BIOINFORMATICS APPLICATIONS NOTE Vol. 29 no. 22 2013, pages 2936–2937
doi:10.1093/bioinformatics/btt515

Sequence analysis Advance Access publication August 31, 2013

Consed: a graphical editor for next-generation sequencing

David Gordon* and Phil Green

Department of Genome Sciences, University of Washington, Seattle, WA 98195, USA

Associate Editor: Michael Brudno

ABSTRACT

Summary: The rapid growth of DNA sequencing throughput in recent years implies that graphical interfaces for viewing and correcting errors must now handle large numbers of reads, efficiently pinpoint regions of interest and automate as many tasks as possible. We have adapted *consed* to reflect this. To allow full-feature editing of large datasets while keeping memory requirements low, we developed a viewer, *bamScape,* that reads billion-read BAM files, identifies and displays problem areas for user review and launches the *consed* graphical editor on user-selected regions, allowing, in addition to long-standing *consed* capabilities such as assembly editing, a variety of new features including direct editing of the reference sequence, variant and error detection, display of annotation tracks and the ability to simultaneously process a group of reads. Many batch processing capabilities have been added.

Availability: The consed package is free to academic, government and non-profit users, and licensed to others for a fee by the University of Washington. The current version (26.0) is available for linux, macosx and solaris systems or as C++ source code. It includes a user's manual (with exercises) and example datasets. http://www.phrap.org/consed/consed.html

Contact: dgordon@uw.edu

Supplementary information: Supplementary data are available at *Bioinformatics* online.

Received on June 13, 2013; revised on August 14, 2013; accepted on August 28, 2013

1 INTRODUCTION

DNA sequencing practice has changed dramatically in recent years. For many organisms, high-quality reference genomes are now available, and current research focuses on identifying variants in particular individuals via 'resequencing,' rather than *de novo* genome assembly. In addition, several high-throughput next-generation sequencing technologies, which generate reads that are generally shorter and/or less accurate than Sanger sequencing reads, have become widely available (Mardis, 2008).

In resequencing applications, newly derived reads are aligned to an existing reference using an alignment program, e.g. Li and Durbin (2009). When no reference is available, the reads must first be assembled into contigs using a *de novo* assembly program, e.g. Zerbino and Birney (2008). (In the following, we use 'assembly' to mean either read alignment to a reference or *de novo* assembly.) Although such programs are reasonably accurate, errors are not uncommon.

Much useful information can be extracted from an imperfect assembly, but accurate sequence is crucial for some applications. Mismapped reads and systematic base-calling errors can confuse SNP-calling programs. Accurate sequence is also needed to reliably identify mutations and when biological features in the sequenced region will be the subject of significant experimental work. As a result, finishing—the process of identifying and correcting assembly errors—remains an important component of many sequencing projects. Reliable finishing generally requires some viewing and manipulation of the assembly by the researcher using a graphical assembly editing program (Dear and Staden, 1991). Several programs allow viewing but not manipulating the assembly, e.g. Schatz *et al.* (2007) and Thorvaldsdottir *et al.* (2013), whereas others allow automated but not manually directed finishing, e.g. Swain *et al.* (2012). To our knowledge, GAP5 (Bonfield and Whitwham, 2010) and *consed* are the only programs that allow editing billion-read assemblies.

Consed (Gordon *et al.*, 1998) was originally developed in the era of Sanger sequencing and used in finishing the human and other genomes; it continues to be widely used, with ~190 citations per year and ~1000 downloads per year. Here we describe recent adaptations to handle large datasets of next-generation reads, as well as new viewing and searching features unrelated to finishing.

2 METHODS

Consed's graphical editor (Gordon, 2003; Gordon *et al.*, 1998), the centerpiece of the *consed* package, has a rich and flexible set of editing and analysis features whose full functionality requires memory-intensive data structures. We have improved its resource management to reasonably handle up to a few million reads (e.g. a 1.7 million-read dataset requires 2.7 GB RAM and 1.5 min for start-up on a desk station), sufficient for most bacterial genomes. To deal with larger assemblies, we implemented a two-step approach in which a limited-feature viewer, *bamScape*, that readily handles several billion reads, is used to identify regions of interest and then to launch *consed's* graphical editor (with full editing and analysis capability) on read sets extracted from these regions.

