Improving Genome Assemblies without Sequencing

Michael Schatz

April 25, 2005
TIGR Bioinformatics Seminar
Assembly Pipeline Overview

1. Sequence shotgun reads
2. Call Bases  \textit{phred/TraceTuner/KB Base Caller}
3. Trim Reads  \textit{lucy}
4. Assemble  \textit{CA/TA/Arachne}
5. Electronic Finishing
   - Second generation base-caller  \textit{AutoEditor}
   - Automatic Gap Closure  \textit{AutoJoiner}
   - Research Techniques
Base-calling in the context of single chromatogram is hard…

but finding base-calling “mistakes” in a multiple alignment is easy.
Signal Parameters

Black dots on the signal curve indicate local maxima and open circles indicate local minima.

- **amplitude (a)**
- **support (b)**
- **minimum difference between amplitude and local minimum (c)**
AutoEditor Results

- Corrects 80% of all discrepant base-calls with an error rate better than 1/8800.

- Increase consensus quality, decrease finishing costs

- Remaining discrepancies highlight assembly problem regions or interesting biological events.

<table>
<thead>
<tr>
<th>Organism</th>
<th>Read length</th>
<th>Corrections</th>
<th>AE errors</th>
</tr>
</thead>
<tbody>
<tr>
<td>Listeria monocytogenes</td>
<td>37420828</td>
<td>145274</td>
<td>4</td>
</tr>
<tr>
<td>Wolbachia sp.</td>
<td>11446011</td>
<td>51163</td>
<td>0</td>
</tr>
<tr>
<td>Burkholderia mallei</td>
<td>47407080</td>
<td>99711</td>
<td>28</td>
</tr>
<tr>
<td>Brucella suis</td>
<td>26629877</td>
<td>112359</td>
<td>2</td>
</tr>
<tr>
<td>Streptococcus agalactiae</td>
<td>23485615</td>
<td>105878</td>
<td>3</td>
</tr>
<tr>
<td>Caviella burnetii</td>
<td>29135115</td>
<td>117232</td>
<td>30</td>
</tr>
<tr>
<td>Campylobacter jejuni</td>
<td>15013845</td>
<td>79237</td>
<td>11</td>
</tr>
<tr>
<td>Chlamyophila caviae</td>
<td>10286694</td>
<td>36972</td>
<td>6</td>
</tr>
<tr>
<td>Dehalococcoides ethenogenes</td>
<td>10724521</td>
<td>46416</td>
<td>12</td>
</tr>
<tr>
<td>Neorickettsia sennetsu Miyayama</td>
<td>8805232</td>
<td>37425</td>
<td>0</td>
</tr>
<tr>
<td>Fibrobacter succinogenes</td>
<td>46463268</td>
<td>196150</td>
<td>4</td>
</tr>
<tr>
<td>Mycoplasma capricolum</td>
<td>9353819</td>
<td>15444</td>
<td>0</td>
</tr>
<tr>
<td>Prevotella intermedia</td>
<td>20084365</td>
<td>94162</td>
<td>3</td>
</tr>
<tr>
<td>Pseudomonas syringae</td>
<td>30369232</td>
<td>177897</td>
<td>46</td>
</tr>
<tr>
<td>Total</td>
<td>346625502</td>
<td>1315320</td>
<td>149</td>
</tr>
</tbody>
</table>

Ask Pawel for more information!
Trimming identifies the regions of good quality for the assembler to use (CLR), as the intersection of the region free of vector (CLV) and the region free of bad quality (CLB).
Quick Assembly Review

The individual reads (green) have been assembled into 2 contigs (blue & yellow). The mate relationship between the reads allows for the contigs to be oriented and the gap size to be estimated.
Sequencing Gap
AutoJoiner Architecture

1. All-vs-All Alignment
2. Analyze Alignments
3. Extend and Join Contigs
4. Contig Fattening
5. AutoEdit Result

- AutoJoin
  - nucmer
  - aj_evaluateOverlaps
  - aj_joinContigs
    - fattenslice
    - zipclap
- autoEditor
All-vs-All Alignment

