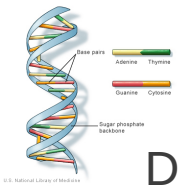
Reducing Genome Assembly Complexity with Optical Maps

AMSC 663
Mid-Year Progress Report
12/13/2011

• Lee Mendelowitz
Lmendelo@math.umd.edu

• Advisor: Mihai Pop
mpop@umiacs.umd.edu
Computer Science Department
Center for Bioinformatics and Computational Biology

Experimental Overview

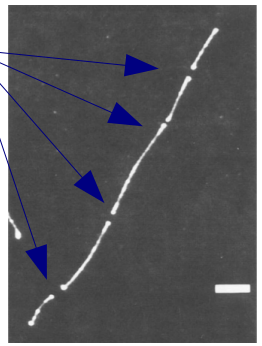


DNA

Optical Map Experiment

(BamHI GGATCC)

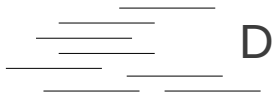
Optical Map



1.937	0.100	C	C
4.713	0.236	C	T
9.742	0.487	A	T
9.241	0.462	T	C
7.734	0.387	G	A
9.186	0.459	G	T
2.069	0.103	A	G
2.886	0.144	C	G
5.084	0.254	T	G
8.260	0.413	C	A
2.300	0.115	C	T
2.232	0.112	T	A
8.209	0.410	C	A
4.122	0.206	T	T

1937	4713	9742	9241	
100	236	487	462	
C C	T A	T T		...

Sequencing Experiment



DNA Reads

~100 bp

Python Script

268	1556	9712	11294	
C T	T C	G C	C A	...

```
contig1 48745 11
268 C T 1556 T C 9712 G C 11294 C A 18254 T C
```

