Reducing Genome Assembly Complexity with Optical Maps

AMSC 664
Final Presentation
5/11/2012

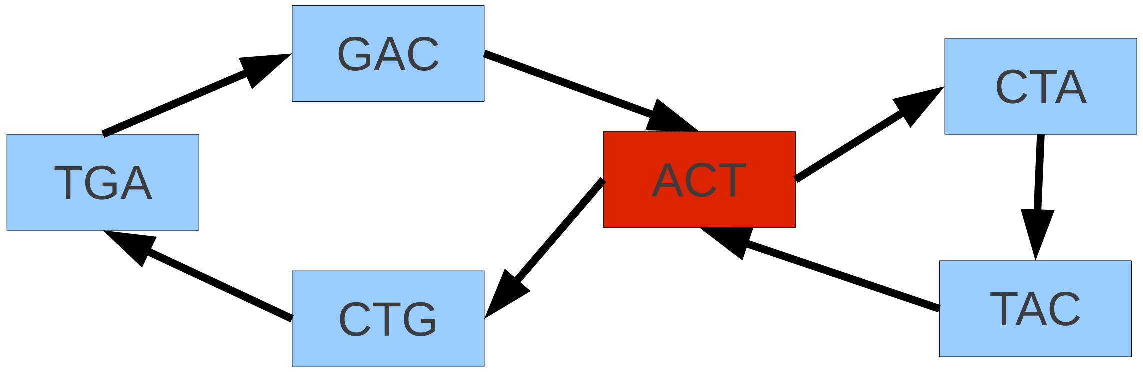
• Lee Mendelowitz
Lmendelo@math.umd.edu

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

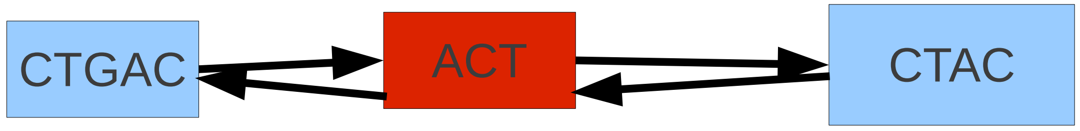
Genome Assembly with de Bruijn Graphs

`Genome = ACTACTGACT, K = 4`

```
ACTACTGACT
ACT
 CTA
  TAC
   ACT
    CTG
     TGA
      GAC
```

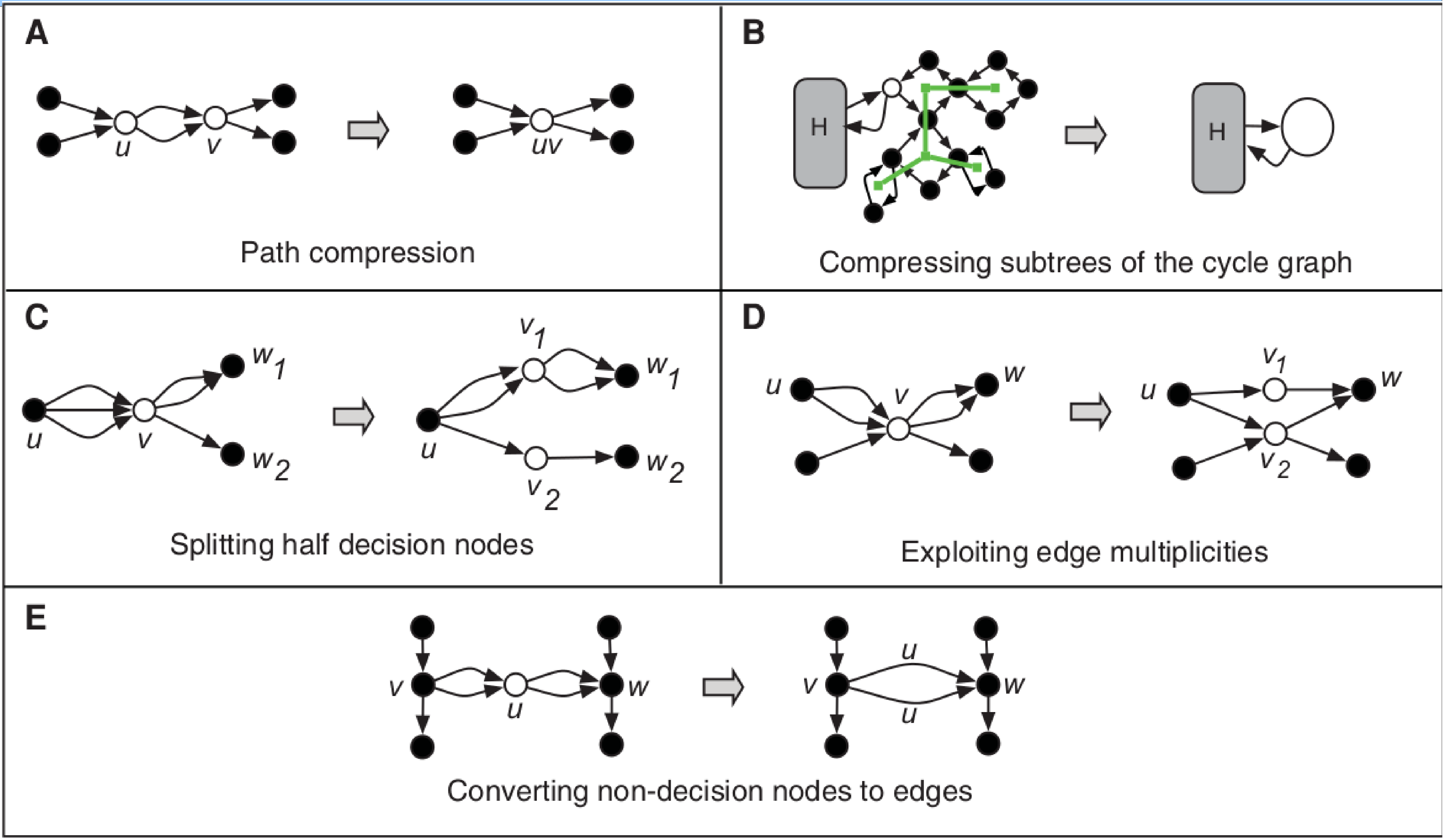


Equivalently:



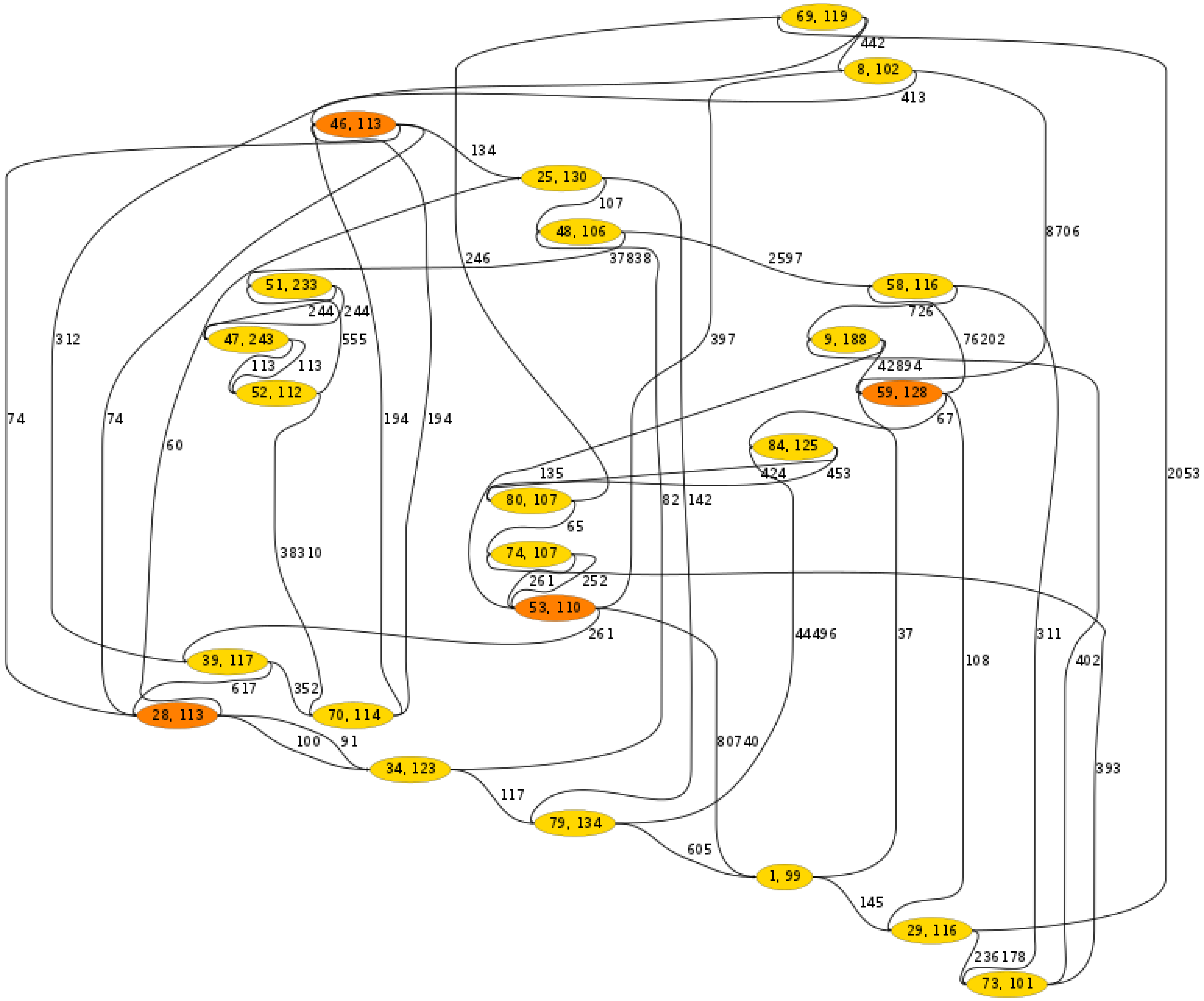
- Multidigraph with one strongly connected component.
- Reconstruction of genome is an Eulerian tour
- In-degree = Out-degree
- Nodes labeled with sequence of length K-1
- Overlaps of K-2 bases
- # of Eulerian tours combinatorial in the number of repeats

