Genome analysis

Bandage: interactive visualization of de novo genome assemblies

Ryan R. Wick1,*, Mark B. Schultz1, Justin Zobel2 and Kathryn E. Holt1

1Department of Biochemistry and Molecular Biology, Bio21 Molecular Science and Biotechnology Institute, University of Melbourne and 2Department of Computing and Information Systems, University of Melbourne, Parkville, Victoria, Australia

*To whom correspondence should be addressed.

Associate Editor: John Hancock

Received on April 20, 2015; revised on May 29, 2015; accepted on June 15, 2015

Abstract

Summary: Although de novo assembly graphs contain assembled contigs (nodes), the connections between those contigs (edges) are difficult for users to access. Bandage (a Bioinformatics Application for Navigating De novo Assembly Graphs Easily) is a tool for visualizing assembly graphs with connections. Users can zoom in to specific areas of the graph and interact with it by moving nodes, adding labels, changing colors and extracting sequences. BLAST searches can be performed within the Bandage graphical user interface and the hits are displayed as highlights in the graph. By displaying connections between contigs, Bandage presents new possibilities for analyzing de novo assemblies that are not possible through investigation of contigs alone.

Availability and implementation: Source code and binaries are freely available at https://github.com/rrwick/Bandage. Bandage is implemented in C++ and supported on Linux, OS X and Windows. A full feature list and screenshots are available at http://rrwick.github.io/Bandage.

Contact: rrwick@gmail.com

Supplementary information: Supplementary data are available at Bioinformatics online.

1 Introduction

Current de novo genome assemblers use graphs, most typically a de Bruijn graph. An ideal graph would contain one distinct path for each underlying sequence, but complexities such as repeated sequences usually prevent this. Instead, assembly graphs contain branching structures, where one node may lead into multiple others. The longest sequences in the graph that can be determined unambiguously are saved as contigs, which are often the final result of de novo assembly (Schatz et al., 2010). However, an assembly graph contains more information because it also has connections between sequences. It can therefore be advantageous to work with assembly graphs instead of contigs.

Bandage facilitates interaction with de Bruijn graphs made by de novo assemblers such as Velvet (Zerbino and Birney, 2008), SPAdes (Bankevich et al., 2012) and Trinity (Grabherr et al., 2011). It displays the graph in a graphical user interface (GUI) using a simple, comprehensible representation. The program is interactive, allowing users to zoom, pan and manually move nodes to focus on areas of interest.

2 Implementation and performance

Bandage is a GUI application written in C++ with Qt, giving it speed, memory efficiency and cross-platform portability. It runs on Linux, OS X and Windows. The Open Graph Drawing Framework library (http://www.ogdf.net/) is used to perform the graph layout using the fast multipole multilevel layout algorithm, which scales well for very large graphs (Hachul and Jünger, 2007).

On a 3 GHz laptop, Bandage can load and display the de novo assembly graph for a bacterial genome (5 Mb) in a few seconds, using <100 MB of RAM. A large graph with hundreds of thousands of nodes, such as a metagenome assembly (100 Mb or more), may take
Banding a visualization (a) Left, ideal bacterial assembly (single contig); right, poor assembly with many short contigs. (b) Left, zoomed-in view of Salmonella assembly; repeated sequences (blTEM and insertion sequence) appear as single nodes with multiple inputs and outputs. Node widths are scaled by read coverage (depth). Right, underlying gene structure deduced from Bandage visualization. (c) 16S rRNA region of a bacterial genome assembly graph, highlighted by Bandage’s integrated BLAST search. Nodes are labelled with their ID numbers and their widths are scaled by coverage. Even though the 16S gene failed to assemble into a single node, the user can manually reconstruct a complete dominant gene sequence from this succession of nodes: 175, 176, 64, 65 and 190.
Conflict of Interest: none declared.

References