Contig restriction map

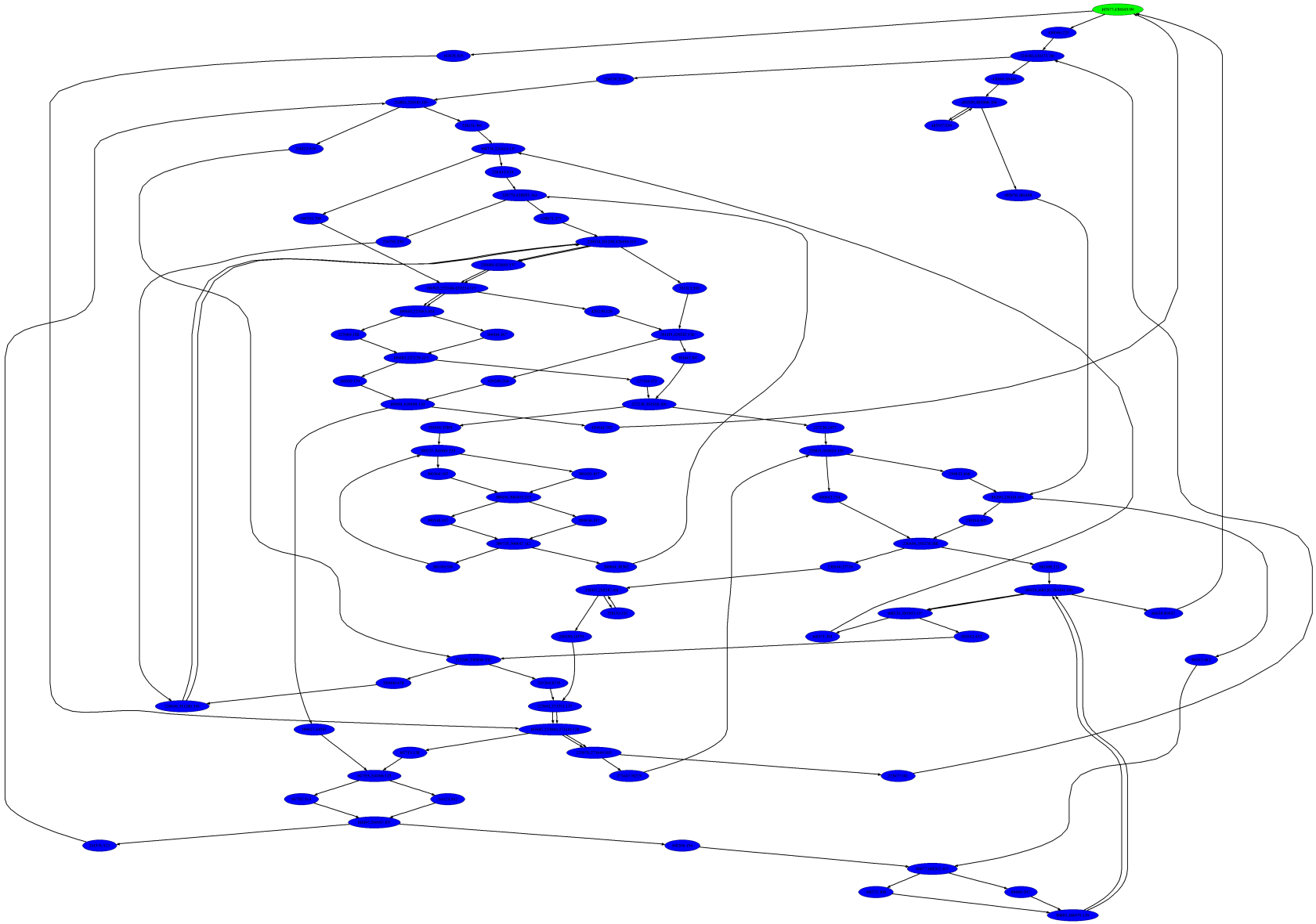
Genome Assembler

Contigs

~ 50 kbp

Assembly Graph

de Bruijn Graph
Mycoplasma genitalium (K=100)



120 edges
84 vertices
- 52 appear 1x
- 28 appear 2x
- 4 appear 3x

Project Schedule & Milestones

- **Phase I (Sept 5 – Nov 27)**
  - Complete code for the contig-optical map alignment tool ☻
  - Test algorithm by aligning user-generated contigs to user-generated optical map ☻
  - Begin implementation of Boost Graph Library (BGL) for working with assembly graphs ☻

Phase II (Nov 27 – Feb 14)

- Finish de Bruijn graph utility functions.
- Complete code for the assembly graph simplification tool
- Test assembly graph simplification tool on simple user-generated graph.

Phase III (Feb 14 – April 1)

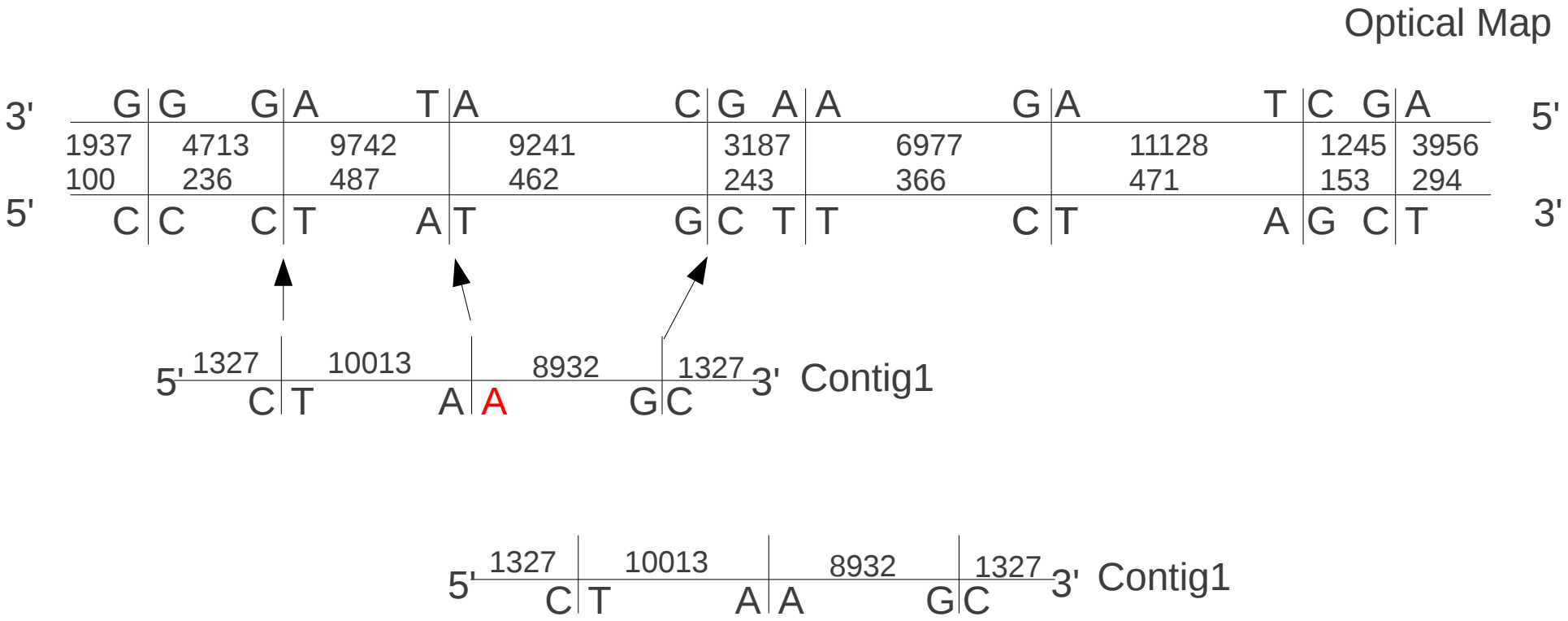
- Validate performance of the contig-optical map alignment tool and the graph simplification tool with archive of de Bruijn graphs for reference bacterial genomes.
- Compute reduction in graph complexities.
- Validate performance using experimentally obtained optical maps + simulated sequence data