The first AutoJoiner!

```plaintext
% show-coords -rcl out.delta | grep X0E835TF
  35001  35998  |  1023   14  |  998  1010  |  95.15  53479  1023  |  1047283847434  X0E835TF
  53419  53479  |  1023   963  |   61    61  | 100.00  53479  1023  |  1047283847434  X0E835TF
  1012  11335  |  285   405  |  122  121  |   90.24  1163  1023  |   X0DA905TF  X0E835TF
   769   980  |  1023   813  |   212    211  |   98.06   980  1023  |   X0E8361TR  X0E835TF

% show-aligns out.delta 1047283847434 X0E835TF
=================================================================
  | Alignments between 1047283847434 and X0E835TF
  | BEGIN alignment [ +1 53419  53479  |  -1 1023  -963 ]
    53419
     tcttgcgcgcttgagaaaccagatcgacttgccgcatcgcgttgctccag
    1023
     tcttgcgcgcttgagaaaccagatcgacttgccgcatcgcgttgctccag

    53468
     ctgacgccatcgg
    974
     ctgacgccatcgg

  | END alignment [ +1 53419  53479  |  -1 1023  -963 ]
=================================================================
```
Why did AutoJoiner make this join?
Contig Extension

Extension Procedure:

- If necessary, extended selected read by aligning trimmed bases to existing consensus.
- Untrim to desired base, promote untrimmed bases to consensus, shift offsets.
“Zip” together contigs by pairwise alignment between consensi.
“Fatten” addition reads in the join region to increase coverage.
What did AutoJoiner do?
AutoJoiner Validation

25%+ of all sequencing gaps closed with 3 mistakes.
Complicating Issues

- Poly-monomer tails
  - Use dust to filter low complexity sequence

- Undetected repeats
  - Require strict agreement with scaffold

- Chimeric reads / Hard Stops
  - Good: Require high alignment similarity.
  - Better: Recognize hard stops by coverage gradients, other clues.
  - Best: Recognize unreliable sequence at chromatogram level.
Pre-Production Techniques

- **Contig Fattening**
  - TVG coverage increased from 5.83X to 6.10X (mean extension: 80.5bp)

- **Contig Growing**
  - Extended 6144 edges in TVG (mean extension: 59.0bp)
Measuring Assembly Quality

- Gross Status
  - scaffold & contig sizes
- Connectedness Status
  - read & clone coverage
- Insert Status
  - mate happiness,
  - library randomness
- Consensus Status
  - Quality Class
  - Consensus Quality Value
- Read Status
  - Correlated SNPs

QC file
Cloe, AssemblyViewer, getCoverage, cvgChop, asmQC
QC file, Assembly Viewer, asmQC
Cloe, getCoverage, getqc
findTcovSnps
Finding Suspicious Regions

% ls
blasted.frg blasted.asm
% /local/asmg/Linux/bin/cavalidate blasted
Doing step 10:  toAmos
Doing step 20:  bank-transact
Doing step 30:  asmQC
Doing step 40:  bank2contig
Doing step 50:  getCoverage
Doing step 60:  findTcovSnps
Doing step 70:  ClusterSnps
Doing step 80:  Load SNP Features
Doing step 90:  Find Surrogates
Doing step 100:  Load Surrogates
Doing step 1000:  Dump Features
Doing step 1010:  Get Suspicious Features
Doing step 1020:  Create Suspicious Regions
% sort -nrk 6 blasted.snp.feat | head
1047283847436 P HIGH_SNP 141991 142712 46 15.67
For a bacterial sample, correlated discrepancies strongly suggest a repeat has been collapsed.
The shrunken mates (orange) suggest the assembly has a deletion from the true sequence.
Local Assembly

