BIOINFORMATICS APPLICATIONS NOTE Vol. 30 no. 23 2014, pages 3405–3407
doi:10.1093/bioinformatics/btu565

Sequence analysis

Advance Access publication August 21, 2014

PyBamView: a browser-based application for viewing short read alignments

Melissa Gymrek[1,2,3]

[1]Whitehead Institute for Biomedical Research, 9 Cambridge Center, Cambridge, MA 02142, [2]Harvard-MIT Division of Health Sciences and Technology, MIT, Cambridge, MA 02139 and [3]Program in Medical and Population Genetics, Broad Institute of MIT and Harvard, Cambridge, MA 02142, USA

Associate Editor: Gunnar Ratsch

ABSTRACT

Summary: Current sequence alignment browsers allow visualization of large and complex next-generation sequencing datasets. However, most of these tools provide inadequate display of insertions and can be cumbersome to use on large datasets. I implemented PyBamView, a lightweight Web application for visualizing short read alignments. It provides an easy-to-use Web interface for viewing alignments across multiple samples, with a focus on accurate visualization of insertions.
Availability and Implementation: PyBamView is available as a standard python package. The source code is freely available under the MIT license at https://mgymrek.github.io/pybamview.
Contact: mgymrek@mit.edu
Supplementary information: Supplementary data are available at *Bioinformatics* online.

Received on May 26, 2014; revised on July 27, 2014; accepted on August 14, 2014

1 INTRODUCTION

The rapid growth of next-generation sequencing (NGS) technologies has led to a wide variety of short read DNA datasets. Manual inspection of sequence alignments is an important aspect of quality control. While the majority of NGS analyses have focused on single nucleotide polymorphisms (SNPs), recent bioinformatics advances allow analysis of more complicated variants, such as small insertions or deletions (Montgomery *et al.*, 2013), larger structural variants (Ye *et al.*, 2009) and short tandem repeats (Gymrek *et al.*, 2012; Highnam *et al.*, 2013). Furthermore, widely used genome engineering techniques, such as the CRISPR-Cas9 system (Cong *et al.*, 2013) can often produce a wide range of complex variants. In these cases, visualization of insertion and deletion events is a particularly critical analysis step.

Current genome browsers, such as the UCSC Genome Browser (Kent *et al.*, 2002) and the Integrative Genomics Viewer (IGV) (Robinson *et al.*, 2011), offer visualization of alignments from SAM/BAM files across multiple samples and integration of many layers of genomics datasets. However, most existing tools have two important limitations. First, most are based on alignments to an ungapped reference sequence, which provides inadequate visualization of insertions. The SAM specification supports a padded reference, which captures multiple sequence alignment information and results in accurate insertion display by most browsers. However, most BAM files consist of pairwise alignments of short reads to a reference and do not use this feature. As a result, insertions are represented by an icon such as a vertical bar, which does not provide any visual information about the size or sequence of the inserted nucleotides. Second, the majority of alignment browsers are cumbersome to use, especially to visualize the large datasets typical of NGS experiments. They either require that the user upload large data files to a remote server or involve complicated installation and large resource requirements to run locally.

Several alignment browsers, such as Bambino (Edmonson *et al.*, 2011), Consed (Gordon and Green, 2013) and the text-based SAMtools (Li *et al.*, 2009) tview, overcome these limitations: they display the sequence of insertions even when using the standard ungapped reference, and are run locally with relatively low system requirements. However, tview does not allow the user to view multiple BAM files at once, and none of these tools allow for exporting alignments as snapshots or for sharing alignments remotely through a Web browser.

Here, I present PyBamView, a lightweight Web application for viewing alignments from BAM files. PyBamView provides alignment visualizations that accurately represent SNP, insertion and deletion events that can easily be exported to create publication-ready figures. It runs locally from the command line with minimal resource requirements and displays alignments in a Web browser. This interface allows users to quickly view alignments locally and to easily share alignments with local or remote collaborators.

2 BASIC USAGE AND FEATURES

PyBamView is a Python-based Web application that is run from the command line. Users provide PyBamView with a directory containing indexed BAM files and an optional reference genome in fasta format:

```
pybamview --bamdir DIRECTORY/WITH/BAMS --ref REF.fa
```

PyBamView will start a small Web server that can be accessed locally in a Web browser. Optional arguments can serve the application over a different address, for instance, for sharing

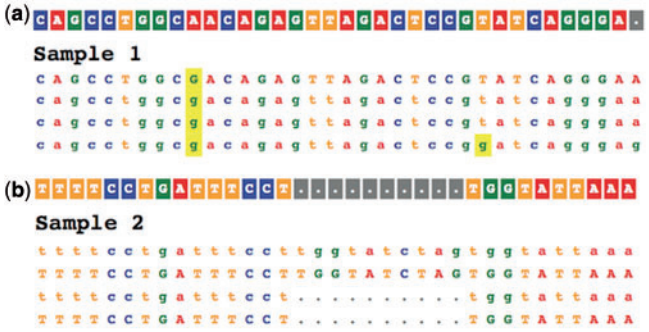

Fig. 1. PyBamView display of sequence variants. In each figure, the reference sequence is shown at the top followed by alignments of each read. Alignments were generated by PyBamView's PDF export feature. (**a**) Mismatches from the reference sequence, due to either SNPs or sequencing errors, are shown as highlighted bases. (**b**) Insertions are shown as gaps in the reference, which allows the length and sequence of the insertion to be easily visualized

the URL with remote collaborators or as a public resource. For instance, adding the options --ip 0.0.0.0 --port 5000 will serve PyBamView over port 5000 via http. The Supplementary Text and program Web site contain a complete description of this feature.

