Ragout – a reference-assisted assembly tool for bacterial genomes

Mikhail Kolmogorov$^1$, Brian Raney$^2$, Benedict Paten$^2$ and Son Pham$^3$

$^1$St. Petersburg University of the Russian Academy of Sciences,
$^2$University of California Santa Cruz, $^3$University of California San Diego

ISMB 2014, Boston
Trans-Siberian Railway

- The longest railroad in the world
- 9248 km
- ~ 15 000 000 railroad ties
A Secret Story

👀 Once Aliens came:
A Secret Story

✿ Once Aliens came:

✿ And they have painted the ties in different colors:
After, they took a lot of pictures:
And after they had been gone, rain has wanished all dyes from the railroad :(

And after they had been gone, rain has vanished all dyes from the railroad :(

Can we now reconstruct the original coloring using those pictures?
And after they had been gone, rain has vanished all dyes from the railroad :(

Can we now reconstruct the original coloring using those pictures?

This is exactly a problem that genome assemblers solve!

- SPAdes
- ABySS
- Velvet
- SOAPdenovo
- SGA
- ...

A Secret Story III
Genome Assembly

- Join short overlapping reads into chromosomes
- Expectation:
Genome Assembly

Join short overlapping reads into chromosomes

Expectation:

Reality:
Complete Sequence?

Jumping libraries:

Long reads:

Still expensive and not as reliable as short reads

Is there any alternative?
Reference-assisted Assembly

- Using a complete genome of another closely-related organism
- Contigs are being aligned on that *reference* genome
Reference-assisted Assembly

- Using a complete genome of another closely-related organism
- Contigs are being aligned on that reference genome

Structural variations?
Rearrangement Approaches

논문:

   - Tries to minimize number of structural variations between two genomes

   - First attempt to use multiple genomes simultaneously
   - One reference and multiple outgroups
   - Still heavily rely on that reference

3. Both approaches may introduce errors
Rearrangement Approaches

  - Tries to minimize number of structural variations between two genomes

- Kim et. al. "Reference-assisted Chromosome Assembly", *PNAS*, 2013
  - First attempt to use multiple genomes simultaneously
  - One *reference* and multiple *outgroups*
  - Still heavily rely on that reference

- Both approaches may introduce errors

- So maybe we need multiple references?
Outline

1. Introduction
2. Quick Overview
3. Algorithm Description
4. Results
5. Further plans
Ragout Recipe

Ragout – Reference-Assisted Genome Ordering UTility

Written in Python/C++

Ingredients:
- Multiple references (in FASTA format)
- Contigs/scaffolds from short-read assembly
- Phylogenetic tree

Output: scaffolds
Outline

1. Introduction
2. Quick Overview
3. Algorithm Description
4. Results
5. Further plans
Genome Representation

- Comparing nucleotide by nucleotide is expensive
- Extract conserved segments (syntenic blocks)
- Assumption: each block is represented exactly once in each genome
Chromosome is represented as an alternating cycle of **directed black** and **undirected red** edges.

- **Black** edges correspond to synteny blocks.
- **Red** edges connect ends of adjacent synteny blocks.
Breakpoint Graphs Are Simple!

Each color defines a perfect matching.
Breakpoint Graphs Are Simple!

Each color defines a perfect matching.
Breakpoint Graphs Are Simple!

Each color defines a perfect matching.
Breakpoint Graphs Are Simple!

Each color defines a perfect matching.
Breakpoint Graphs Are Simple!

Each color defines a perfect matching
Breakpoint Graphs Are Simple!

Each color defines a perfect matching
Breakpoint Graphs Are Simple!

Each color defines a perfect matching
Breakpoint Graphs Are Simple!

Each color defines a perfect matching
Incomplete Breakpoint Graph

Some adjacencies are missing
Incomplete Breakpoint Graph

Some adjacencies are missing
Find missing edges

= Recover perfect matching

There are multiple variants of such matching

How to find the correct one?
States of Adjacencies

\[ a b c d \quad a c b d \]

- State = adjacent vertex
- State of \( c^t \): \( d^t \rightarrow a^h \)
- Rearrangements change states of adjacencies
Choose an arbitrary perfect matching
Choose an arbitrary perfect matching
Pick a vertex from the graph
Choose an arbitrary perfect matching
Pick a vertex from the graph
Label tree nodes as states of chosen vertex in genomes
The tree represents evolution of breakpoint states
Find scenario with minimum number of changes

Associated cost for graph vertex $u$ and tree $T$:

$$P(u, T) = \sum_{\text{branch } (i, j), i \neq j} W(\text{branchlength})$$
Optimal Contigs Order

Cost for a complete graph $G$: $\sum_{u \in G} P(u, T)$

Want a prefect matching which minimizes this cost

An efficient solution:

- Node weight $\rightarrow$ edge weight
- Find minimum weight perfect matching
- Blossom algorithm in $O(n^4)$
Iterative Assembly

(a) Locally consistent

(b) Locally inconsistent

- Solve the dilemma about choice of synteny block size
- Merge scaffolds with different precision into one assembly
Refinement with Assembly Graph

- Incorporate very small/repetitive contigs
- Analogously to repeat resolution in short-read assembly
Outline

1. Introduction
2. Quick Overview
3. Algorithm Description
4. Results
5. Further plans
# Toy Test – One *E. Coli* Reference

