Sequence analysis

NGSView: an extensible open source editor for next-generation sequencing data

Erik Arner*, Yoshihide Hayashizaki and Carsten O. Daub

ABSTRACT

Summary: High-throughput sequencing technologies introduce novel demands on tools available for data analysis. We have developed NGSView (Next Generation Sequence View), a generally applicable, flexible and extensible next-generation sequence alignment editor. The software allows for visualization and manipulation of millions of sequences simultaneously on a desktop computer, through a graphical interface. NGSView is available under an open source license and can be extended through a well documented API.

Availability: http://ngsview.sourceforge.net

Contact: arner@gsc.riken.jp

1 INTRODUCTION

The emergence of next-generation sequencing platforms (Holt and Jones, 2008, Shendure and Ji, 2008) imposes increasing demands on the bioinformatics methods and software. High-throughput sequencing technologies introduce novel demands on tools available for data analysis. We have developed NGSView (Next Generation Sequence View), a generally applicable, flexible and extensible next-generation sequence alignment editor. The software allows for visualization and manipulation of millions of sequences simultaneously on a desktop computer, through a graphical interface. NGSView is available under an open source license and can be extended through a well documented API.

Availability: http://ngsview.sourceforge.net

Contact: arner@gsc.riken.jp

2 METHODS

NGSView is an extension of DNPtrapper (Arner et al., 2006), our previously developed alignment editor designed for analysis of Sanger reads from complex repeat regions. The code has been extensively refactored in order to meet the requirements of next generation sequencing data. It is implemented in C++ using the Qt (http://www.qtsoftware.com/) GUI toolkit for visualization and Berkeley DB (http://www.oracle.com/) as the back end database. The RAM required to run the software is low and independent of project size. This is achieved by reading data from disk at request rather than keeping data cached in the main application; a layered database design also ensures that disk access lag is kept at a minimum.

The fast pace of development and increasingly shorter time-to-market for sequencing platforms furthermore introduces additional demands on software generality, flexibility and extensibility. In order to avoid a lag between sequencing technology development and available analysis methods, it is a great advantage if existing tools are sufficiently general and easy to modify for a re-adaptation to appearing technologies.

Recently, tools specifically designed for visualizing next-generation sequencing data have been introduced (Bao et al., 2009; Huang and Marth, 2008). While these applications go a long way toward fulfilling the visualization needs of next-generation sequencing projects, they lack either in generality, flexibility or extensibility—they have strong couplings to specific sequencing platforms or limits in the amount of data they can handle, they offer limited means of editing and manipulating data, and their source code are either closed source or lack a well-defined application program interface (API).

We introduce NGSView (Next Generation Sequence View), an open source alignment editor and visualization tool, designed to address the issues mentioned above. It provides generality in being able to handle sequence data of any format and virtually any size, flexibility in allowing extensive editing options in addition to visualization and extensibility by being released under an open source license with a well-documented API. Using NGSView, it is possible to very quickly go from a zoomed in sequence level view to a zoomed out view of an entire chromosome, and editing operations can be performed on any subset of reads defined by the user.

*To whom correspondence should be addressed.

© The Author(s) 2009. Published by Oxford University Press. This is an Open Access article distributed under the terms of the Creative Commons Attribution Non-Commercial License (http://creativecommons.org/licenses/by-nc/2.5/uk/) which permits unrestricted non-commercial use, distribution, and reproduction in any medium, provided the original work is properly cited.

Downloaded from https://academic.oup.com/bioinformatics/article-abstract/26/1/125/182206 by guest on 30 April 2019
3 RESULTS

Here, we introduce the key elements of NGSView. Additional detailed documentation including screen shots is available at the software web site listed above.

3.1 Generality

NGSView is a general sequence viewer in the sense that it assumes very little about the sequencing platform(s) used in a project, and simultaneously can handle sequence data of a wide range of sequence lengths and types. For basic visualization functionality, the only assumed property of a sequence is that it has a spatial occupation in an alignment, meaning that it has a start, an end and a row. All additional information about the sequence—including but not limited to the nucleotide sequence (or color space sequence in the case of SOLiD data), quality values, SNP locations, mate pair information and meta data—are stored as feature data coupled to the sequence, with general and configurable methods for how to visualize different categories of feature data. This means that anything with spatial properties that can be expressed in terms of row, start and end (with optional additional features), can be visualized and manipulated in the software.

Additional generality is provided in the amount of data that NGSView can handle. The use of Berkeley DB as back end allows for very fast disk retrieval and enables scrolling through millions of reads with no lag at a zoomed in level, as well as visualization of millions of reads simultaneously at a zoomed out level.

While NGSView is not intended to replace genome browsers like UCSC (Kuhn et al., 2009) and Ensembl (Hubbard et al., 2009), the general capability of displaying spatial data described above enables analysis of sequencing data in the context of annotation data. The included all purpose parser, which includes GFF parsing capability, facilitates inclusion and visualization of various types of annotation data into the viewer.

3.2 Flexibility

In NGSView, each element in the viewer is a bona fide object which can be selected and manipulated independently or in combination with other elements present in the same view. Different highlighting, browsing, scrolling and sorting operations are available (based on, e.g. SNP content, mate pair information, expression or other annotation data) for any subset of sequences selected, as well as other types of data manipulation and editing, and exporting to different file formats. NGSView also includes a user-configurable feature data type, which can be accessed by general sorting and highlighting methods available in the viewer.

In contrast to other next-generation viewers, NGSView provides additional flexibility in allowing editing operations such as cut, copy and paste, as well as dragging and dropping of sequences into any position. It is possible to create new contigs from subsets of the data as the user sees fit, thus enabling a sand box approach where different editing operations can be tried out without compromising the integrity of the original alignment.

3.3 Extensibility

As mentioned above, NGSView includes a user-configurable feature data type, allowing users to include additional feature data types into the NGSView input XML in a straightforward way. The package also comes with a documented API, including a framework for adding data types and operations in a well-defined manner. Details about extending the program, including skeleton code, are available at the NGSView web site. By releasing the source code under an open source license, we hope that additional members of the bioinformatics community will feel encouraged to contribute to further development of the software and API.

3.4 Additional features

NGSView can handle gapped alignments. Differential expression of reads, e.g. from case/control or time course experiments, can also be visualized.

ACKNOWLEDGEMENTS

The authors wish to acknowledge Erik Sjöland, who provided implementations and designs of key parts of the previous version of this software (DNPTrapper).

Funding: Research Grant for RIKEN Omics Science Center from MEXT (to Y.H.).

Conflict of Interest: none declared.

REFERENCES


Korbel, J.O. et al., 2009 PDEm: a computational framework with simulation-based error models for inferring genomic structural variants from massive paired-end sequencing data. Genome Biol., 10, R23.


Li, H. et al., 2009 The Sequence Alignment/Map format and SAMtools. Bioinformatics, 25, 2076–2079.

Li, R. et al., 2009 SNP detection for massively parallel whole-genome resequencing. Genome Res., 19, 1124–1132.