Graph Simplification Operations

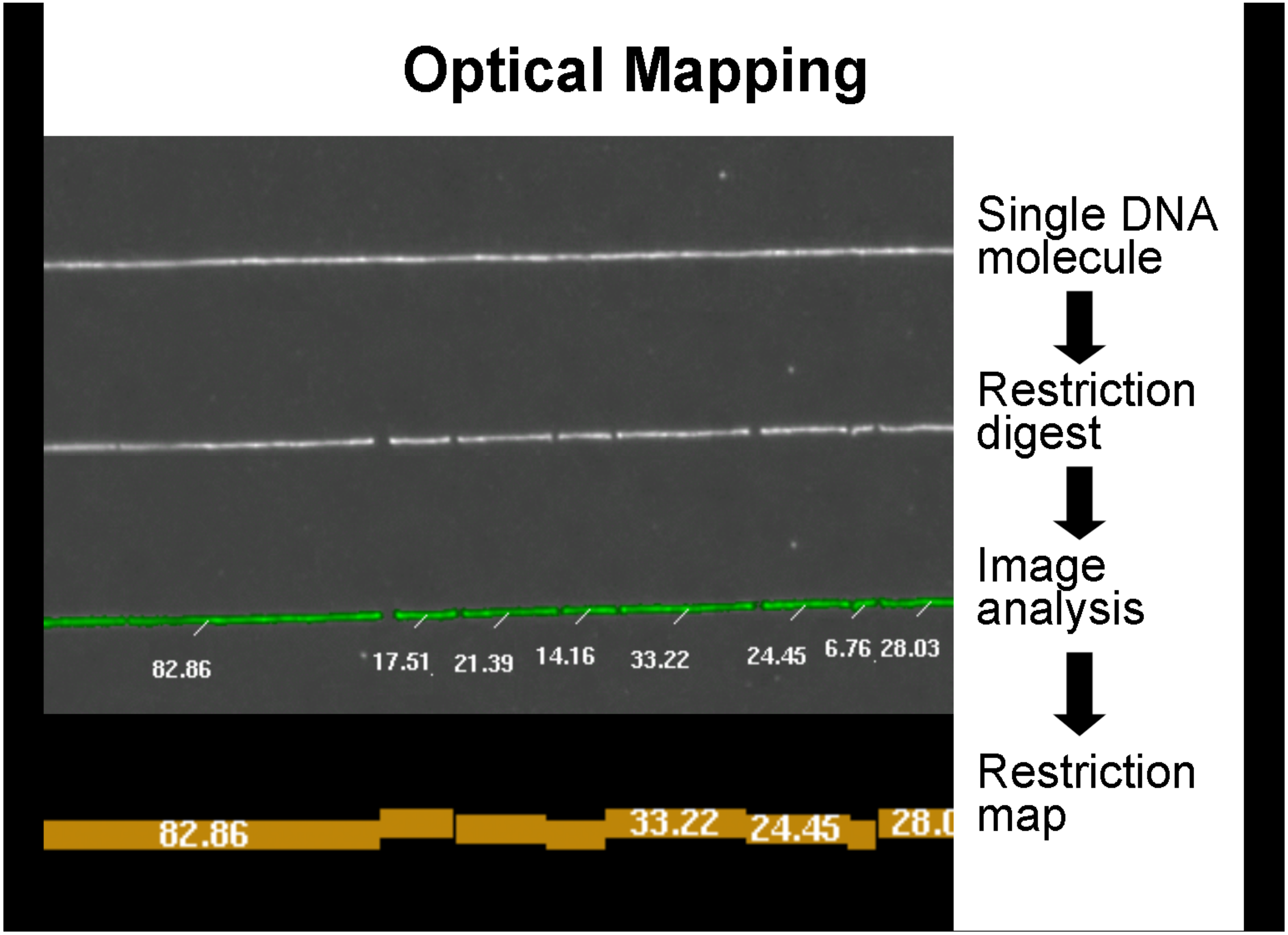

A
u
v
uv
Path compression
B
H
H
Compressing subtrees of the cycle graph
C
u
v
w1
w2
v1
v2
u
w1
w2
Splitting half decision nodes
D
u
v
w
u
v1
v2
w
Exploiting edge multiplicities
E
v
u
w
v
u
u
w
Converting non-decision nodes to edges

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

de Bruijn Graph
Mycoplasma genitalium (K=100)



Experimental Overview



Project Goals

- Develop the **Contig-Optical Map Alignment Tool.**
  - Aligns contigs to an optical map based on restriction pattern with sequence information.
  - Evaluate significance of alignments through a permutation test.

- Develop the **Graph Simplification Tool,** with functionality to:
  - Read and write graphs to/from files.
  - Count the number of unique shortest paths between two nodes.
  - Modify the graph by replacing a selected path with a single edge.
  - Simplify the graph through path compression.

- Develop a **Pipeline**:
  - Integrate the Contig-Optical Map Alignment Tool and Graph Simplification Tool.
  - Generates simulated optical maps.
  - Evaluate the correctness of the graph simplification operations
  - Write debug level logs files and summary files to disk.
  - Submit jobs to Condor cluster.

- **Validate** pipeline on dataset of 351 prokaryotic reference genomes.

Project Schedule & Milestones

Phase I (Sept 5 – Nov 27)

- Complete code for the contig-optical map alignment tool (C++)
- Test algorithm by aligning user-generated contigs to user-generated optical map
- Begin implementation of networkx for working with assembly graphs

Phase II (Nov 27 – Feb 14)

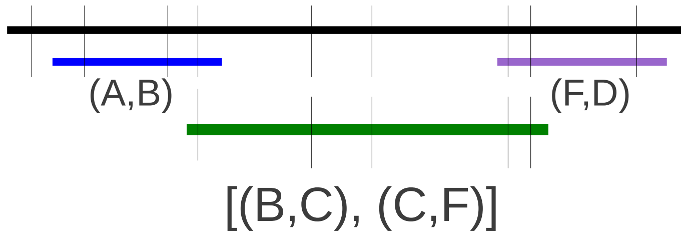
- Finish de Bruijn graph utility functions.
- Complete code for the assembly graph simplification tool (Python)
- Test assembly graph simplification tool on simple user-generated graph.
- Implement parallel implementation of the contig-optical map alignment tool using OpenMP

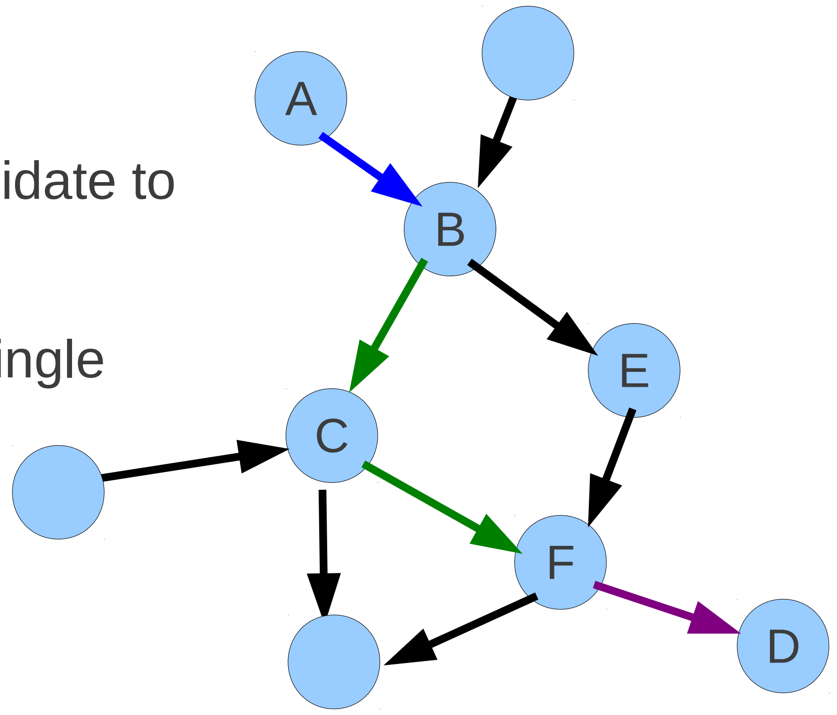
Phase III (Feb 14 – May 8)

- Integrate alignment tool and graph simplification tool into a single pipeline (Python)
- 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.

Algorithmic Recipe

1. Align contigs (graph edges) to optical map
2. Tile uniquely aligned contigs across optical map
3. Find shortest paths between aligned contig neighbors.
4. Select unique shortest paths as gap closure candidates.
5. Perform global alignment of gap closure candidate to the optical map and accept/reject path.
6. Replace accepted paths in the graph with a single edge.
7. Perform path compression.
8. Evaluate graph correctness





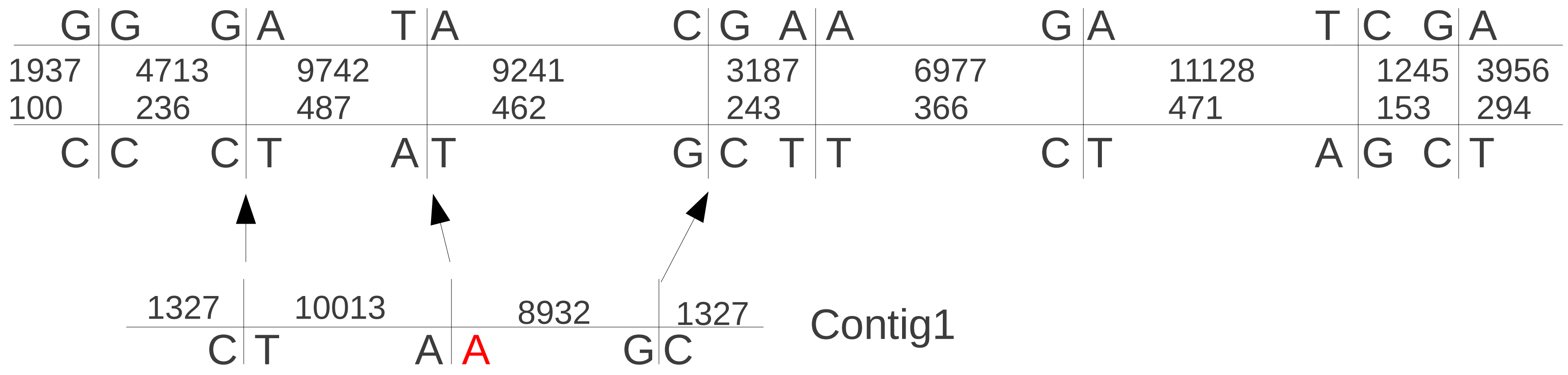
Contig-Optical Map Alignment Tool

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:

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



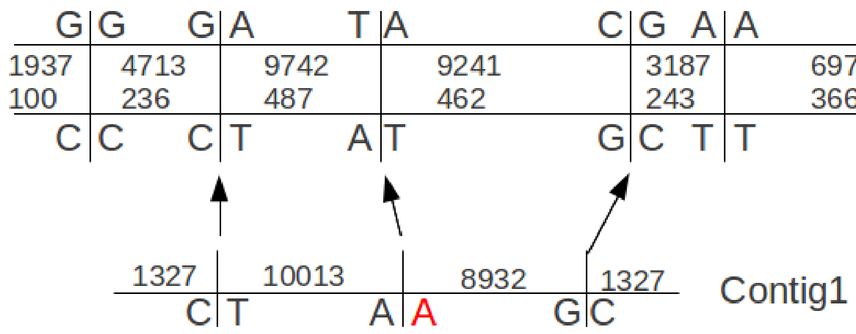
Scoring Alignments

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

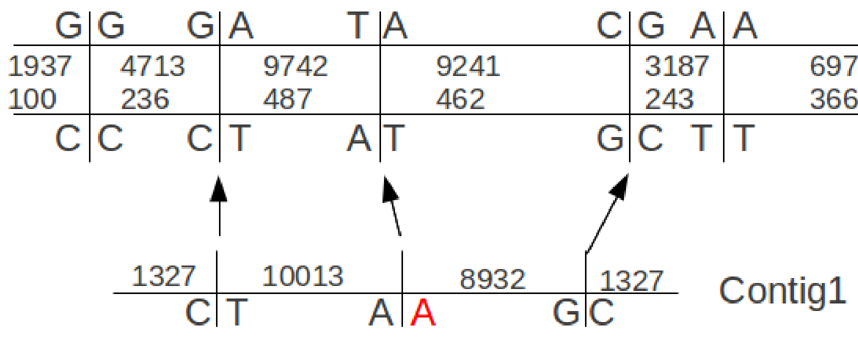
Alignment score:

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

The best match is given by the lowest score.



