BIOINFORMATICS APPLICATIONS NOTE
Vol. 28 no. 9 2012, pages 1276–1277
doi:10.1093/bioinformatics/bts122

Sequence analysis

Advance Access publication March 13, 2012

SAMSCOPE: an OpenGL-based real-time interactive scale-free SAM viewer

Kris Popendorf and Yasubumi Sakakibara*

Biosciences and Informatics, Keio University, 3-14-1 Hiyoshi, Yokohama, 223-8522, Japan

Associate Editor: Alex Bateman

ABSTRACT

Summary: Existing SAM visualization tools like 'samtools tview' (Li *et al.*, 2009) are limited to a small region of the genome, and tools like Tablet (Milne *et al.*, 2010) are limited to a relatively small number of reads and may fail outright on large datasets. We need to visualize complex ChIP-Seq and RNA-Seq features such as polarity as well as coverage across whole 3 Gbp genomes such as Human. We have addressed these problems in a lightweight visualization system called SAMSCOPE accelerated by OpenGL. The extensive pre-processing and fast OpenGL interface of SAMSCOPE provides instantaneous and intuitive browsing of complex data at all levels of detail across multiple experiments.

Availability and implementation: The SAMSCOPE software, implemented in C++ for Linux, with source code, binary packages and documentation are freely available from http://samscope.dna.bio.keio.ac.jp.

Contact: yasu@bio.keio.ac.jp

Received on October 17, 2011; revised on January 28, 2012; accepted on March 7, 2012

1 INTRODUCTION

Next-generation sequencing (NGS) workflows often involve mapping reads onto reference genomes using tools such as SHRiMP2 (David *et al.*, 2011), Bowtie (Langmead *et al.*, 2009) and others. Mapping determines the likely point of origin (or origins) of a given read. Despite the multitude of different mapping methods and software, the SAM (*Sequence Alignment/Map*) format (Li *et al.*, 2009) and the associated binary encoding (BAM) have emerged as the *lingua franca* of NGS mapping file formats. For many projects using SAM data, it is desirable to visually inspect the results of mapping for quality control and exploration. However, because a single NGS run can provide millions of reads from billions of bases of genome sequence, simply opening up a SAM file and making sense of the content is a non-trivial problem. The two main problems in visualization of a SAM dataset are:

(1) Sam files are structured in terms of reads. To calculate coverage of a given base, we have to look at *each read* and see which bases it maps to, then count how many times that base has been mapped.

(2) Most computers only have 1000–2000 pixel wide displays with which to visualize billions of data points.

*To whom correspondence should be addressed.

As visualization is a common need, a variety of tools have been introduced to view SAM data in different ways. For example 'samtools tview' (Li *et al.*, 2009) provides an interactive text-based viewer which shows each base of each read and reference genome as a character in a text terminal. This can be useful for inspecting narrow regions (~100 bases) with fewer reads than terminal rows (~30), but not helpful for examining larger regions or deeper coverage. 'Tablet' (Milne *et al.*, 2010) and its close cousin 'IGB' (Nicol *et al.*, 2009) both provide Java-based graphical interfaces for drawing reads against a reference sequence where each base is represented as a colored rectangle. Both can summarize overall coverage with a secondary visualization track. However, both Tablet and IGB draw each read similar to 'samtools tview', limiting their speed and effectiveness when a large number of reads would be in view. 'Integrative Genomics Viewer' (IGV; Robinson *et al.*, 2011) can load SAM/BAM files providing detailed inspection capabilities, and provides a 'mean coverage' track given proper pre-processing, which can scale to arbitrary genome sizes. For applications like ChIP-Seq (Pepke *et al.*, 2009) or RNA-Seq however, '*mean* coverage' is not necessarily helpful and hard to use at large scales, as most coverage values are at zero or near zero.

We needed a flexible method to visualize and interactively inspect various features from large numbers of reads across mammalian-scale genomes while addressing the problems above, so we developed our own approach in SAMSCOPE.