BamScape takes as input a reference sequence and a BAM (Li *et al.*, 2009) file of reads aligned to it. Resource requirements are modest, e.g. ~300 MB RAM and 10 s to start up and display an 800 KB region from a BAM file of 2 billion human reads, and <5 s to jump to other locations. The Reads vs Reference Window (Supplementary Fig. S1) displays read depth, depth of inconsistent mate pairs (i.e. those with anomalous relative orientation or map location) and read-reference discrepancy rate. Potential 'problem sites' (misassemblies or sequence variants) are found using user-defined thresholds for these variables (Supplementary Fig. S2). Problem sites are added to an interactive list (i.e. such that clicking on a list item scrolls the window to that location). At any region of interest, the user can click to bring up *consed's* graphical editor (taking e.g. ~12 s for a

*To whom correspondence should be addressed.

~20 KB region of coverage depth ~25×), examine the read data in greater detail and, if desired, edit the reference sequence or the assembly in these regions. (As *consed's* graphical editor is restricted to datasets of a few million reads, this method can currently fix misassemblies of regions at most this size.) Edits to various locations in the reference may be made in one or several editing sessions, and *consed* can create a new version of the reference that reflects all edits. This can then be used with a read alignment program to create a new BAM alignment file.

Tracks [as in the UCSC Genome Browser (Kent *et al*., 2002)] can be shown in the Aligned Reads Window (Supplementary Fig. S3) either as a graph, as genes with indications of untranslated regions, introns and the translated amino acid sequence or as bars with grayscale indicating quantitative data (e.g. conservation scores).

A *tag* is a label attached to a region of a read or reference sequence (Gordon *et al*., 1998). We have expanded the tag feature to allow comments, user-defined tag types and user-specified tag fields that may contain numbers, text or references to other tags. *Consed* has flexible tag-search capabilities and generates interactive lists of the tags that are found.

Consed detects putative SNPs and indel polymorphisms and calculates genotype qualities using the method of Li *et al*. (2008), producing an interactive list that allows putative variants to be viewed along with their supporting read data. A VCF format (Danecek *et al*., 2011) report can also be generated in batch.

The *consed* graphical editor provides several capabilities (alternative to those in *bamScape*) for misassembly detection and correction. The Assembly View Window (Supplementary Fig. S4) (Nielsen *et al*., 2010) gives a close-up view of potentially misassembled regions showing the order and orientation of contigs in scaffolds, read depth, clusters of inconsistent mate pairs and sequence similarity. The Highly Discrepant Positions Window (Supplementary Fig. S5) shows an interactive list of locations where multiple reads disagree with the consensus or reference sequence. An interactive list of high- (or low-) depth coverage regions can be generated.

The user can select a read to be placed near the top of the Aligned Reads Window (Supplementary Fig. S3) and determine, by visually comparing it to other reads at informative sites, which reads belong with it and which do not; the latter can then be moved to another location. Reads can be removed from the contig by right-clicking on read names, by clicking on inconsistent mate pairs in Assembly View, by highlighting read names (see later in the text) and requesting that highlighted reads be removed or by supplying a list of reads to a batch program. A group of reads can be removed together into a single contig that preserves alignments, removed individually into separate contigs for each read, or deleted entirely from the assembly. Optionally, mates of reads can also be removed. The removed reads can either be added to a different location using the join feature (Gordon *et al*., 2001) or reassembled by clicking 'miniassembly'.

Contig joins can be made in any of three ways: manually, using sequence similarity as shown in Assembly View or as found with the search-for-string feature, by clicking to display the alignment in the Compare Contigs Window (Supplementary Fig. S6, bottom) and then clicking 'join'; in semiautomated fashion using an interactive list of potential joins (Supplementary Fig. S6, top) generated in batch by *consed's autoreport* program; or in fully automated batch mode accepting all joins recommended by *autoreport*.