Alignment Algorithm



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

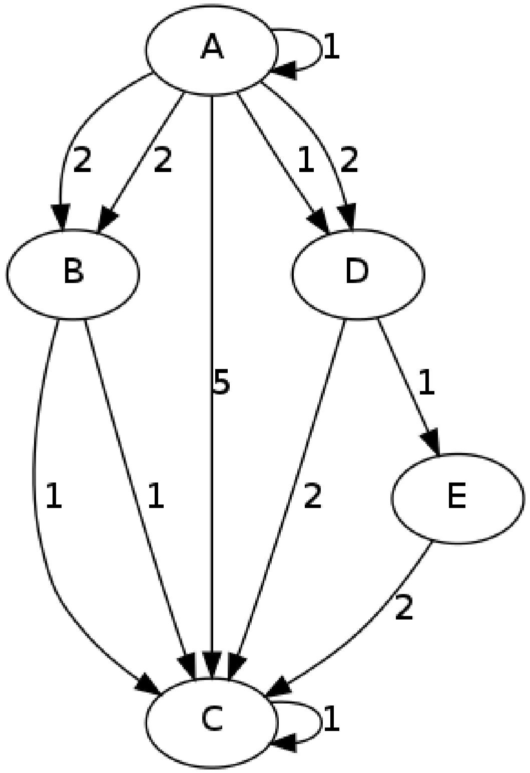
- S_{ij} : Score of the best alignment of contig through ith fragment with optical map through jth fragment.
- Find S_{ij} by extending a previously scored alignment $S_{i',j'}$ where $0 \leq i' < i$, $0 \leq j' < j$.

$$S_{ij} = \min_{0 \le k \le i, 0 \le l \le j} \underbrace{C_r \times (i-k+j-l)}_{\text{Missed restriction sites}} + \underbrace{C_s \times d_{ij}}_{\text{Sequence Edit Distance}} + \underbrace{\frac{(\sum_{s=k}^{i} c_s - \sum_{t=l}^{j} o_t)^2}{\sum_{t=l}^{j} \sigma_t^2}}_{\text{Chi-Square}} + \underbrace{S_{(k-1)(l-1)}}_{\text{Prefix alignment score}}$$

Assembly Graph Simplification Tool

Count Number of Shortest Paths

- **Goal:** Count the number of unique shortest paths from source node to target node.
- Dijkstra's algorithm: O(E + V log(V))
  - Store examined nodes with tentative distances in a priority queue.
  - Store set of visited nodes.
- For each node store a set of predecessors on shortest paths from source.



Distance from A:
A: 0
B: 2
D: 1
C: 3
E: 2

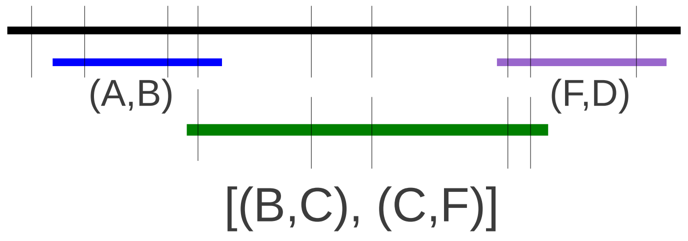
Predecessors
A: [ ]
B: [A]
D: [A]
C: [B,D]
E: [D]

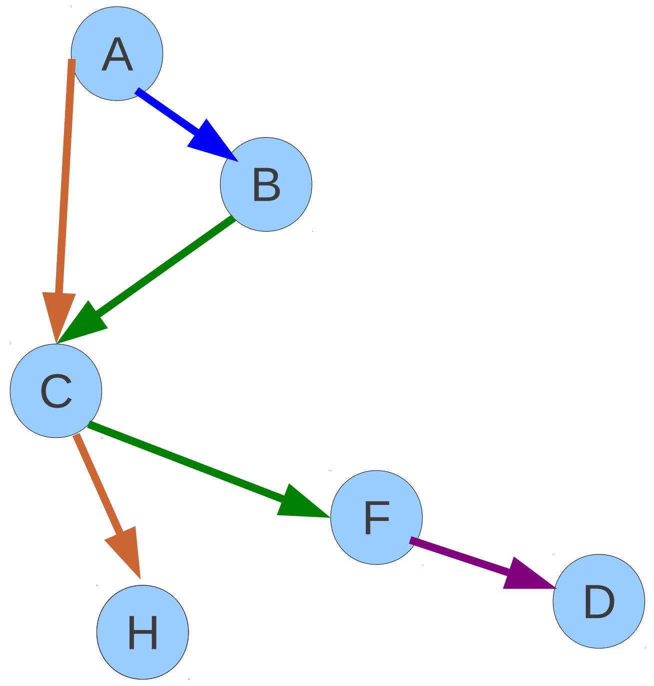
Node Paths: [A,B,C], [A,D,C]
Edge Paths: [(A,B,0), (B,C,0)]
[(A,B,0), (B,C,1)]
[(A,B,1), (B,C,0)]
[(A,B,1), (B,C,1)]
[(A,D,0), (D,C,0)]

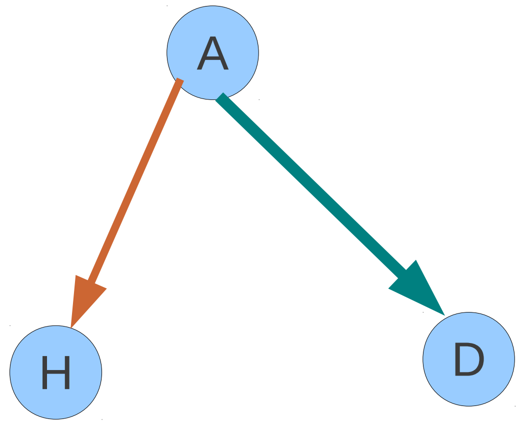
Edge denoted by (Node 1, Node 2, Edge Key)

Graph Simplification

- Replace a given path with a single edge.
- Delete any disconnected nodes.
- Perform path compression. (A → C → H)
- Assert validity of the graph





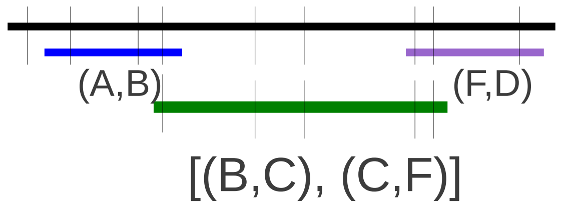


Validate on 351 Prokaryotic Genomes

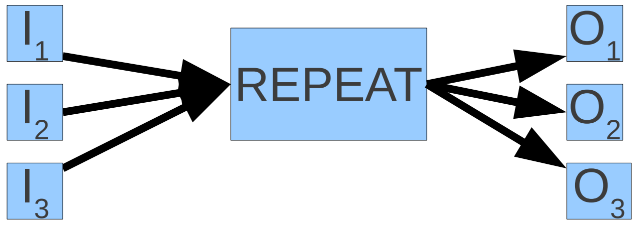
- Simulate optical maps from reference genomes.
  - Enzyme = BamHI (GGATCC), K=100, Fragment Variance = 0.3 * Fragment Length
    - No error
    - Low error (sizing error s.d = 1%, 10% substitutions, 5% missing sites)
    - High error (sizing error s.d. = 5%, 20% substitutions, 10% missing sites)

- Evaluate **alignment correctness**:
  - Alignment within 0.1% of true contig location

- Evaluate **path correctness** for selected closure paths using longest common subsequence.
  - True path: [(A,B,0), (B,C,1), (C,F,0), (F,D,2)]
  - Selected Path: [(A,B,0), (B,C,1),(C,E,1),(E,F,0),(F,D,2)]
  - Common path length from edges (A,B,0) + (F,D,2)
  - Path correctness is ratio of common length to true length



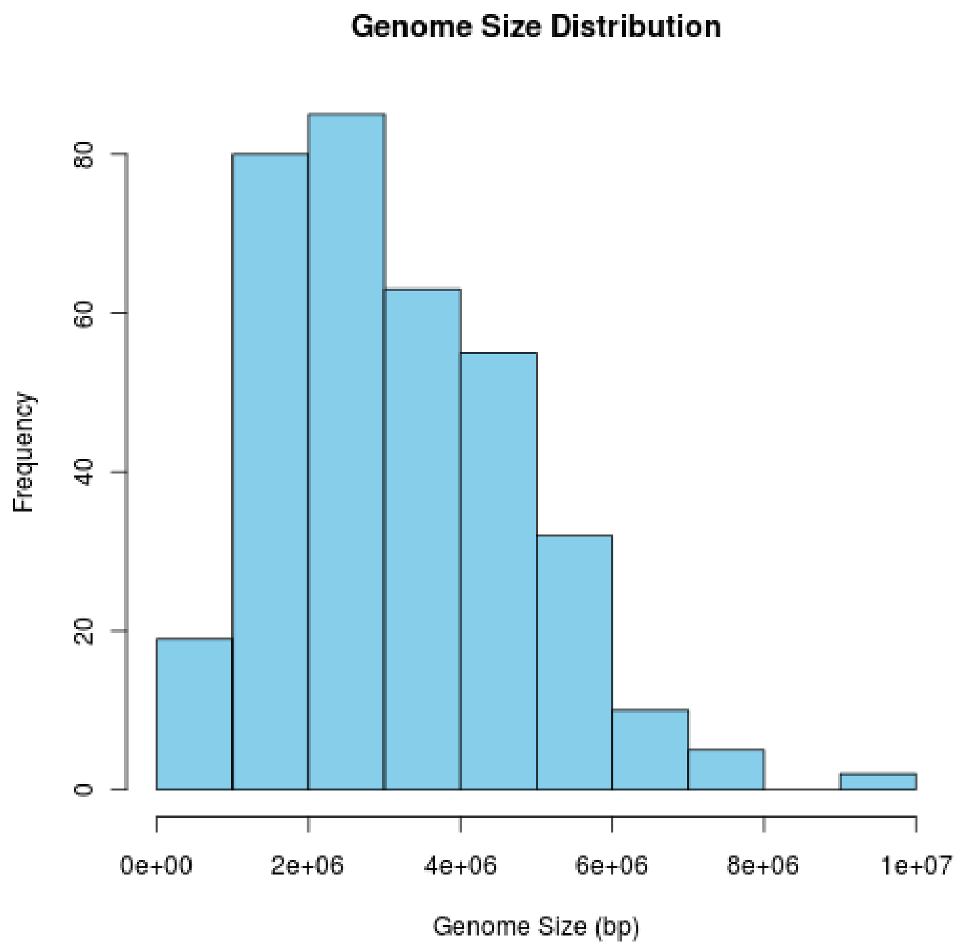
- Evaluate reduction in complexity. Example: a = 3

