MetagenomeScope

Web-Based Hierarchical Visualization of Metagenome Assembly Graphs

Marcus Fedarko[1, 2], Jay Ghurye[1, 2], Todd Treangen[2], and Mihai Pop[1, 2]

[1]Department of Computer Science, University of Maryland, College Park

[2]Center for Bioinformatics and Computational Biology, University of Maryland, College Park

mfedarko@umd.edu, jayg@cs.umd.edu, treangen@umd.edu, mpop@umiacs.umd.edu

http://ter.ps/e5r
Link to demo.

Motivation

Complexities such as repetitive sequences and sequencing errors can create branches and cycles in assembly graphs. These graphs thus often require manual examination after their generation to resolve ambiguous connections and correct assembly errors [6]. This has brought about the need for tools that can visualize assembly graphs effectively, displaying relevant biological metadata and graph structural information alike in a readily accessible manner. Furthermore, there is a documented dearth of hierarchical visualization tools that allow the user to navigate "from the large structure down to the base level [of the assembly graph]" [4].

To fulfill this need we present MetagenomeScope, an interactive web-based tool for the visualization of assembly graphs. MetagenomeScope contains a number of features intended to aid bioinformaticians in exploratory analysis of these graphs at both coarse and fine levels of complexity.

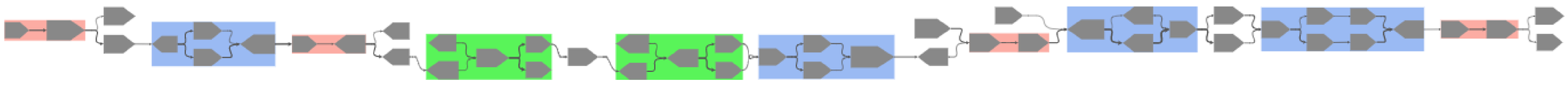


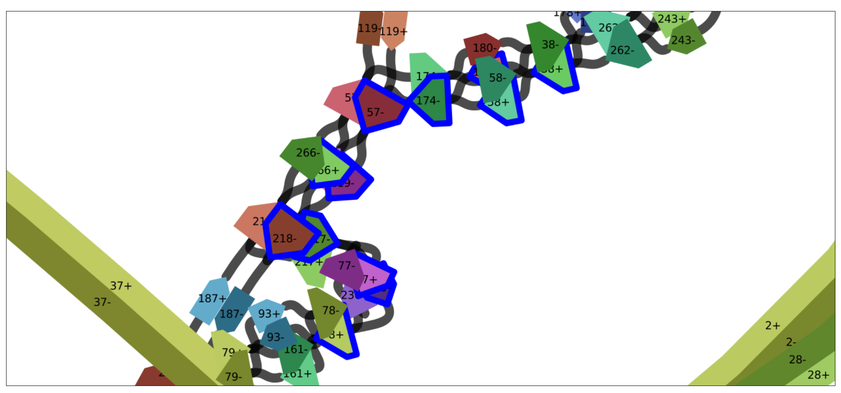
Figure 1: Seventeenth largest connected component of a human metagenome assembly graph (accession ID SRS049959), visualized in MetagenomeScope. Gray pentagons represent *contigs*, fragments of DNA pieced together by an assembler; edges correspond to overlaps between contigs.

Implementation and Availability

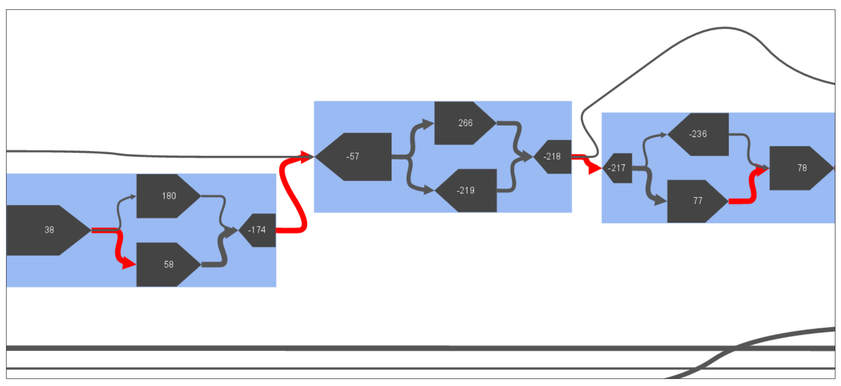
MetagenomeScope is composed of two software components—a **preprocessing script** (written in Python and C++) and a client-side **viewer interface**. The preprocessing script takes an assembly graph file as input and performs layout on its connected components using Graphviz' [1] *dot* and (for portions of the SPQR decomposition mode) *sfdp* layout tools, generating a SQLite3 database file that can be loaded in the viewer interface. The viewer interface uses Cytoscape.js [2] to visualize these graphs accordingly. Although MetagenomeScope was designed to be useful in a metagenomic context, the tool can also be used to visualize single-genome assembly graphs.

