De novo assembly of NGS data

Sandra Smit
WUR bioinformatics
De novo assembly. Why?
De novo assembly. Why?

Read lengths are much shorter than even the smallest genome.
De novo assembly. Why?

- No reference genome available
- Not all reads map to the reference

Platforms:
- 454
- Illumina
- SOLiD
- PacBio
Assemble this!
Find overlap
Layout of the reads
Layout of the reads

the
thequ
heq
qui
quic
quickb
uickb
ckbro
ckbrow
kbr
bro
ownfo
nfo
nfox
fox
oxjum
umps
umpso
sove
ver
erthel
rthela
thel
thelaz
helaz
lazydo
azy
azyd
ydo
ydog

thequickbrownfoxjumpsoverthelazydog
Overlap-layout-consensus approach

TACCTGGAACCTGAAAAGAAGGCTTACTGGAGCCGCTGGCAGTG

TGCGTGGGATCTCGGCGAAATTCTTTTGCCGCA

GTGATGGTATGCGCACCTTGCCTGGGATCTC

CCGCTGGCAGTGACGGAACGGCTGGCCATTATCTCGGTGGTAGGTGATGGGTATGCG
Overlap-layout-consensus approach

TACCTGGAACTGAAAGAAGGCTTACTGGAGCCGCTGGCAGTG

TGCOTGGGATCTCGGCGAATATTCTTTGCCGCA

GTGATGGTATGCGCACCTTGCOTGGGATCTC

CCGCTGGCAGTGACGGAACGGCTGGCCATTATCTCGGTGGTAGGTGATGGTATGCG

All versus all sequence similarity comparison
Overlap-layout-consensus approach

Alignment of sequences with high sequence identity
Overlap-layout-consensus approach
Greedy extension

Greedily join reads that are most similar to each other
Graph-based OLC

Reads = nodes

TACCTGGAACTGAAAGAAGGCTTACTGGAGCCGCTGGCAGTG

TGCGTGGGATCTCGGCGAAATTCTTTGCGGCA

GTGATGGGTATGCGCACCTTGCGTGTTGATCTC

CCGCTGGCAGTGACGGAACGGCTGGCCATTATCTCGGTGGTAGGTATGGTATGCG

1

2

3

4
Graph-based OLC

TACCTGGAACTGAAAGAAGGCTTACTGGAGCCGCTGGCAGTG

TGCGTGGGATCTCGGCGAAATTCTTTTGCCGCA

GTGATGGTATGCGCACCTTGCGTGGGATCTC

CCGCTGGCAGTGACGGAACGGCTGGCCATTATCTCGGTGGTAGGTGATGGTATGCG

Reads = nodes
Overlap = edges
Graph-based OLC

Find Hamiltonian path = path that visits each node exactly once
OLC disadvantages

- All versus all comparison is computationally expensive
- Data volumes keep growing
- Short-read technologies (less overlap)
- No efficient algorithm to find Hamiltonian path
The de Bruijn graph approach

• Solving the assembly problem with De Bruijn graphs
  – Nodes are $k$-mers
  – Edges are overlaps between $k$-mers on $k-1$ positions

• A $k$-mer is a (short) substring of a read
  – A read of $n$ bp consists of $(n - k + 1)$ $k$-mers
  – Example: a read of 75 bp consists of 45 (overlapping) 31-mers

TGACCAGTG
TGAC
GACC
ACCA
CCAG
....
Generating $k$-mers from reads
Graph-based assembly of $k$-mers

Find Eulerian path = path that visits every edge exactly once
K-mer extraction and graph creation happen simultaneously.