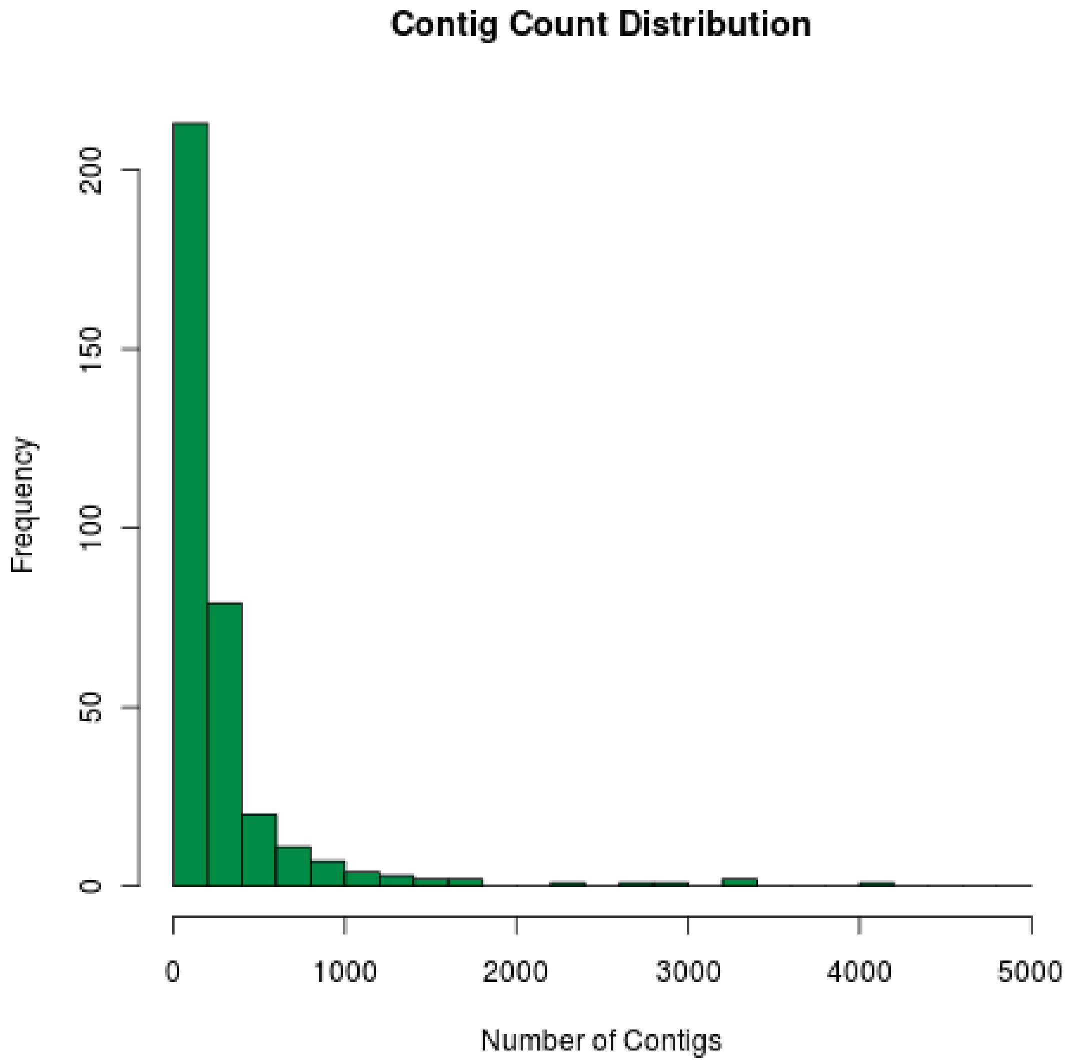


$$C(v) = \sum_{i=2}^{a} i = \frac{a(a+1)}{2} - 1$$

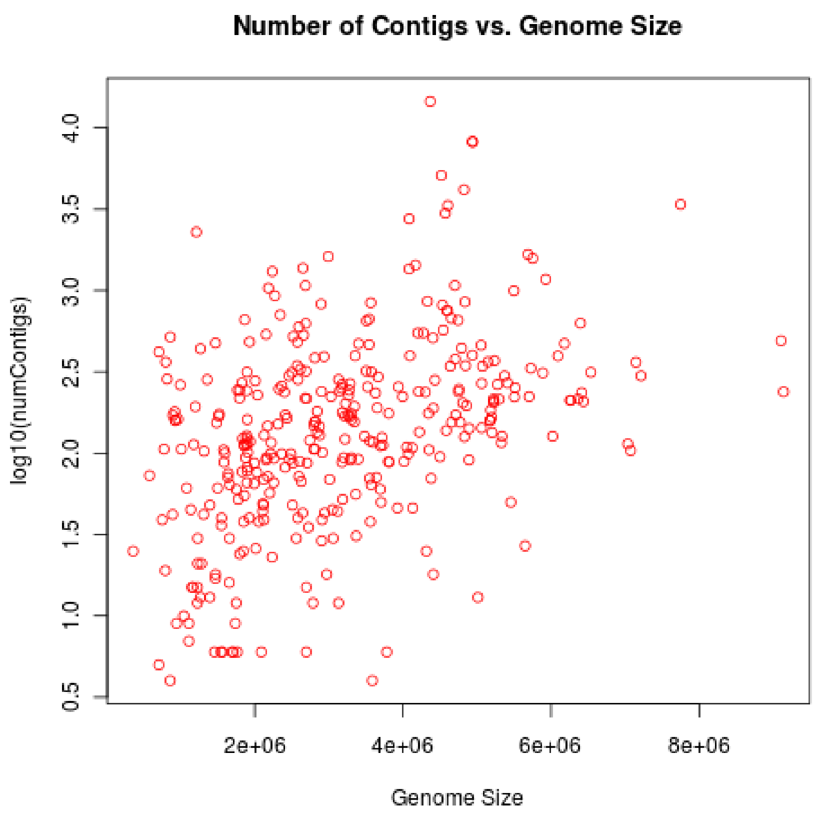
$$C(G) = \sum_{v \in V} C(v)$$

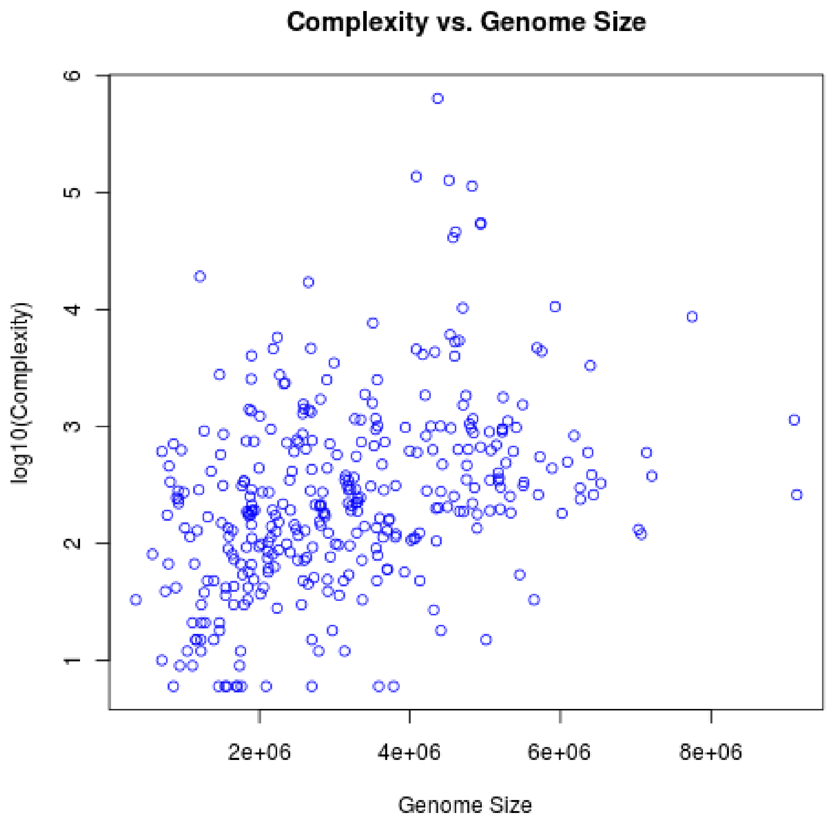
Validation Data Set: 351 Genomes



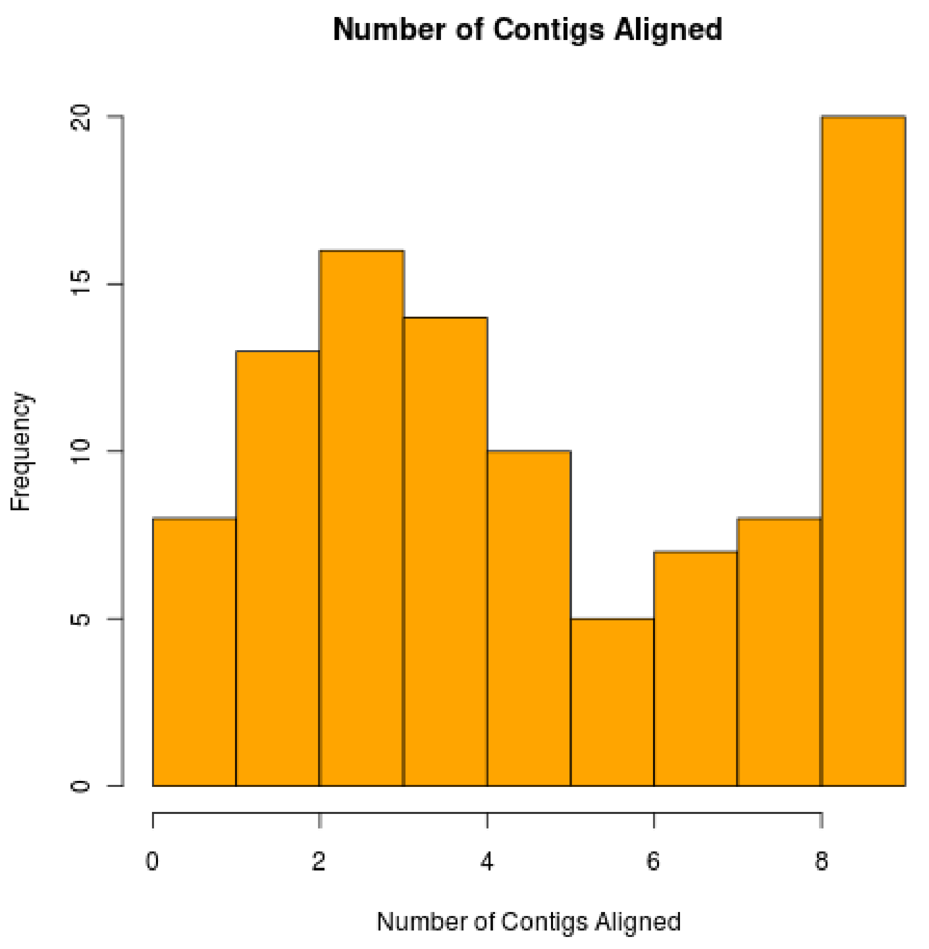


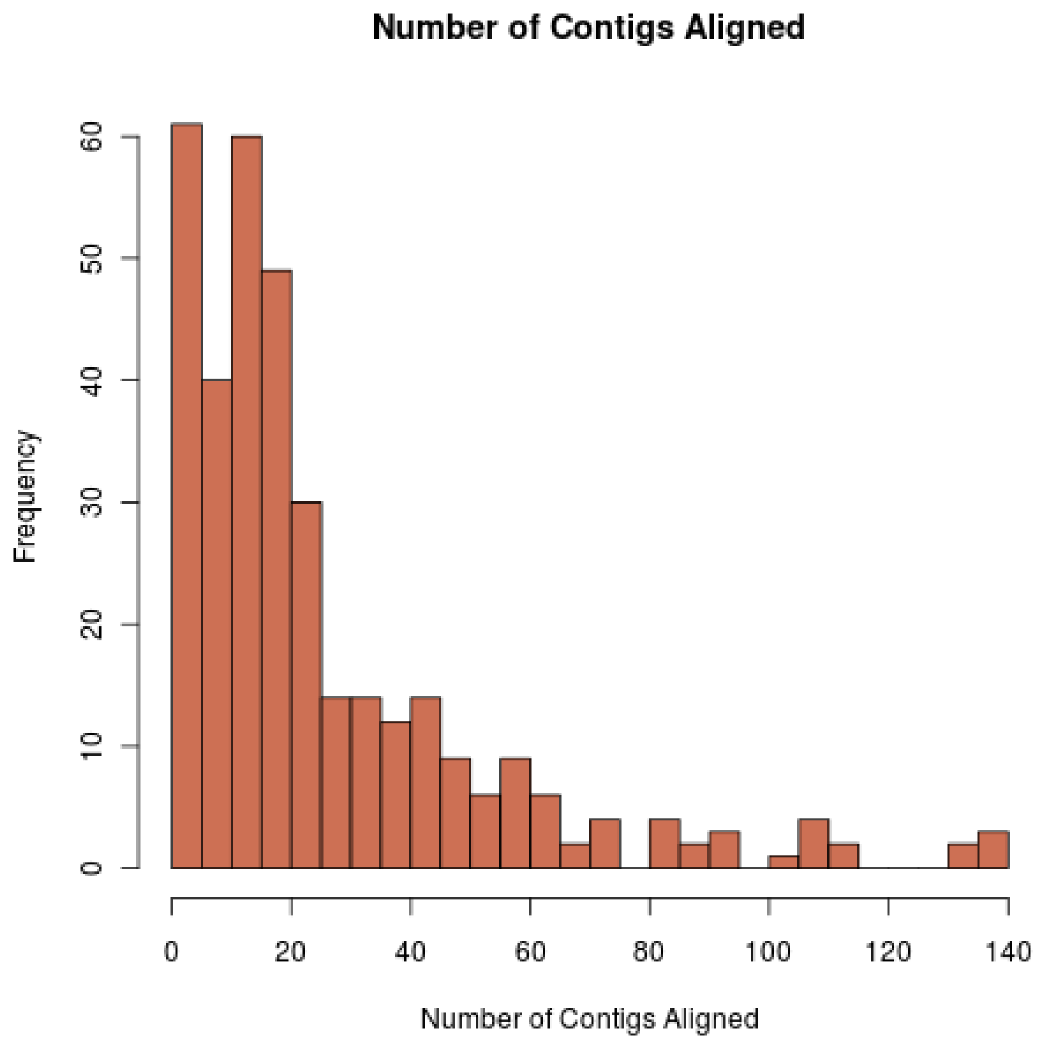
Validation Data Set: 351 Genomes





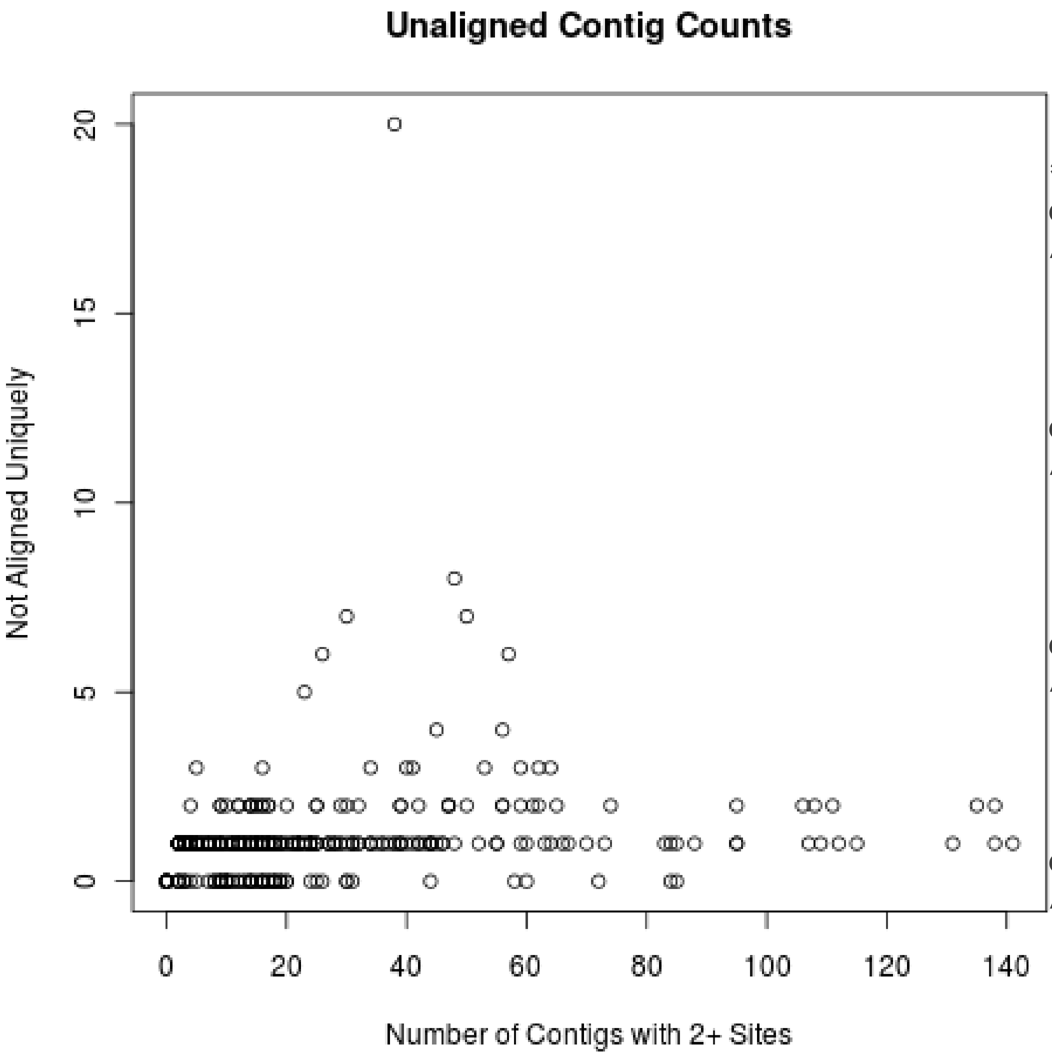
Alignment Results:
(Error Free Optical Map)





- All aligned contigs have an alignment in correct position (within 0.1% of true location)

Alignment Results:
(Error Free Optical Map)



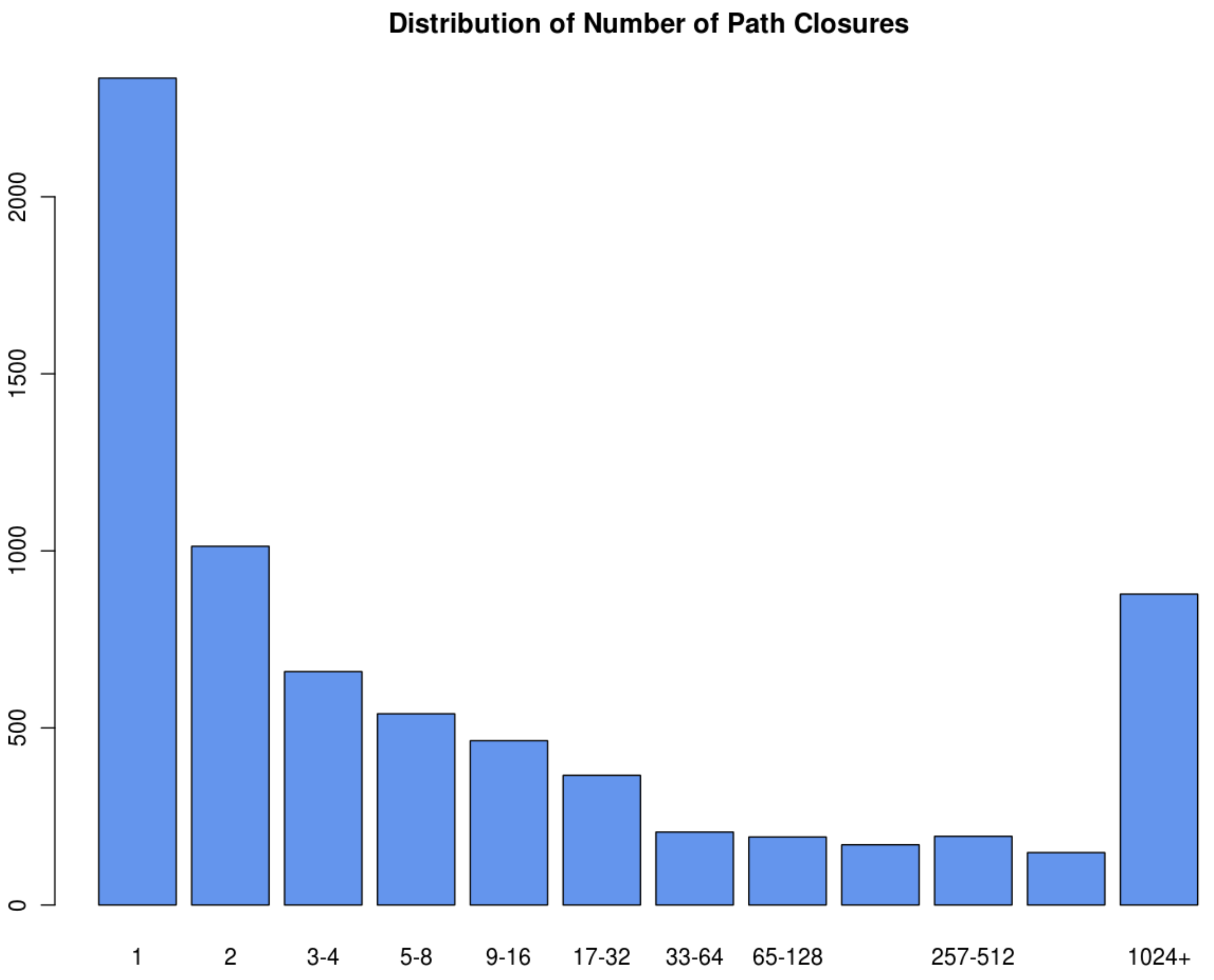
Outlier: Nocardia farcinica (NC_006361)

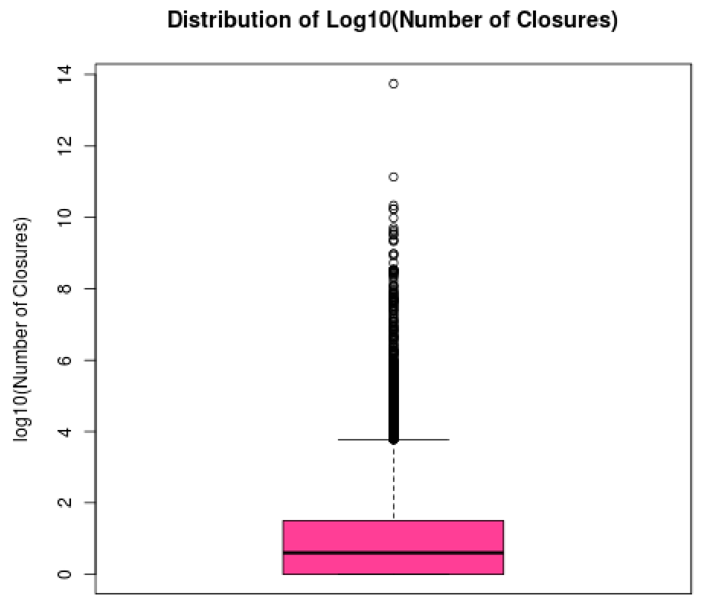
· Many contigs with “uninformative” restriction pattern

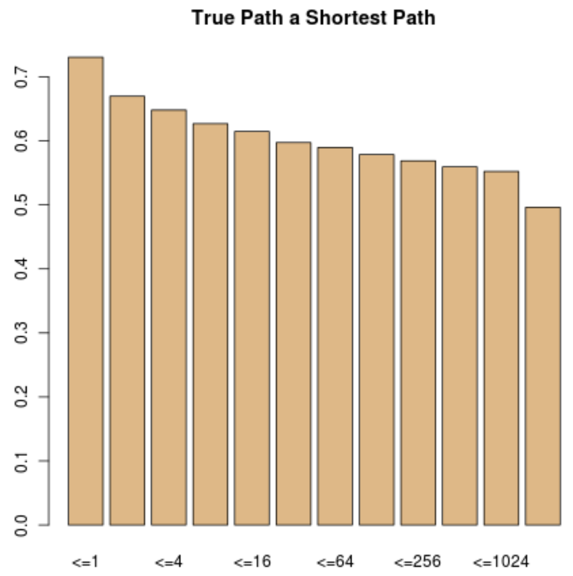
```
*********************************************
edge_40_88_0 311 0.0
/cbcb/project-scratch/lmendelo/debruijn/condor/no
                   Contig Frags | Optical Frags
                220 = 220   G;G | 4246 = 4246 G;G
                  12 = 12   C;G | 12 = 12   C;G
                  79 = 79       | 789 = 789
edge_40_88_0 311 0.0
/cbcb/project-scratch/lmendelo/debruijn/condor/no
                   Contig Frags | Optical Frags
                220 = 220   G;G | 1431 = 1431 G;G
                  12 = 12   C;G | 12 = 12   C;G
                  79 = 79       | 1223 = 1223
edge_40_88_0 311 0.0
/cbcb/project-scratch/lmendelo/debruijn/condor/no
                   Contig Frags | Optical Frags
                220 = 220   G;G | 5221 = 5221 G;G
                  12 = 12   C;G | 12 = 12   C;G
                  79 = 79       | 2664 = 2664
edge_40_88_0 311 0.0
/cbcb/project-scratch/lmendelo/debruijn/condor/no
                   Contig Frags | Optical Frags
                220 = 220   G;G | 702 = 702   G;G
                  12 = 12   C;G | 12 = 12   C;G
                  79 = 79       | 3076 = 3076
```