Phase IV (time permitting)

- Implement parallel implementation of the contig-optical map alignment tool using OpenMP
- Explore possibility of using the parallel Boost Graph Library.
- Test graph simplification tool on assembly graph produced by a de Bruijn graph assembler.

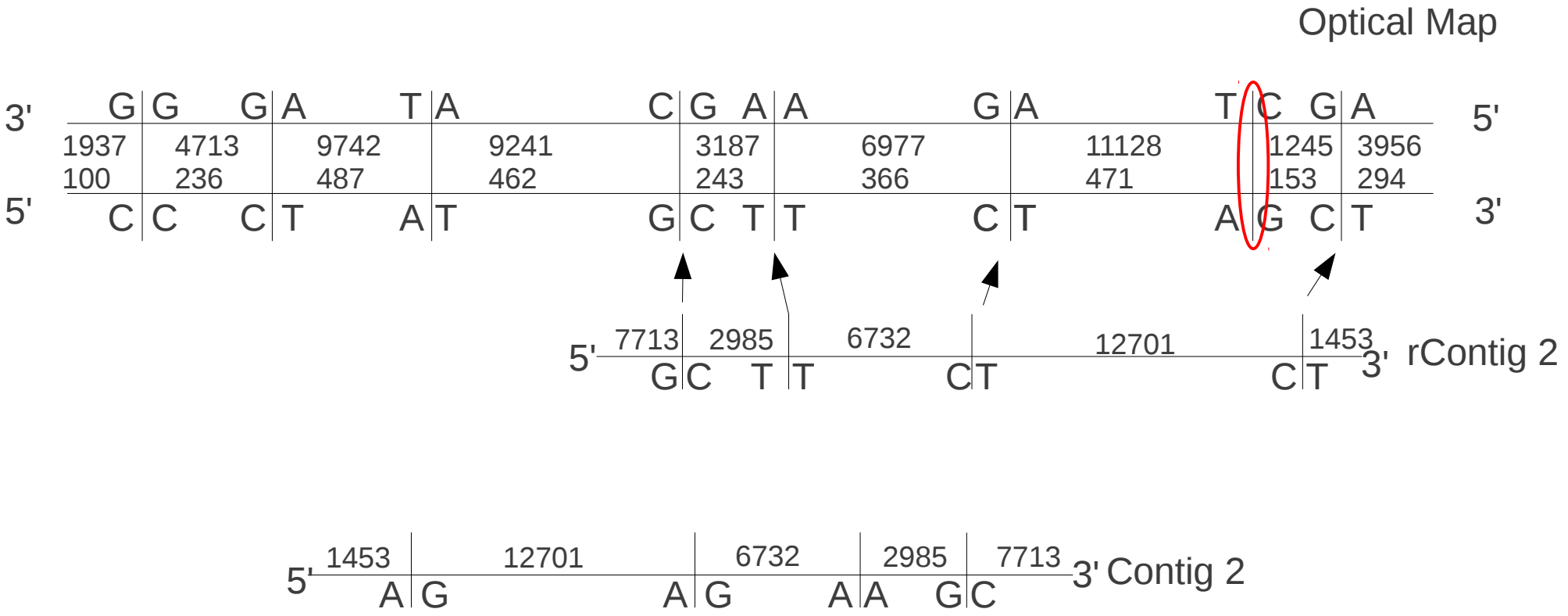
Contig Optical Alignment Tool

Goal: Find the best alignment to the optical map for each contig and evaluate significance of the alignment.