MetagenomeScope is licensed under the GNU General Public License, version 3.0. Its code is publicly available on GitHub at `https://github.com/marbl/MetagenomeScope`.

Related Work



(a) Screenshot captured in Bandage v0.8.1 of a specified region of an E. coli assembly graph. The graph was drawn as a "double graph" in Bandage's linear layout mode with otherwise default settings. Selected nodes are outlined in blue by Bandage.



(b) Screenshot captured in MetagenomeScope's "standard mode" of the same region of the same graph as in Fig. 2a. The collections of contigs highlighted in blue have been flagged as "bubbles," regions of the assembly graph exhibiting a converge→diverge→converge pattern [3, 5].

Figure 2: Screenshots of the same region of an E. coli assembly graph (`E_coli_LastGraph`) distributed with Bandage, rendered in Bandage (Fig. 2a) and MetagenomeScope (Fig. 2b). In both screenshots, the visualization interface has been zoomed to fit the collection of selected contigs 38+, 180+, 58+, 174-, 57-, 266+, 219-, 218-, 217-, 77+, 236-, and 78+. Bandage's depiction of the region is difficult to analyze, and does not yield much biological insight into the selected contigs aside from their relative size and orientation. The drawing produced by MetagenomeScope, however, makes clear these contigs' relations to each other in the assembly graph.

Features

Hierarchical layout

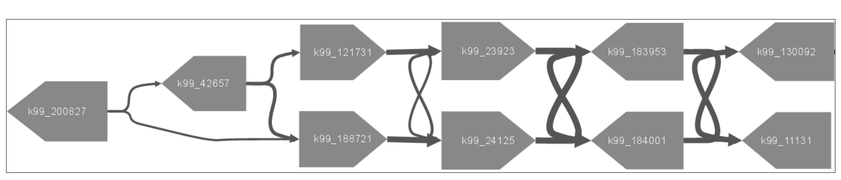
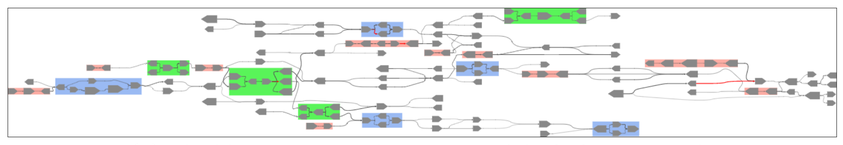
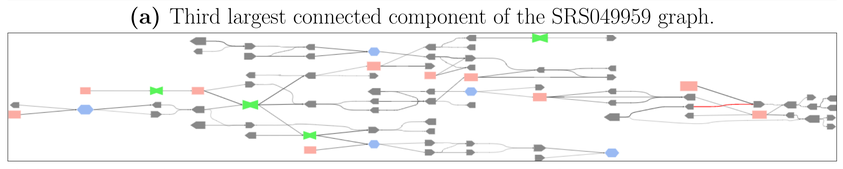


Figure 3: Screenshot of MetagenomeScope's standard mode, focused on another region of the SRS049959 graph. From a biological perspective, the intersecting paths through this region of the graph might indicate variation between two otherwise similar DNA samples contained in this metagenome.

Structural pattern layout grouping and highlighting



(a) Third largest connected component of the SRS049959 graph.



(b) Same region of the graph as in Fig. 4a, but with all identified structural patterns collapsed.

Figure 4: MetagenomeScope's standard mode visualizations of the third largest connected component of the SRS049959 graph. As is demonstrated in Fig. 4b, structural pattern collapsing can significantly reduce the visual complexity of the display, thus simplifying manual analysis.