Number of Shortest Paths

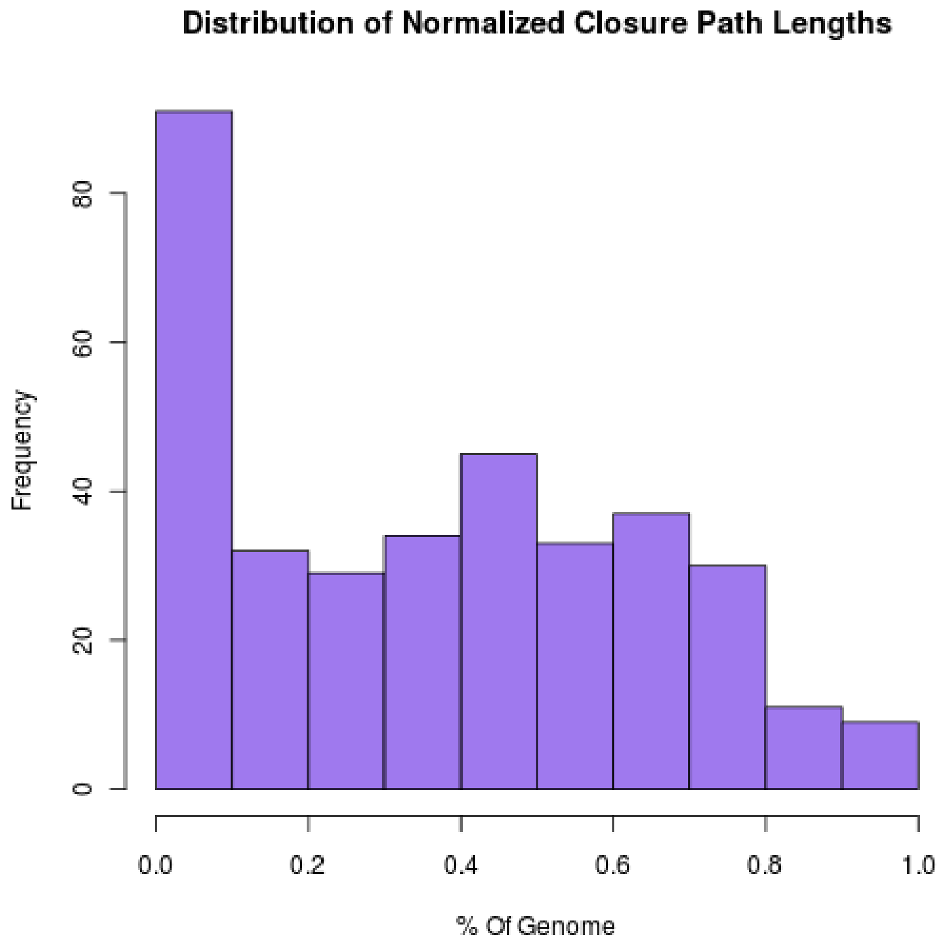
(Error Free Optical Map)

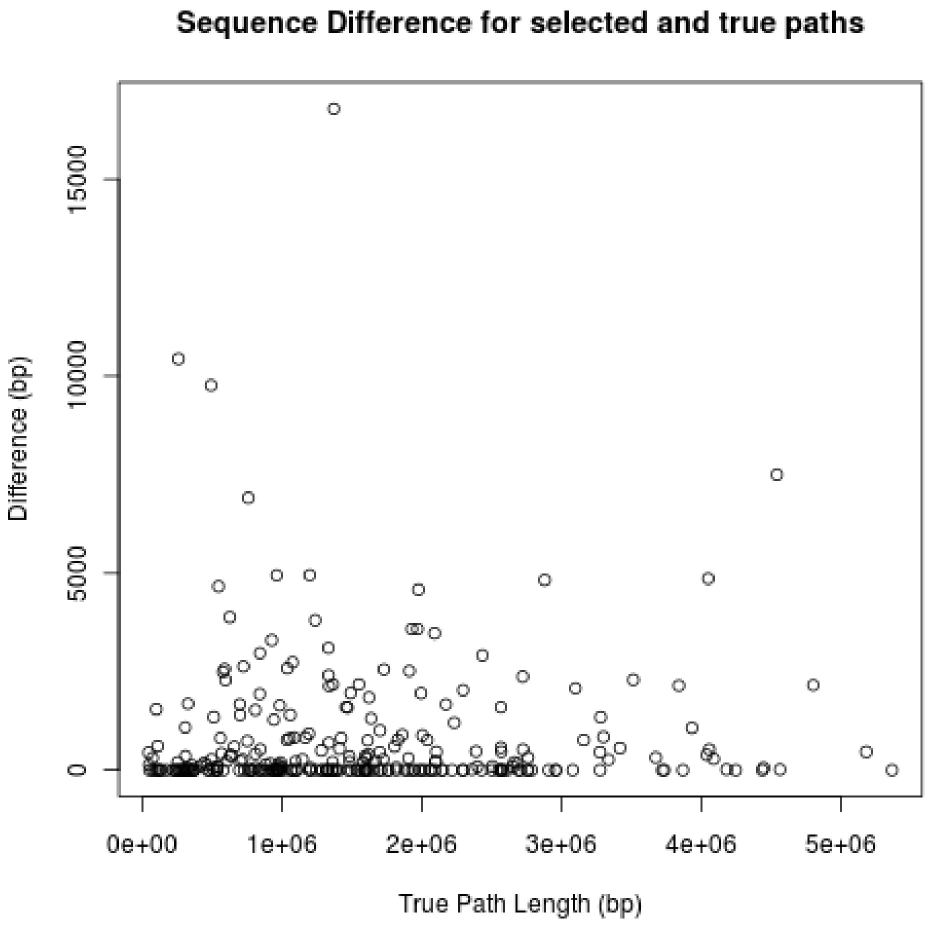




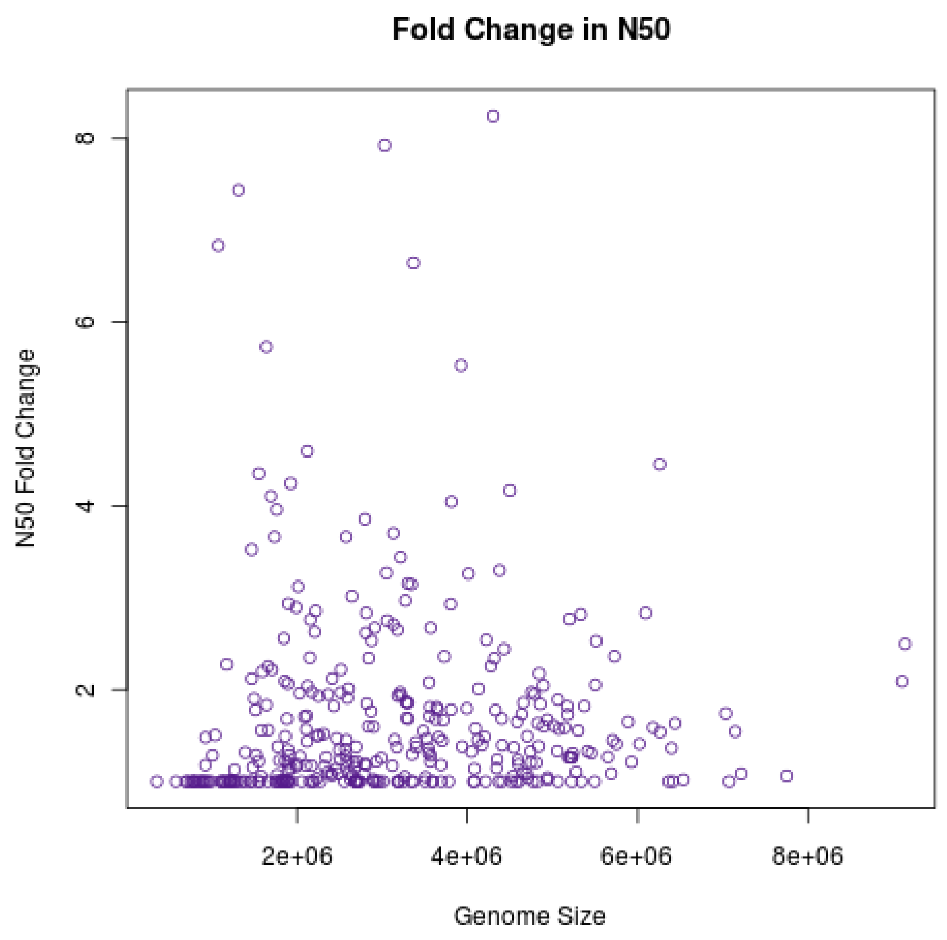


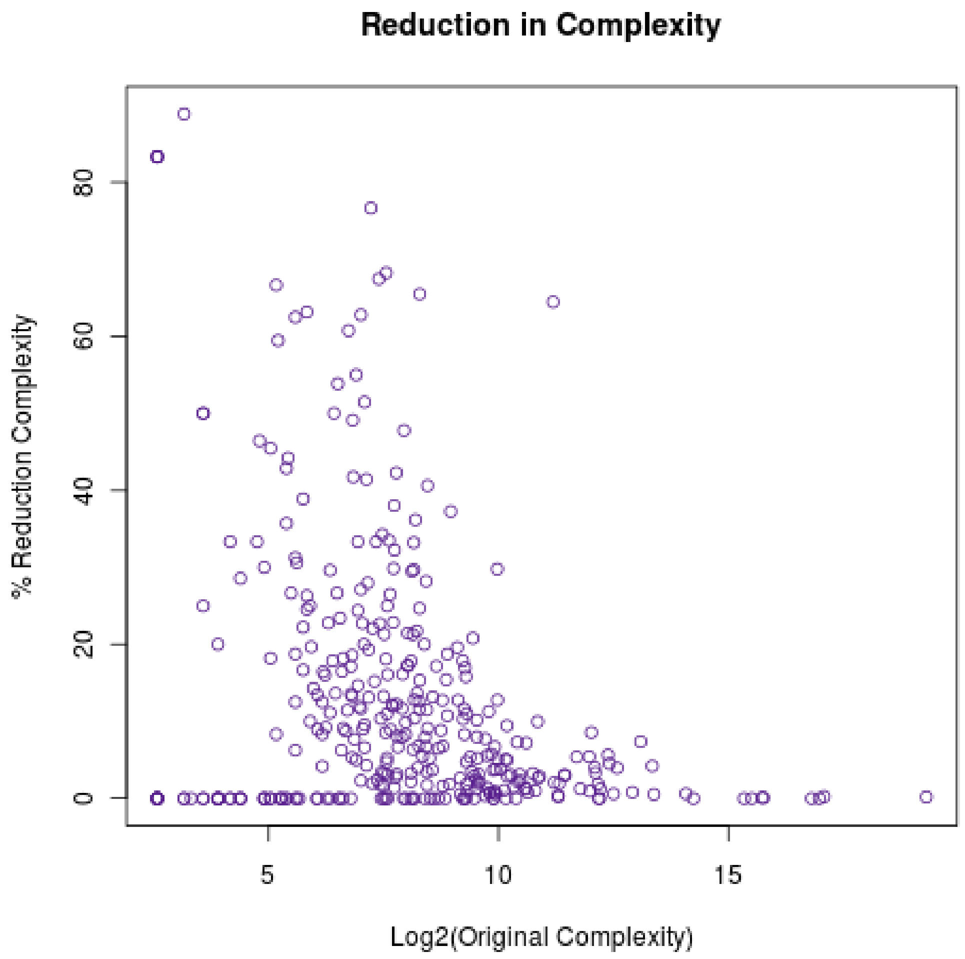
Accepted Path Closures
(Error Free Optical Map)





