

# Alview: Portable Software for Viewing Sequence Reads in BAM Formatted Files



Richard P. Finney, Qing-Rong Chen, Cu V. Nguyen, Chih Hao Hsu, Chunhua Yan, Ying Hu, Massih Abawi, Xiaopeng Bian and Daoud M. Meerzaman

Computational Genomics Research Group, Center for Bioinformatics and Information Technology, National Cancer Institute, Bethesda, MD, USA.

**ABSTRACT:** The name *Alview* is a contraction of the term *Alignment Viewer*. Alview is a compiled to native architecture software tool for visualizing the alignment of sequencing data. Inputs are files of short-read sequences aligned to a reference genome in the SAM/BAM format and files containing reference genome data. Outputs are visualizations of these aligned short reads. Alview is written in portable C with optional graphical user interface (GUI) code written in C, C++, and Objective-C. The application can run in three different ways: as a web server, as a command line tool, or as a native, GUI program. Alview is compatible with Microsoft Windows, Linux, and Apple OS X. It is available as a web demo at <https://cgwb.nci.nih.gov/cgi-bin/alview>. The source code and Windows/Mac/Linux executables are available via <https://github.com/NCIP/alview>.

**KEYWORDS:** genomics, short read, alignment, visualization, BAM, open source

**CITATION:** Finney et al. Alview: Portable Software for Viewing Sequence Reads in BAM Formatted Files. *Cancer Informatics* 2015:14 105–107 doi: 10.4137/CIN.S26470.

**TYPE:** Software or Database Review

**RECEIVED:** March 24, 2015. **RESUBMITTED:** May 24, 2015. **ACCEPTED FOR PUBLICATION:** May 26, 2015.

**ACADEMIC EDITOR:** J.T. Efrid, Editor in Chief

**PEER REVIEW:** Six peer reviewers contributed to the peer review report. Reviewers' reports totaled 1,267 words, excluding any confidential comments to the academic editor.

**FUNDING:** This study is supported by the NCI of the U.S. National Institutes of Health. The authors confirm that the funder had no influence over the study design, content of the article, or selection of this journal.

**COMPETING INTERESTS:** Authors disclose no potential conflicts of interest.

**CORRESPONDENCE:** [finneyr@mail.nih.gov](mailto:finneyr@mail.nih.gov)

**COPYRIGHT:** © the authors, publisher and licensee Libertas Academica Limited. This is an open-access article distributed under the terms of the Creative Commons CC-BY-NC 3.0 License.

Paper subject to independent expert blind peer review. All editorial decisions made by independent academic editor. Upon submission manuscript was subject to anti-plagiarism scanning. Prior to publication all authors have given signed confirmation of agreement to article publication and compliance with all applicable ethical and legal requirements, including the accuracy of author and contributor information, disclosure of competing interests and funding sources, compliance with ethical requirements relating to human and animal study participants, and compliance with any copyright requirements of third parties. This journal is a member of the Committee on Publication Ethics (COPE).

Published by Libertas Academica. Learn more about this journal.

## Introduction

New, large genomic data sets are providing more in-depth insights into the diagnosis and treatment of disease. In the past decade, new and innovative methods have continued to add value to the underlying data and uncover the secrets of the genome. Visual data inspection by experienced researchers is an important quality control element in the analytical process. Additionally, data visualization helps one to prioritize downstream analysis and verification steps. Unfortunately, this part of the process is tedious and time consuming, and the increasing volumes of high-throughput sequencing data of various types and platforms are proving to be a major analytical challenge. Here, we report a visualization tool that allows researchers to explore their data at a very rapid speed and significantly reduce the burden of reviewing tens and hundreds of thousands of variant calls. Areas with systematic read errors can be quickly identified, and inefficient attempts to verify results in noisy regions can be avoided.