Integrated scaffold display

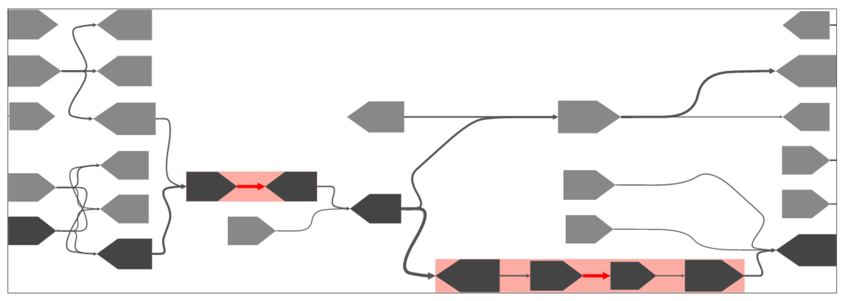
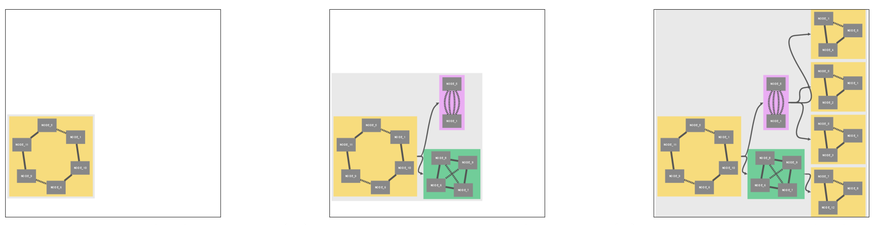


Figure 5: Region of a biofilm assembly graph visualized in MetagenomeScope. Contigs contained within a selected scaffold are colored darker than other contigs to indicate their selection status.

SPQR tree mode



(a) Fully collapsed SPQR tree. **(b)** Partially collapsed SPQR tree. **(c)** Fully uncollapsed SPQR tree.

Figure 6: Screenshots of a SPQR tree visualized in MetagenomeScope's "explicit" SPQR tree decomposition mode, in various states of collapsedness. Fig. 6a shows the tree as it is displayed upon first being drawn, collapsed to its root metanode. Fig. 6b shows the tree after its root metanode has been uncollapsed. Fig. 6c shows the tree after both children of the root metanode have been uncollapsed. Each additional uncollapsing operation reveals further information in the tree, indicating paths through the underlying biconnected component structure in increasing levels of detail. The graph shown here was taken from the Fig. 2a graph in [5].

Future Work

- Parallelizing layout operations across connected/biconnected components
- Using treemaps to indicate relative sizes of connected components
- Further utilizing SPQR tree decomposition to purposefully decompose regions of the graph spanning large horizontal distances in a linearized layout

Acknowledgements

The authors were supported in part by the NIH, grant R01-AI-100947, the NSF, grant IIS-1117247, and the Navy Research Laboratories, cooperative agreement N00173162C001, all to MP. Travel to this conference for MF was supported by the Rita Colwell Travel Fellowship.

This poster was designed using the "a0poster Portrait Poster" LaTeX template, available online at `https://latextemplates.com/template/a0poster-portrait-poster`.

References

[1] John Ellson, Emden Gansner, Lefteris Koutsofios, Stephen C. North, and Gordon Woodhull. *Graphviz— Open Source Graph Drawing Tools*, pages 483–484. Springer Berlin Heidelberg, Berlin, Heidelberg, 2002.

[2] Max Franz, Christian T. Lopes, Gerardo Huck, Yue Dong, Onur Sumer, and Gary D. Bader. Cytoscape.js: a graph theory library for visualisation and analysis. *Bioinformatics*, 32(2):309, 2016.

[3] Jason R. Miller, Sergey Koren, and Granger Sutton. Assembly algorithms for next-generation sequencing data. *Genomics*, 95(6):315–327, 2010.

[4] Gene Myers, Mihai Pop, Knut Reinert, and Tandy Warnow. Next Generation Sequencing (Dagstuhl Seminar 16351). *Dagstuhl Reports*, 6(8):91–130, 2017.

[5] Jurgen F. Nijkamp, Mihai Pop, Marcel J. T. Reinders, and Dick de Ridder. Exploring variation-aware contig graphs for (comparative) metagenomics using marygold. *Bioinformatics*, 29(22):2826, 2013.

[6] Adam M. Phillippy, Michael C. Schatz, and Mihai Pop. Genome assembly forensics: finding the elusive mis-assembly. *Genome Biology*, 9(3):R55, 2008.