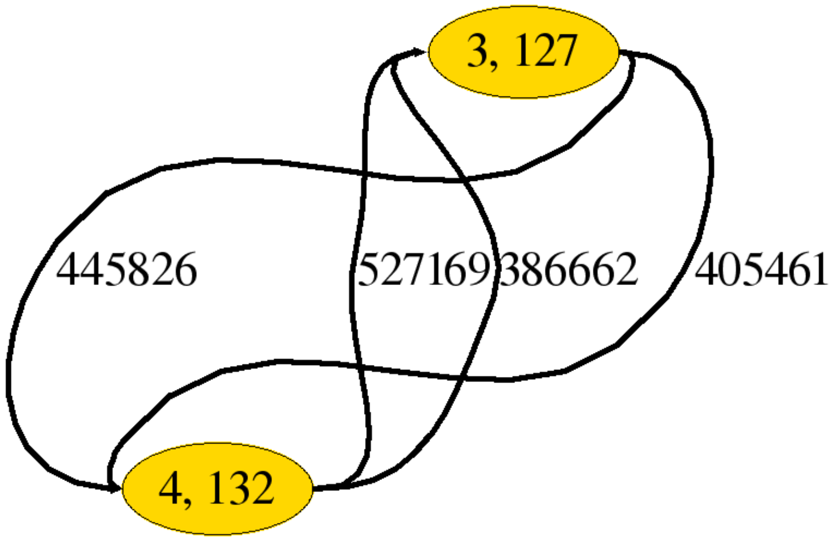
Improvements To Assembly
(Error Free Optical Map)



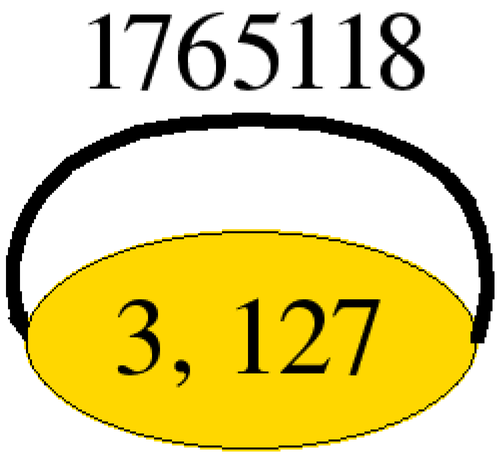


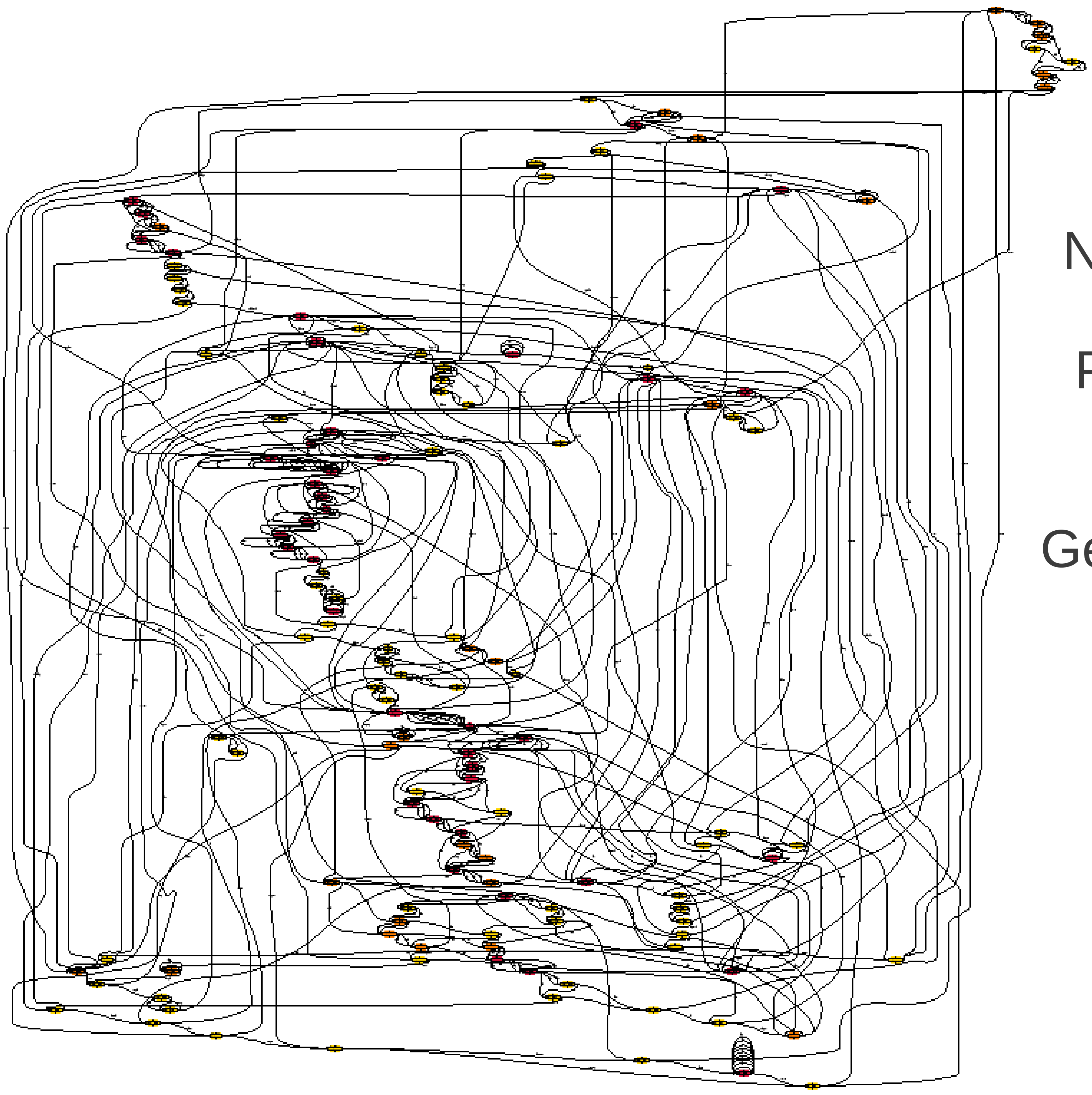
NC_000868 Pyrococcus abyssi
(Error Free Optical Map)

