Expandable de Novo Genome Assembler for Short-Read Sequence Data

Nikolay Vyahhi Sergey Nurk Anton Bankevich Max Alekseyev Pavel Pevzner

Algorithmic Biology Lab, Academic University of the Russian Academy of Sciences, St. Petersburg, Russia

http://bioinf.spbau.ru/en/assembler

Abstract

De novo genome sequence assembly is the essential step to reveal genomic sequences of different species world-wide. Currently there exists various genome assemblers for short-read NGS data, such as Velvet, SOAPdenovo, ALLPATH, ABySS and others.

We present new open-source de Bruijn graph-based assembler currently in development on C++, which uses novel algorithmic ideas such as context-free graph approach and also have agile and expandable software architecture. It requires affordable amount of memory and computations while giving high quality results. It provides solid basis for single-cell and mammalian assemblers in the near future.

In the Poster

1. De Bruijn graph construction, index and callbacks.
2. Error correction and corruption based on topology of the de Bruijn graph (e.g. tip clipping, bulge removal).
3. Efficient data structures for storing and handling of the additional information in the de Bruijn graph (e.g. coverage or distances between pairs of edges).

Graph Construction

- Vetexes: K-mers. Edges: $(K+1)$-mers.
- All simple pathes (i.e. without branchings) are condensed.
- Hash-index $(K+1)$-mer $\rightarrow (edge, offset)$.
- 2 bits per nucleotide in sequences.
- Every edge/vertex knows it's reverse-complementary edge/vertex.

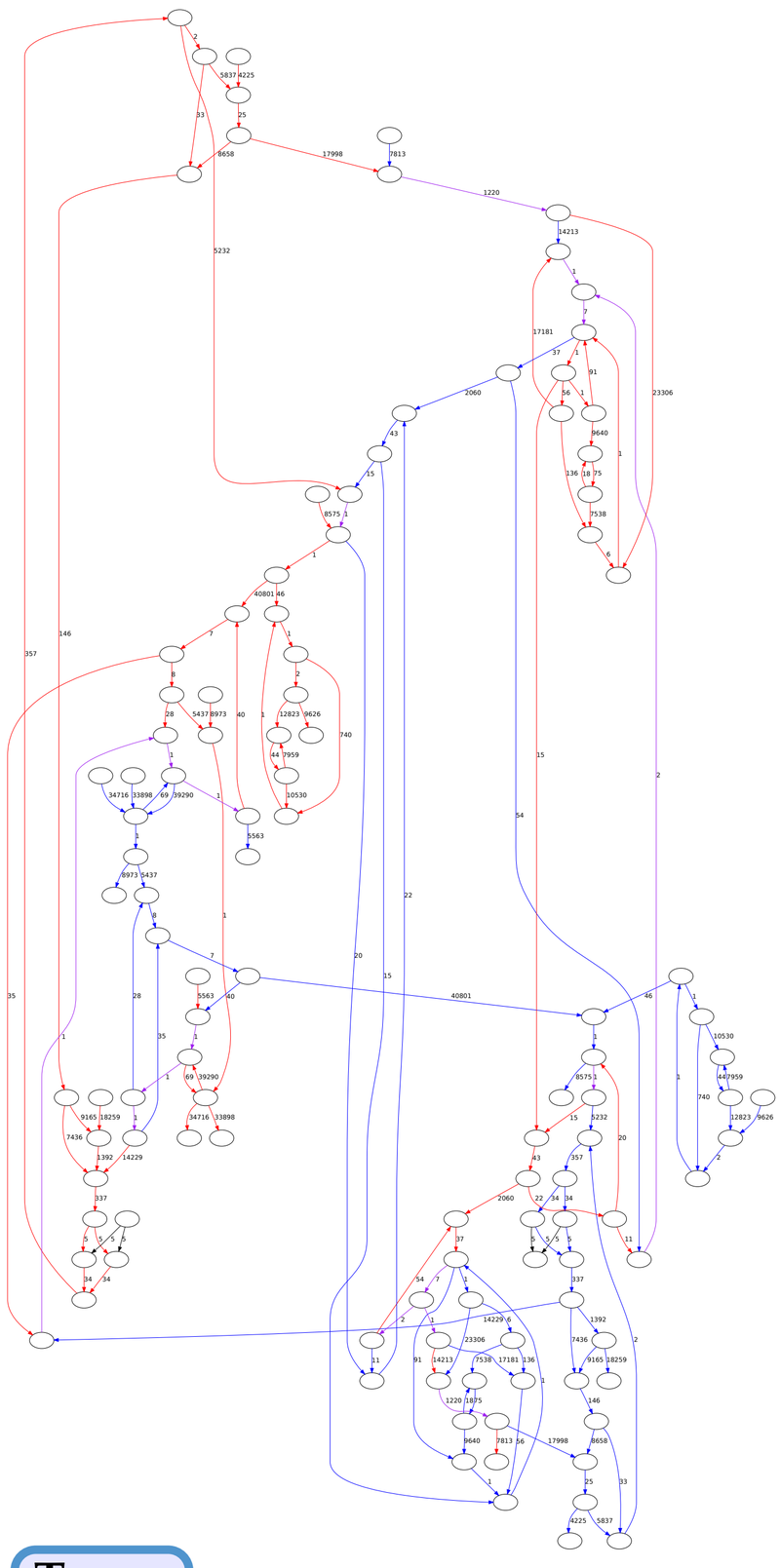
Graph Operations

- Addition/deletion of edge/vertex.
- Concatenation of a path.
- Split of an edge in a given position.
- Projection of one edge (with all additional info) to another edge.