Contig Optical Alignment Tool

Goal: Find the best alignment to the optical map for each contig and evaluate significance of the alignment.

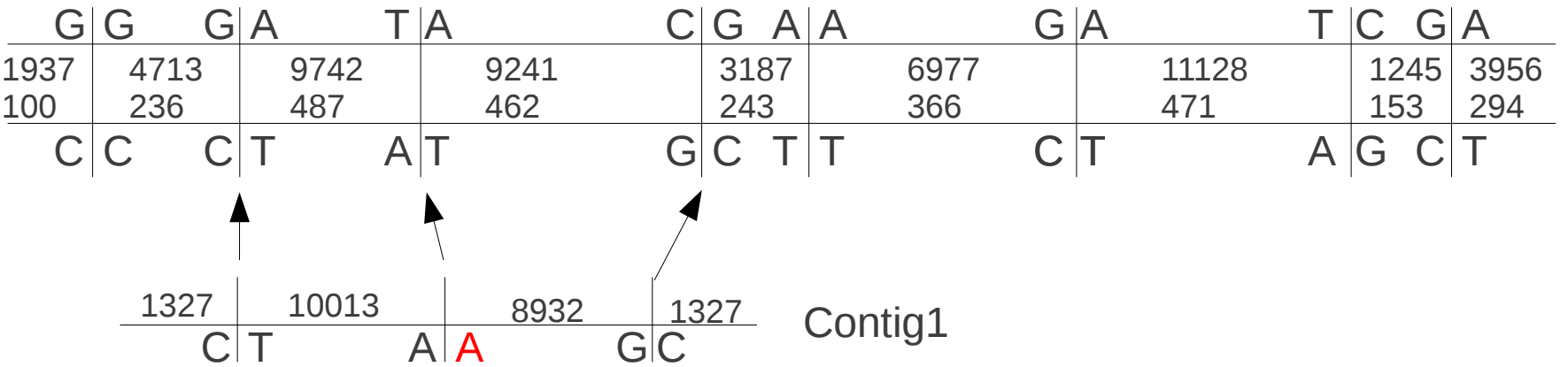


Scoring Alignments

- o_i: Optical restriction fragment mean length
- σ_i: Optical restriction fragment standard deviation
- c_i: contig restriction fragment length

χ^2 scoring function for alignment of contig at position j of optical map:

$$S_{\chi^2} = \sum_{i=1}^{n} \left(\frac{c_i - o_{j+i}}{\sigma_i} \right)^2$$



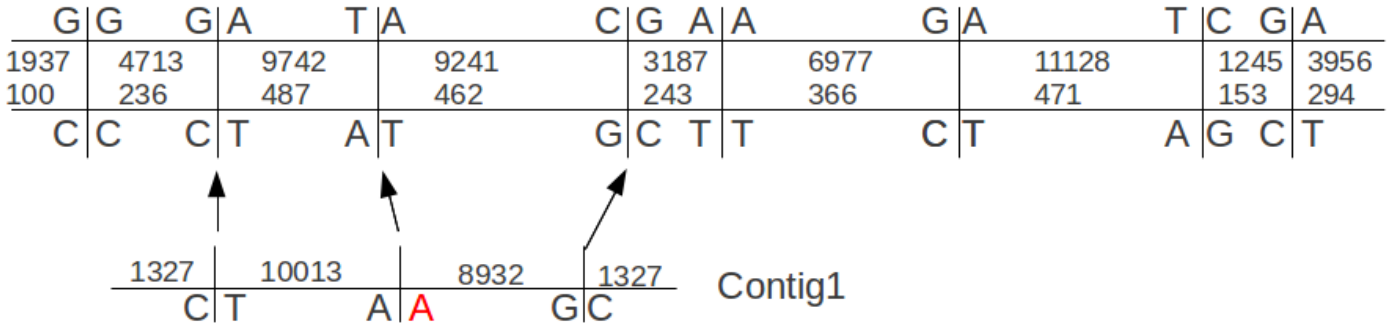
Scoring Alignments

- d_i: edit distance at ith aligned restriction site
- m_r: number of missed restriction sites of alignment
- C_r, C_s: constant weights

Alignment score:

$$S = S_{\chi^2} + C_r \times m_r + C_s \times \sum_{i=1}^{n-1} d_i$$

The best match is given by the lowest score.