Error correction is an important step.
Using paired-end information

DNA molecule

shotgun reads

paired-end reads

scaffolds
Strategies & Assemblers

- **Solexa reads**
- **454 reads**
- **Solid+Solexa+Helicos**
- **454+Solexa**

### Years
- 2010
- 2009
- 2008
- 2007
- 2006
- 2005

### Algorithms
- **Greedy-extension**
- **OLC**
- **De Bruijn Graph**
- **Hybrid**
- **Other Strategies**

- **QSRA**
- **Forge**
- **Shorty**
- **CABOG**
- **Edena**
- **VCAKE**
- **SHARCGS**
- **SSAKE**
- **Newbler**
- **ALLPATHS-LG**
- **ALLPATHS**
- **SOAPdenovo**
- **ABySS**
- **Euler-LiSR**
- **Velvet**
- **Euler-SR**
- **Taipan**
- **Ray**
- **Contrail**
- **LOCAS**
- **SR-ASM**
- **Segcons**
- **MIRA3**
- **PCAP long-read assembler**

*Zhang et al. 2011*
Just push the button?

Assemble
Complicating factor: DNA structure

DNA is double stranded

DNA contains palindromes etc.

Palindrome

```
T T A G C A C G T G C T A A
A A T C G T G C A C G A T T
```

Mirror repeat

```
T T A G C A C C A C G A T T
A A T C G T G T C T A A
```
Complicating factor: repeats

Tomato has a large, repetitive, genome
- 200-250 Mb gene-rich euchromatin
- 600-700 Mb highly repetitive heterochromatin

- Multiple overlaps in overlap-layout-consensus approach
- Multiply connected nodes in graph-based approach
Complicating factor: sequencing errors

Technologies have different properties and error profiles

Sequence-dependent coverage bias

Nonuniform error rates

Kircher & Kelso, 2010
Data filtering is essential to remove:
- Sequencing errors
- Homopolymer stretches
- Low-quality bases

Use for example a k-mer table
- assumption: k-mer table represents the complete genome
Complicating factor: heterozygosity

- Most eukaryotes are diploid
  - i.e. have two matching sets of chromosomes
- Two alleles for each gene
- Alleles are the same: homozygous
- Alleles are different: heterozygous

- During assembly two copies of the chromosomes are compressed into single genomic sequence
Heterozygosity illustrated by k-mers

Assumptions
- the reads are randomly sampled from the genome
- the k-mer table represents the complete genome
$k$-mers – homozygous potato ($k = 31$)
Complicating factor: heterozygosity

• Heterozygous $k$-mers create “bubbles” in the graph
  – Every SNP and in/del creates a bubble

• Solutions
  – Pinching bubbles: loss of diversity
  – Breaking graph: loss of contiguity
Complicating factor: ploidy

- Ploidy is the number of sets of chromosomes in a biological cell
- Diploid: two sets of chromosomes
  - Human cells
- Polyploidy means more than two sets of chromosomes per nucleus
- Tetraploid: four sets of chromosomes
  - Common in plants
Assembly is not so easy
Assembly workflow

- NGS sequencing/data production
- Data cleaning/filtering/error correction
- *De novo* assembly into contigs and (super)scaffolds
- Assembly finishing/gap filling
- Assembly validation
Assembly workflow

1. **NGS sequencing/data production**
2. Data cleaning/filtering/error correction
3. *De novo* assembly into contigs and (super)scaffolds
4. Assembly finishing/gap filling
5. Assembly validation
Sequencing strategy

Sequencing strategy: which sequencing data is necessary for my project? How can we balance the benefits and the costs?

Technologies have different properties and error profiles

What is the contribution of different input data to an assembly?
Sequencing strategy

- remove matepairs
  - 20kb
  - 8kb
  - 3kb

- 454 shotgun + matepair

- add Sanger data
  + SCE
  + SCE + SSG

- remove coverage
  - 25%
  - 50%
Sequencing strategy

<table>
<thead>
<tr>
<th># scaffolds</th>
<th>average length</th>
<th>N50 index</th>
<th>N95 index</th>
<th>% N</th>
</tr>
</thead>
<tbody>
<tr>
<td>454 - 50%</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>454 - 25%</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>454 - 3kb</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>454 - 8kb</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>454 - 20kb</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>454</td>
<td>3,974</td>
<td>197 kb</td>
<td>68</td>
<td>7.4</td>
</tr>
<tr>
<td>454 + SCE</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>454 + SCE + SSG</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
Assembly workflow