Callback technique: each time a modifying operation is performed corresponding event is triggered and all objects that listen to graph events are notified about what exactly happened to the graph. It allows to implement any additional information as an external structure and hide its logic from the graph and from processing algorithms.

Errors Handling

- **Tip Clipping**.
  We iterate through all edges of the de Bruijn graph in order of increasing length.
  If current edge topologically looks like tip and has alternative path with good enough $coverage$ this edge will be removed.
- **Simple Bulge Removal**.
  Simple bulge is a rather short edge e with an alternative path P between its start and end of length $\approx length(e)$
  Our current approach is iterative removal of simple bulges. This strategy allows to delete pretty complex bulge structures and appeared to work pretty well if we alternate it with low coverage edge removal.
- **Removal of Low-Coverage Edges** with small length (as erroneous).



Additional Info

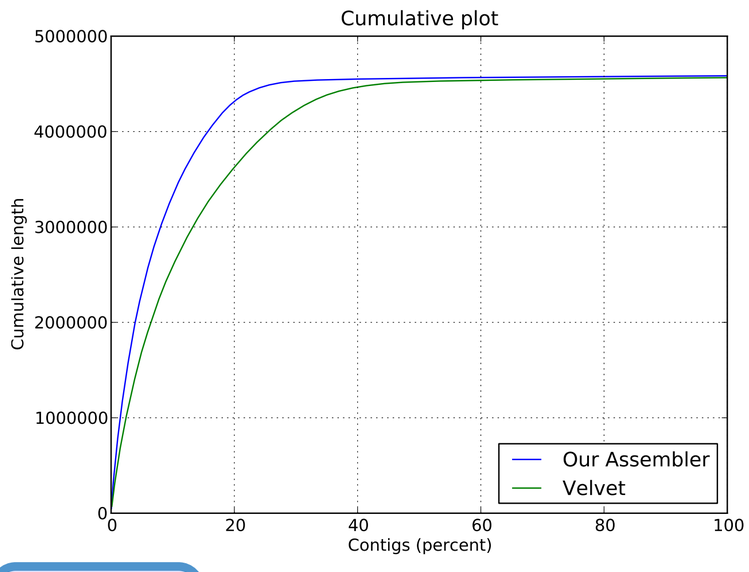
- **Coverage**.
  For edge it's the number of $(K+1)$-mers from this edge which was in initial genomic reads.
  Coverage is additive, means sum of coverages of two edges is the coverage of the concatenation of this two edges.
- **Distances between Edges** (usually takes from mate-reads or long reads).

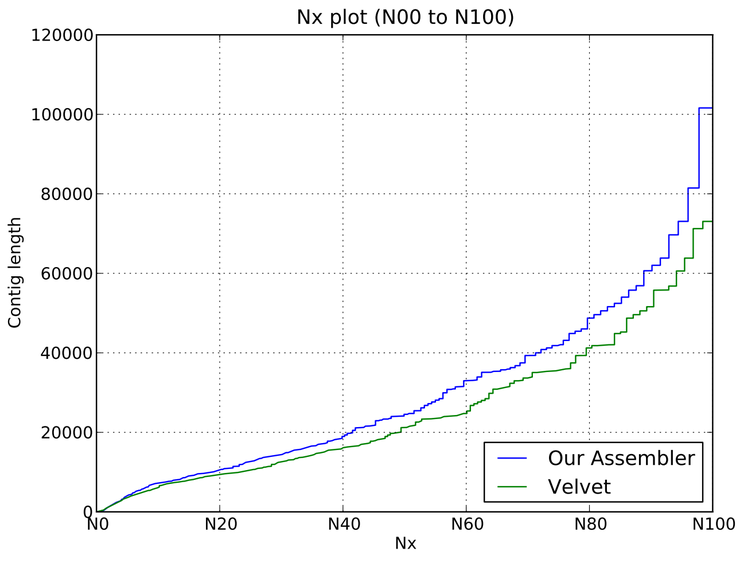
$$(edge, edge, distance) \rightarrow level\ of\ certainty\ (weight)$$

  Insert length has large dispersion value thus clustering of some sort is required. There are two types of clustering each of which are implemented: online clustering and offline clustering.

Evaluations

- E.Coli MG1655-K12 bacteria (4.6Mbp genome)
- 14M Illumina paired-end reads of size = 100bp, gap $\approx$ 20bp (total: 6 Gb of FASTQ).
- Error Correction with Quake (before assembly).
- $\approx$ 1 hour on a laptop, 1 Gb RAM, no HDD.





Future

0. Repeat resolving.
1. Single Cell (Bacterial) Assembler.
2. Mammalian Genomes Assembler (see the poster of *Mikhail Dvorkin* and *Alexander Kulikov*: Earmark graph approach to de novo genome assembly).
3. Transcriptome Assembler, Cancer Genomes Assembler and other customizations...

Team

Dmitrij Antipov, Anton Bankevich, Mikhail Dvorkin, Alexander Kulikov, Sergey Nurk, Alexander Sirotkin, Nikolay Vyahhi, Max Alekseyev, Pavel Pevzner.

Supported by "megagrant"
(Ministry of Science and Education, Russia).