Levenshtein Edit Distance (Wagner-Fischer Algorithm)

- Similarity measure between strings
- Allowed edits: Substitution, Deletion, Insertion

a = “ACTGG” b =“CTTCG”

	-	C	T	C	C	G
-	0	1	2	3	4	5
A	1					
C	2					
T	3					
G	4					
G	5					

- $D_{i,j}$: edit distance of a[0:i] and b[0:j]
- $D_{i,0} = i$ and $D_{j,0} = j$
- $D_{i,j} = D_{(i-1),(j-1)}$ if a[i] == b[j]
- $D_{i,j} = \min (D_{(i-1),(j-1)}+1, D_{i,(j-1)}+1, D_{(i-1),j}+1)$ if a[i] != b[j]

Substitution Insertion Deletion

Levenshtein Edit Distance

- $D_{i,j} = D_{(i-1),(j-1)}$ if a[i] == b[j]
- $D_{i,j}$ = Min ($D_{(i-1),(j-1)}$+1 , $D_{i,(j-1)}$+1, $D_{(i-1),j}$+ 1) if a[i] != b[j]

	-	C	T	C	C	G
-	0	1	2	3	4	5
A	1	1	2	3		
C	2	1	2	2		
T	3	2	1	**2**		
G	4	3	2			
G	5	4	3			

Want to edit "ACT to "CTC" with minimum number of edits.

- Option 1: Edit "AC" to "CT" and Substitute "C" for "T"
  - D("ACT", "CTC") = D("AC", "CT") + 1 = 3
- Option 2: Edit "ACT" to "CT" and Insert "C"
  - D("ACT", "CTC") = D("ACT", "CT") + 1 = 2
- Option 3: Edit "AC" to "CTC" and Delete "T"
  - D("ACT", "CTC") = D("AC", "CTC") + 1 = 3

Insertion
Deletion
Substitution
Match

Answer: Edit "ACT" to "CT and Insert C

A C T -

- C T C

D("ACT", "CTC") = D("ACT","CT") + 1 = 2

Levenshtein Edit Distance

- $D_{i,j} = D_{(i-1),(j-1)}$ if a[i] == b[j]
- $D_{i,j} = \text{Min}\ (\ D_{(i-1),(j-1)}+1\ ,\ D_{(i),(j-1)}+1,\ D_{(i-1),(j-1)}+\ 1\ )$ if a[i] != b[j]

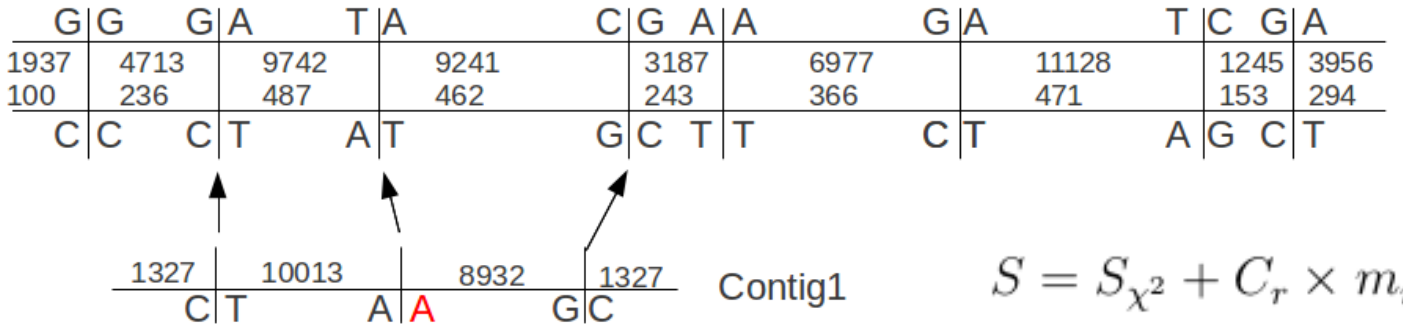
	-	C	T	C	C	G
-	**0**	1	2	3	4	5
A	**1**	1	2	3	4	5
C	2	**1**	2	2	3	4
T	3	2	**1**	**2**	3	4
G	4	3	2	2	**3**	3
G	5	4	3	3	3	**3**

