GENOME ASSEMBLY
(BACKGROUND AND STRATEGY)

TEAM-3 : GROUP-1
Will A Hutwagner
Mani Jain
Prerna Jain
Yini Yang
Sinjini Gupta
Sreenivasan AC
Contents:

1. Overview of the whole Project
2. Introduction
3. Quality Analysis and Control
4. De Novo Genome Assembly Algorithms
5. De Novo Assembly Softwares
6. Post Assembly Quality Analysis
7. Genome Assembly Pipeline
8. Work Delegation
Overview of whole project:

**AIM**: To identify the source/origin of the given foodborne illness.

We are given the sequence reads and their quality scores. The procedure to be followed is:

- **Genome Assembly**
- Gene Prediction
- Functional Annotation
- Comparative Genomics
- Predictive Webserver
Contents:

1. Overview of the whole Project
2. Introduction
3. Where we are
4. Quality Analysis and Control
5. De Novo Genome Assembly Algorithms
6. De Novo Assembly Softwares
7. Post Assembly Quality Analysis
8. Genome Assembly Pipeline
9. Work Delegation
What is Genome Assembly?

DEFINITION:

Genome assembly refers to the process of taking a large number of short DNA sequences and putting them back together to create a representation of the original chromosomes from which the DNA originated.

Fig 2.1  Representation of Genome Assembly

Image source: https://www.biostars.org/p/253222/
How to Assemble a Genome?

Genome Assembly is conducted through 5 main steps of the procedure:

➔ SEQUENCING
➔ QUALITY ANALYSIS
➔ QUALITY CONTROL
➔ GENOME ASSEMBLY
➔ GENOME ASSEMBLY METRICS
Sequencing

Genome sequencing is figuring out the order of DNA nucleotides, or bases, in a genome—the order of As, Cs, Gs, and Ts that make up an organism's DNA.

Fig 2.2  Digital representation of Genome

ILLUMINA Sequencing: Possible errors

- ILLUMINA produces very short reads (50-300 bp), which limits the capability to resolve complex regions with repetitive sequences. It also makes other computations like sequencing entire RNA transcripts difficult or impossible.
- Overall error rates are very low, but the errors are not evenly distributed across positions and nucleotides, creating a notable bias.
Contents:

1. Overview of the whole Project
2. Introduction
3. Quality Analysis and Control
4. De Novo Genome Assembly Algorithms
5. De Novo Assembly Softwares
6. Post Assembly Quality Analysis
7. Genome Assembly Pipeline
8. Work Delegation
Quality Evaluation:

It's very common to have some reads within a sample, or some positions across all reads that are low quality and should be discarded.

Pre-Processing:
- Trim reads
- Exclude low quality reads
- Contaminations

FastQC: quality control checks of raw reads
Trimmomatic:

WHAT IS TRIMMOMATIC?

- All in one pre-assembly tool
- Trim
- Technical sequences removal
- Quality filtering

WHY DO WE CHOOSE IT?

- Pair-aware, optimized for Illumina NGS data;
- Ability to choose a set of processing steps in a user-defined order;
- Steps included could be repeated
- Compatible with other tools using shell pipeline;
- Fast, flexible and efficient.
FastQC:

Before Trim

After Trim
Contents:

1. Overview of the whole Project
2. Introduction
3. Quality Analysis and Control
4. **De Novo Genome Assembly Algorithms**
5. De Novo Assembly Softwares
6. Post Assembly Quality Analysis
7. Genome Assembly Pipeline
8. Work Delegation
The Genome Assembly Problem

1. Reference Based
2. De Novo
   A. Greedy
   B. Graph Based Approaches
      i. Overlap, Layout, Consensus
      ii. De Bruijn Graphs
1. Hybrid

Fig 4.1 Steps in De Novo Genome Assembly

Image source: J Commins et al
Overlap, Layout, Consensus Algorithm

- Construct overlap graph directly from reads
- Each read is a vertex - edge is added if they overlap
- Graph is arranged in appropriate layout of reads.
- **Consensus** from joining read sequences and merging overlaps
- Shortest common superstring problem - NP complete, not feasible for large no. of reads

*Fig 4.2 Genome assembly from reads OLC algorithm*
*Image source: J Commins et al*
De Bruijn Graphs

- Each node in the graph = kmer
- Connect if there is a k-1 overlap
- Eulerian walk - you can visit each edge of the graph exactly once
- Typically ‘cleaned’ before analysis (remove portions which have low coverage)

An Eulerian path approach to DNA fragment assembly

Pavel A. Pevzner*, Haixu Tang†, and Michael S. Waterman‡**

*Department of Computer Science and Engineering, University of California, San Diego, La Jolla, CA; and Departments of †Mathematics and ‡Biological Sciences, University of Southern California, Los Angeles, CA

Contributed by Michael S. Waterman, June 7, 2001

(a) aacccgg ccggtt

(b) aacc – accg – cccg – cggg – ccgt – ggtt

(c)

Fig 4.3 Assembly of reads using DBG

Image source: Miller et al
An Example

AAABBBBA

take all 3-mers: AAA, AAB, ABB, BBB, BBA

form L/R 2-mers: AA, AA, AA, AB, AB, BB, BB, BB, BB, BA

L R L R L R L R L R

Let 2-mers be nodes in a new graph. Draw a directed edge from each left 2-mer to corresponding right 2-mer:

Each edge in this graph corresponds to a length-3 input string

Image source: Ben Langmead
Contents:

1. Overview of the whole Project
2. Introduction
3. Quality Analysis and Control
4. De Novo Genome Assembly Algorithms
5. **De Novo Assembly Softwares**
6. Post Assembly Quality Analysis
7. Genome Assembly Pipeline
8. Work Delegation
Genome Assembly Softwares:

- ABySS
- SKESA
- Unicycler (SPAdes)
- IDBA-UD

Table: 1

<table>
<thead>
<tr>
<th>Assemble</th>
<th>Average N50</th>
<th>Mismatches per 100kb</th>
</tr>
</thead>
<tbody>
<tr>
<td>SKESA</td>
<td>195141</td>
<td>0.40</td>
</tr>
<tr>
<td>SPAdes</td>
<td>131823</td>
<td>3.21</td>
</tr>
<tr>
<td>AByss</td>
<td>125993</td>
<td>0.98</td>
</tr>
<tr>
<td>IDBA</td>
<td>119363</td>
<td>5.17</td>
</tr>
</tbody>
</table>

Table 1 Source: 
http://cab.spbu.ru/software/spades/
**Abyss**

*ABySS: A parallel assembler for short read sequence data*

Jared T. Simpson, Kim Wong, Shaun D. Jackman, Jacqueline E. Schein, Steven J. M. Jones, and Iman Bizo

- *De novo*, parallel, paired-end sequence assembler
- Uses DeBruijn graphs for the assembly process
- Designed for short reads
- Single-k genome assembler
- Generates substrings of length k from the sequence reads and processes to remove read errors
- Uses mate-pair information to extend contigs by resolving ambiguities in overlaps

- Pros: produces short but correct scaffolds, parallelize the assembly, can cope with large genome assembly
- Cons: do not vary k-mer length, relatively slow
SKESA

- De-novo sequence read assembler for cultured single isolate genomes
- Uses DeBruijn graphs for the assembly process
- Multiple k-mer sizes are used
- Performs trimming of reads
- Designed to create breaks at repeat regions in the genome

- Pros: fast, multithreaded, produces reproducible assemblies
- Cons: works best only for haploid genomes, works best only on reads generated from illumina sequencing technology, no inbuilt scaffolding
SPAdes

- De-novo sequence read assembler for small genomes
- Employs multi-sized DeBruijn graphs for the assembly process
- Construction of assembly graph, k-bimer adjustment, paired assembly graph construction and contig construction
- Uses Bayes Hammer software for error correction

- Pros: works with Illumina or Ion Torrent read, capable of producing hybrid assemblies
- Cons: not suited for larger genomes
Unicycler

For short reads, Unicycler uses SPAdes to construct a DBG for a series of kmer sizes. It chooses assembly with minimum contigs and dead ends. It removes ‘contamination’ by removing low depth reads. It uses Bowtie2 and Pilon to polish assemblies. It has built-in scaffolding.

Pros: optimal layout, low mis-assembly rates, can cope with repetitive genomes, easy to use, scaffolding in-built

Cons: run time can be high
IDBA-UD

- Made for short reads with uneven sequence depths
- Iterates small k-mer to large k-mer (multi-k genome assemblers)
- Short and low depth contigs are removed from each iteration
- PE reads are aligned to contigs and assembled locally to fill some missing k-mers in low-depth regions

Pros: to make fewer gaps and branches, forms long contigs in low and high-depth regions

Cons: Takes long time, had segmentation faults
## Comparing Assemblers

<table>
<thead>
<tr>
<th></th>
<th>Abyss</th>
<th>IDBA</th>
<th>Unicycler</th>
<th>Skesa</th>
</tr>
</thead>
<tbody>
<tr>
<td># contigs (&gt;= 0 bp)</td>
<td>22</td>
<td>1833</td>
<td>11</td>
<td>21</td>
</tr>
<tr>
<td># contigs (&gt;= 50000 bp)</td>
<td>8</td>
<td>11</td>
<td>8</td>
<td>8</td>
</tr>
<tr>
<td>Largest Contig</td>
<td>1458069</td>
<td>707078</td>
<td>1457249</td>
<td>1458574</td>
</tr>
<tr>
<td>Total length</td>
<td>2894398</td>
<td>2908733</td>
<td>2883340</td>
<td>2907548</td>
</tr>
<tr>
<td>Total length (&gt;= 50000 bp)</td>
<td>2771275</td>
<td>2729844</td>
<td>2800416</td>
<td>2802305</td>
</tr>
<tr>
<td>N50</td>
<td>1458069</td>
<td>376626</td>
<td>1457249</td>
<td>1458574</td>
</tr>
<tr>
<td>L50</td>
<td>1</td>
<td>3</td>
<td>1</td>
<td>1</td>
</tr>
<tr>
<td>GC (%)</td>
<td>37.9</td>
<td>37.89</td>
<td>37.88</td>
<td>37.89</td>
</tr>
</tbody>
</table>
Metrics

Low number contigs are better

Longer contig length is better

Higher N50 better

Lower L50 better

Lower mismatched reads
Quality analysis softwares:

QUAST

Quality Assessment Tool for Genome Assemblies

LAP

LAP (Log Average Probability) - de novo probabilistic measure of assembly quality
Analysis pipeline:

Data → FastQC and MultiQC → Trimmomatic → SKESA → Unicycler → Abyss → SPAdes → Draft Assembly

Draft Assembly → QUAST "LAP Merged Assembly"
Delegation: who does what...

- Pipeline development - equal contribution.
- Each person does the whole pipeline so that everyone gets experience
- Wiki Page - Sectioned student-wise
References


Questions?