1. NGS sequencing/data production
2. Data cleaning/filtering/error correction
3. *De novo* assembly into contigs and (super)scaffolds
4. Assembly finishing/gap filling
5. Assembly validation
Assembly workflow

NGS sequencing/data production

Data cleaning/filtering/error correction

*De novo* assembly into contigs and (super)scaffolds

Assembly finishing/gap filling

Assembly validation
Genome finishing

From initial draft to complete genome

• Closing gaps
  – IMAGE pipeline, Tsai et al. 2010
  – GapFiller, Boetzer and Pirovano 2012
• Establish order and orientation of contigs/scaffolds
  – E.g. FISH experiments
• Base error correction
  – k-mer correction
• Contamination removal

Chain et al., Science 2009
Genome Project Standards in a New Era of Sequencing
Gap closure

1. Align the paired end reads onto draft sequence
2. Local assembly of the aligned reads; new contigs are produced
3. Gaps are now shortened. Repeat the whole procedure in a few iterations
4. The gap is now closed

IMAGE pipeline, Tsai et al. 2010
Assembly workflow

1. NGS sequencing/data production
2. Data cleaning/filtering/error correction
3. De novo assembly into contigs and (super)scaffolds
4. Assembly finishing/gap filling
5. Assembly validation
Assembly quality

- Number and length of contigs/scaffolds
  - N50
- Number of gaps/Ns
- GC content
- Coverage and completeness
- Structural consistency
  - Paired-end data
  - Physical and genetic maps
- Error rate
  - Known sequences (BAC clones, ESTs, etc.)

50% of assembly
N50 length: 18 Kb
N50 index: 4
## Basic stats: exercise

- Calculate
  - N50 contig size and index
  - N90 contig size and index

<table>
<thead>
<tr>
<th>Contigs (length)</th>
<th>250</th>
<th>150</th>
<th>120</th>
<th>100</th>
<th>90</th>
<th>80</th>
<th>80</th>
<th>70</th>
<th>40</th>
<th>20</th>
</tr>
</thead>
</table>
Basic stats: exercise

- Total assembly size = 1000
- Sort from large to small
- N50 lookup value = 500
- N50 contig size = 120
- N50 contig index = 3
- N90 contig size = 70
- N90 contig index = 8

<table>
<thead>
<tr>
<th>Contigs (length)</th>
</tr>
</thead>
<tbody>
<tr>
<td>250</td>
</tr>
<tr>
<td>150</td>
</tr>
<tr>
<td><strong>120</strong></td>
</tr>
<tr>
<td>100</td>
</tr>
<tr>
<td>90</td>
</tr>
<tr>
<td>80</td>
</tr>
<tr>
<td>80</td>
</tr>
<tr>
<td><strong>70</strong></td>
</tr>
<tr>
<td>40</td>
</tr>
<tr>
<td>20</td>
</tr>
</tbody>
</table>

- 50% at 120
- 90% at 70
Summary

NGS sequencing/data production

Data cleaning/filtering/error correction

*De novo* assembly into contigs and (super)scaffolds

Assembly finishing/gap filling

Assembly validation
Assembly of large genomes using second-generation sequencing
Schatz et al.
Genome Res. 2010

Assembly algorithms for next-generation sequencing data
Miller et al.
Genomics 2010

High-throughput DNA sequencing -- concepts and limitations
Kircher and Kelso
Bioessays 2010
Acknowledgements

• Bioinformatics (WU)
• Applied bioinformatics (PRI)
• Bioinformatics ASG

The International Tomato Genome Sequencing Consortium

Potato Genome Sequencing Consortium