td>64</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

- SOAPdenovo and ALLPATHS came out neck-and-neck followed closely behind by SGA, Celera Assembler, ABySS
- My recommendation for “typical” short read assembly is to use ALLPATHS
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
Whole Genome Alignment with MUMmer

Slides Courtesy of Adam M. Phillippy
amp@umics.umd.edu
WGA visualization

• How can we visualize *whole* genome alignments?

• With an alignment dot plot
  – \( N \times M \) 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
Seed and Extend

How can quickly find large alignments?

1. Find short exact matches
   - using a suffix tree
2. Cluster exact matches
   - using size, gap and distance parameters
3. Extend clusters & report alignments
   - using modified Smith-Waterman algorithm
WGA example with nucmer

*Yersina pestis CO92 vs. Yersina pestis KIM*

- High nucleotide similarity, 99.86%
  - Two strains of the same species
- Extensive genome shuffling and highly repetitive
  - Global alignment will not work

```
nucmer  -maxmatch  CO92.fasta  KIM.fasta
-maxmatch     Find maximal exact matches (MEMs)
```

```
delta-filter  -l  out.delta  >  out.filter.m
-n            Many-to-many mapping
```

```
show-coords  -r  out.delta.m  >  out.coords
-r            Sort alignments by reference position
```

```
mummerplot  --large  --layout  out.delta.m
--large      Large plot
--layout     Nice layout for multi-fasta files
```
http://mummer.sourceforge.net
Thank You

http://schatzlab.cshl.edu
@mike_schatz / #BTG2012