<table>
<thead>
<tr>
<th></th>
<th>Ragout</th>
<th>MCM</th>
<th>OSLay</th>
</tr>
</thead>
<tbody>
<tr>
<td>Scaffolds</td>
<td>1</td>
<td>1</td>
<td>8</td>
</tr>
<tr>
<td>Contigs (coverage)</td>
<td>129 (97.9%)</td>
<td>77 (97.6%)</td>
<td>80 (96.7%)</td>
</tr>
<tr>
<td>Miss-ordered</td>
<td>0</td>
<td>0</td>
<td>1</td>
</tr>
</tbody>
</table>

🔍 One *E. Coli* reference without rearrangements

🔍 #Contigs – 156 (98.18%)
## Assembly with Rearrangements – Four *H. Pylori* References

<table>
<thead>
<tr>
<th>#References</th>
<th>Scaffolds</th>
<th>Contigs (cov.)</th>
<th>Miss-ordered</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Ragout</strong></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>1</td>
<td>2</td>
<td>91 (97.7%)</td>
<td>6</td>
</tr>
<tr>
<td>2</td>
<td>2</td>
<td>95 (97.8%)</td>
<td>1</td>
</tr>
<tr>
<td>3</td>
<td>1</td>
<td>95 (97.8%)</td>
<td>1</td>
</tr>
<tr>
<td>4</td>
<td>1</td>
<td>93 (97.6%)</td>
<td>0</td>
</tr>
<tr>
<td><strong>RACA</strong></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>2</td>
<td>3</td>
<td>35 (83.6%)</td>
<td>2</td>
</tr>
<tr>
<td>3</td>
<td>2</td>
<td>35 (83.6%)</td>
<td>1</td>
</tr>
<tr>
<td>4</td>
<td>2</td>
<td>35 (83.8%)</td>
<td>1</td>
</tr>
</tbody>
</table>

Four *H. Pylori* references with rearrangements

# Contigs – 183 (98.57%)
A hybrid approach for the automated finishing of bacterial genomes

Ali Bashir1,2,7, Aaron A Klammer1,7, William P Robins3, Chen-Shan Chin1, Dale Webster1, Ellen Paxinos1, David Hsu1, Meredith Ashby1, Susana Wang1, Paul Peluso1, Robert Sebra1, Jon Sorenson1, James Bullard1, Jackie Yen1, Marie Valdovino1, Emilia Mollova1, Khai Luong1, Steven Lin1, Brianna LaMay1, Amruta Joshi1, Lori Rowe1, Michael Frace4, Cheryl L Tarr4, Maryann Turnsek4, Brigid M Davis5,6, Andrew Kasarskis1, John J Mekalanos4, Matthew K Waldor3,5,6 & Eric E Schadt1,2

- 40 bp non-paired Illumina reads
- Roche 454 reads
- PacBio reads
Long Reads or ...

A hybrid approach for the automated finishing of bacterial genomes

Ali Bashir¹,², mindset, Aaron A Klammer¹, Matthew P Robins³, Chen-Shan Chin¹, Dale Webster¹, Ellen Paxinos¹, David Hsu¹, Meredith Ashby¹, Susana Wang¹, Paul Peluso¹, Robert Sebra¹, Jon Sorenson¹, James Bullard³, Jackie Yen¹, Marie Valdovino¹, Emilia Mollova¹, Khai Luong¹, Steven Lin¹, Brianna LaMay¹, Amruta Joshi¹, Lori Rowe¹, Michael Frace⁴, Cheryl L Tarr⁴, Maryann Turnsek⁴, Brigid M Davis⁵, Andrew Kasarskis¹, John J Mekalanos³, Matthew K Waldor³,⁵,⁶ & Eric E Schadt¹,²

-many bp non-paired Illumina reads
-
Roche 454 reads?
-
PacBio reads?
-
Can we replace long reads with Ragout here?
### Long Reads or Reference-assisted Assembly?

<table>
<thead>
<tr>
<th>#References</th>
<th>Scaffolds</th>
<th>Contigs (cov.)</th>
<th>Miss-ordered</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Ragout</strong></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>1</td>
<td>3</td>
<td>185 (94.8%)</td>
<td>3</td>
</tr>
<tr>
<td>2</td>
<td>2</td>
<td>179 (94.7%)</td>
<td>4</td>
</tr>
<tr>
<td>3</td>
<td>2</td>
<td>174 (94.7%)</td>
<td>0</td>
</tr>
<tr>
<td><strong>RACA</strong></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>2</td>
<td>6</td>
<td>124 (85.8%)</td>
<td>0</td>
</tr>
<tr>
<td>3</td>
<td>3</td>
<td>127 (90.0%)</td>
<td>0</td>
</tr>
</tbody>
</table>

- Three *V. Cholerae* references with rearrangements
- #Contigs – 1407 (96.89%)
- Results are shown without refinement (poor assembly quality)
Outline

1. Introduction
2. Quick Overview
3. Algorithm Description
4. Results
5. Further plans
New results & further plans

Assembly of *Drosophila yakuba* with three other *Drosophila* species:

<p>| | |</p>
<table>
<thead>
<tr>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>Scaffolds</td>
<td>10</td>
</tr>
<tr>
<td>Contigs</td>
<td>1538  (94.92%)</td>
</tr>
<tr>
<td>Miss-ordered</td>
<td>26</td>
</tr>
<tr>
<td>Contigs N50</td>
<td>162 216</td>
</tr>
<tr>
<td>Scaffolds N50</td>
<td>30 316 814</td>
</tr>
</tbody>
</table>

Assembly of multiple mouse lines

Capturing rearrangements with assembly graph

Illumina BaseSpace integration
Acknowledgements

Travel funding was generously provided by Akamai Technologies

http://fenderglass.github.io/Ragout