Bioinformatics, 32(21), 2016, 3321–3323
doi: 10.1093/bioinformatics/btw379
Advance Access Publication Date: 4 July 2016
Applications Note

OXFORD

Genome analysis

Icarus: visualizer for *de novo* assembly evaluation

Alla Mikheenko[1], Gleb Valin[2], Andrey Prjibelski[1], Vladislav Saveliev[1] and Alexey Gurevich[1,*]

[1]Center for Algorithmic Biotechnology, Institute of Translational Biomedicine, St. Petersburg State University, St. Petersburg, Russia, 199034 and [2] Department of Mathematics and Information Technology, St. Petersburg Academic University, Russian Academy of Sciences, St. Petersburg, Russia, 194021

*To whom correspondence should be addressed.

Associate Editor: Bonnie Berger

Received on April 13, 2016; revised on May 25, 2016; accepted on June 10, 2016

Abstract

Summary: Data visualization plays an increasingly important role in NGS data analysis. With advances in both sequencing and computational technologies, it has become a new bottleneck in genomics studies. Indeed, evaluation of *de novo* genome assemblies is one of the areas that can benefit from the visualization. However, even though multiple quality assessment methods are now available, existing visualization tools are hardly suitable for this purpose. Here, we present Icarus—a novel genome visualizer for accurate assessment and analysis of genomic draft assemblies, which is based on the tool QUAST. Icarus can be used in studies where a related reference genome is available, as well as for non-model organisms. The tool is available online and as a standalone application.
Availability and Implementation: http://cab.spbu.ru/software/icarus
Contact: aleksey.gurevich@spbu.ru
Supplementary information: Supplementary data are available at *Bioinformatics* online.

1 Introduction

Dozens of assembly algorithms have been developed in recent years, unfortunately none of them is perfect. Several evaluation studies (Bradnam *et al.*, 2013; Earl *et al.*, 2011; Magoc *et al.*, 2013; Salzberg *et al.*, 2012) showed that genome size, coverage depth, sequencing technology and many other aspects may dramatically affect quality of genome assembly. Thus, it is challenging to create a universal assembly approach, and an everyday tool for assembly evaluation and comparison is needed for choosing the best algorithm for a specific experiment.

QUAST (Gurevich *et al.*, 2013) is a commonly used tool for quality assessment of genome assembly. It computes a full range of assembly quality metrics such as N50, number of erroneous contigs, covered genes, etc. One of the main drawbacks of the tool is a lack of proper visualization of assemblies and their alignment to the reference genome.

Genome browsers have proven to be instrumental in genomic studies (Nielsen *et al.*, 2010). It is not a coincidence that there are now a great variety of interactive visualization tools, which include UCSC Genome Browser (Speir *et al.*, 2016), Integrative Genome Viewer (Robinson *et al.*, 2011) and many others. The most successful genome browsers are adapted to specific analysis goals, and it is unlikely that a universal tool for genomics analysis is practical (Nielsen *et al.*, 2010).

Special classes of interactive graphical tools are genome annotating sofware, such as Web Appolo (Lee *et al.*, 2013), and sequence finishing applications, such as Consed (Gordon *et al.*, 1998). These programs allow users not only to browse genome sequences, but also to edit them in place. However, the later examples are more complicated in use than regular genome browsers, and intented for experienced users. In addition, some of these tools are proprietary software, which also limits its usage by a wider community.

We present Icarus, a lightweight browser for draft assemblies exploration and evaluation. Icarus is integrated in QUAST and complements its output with interactive visualizations of assembly alignments to the reference genome, and various features detected

by QUAST (e.g. misassemblies). Icarus is an open-source software and it is freely available as a web tool and as a command-line utility.

2 Methods

Icarus pipeline consists of the following steps (see Supplementary Fig. 1):

1. Running QUAST: aligning contigs to the reference (if available), gene finding, detecting assembly errors;
2. Post-processing: detecting similarities between the assemblies;
3. Creating JavaScript-based web pages.

Icarus outputs two types of interactive HTML files: contig alignment viewer and contig size viewer. Each viewer contains at least two panes: an assembly overview with all contigs shown at full scale, and a pane where users can zoom into any region of interest. Both panes display all assemblies in several tracks next to each other. The viewers support standard genome browser functionality: navigating, dragging, zooming, clicking on elements for detailed information such as contig name, type and length. Specific features of each viewer are described below. Examples of Icarus output for assemblies of *B.impatiens*, *S.aureus* and CAMI (http://cami-challenge.org) metagenomic dataset are demonstrated in the Supplementary Material and on our website.

2.1 Contig alignment viewer

This type of viewer is available only if a reference genome is provided. If the genome consists of large chromosomes ($\geq$50 Mb), each sequence is displayed in a separate viewer. This is also true for multiple reference genomes (see Mikheenko *et al.*, 2016). An example of the viewer is given in Figure 1.

The contig alignment viewer places contigs according to their mapping to the reference genome produced by Nucmer aligner (Kurtz *et al.*, 2004). The color scheme is designed to differentiate between correct contigs (green or blue) and contigs with assembly errors (*misassemblies*, red or orange). Misassembled contigs are broken into correctly aligned blocks, so users can easily identify and switch between blocks of the same erroneous contig. The side of the block with the misassembly event is highlighted. Icarus supports all types of misassembly events detected by QUAST (relocations, inversions, etc). Users can show and hide all blocks containing specific types of misassemblies.