The Web browser displays a list of all samples contained in the BAM files provided. Users can select one or more samples to open in the genome-browser view. This consists of a reference track, followed by collapsible alignment tracks containing reads for each sample. While there is theoretically no limit to the number of samples analyzed, PyBamView can reasonably display five low to moderate coverage samples at once.

Users can navigate to the genomic region of interest by entering the genomic coordinate into the search bar (e.g. chr1:10000). In the default view, base pair differences from the reference genome are highlighted, allowing easy identification of SNPs and potential sequencing errors (Fig. 1a). A deleted base pair is indicated by a '.' in the alignment, and an insertion as a '.' in the reference sequence (Fig. 1b). This allows easy visualization of the sequence and size of inserted bases, which is not currently possible with most alignment browsers (Supplementary Fig. S1). Users can zoom out up to 100× to easily visualize large insertions or deletions spanning hundreds or thousands of bases. Additional features are described in the Supplementary Text.

3 EXAMPLE USE CASES

Alignment visualization is a critical step of any sequencing experiment. Here, I show three examples where PyBamView provides useful visualization of sequence variants. Use cases are not limited to these examples and can theoretically include any '-seq' experiment that can be represented by a BAM file.

First, it provides accurate visualizations of different length insertions, such as different alleles of a tandem repeat (Supplementary Fig. S2a). Furthermore, zooming out allows for visualization of large repeat expansions, such as a 60bp CAG expansion in Huntington's Disease (Supplementary Fig. S2b, simulated 250bp reads).

Second, it can be used to analyze variation across samples. This is useful in such analyses as comparing matched tumor versus normal samples or looking for mutations in affected versus non-affected individuals in disease genetic studies. Supplementary Figure S3 shows example comparisons of individuals at a SNP, small insertion and a large deletion spanning several kb.

Third, it can visualize complex mutations generated by genome engineering technologies such as CRISPR-Cas9 (Cong *et al.*, 2013). Dissecting these mutations requires adequate visualization of indels. An example alignment from a CRISPR library is shown in Supplementary Figure S4.

4 IMPLEMENTATION

PyBamView is implemented as a Python-based Web application using the Flask library. Alignments are processed using a Python backend, which then generates HTML, Cascading Style Sheets (CSS) and JavaScript files that are displayed in the Web browser.

PyBamView takes advantage of BAM and fasta indexing to avoid loading large files into memory. It uses the pysam and pyfasta libraries for parsing BAM and fasta files, respectively. Both libraries use efficient index data structures, which allow them to quickly fetch data from specific genomic regions of interest. Read alignments are parsed from the CIGAR strings in the BAM file and are displayed as simple HTML tables as Scalable Vector Graphics (SVG) elements using Javascript. All CIGAR options reported in the SAM specification, including the padded reference option, are supported (Supplementary Fig. S5).

5 CONCLUSION

As the use of NGS to analyze complex genomic events grows, there is a critical need for accurate and easy-to-use visualization tools. PyBamView provides a simple, yet powerful, interface for alignment visualization that facilitates collaborative data analysis.

ACKNOWLEDGEMENTS

The author would like to acknowledge members of the Erlich lab, Alon Goren, and Roy Ronen for helpful feedback, and Assaf Gordon for valuable programming guidance.

Funding: This work was supported by a National Defense Science and Engineering Graduate Fellowship (32 CFR 168a).

Conflict of interest: none declared.

REFERENCES

Cong,L. *et al.* (2013) Multiplex genome engineering using CRISPR/Cas systems. *Science*, **339**, 819–823.

Edmonson,M.N. *et al.* (2011) Bambino: a variant detector and alignment viewer for next-generation sequencing data in the SAM/BAM format. *Bioinformatics*, **27**, 865–866.

Gordon,D. and Green,P. (2013) Consed: a graphical editor for next-generation sequencing. *Bioinformatics*, **29**, 2936–2937.

Gymrek,M. *et al.* (2012) lobSTR: a short tandem repeat profiler for personal genomes. *Genome Res.*, **22**, 1154–1162.

Highnam,G. *et al*. (2013) Accurate human microsatellite genotypes from high-throughput resequencing data using informed error profiles. *Nucleic Acids Res*., **41**, e32.

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

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

Montgomery,S.B. *et al*. (2013) The origin, evolution, and functional impact of short insertion-deletion variants identified in 179 human genomes. *Genome Res*., **23**, 749–761.

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

Ye,K. *et al*. (2009) Pindel: a pattern growth approach to detect break points of large deletions and medium sized insertions from paired-end short reads. *Bioinformatics*, **25**, 2865–2871.