

# De Bruijn Graph Assembler

Kevin Boehme      Shaun Miller      Ken Reese

October 16, 2014

## 1 Methodology

Our error handling approach was quite simple. Once the reads were read in from the FASTA format and our algorithm broke them into the specified kmer length, we traversed the list of kmers looking for kmers that only appeared once. This effectively checked the support for each kmer in our list. If a kmer was found only once it would suggest a genotyping error and as such would be thrown out. If on the other hand, the kmer was supported multiple times, this suggests that it was correctly genotyped and was subsequently kept. Our algorithm currently does not attempt to bridge non-branching nodes.

## 2 Quality Analysis

In order to analyze the quality of our assembly we relied on a few, informative metrics including average contig size, N50, number of contigs, and the maximum contig size. These metrics provided a solid foundation for determining the quality of our assembly as we adjusted the kmer length as well as providing a tractable means of comparing our assembly to other De Bruijn graph assemblers (such as Velvet).

### Real Dataset, Small, klen = 42

Mean contig size:	382
N50:	1015
Number of Contigs	3
Largest Contig Size	1015

**3 Comparison**

**4 BLAST Results**

**5 Assembler Improvements**

**Appendix A — Charts**

**Appendix B — Notes**