If several assemblies are provided, Icarus identifies all contigs that are similar in most of the assemblies and colors them in blue (for correct contigs) or orange (for misassembled ones). This feature helps researchers and assembly algorithms developers to see analogies between various assembly approaches. Details of Icarus similarity identification algorithm and examples of its performance are presented in the Supplementary Material.

The viewer can additionally visualize genes, operons and read coverage distribution along the genome using two additional tracks. The annotation track helps to understand which assembly contains more functional elements, and whether any of them are not covered by any assemblies at all. The coverage distribution track helps to monitor behaviour of assembly algorithms in regions of extremely high or low coverage.

2.2 Contig size viewer

This type of viewer shows contigs sorted by size in descending order. This ordering is suitable for comparing the largest contigs (which are the most interesting ones in most genomic studies). The viewer also labels the commonly used assembly quality statistics, N50 and N75. If the approximate genome length for the organism is known,

Fig. 1. An example of the contig alignment viewer for ABySS (Simpson *et al.*, 2009), SPAdes (Bankevich *et al.*, 2012) and Velvet (Zerbino and Birney, 2008) assemblies of *S.aureus* single-cell dataset from Chitsaz *et al.* (2011). The top grey panel shows Icarus controls for moving and zooming, and it also includes checkboxes for showing and hiding all types of detected misassemblies. The right grey panel presents details of the selected block. In this example, the highlighted block is a fragment of the misassembled contig *NODE_5*, which consists of three misassembled blocks. Users can expand detailed information on these blocks on this panel, or switch to any of them in the detailed view pane. The main viewer section is divided into four panels. From top to bottom: detailed assembly view, detailed read coverage (minimized to 'Show read coverage' button here), assembly overview and read coverage overview. Most of the contigs are correct (colored green and blue), but the assembly overview panel clearly shows that there are two regions where all three assemblers generate erroneous contigs (misassemblies, colored red and orange). One of these regions is depicted on the detailed assembly view. The orange color of ABySS and SPAdes misassembled blocks means that they have similar mappings to the reference. Worth noting that Velvet also made a misassembly at the same position, but its contig is shorter

NG50 and NG75 are also labelled (see Gurevich *et al.* (2013) for details). If the full reference genome is provided, the viewer shares the coloring scheme with the contig alignment viewer in order to highlight correctly mapped contigs (green blocks), unaligned contigs (grey blocks) and misassembled contigs (red blocks). Misassembly breakpoints are marked with dashed lines. All blocks are linked to their representations in the alignment viewer, allowing to quickly navigate between the viewers.

3 Conclusion

Inspired by genomic data browsing software, we developed Icarus—a browser for quality assessment and comparison of draft assemblies. Icarus engine is based on QUAST, a tool for de novo genome assembly evaluation, and complements its output with interactive visualizations of contig alignments, features and errors. As an open source project, our web-based and command-line tool can be easily utilized by the research community. We believe that Icarus will also help developers of assembly algorithms to improve their methods.

Acknowledgement

We would like to thank Shaun Jackman for substantial feedback on our software.

Funding

This work was supported by St. Petersburg State University, St. Petersburg, Russia [grant number 15.61.951.2015].

Conflict of Interest: none declared.

References

Bankevich,A. *et al.* (2012) SPAdes: a new genome assembly algorithm and its applications to single-cell sequencing. *J. Comput. Biol.*, **19**, 455–477.

Bradnam,K. *et al.* (2013) Assemblathon 2: evaluating de novo methods of genome assembly in three vertebrate species. *Gigascience*, **2**, 10.

Chitsaz,H. *et al.* (2011) Efficient de novo assembly of single-cell bacterial genomes from short-read data sets. *Nat. Biotechnol.*, **29**, 915–921.

Earl,D. *et al.* (2011) Assemblathon 1: a competitive assessment of de novo short read assembly methods. *Genome Res.*, **21**, 2224–2241.

Gordon,D. *et al.* (1998) Consed: a graphical tool for sequence finishing. *Genome Res.*, **8**, 195–202.

Gurevich,A. *et al.* (2013) QUAST: quality assessment tool for genome assemblies. *Bioinformatics*, **29**, 1072–1075.

Kurtz,S. *et al.* (2004) Versatile and open software for comparing large genomes. *Genome Biol.*, **5**, R12.

Lee,E. *et al.* (2013) Web Apollo: a web-based genomic annotation editing platform. *Genome Biol.*, **14**, R93.

Magoc,T. *et al.* (2013) GAGE-B: an evaluation of genome assemblers for bacterial organisms. *Bioinformatics*, **29**, 1718–1725.

Mikheenko,A. *et al.* (2016) MetaQUAST: evaluation of metagenome assemblies. *Bioinformatics*, **32**, 1088–1090.

Nielsen,C. *et al.* (2010) Visualizing genomes: techniques and challenges. *Nat. Methods*, **7**, S5–S15.

Robinson,J. *et al.* (2011) Integrative genomics viewer. *Nat. Biotechnol.*, **29**, 24–26.

Salzberg,S. *et al.* (2012) GAGE: A critical evaluation of genome assemblies and assembly algorithms. *Genome Res.*, **22**, 557–567.

Simpson,J. *et al.* (2009) ABySS: a parallel assembler for short read sequence data. *Genome Res.*, **19**, 1117–1123.

Speir,M. *et al.* (2016) The UCSC Genome Browser database: 2016 update. *Nucleic Acids Res.*, **44**, D717–D725.

Zerbino,D. and Birney,E. (2008) Velvet: algorithms for de novo short read assembly using de Bruijn graphs. *Genome Res.*, **18**, 821–829.