% run_CA -local -noedit -noupload local.frg -dir ca-0.003 -e 0.003
% nucmer 1047283847436.fasta ca-0.003/local.fasta
% /local/asmg/Linux/bin/mummerplot out.delta -R 1047283847436.fasta -Q ca-0.003/
   local.fasta -layout -filter
Resolved Repeat

Unfortunately, size violated mates are only a clue.
Ask Mihai for current research techniques.
## Final Results

<table>
<thead>
<tr>
<th></th>
<th>Original Assembly</th>
<th>Reassembled</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>TotalScaffolds</strong></td>
<td>21</td>
<td>5</td>
</tr>
<tr>
<td><strong>TotalSpanOfScaffolds</strong></td>
<td>4770228</td>
<td>4819528</td>
</tr>
<tr>
<td><strong>IntraScaffoldGaps</strong></td>
<td>51</td>
<td>33</td>
</tr>
<tr>
<td><strong>MeanSequenceGapSize</strong></td>
<td>410.18</td>
<td>96.94</td>
</tr>
</tbody>
</table>

**[Top5Scaffolds]**

<table>
<thead>
<tr>
<th>Rank</th>
<th>Length</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>25:2156009:2164583:357.25</td>
</tr>
<tr>
<td>2</td>
<td>13:826284:830667:365.25</td>
</tr>
<tr>
<td>3</td>
<td>3:437076:439209:1066.50</td>
</tr>
<tr>
<td>4</td>
<td>3:333768:334012:122.00</td>
</tr>
<tr>
<td>5</td>
<td>5:310971:311756:196.25</td>
</tr>
</tbody>
</table>

**[Top5Contigs]**

<table>
<thead>
<tr>
<th>Rank</th>
<th>Length</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>5939:536280</td>
</tr>
<tr>
<td>2</td>
<td>2751:259499</td>
</tr>
<tr>
<td>3</td>
<td>3005:238048</td>
</tr>
<tr>
<td>4</td>
<td>2199:220720</td>
</tr>
<tr>
<td>5</td>
<td>2509:196450</td>
</tr>
</tbody>
</table>

Expected Genome Size: 4.85Mb - 5.04Mb
Research Directions

- AutoEditor 2.0: Better results, better engineering
- Context Based trimming
  - Partial Overlaps
  - Reference sequence
- Advanced CA Techniques
  - Contained Stones bug fixes
  - Blasting Degenerate and Surrogate Unitigs
  - Assembling in the gap
- Arachne & Other Assemblers
- Assembler Reconciliation
- AMOS Framework
- Assembly Forensics
- Assembly Visualization / Navigation
Conclusions

- Overriding strategy: Start conservatively, and iteratively build as more information becomes available.

- 95.5% - 99.2% of genome in a single scaffold not typical yet, but it could be.
  - Be aware of potential size/quality tradeoffs, though.

- Assembly is complicated by genome structure, repeat characteristics, data quality, data management- one size does not fit all, ask for help.
  - Use Data Support!
Acknowledgements

- Steven Salzberg
- Martin Shumway
- Jason Miller
- Pawel Gajer
- Art Delcher
- Mihai Pop
- Adam Phillippy
- WGA
- SE
- Data Support
- Jane Carlton
- Vish Nene
The individual reads (green) have been assembled into 2 contigs (blue & yellow). The mate relationship between the reads allows for the contigs to be oriented and the gap size to be estimated.
1. An all-vs-all pairwise alignment between the full range sequences from the flanking contigs is computed.
2. The alignments are tested for consistency with the scaffold and for being of sufficient quality. If any alignments satisfy the requirements, the best alignment (blue) is selected for joining the contigs.
3. The contigs are extended by extending the selected reads beyond their original clear range to the desired position. If necessary, the reads are first aligned to the existing consensus.
4. The contigs are joined by aligning the newly extended consensi. Alignment gaps inserted into the conseni are promoted into the appropriate positions in the underlying multiple alignment. The joined contig (orange) replaces the original two in the scaffold.
5. The join region is fattened to increase the depth of coverage and enhance the consensus quality.