Original Graph



Final Graph



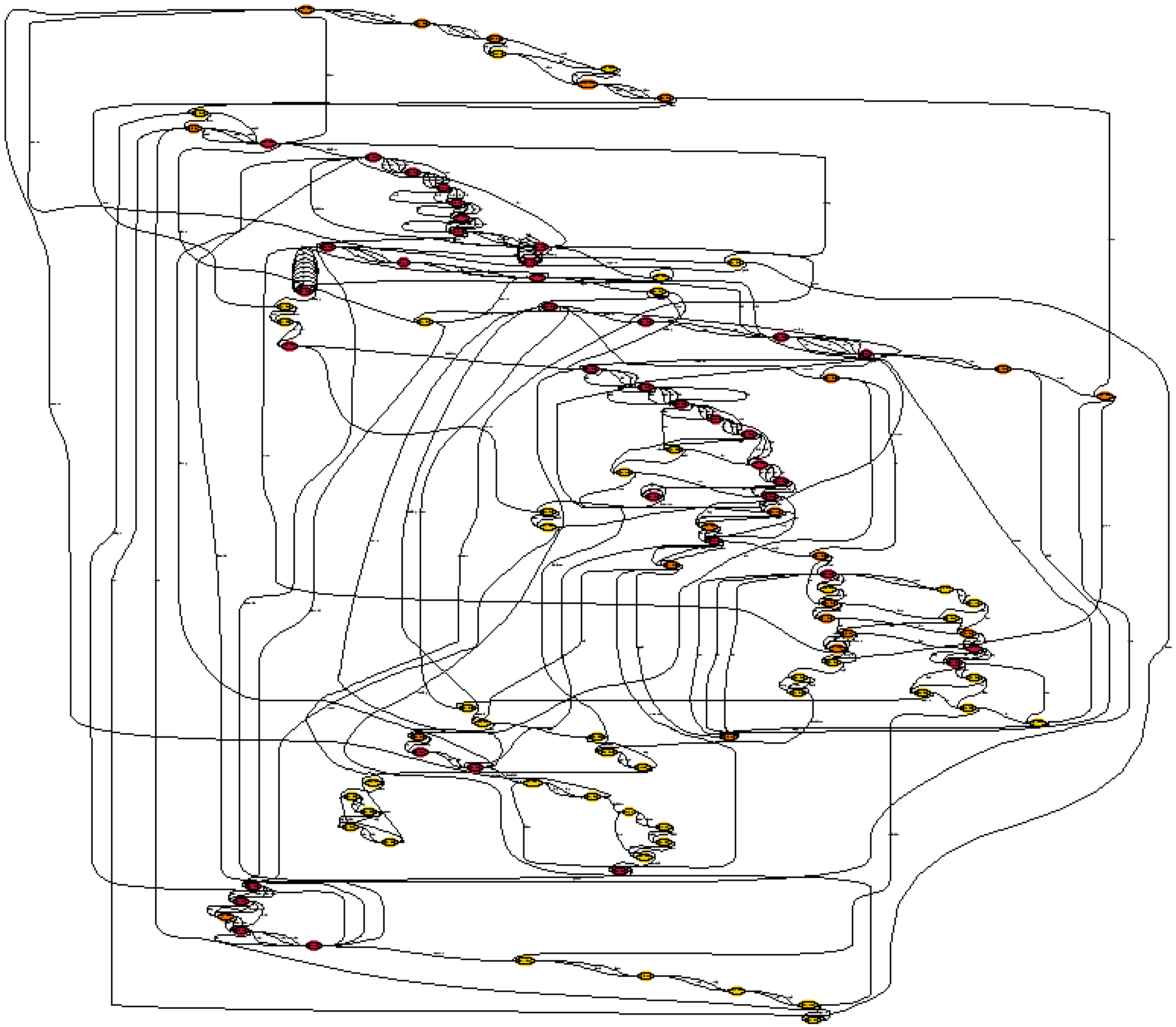


NC_005823

Pyrococcus abyssi

Genome size: 4.3 Mbp

Nodes: 134
Edges: 415
N50: 55,117
Complexity: 1007



NC_005823

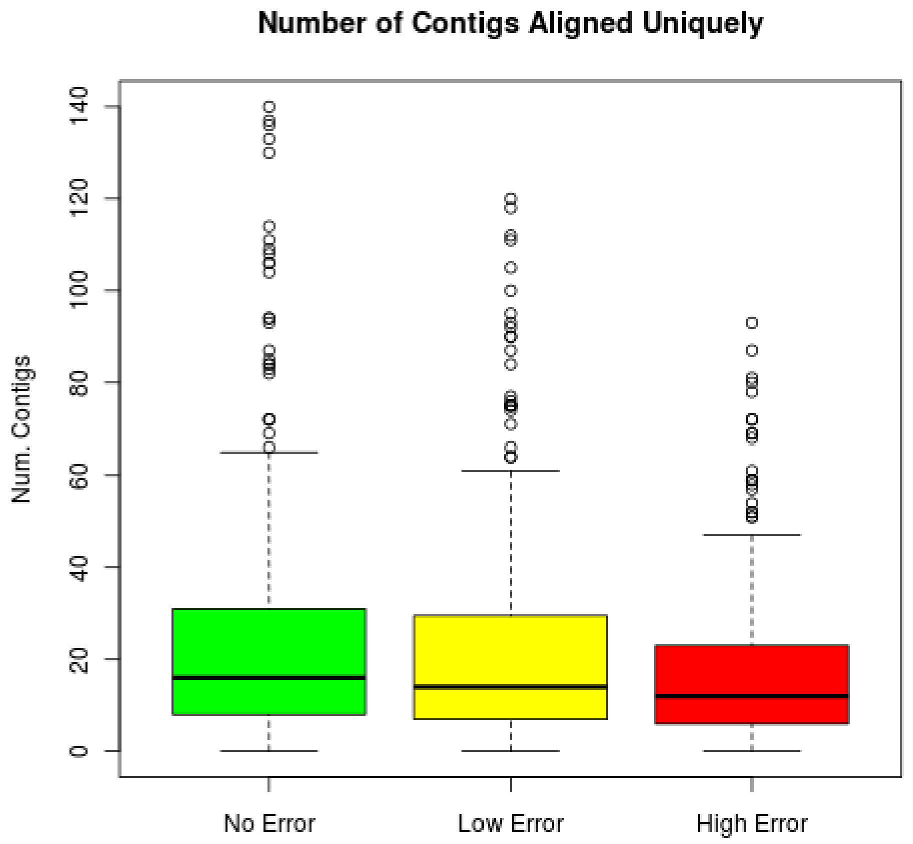
Pyrococcus abyssi

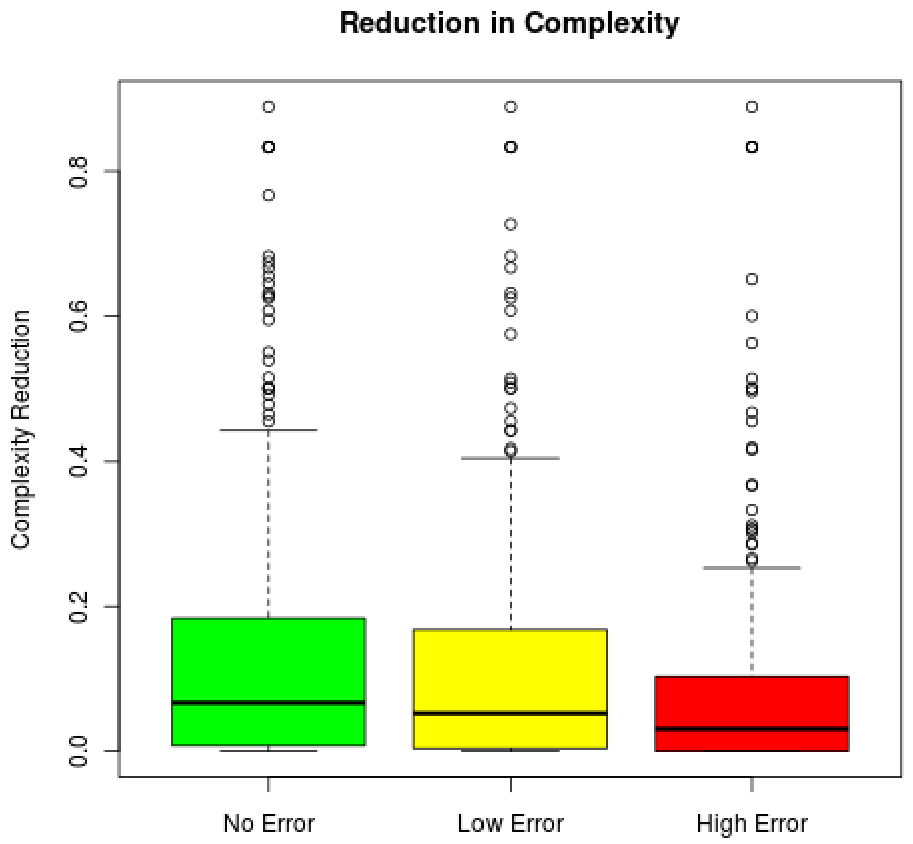
Genome size: 4.3 Mbp

Nodes: 104
Edges: 309
N50: 124,312
Complexity: 707

Incorrect: 2,373 out of 2,722,585

Results Across Error Settings

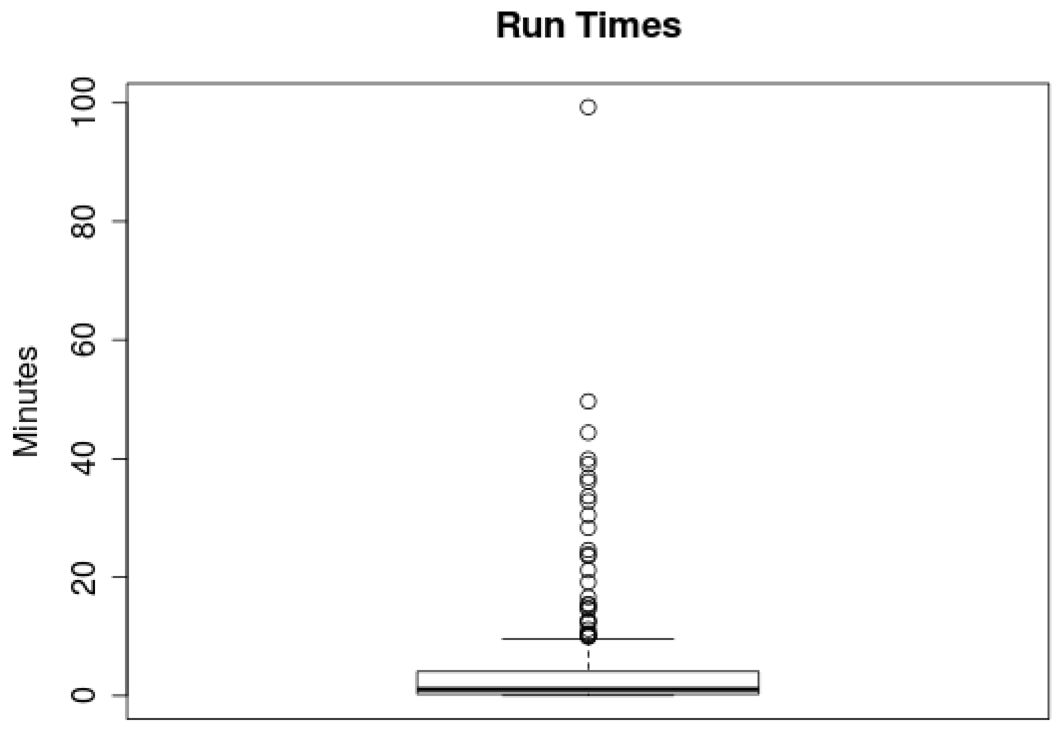




Run Times

CBCB Condor Cluster:
24 nodes
12 cores, 48 GB RAM

Mean run time ~ 4 minutes
Median run time ~ 1 minute



Conclusions & Potential Improvements

Conclusions

- Unique shortest path heuristic works well (when a unique shortest path exists).
- Many contigs are "unalignable" due to lack of restriction sites or uninformative restriction patterns.
- Most of the repeat structure of the genome is contained in a small fraction of the genome.

Potential Improvements

- Choose the most informative restriction enzyme for the genome.
- Use multiple rounds of contig alignment and graph simplification.
- Combine paired read information with optical maps.
- Use multiple optical maps.

Deliverables

- Source code for contig-optical map alignment tool
- Source code for graph simplification tool
- Source code for pipeline
- Log files & summary files for simulations
- Written report

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.

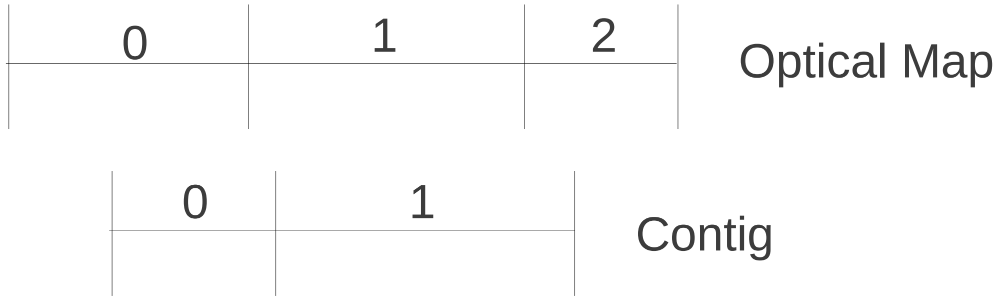
Schatz, M. C., Delcher, A. L., & Salzberg, S. L. (2010). Assembly of large genomes using second-generation sequencing. Genome research, 20(9), 1165-73.

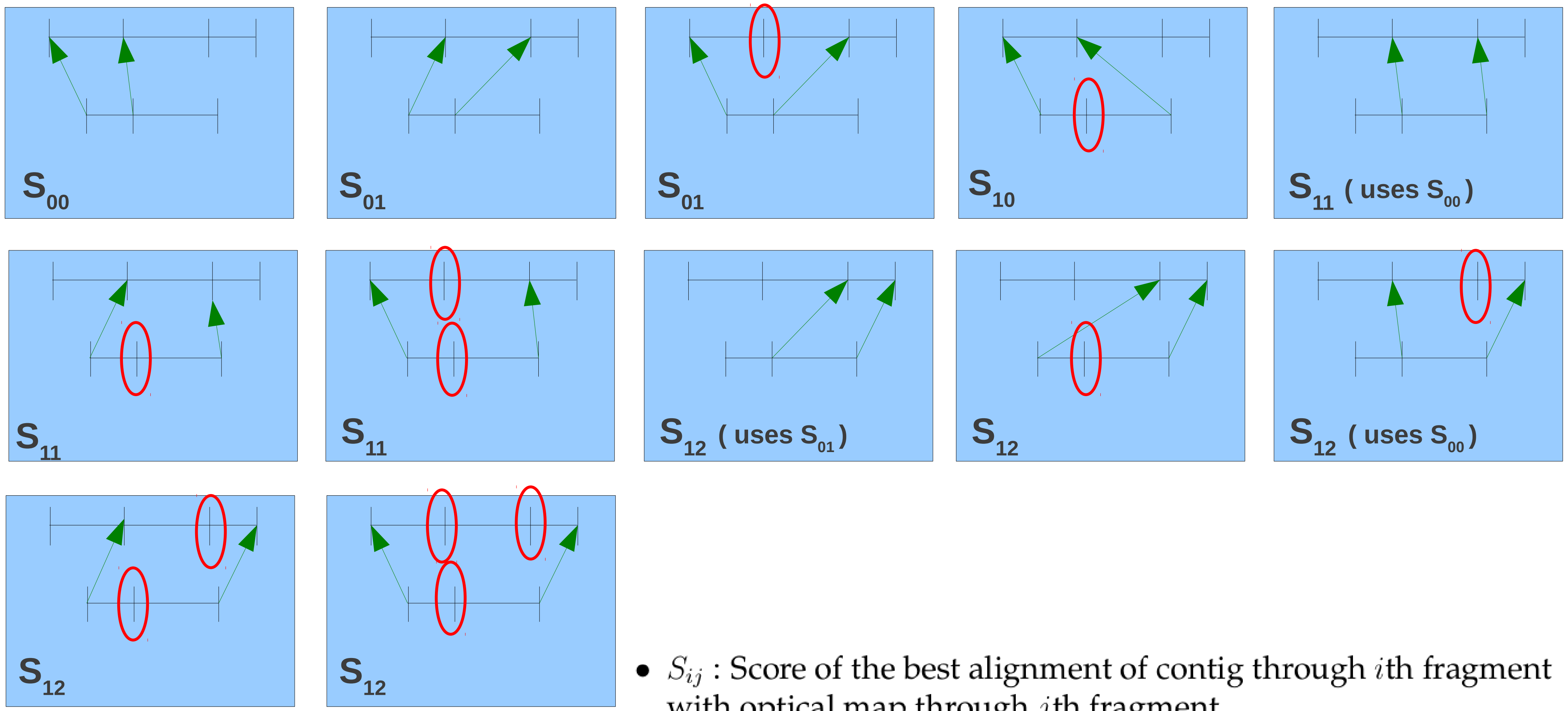
Valouev, A., Li, L., Liu, Y.-C., Schwartz, D. C., Yang, Y., Zhang, Y., & Waterman, M. S. (2006). Alignment of optical maps. Journal of Computational Biology, 13(2), 442-62. doi:10.1089/cmb.2006.13.442

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.

Alignment Algorithm

S_{00}	S_{01}	$\cancel{S_{02}}$
S_{10}	S_{11}	S_{12}




S_{00}
S_{01}
S_{01}
S_{10}
S_{11} (uses S_{00})
S_{11}
S_{11}
S_{12} (uses S_{01})
S_{12}
S_{12} (uses S_{00})
S_{12}
S_{12}

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