## Features and Methods

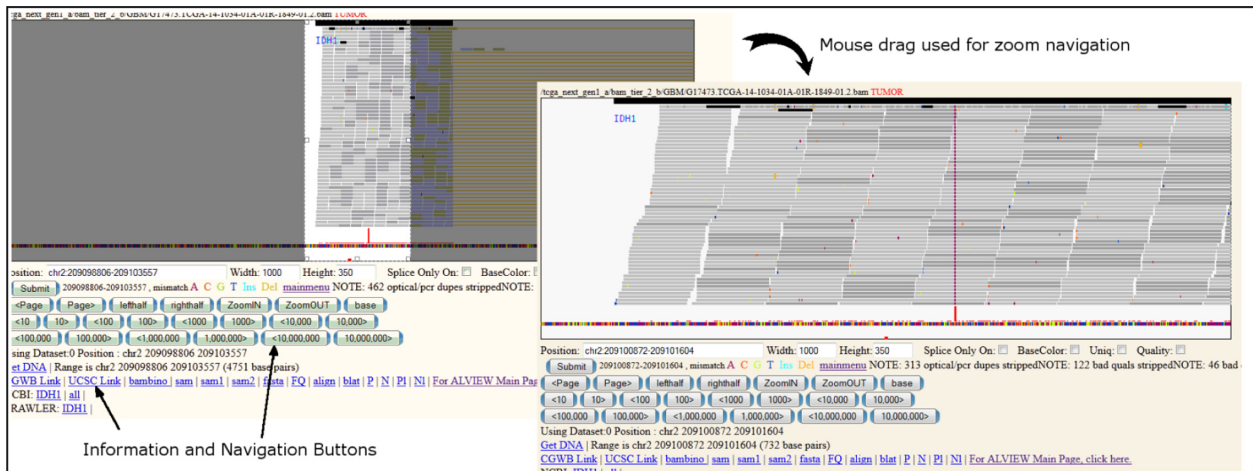
Alview is a fast and portable visualization tool. The core code interfaces with Heng Li et al's SAMtools Library<sup>1</sup> for parsing BAM files. The program is written in platform-independent C. Peculiarities specific to an operating system are isolated with `if defined (ifdef)` directives; so, for instance, when Microsoft

Visual C provides alternate support for a portable operating system interface (POSIX) standard function, a handcrafted, native interface work around is supplied.

For graphical user interface (GUI) frameworks, Alview uses WIN32 interface for Windows, the GTK2 interface for Linux, and BSD Unix-based systems and Cocoa for Apple Mac OS X. SAMtools<sup>1</sup> is written to POSIX standards, but different Microsoft Visual compilers provide various levels of support for these UNIX style standards. As a result, the source code for third-party libraries that were modified for Windows is provided to facilitate compiling and linking Alview on Windows.

The main code for Alview, in the file *alviewcore.cpp*, is written to be portable between operating systems and emphasizes speed of execution. The code can be compiled as a stand-alone executable and must be linked with the `zlib`<sup>2</sup> and SAMtools<sup>1</sup> libraries. Sequence reads are processed via custom SAMtools callback functions arranged in in-memory data structures and represented by an aesthetic, annotated image. The image is then output to the screen as a native graphics object or to the disk as a standard image format file.

Alview can also be compiled as a webserver daemon that uses the common gateway interface (CGI)<sup>3</sup> standard. The CGI version produces interactive html output and uses dynamic HTML5<sup>4</sup> features, including zoom in by selection via a



**Figure 1.** Information and navigation in Alview – upper left is original and lower right is zoom in via mouse drag to examine SNP. Various navigation buttons and information blocks assist in browsing BAM files.

jQuery<sup>5</sup> library. The CGI webserver Alview version loads a list of permitted-to-access BAM files from a user-maintained text file; so custom lists of BAM files of interest are easy to generate and use. The source code is free and open to modification so that users and local system operators can implement their own security.

The Alview CGI webserver version provides modifiable URL access, so that, for instance, cells in a spreadsheet can link to viewable results for any sample or location. A user-generated custom HTML file can link to specific samples and regions. Stand-alone Alview accepts parameters that specify BAM file name and genomic coordinates. Invoking Alview in a script can create a slideshow of interesting regions. For example, fields in a single nucleotide polymorphism (SNP) detection output file can be used to specify a series of calls to Alview to generate images for each purported polymorphism or mutation. The results can be quickly and easily reviewed by researchers. Users can generate text to annotate the slideshow images. A template is provided for command line creation of slide shows. The burden of reviewing ten and hundreds of thousands of mutation calls can therefore be significantly reduced.