Insertion
Deletion
Substitution
Match

Answer: 3 Edits

A C T - G G
- C T C C G

Alignment Algorithm



$$S = S_{\chi^2} + C_r \times m_r + C_s \times \sum_{i=1}^{n-1} d_i$$

- S_{ij} : Score of the best alignment of contig through ith fragment with optical map through jth fragment.
- Find S_{ij} by extending a previously scored alignment $S_{i',j'}$ where $0 \leq i' < i, 0 \leq j' < j$.

$$S_{ij} = \min_{0 \leq k \leq i, 0 \leq l \leq j} C_r \times (i-k+j-l) + C_s \times d_{ij} + \frac{(\sum_{s=k}^{i} c_s - \sum_{t=l}^{j} o_t)^2}{\sum_{t=l}^{j} \sigma_t^2} + S_{(k-1)(l-1)}$$

Missed restriction sites

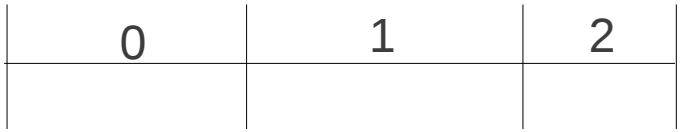
Sequence Edit Distance

Chi-Square

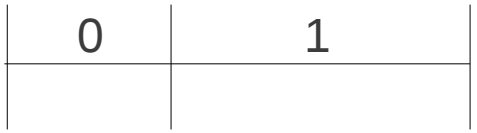
Prefix alignment score

Alignment Algorithm

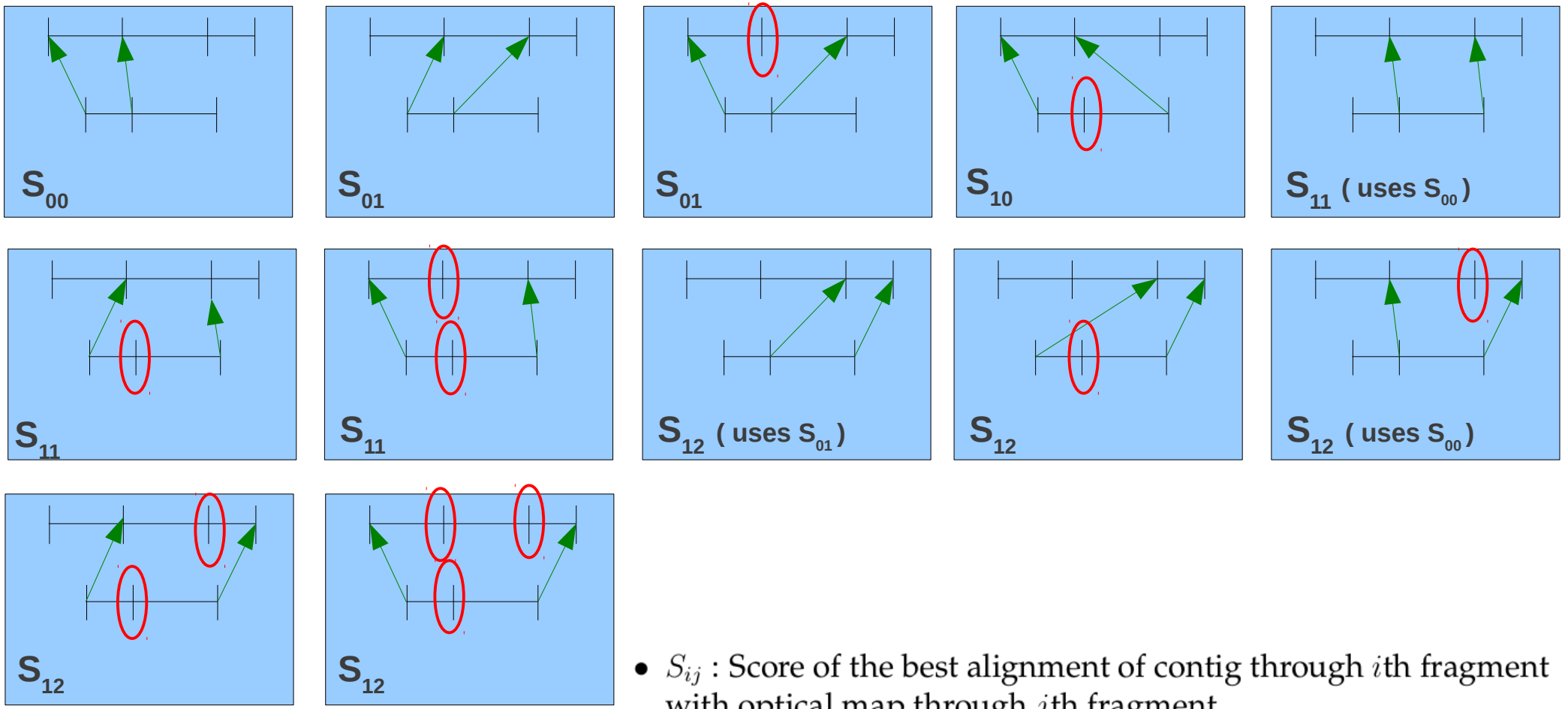
S_{00}	S_{01}	S_{02}
S_{10}	S_{11}	S_{12}