False joins can be corrected using the tear function (Gordon *et al*., 2001), which now allows the user to sort the reads into two new contigs while looking at base discrepancies.

Consed can pick PCR or walking primers for closing gaps between contigs (either under manual control or in batch mode). Once new reads have been generated, a button click or a script running in batch adds them to the assembly by running *cross_match* to find the best gapped alignment of each read against the existing consensus sequence, incorporating the read at the aligned location. Optionally, reads can be targeted to an approximate location. An interactive list of newly added reads is displayed. A batch feature corrects and extends the contig sequences where appropriate.

Consensus bases and base qualities (Ewing and Green, 1998) can optionally be recalculated following changes to the assembly. The interactive list 'questionable consensus bases' indicates potential errors in the consensus sequence (identified using a choice of three different algorithms involving read frequency, quality and strand) for user review and editing. The consensus can be trimmed.

Reads can be grouped by highlighting their names and then operated on as a group. Highlighting can be done in various ways (e.g. by clicking on read names or by specifying a sequence at a particular location). All reads at a particular reference position can be edited at once to have the same base. All read bases to the left (or right) of a particular position can be changed to X's (indicating vector).

Additional new features are described in Supplemental information.

ACKNOWLEDGEMENTS

The work described here reflects suggestions, ideas and requests from many users including Amr Abouelleil, Don Bovee, Yaron Butterfield, Bob Fulton, Eugene Goltsman, Don Hutchison, Sven Klages, Jim Knight, Rachel Maupin, Mike McLellan, Margaret Priest, Jeremy Schmutz, Lori Spiegel, Cindy Strong, Jeremy Wheeler and Shiaw-Pyng Yang. We particularly thank Pat Minx and Aye Wollam for extensive help over many years.

Funding: National Institutes of Health grant R01HG005710.

Conflict of interest: *Consed* is licensed for a fee to commercial users by the authors' employer (University of Washington).

REFERENCES

Bonfield,J.K. and Whitwham,A. (2010) Gap5—editing the billion fragment sequence assembly. *Bioinformatics*, **26**, 1699–1703.

Danecek,P. *et al*. (2011) The variant call format and VCFtools. *Bioinformatics*, **27**, 2156–2158.

Dear,S. and Staden,R. (1991) A sequence assembly and editing program for efficient management of large projects. *Nucleic Acids Res*., **19**, 3907–3911.

Ewing,B. and Green,P. (1998) Base-calling of automated sequencer traces using phred. II. Error probabilities. *Genome Res*., **8**, 186–194.

Gordon,D. (2003) Viewing and editing assembled sequences using Consed. *Curr. Protoc. Bioinformatics*, Chapter 11, Unit 11.12.

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

Gordon,D. *et al*. (2001) Automated finishing with autofinish. *Genome Res*., **11**, 614–625.

Kent,W.J. *et al*. (2002) The human genome browser at UCSC. *Genome Res*., **12**, 996–1006.

Li,H. and Durbin,R. (2009) Fast and accurate short read alignment with Burrows-Wheeler transform. *Bioinformatics*, **25**, 1754–1760.

Li,H. *et al*. (2009) The sequence alignment/map format and SAMtools. *Bioinformatics*, **25**, 2078–2079.

Li,H. *et al*. (2008) Mapping short DNA sequencing reads and calling variants using mapping quality scores. *Genome Res*., **18**, 1851–1858.

Mardis,E.R. (2008) Next-generation DNA sequencing methods. *Annu. Rev. Genomics Hum. Genet*., **9**, 387–402.

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

Schatz,M.C. *et al*. (2007) Hawkeye: an interactive visual analytics tool for genome assemblies. *Genome Biol*., **8**, R34.

Swain,M.T. *et al*. (2012) A post-assembly genome-improvement toolkit (PAGIT) to obtain annotated genomes from contigs. *Nat. Protoc*., **7**, 1260–1284.

Thorvaldsdottir,H. *et al*. (2013) Integrative Genomics Viewer (IGV): high-performance genomics data visualization and exploration. *Brief. Bioinform*., **14**, 178–192.

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