The source code is available at GitHub.<sup>6</sup> The README file there points to links for selected executables and complete download packages that include the associated reference genome data. A live webserver version of Alview for examining public human cancer short-read datasets is available at <https://cgwb.nci.nih.gov/cgi-bin/alview>. The core source code for Alview is in the public domain. It uses some permissive free software licensed libraries. Alview source code and executables for several operating systems are available at the National Cancer Institute (NCI)/National Cancer Informatics Program’s (NCIP’s) GitHub site: <https://github.com/NCIP/alview>. Developers may modify Alview as they wish. NCI retains the copyrights to “National Cancer Institute” and associated images, which may not be used in forked projects.

## Results

Alview provides a solid substructure that allows for various types of access to short-read data across different operating systems. Figure 1. demonstrates the various navigation and information buttons available in the web version of Alview and shows how selection via mouse provides zoom in capabilities. Alview is a trim, fast, precise tool and complements existing programs such as the Integrated Genomics Viewer (IGV),<sup>7</sup> BamView,<sup>8</sup> and GBrowse 2.0.<sup>9</sup> The benefits of Alview are extreme speed and a sharp focus on exploring short reads.

Comparison of Alview with other programs should not be judged solely on benchmarks. Compounding factors include operating system cache effects and internet congestion. Different implementation philosophies can influence memory usage and performance but provide useful alternative paths to solving similar problems.

IGV provides much more functionality than Alview by supporting many other input file types other than BAM sequence read files. IGV’s Java implementation provides *write once, run anywhere* portability via implementations of the Java virtual machine. Alview’s implementation relies on low-level operating system and native GUI toolkit API calls. Alview provides extreme speed but is difficult to develop and maintain.

IGV requires registration for download for running off of disk, whereas Alview does not. Desktop IGV may require internet for full, easy, simple operation, whereas Alview does not require network connection (though it may call user-invoked external webpages). Alview operation does not log any user activity. On a Windows 7 Intel Core i5–2400 CPU at 3.10 GHz and 8 GB RAM, restarts of IGV v2.3 took from 12 to 18 seconds. Restarts of Alview took a small fraction of one second. For a small view of a genomic region, the Java Platform SE Binary for IGV took up 292 Mb, while Alview took up 11 Mb.



## Author Contributions

Design and coding: RF. Design and testing: CN, CH, CY, YH, MA, XB and Project management: DM. All authors reviewed and approved of the final manuscript.

## REFERENCES

1. Li H, Handsaker B, Wysoker A, et al. The sequence alignment/map format and SAMtools. *Bioinformatics*. 2009;25(16):2078–9.
2. L. Peter Deutsch, Jean-Loup Gailly, ZLIB Compressed Data Format Specification version 3.3, Internet RFC 1950, May 1996.
3. Coar K, Robinson D. The WWW common gateway interface version 1.1; 1999. Work in Progress.
4. Hickson I, Hyatt D. *HTML5: A Vocabulary and Associated APIs for HTML and XHTML*. Beijing: W3C Working Draft; 2011.
5. De Volder K. JQuery: a generic code browser with a declarative configuration language. In: Hentenryck PV, ed. *Practical Aspects of Declarative Languages*. Berlin: Springer; 2006:88–102.
6. Richard Finney, et.al. Alview Package, Version 1.0, (2015), National Cancer Institute, Bethesda, Maryland, USA. <https://github.com/NCIP/alview>
7. Robinson JT, Thorvaldsdóttir H, Winckler W, et al. Integrative genomics viewer. *Nat Biotechnol*. 2011;29(1):24–6.
8. Carver T, Böhme U, Otto TD, Parkhill J, Berriman M. BamView: viewing mapped read alignment data in the context of the reference sequence. *Bioinformatics*. 2010;26(5):676–7.
9. Stein LD. Using GBrowse 2.0 to visualize and share next-generation sequence data. *Brief Bioinform*. 2013;14(2):162–71.