Optical Map



Contig



- S_{ij} : Score of the best alignment of contig through ith fragment with optical map through jth fragment.

Alignment Algorithm

- Complexity: $\mathcal{O}(m^2n^2)$ for m contig fragments and n optical map fragments.
- Must double the number of fragments for a circular genome.
- Must try aligning a contig and its reverse complement.
- Reject any alignment of a contig fragment of length c to an optical fragment of length o with standard deviation σ if:

$$|c - o| > C_\sigma \sigma$$

Evaluating Alignments

• Can evaluate how significant an alignment is between a contig and the optical map through a **permutation test**

- Permute the restriction fragments of the contig and determine the best alignment score of the permuted contig
- 500 samples from space of permuted contigs
- Evaluate the probability that a permuted contig aligns better to the optical map than the original contig.

$$P(\text{alignment score of permuted contig} \geq \text{alignment score of original contig})$$

Validations/Results

Test 1:

- Randomly generated optical map (small standard deviation), n=100
- 10 extracted contigs (both forward and reverse, no errors)
- 10 random contigs
- Permutation test off

- $C_\sigma = 5$
- $C_r = C_s = 12,500$

Result:

- 10 extracted contigs mapped to correct location
- 10 random contigs mapped with poor quality

True Contig:

```
Matching with standard deviation threshold: 5
contig5 75808 1 19 35
0 0 0 0 1
G 1774,1 C ; T 1288,1 T ; C 8156,1 G ; C 1582,1 C ; A 6960,1 T ; C 3754,1 C ;
; T 1415,1 A ; T 7223,1 T ; T 8010,1 C ; A 1719,1 A
 205 C; T 1288 T; C 8156 G; C 1582 C; A 6960 T; C 3754 C; T 4155 C; G 4452 C;
CTTCGCCATCCTCGCCTTGCAAGACTATTTCA
CTTCGCCATCCTCGCCTTGCAAGACTATTTCA
```

Random Contig:

```
Matching with standard deviation threshold: 5
contig17 83389 1 92 102
25 1 0 1 1
A 8440,1 A ; A 9851,1 G  T 9971,1 G  C 5105,1 C  T 3447,1 G
 7028 A; G 5009 G A 3718 A T 3385 T T 7096 C T 3052 A T 3081
AA
AG
```

Validations/Results

Test 2:

- Randomly generated optical map (standard deviation up to 5%), n=400
- 30 extracted contigs
  - Both forward and reverse
  - 10% substitution error rate
  - 10% false site / missing site rate
- 10 random contigs
- Permutation test on

- $C_\sigma = 5$
- $C_r = C_s = 12,500$

Result:

- 30 true contigs aligned to correct location
- 1 of 10 random contigs aligned with significance (False Positive):

```
Matching with standard deviation threshold: 5
contig37 40185 0 373 380
4 2 67.7077 0.03 1
G 4232,212 A ; A 4988,249 C ; T 9632,482 T ; A 8667,433 C ; C 8217,411 G  C 1560,100 A ; G 4026,201 T ; A 7448,372 C
 242 A; A 3966 C; T 5335 A C 3225 T; A 7497 C; A 3945 T C 7861 A; G 2689 C T 2131 A; A 3294
AACTTACCAGTA
AACTTACAAGAA
{}
```

Validations/Results

False positive with $C_r = C_s = 12{,}500$....

```
Matching with standard deviation threshold: 5
contig37 40185 0 373 380
4 2 67.7077 0.03 1
G 4232,212 A ; A 4988,249 C ; T 9632,482 T ; A 8667,433 C ; C 8217,411 G  C 1560,100 A ; G 4026,201 T ; A 7448,372 C
 242 A; A 3966 C; T 5335 A C 3225 T; A 7497 C; A 3945 T C 7861 A; G 2689 C T 2131 A; A 3294
AACTTACCAGTA
AACTTACAAGAA
{}
```

… becomes true negative with $C_r = 5$, $C_s = 3$

```
Matching with standard deviation threshold: 5
contig37 40185 1 204 210
9 1 0.430057 0.978 1
C 6456,323 A  C 9258,463 G ; C 3879,194 T ; G 4775,239 A  G 2810,141 T ; A 2684,134 C  G 8003,400 A
 3294 T T 2131 A G 2689 C T 7861 G; A 3945 T; G 7497 T; A 3225 G T 5335 A G 3966 T T 242
GCTGTA
GATGTA
```

...but these constants introduce a new false positive.

```
Matching with standard deviation threshold: 5
contig39 49493 0 88 95
4 7 2.72827 0.046 1
A 9722,486 T ; C 7093,355 C  C 1926,100 T ; G 8401,420 C ; C 9405,470 A ; T 8148,407 A ; T 9232,462 C ; T
 1991 C; C 5170 T G 3920 T; C 8817 A; A 8923 A; A 8023 C; T 4460 T T 2164 G G 2261 C; G 3764
TCTGCCATATCT
CCTCAAAACTCG
{}
```

Project Schedule & Milestones

Phase I (Sept 5 – Nov 27)

- Complete code for the contig-optical map alignment tool ☻
- Test algorithm by aligning user-generated contigs to user-generated optical map ☻
- Begin implementation of Boost Graph Library (BGL) for working with assembly graphs ☻

Phase II (Nov 27 – Feb 14)

- Finish de Bruijn graph utility functions.
- Complete code for the assembly graph simplification tool
- Test assembly graph simplification tool on simple user-generated graph.

Phase III (Feb 14 – April 1)

- Validate performance of the contig-optical map alignment tool and the graph simplification tool with archive of de Bruijn graphs for reference bacterial genomes.
- Compute reduction in graph complexities.
- Validate performance using experimentally obtained optical maps + simulated sequence data

Phase IV (time permitting)

- Implement parallel implementation of the contig-optical map alignment tool using OpenMP
- Explore possibility of using the parallel Boost Graph Library.
- Test graph simplification tool on assembly graph produced by a de Bruijn graph assembler.

References

Kingsford, C., Schatz, M. C., & Pop, M. (2010). Assembly complexity of prokaryotic genomes using short reads. BMC bioinformatics, 11, 21.

Nagarajan, N., Read, T. D., & Pop, M. (2008). Scaffolding and validation of bacterial genome assemblies using optical restriction maps. Bioinformatics (Oxford, England), 24(10), 1229-35.

Pevzner, P. a, Tang, H., & Waterman, M. S. (2001). An Eulerian path approach to DNA fragment assembly. Proceedings of the National Academy of Sciences of the United States of America, 98(17), 9748-53.

Samad, a, Huff, E. F., Cai, W., & Schwartz, D. C. (1995). Optical mapping: a novel, single-molecule approach to genomic analysis. Genome Research, 5(1), 1-4.

Schatz, M. C., Delcher, A. L., & Salzberg, S. L. (2010). Assembly of large genomes using second-generation sequencing. Genome research, 20(9), 1165-73.

Wetzel, J., Kingsford, C., & Pop, M. (2011). Assessing the benefits of using mate-pairs to resolve repeats in de novo short-read prokaryotic assemblies. BMC bioinformatics, 12, 95.