3D computer graphics addresses a similar problem when drawing textures on distant 3D objects: how to efficiently generate a reasonable approximation of millions of points of color data into one screen pixel. A solution known as MIP mapping (Williams, 1983) has become a mainstay of modern 3D rendering; in it a series of filtered copies of each texture are pre-calculated at exponentially decreasing resolutions. Thus, when a distant object is rendered, rather than sampling millions of points to calculate the combined contribution to one screen pixel, an approximation is achieved with just a few samples from a lower resolution copy. We apply the MIP map concept to genome visualization in SAMSCOPE.

2 METHODS AND IMPLEMENTATION

SAMSCOPE adopts a layer-based display model, where each layer reflects a SAM mapping feature, such as coverage. Layers are stored as BAM MIP Maps ('BIPs') on disk in a binary format allowing instantaneous navigation with minimal memory requirements. Multiple layers can be displayed simultaneously as different colors, and in multiple synchronized windows. This layer-based design makes it simple to display results from multiple SAM files as different layers, and visually compare results from different experiments. SAMSCOPE supports a variety of feature calculations

for different applications. For example, in ChIP-Seq the difference in number of reads mapped to the forward strand compared with the reverse strand (what we term 'polarity') results in a characteristic zero-crossing inverted-peak pair which, in conjunction with coverage, often reflects protein binding sites (Pepke *et al.*, 2009). SAMSCOPE allows easy composition of data layers into compound data layers, such as polarity; 'forward' and 'reverse' count layers are generated, from which overall 'coverage' (the sum) and 'polarity' (the difference) are derived (shown in Fig. 1B). This design allows for fast and memory efficient pre-processing of huge datasets, as data primitives can be stored on disk and accessed at random as needed.

To generate BIPs we adapt the traditional graphics MIP map approach to sampling 1D data series. The pre-processing algorithm is as follows:

(1) Calculate the full resolution series with one value for each base of reference genome, forming a series of columns with one value each.

(2) Merge data from adjacent columns generating half the number of composite columns. To merge two columns, combine values from each column as value/frequency tuples. For example, if the value '3' occurred twice in each column, a single tuple {3,4} would be retained.

(3) If the number of value/frequency tuples in the current column exceeds a user adjustable maximum, retain a subset; the most frequent values are prioritized, along with the extreme maximum and minimum values.

(4) The merged column data and an index reference for fast random lookup are stored to disk.

(5) Steps 2–4 are repeated until only one column remains.

To render a given region of data, a set of columns is selected to match the pixel width of the display window. For example, if rendering a 50 Mbp region in a 1024 pixel wide window, data from the 16th 'column merge' iteration is selected, such that each column represents a sample from the underlying $2^{16} = 65\,536$ bases. Thus, the display can be rendered with data from 763 precomputed columns. This approach has some practical benefits:

- To draw one screen of P pixels, at most $O(P)$ values must be read and drawn. Fast rendering allows us to abandon scroll bars, and adopt an intuitive mouse-based 'pan and zoom' interface familiar to users of Google Maps (http://maps.google.com).
- The pre-processed data retains multiple values per column, allows SAMSCOPE to change the rendering style at runtime to a representative spectrum of values at each column, or an average value, or maxima and minima or other arbitrary effects.
- Complex features (such as peaks in ChIP-Seq and RNA-Seq) are visible at genome-scale, and not obscured by techniques like averaging as used in previous tools like IGV and MAQ (Li *et al.*, 2008).
- Just as BIP display only needs a few columns in memory at a time, BIP generation only needs the values of two pairs of columns. Thus SAMSCOPE can run perfectly well on an average laptop computer.
- Finally, because empty columns do not need to store any data values, and because BIP files are stored as 'sparse files' on disk, BIP files are well suited for sparse data like exomes or ChIP-Seq.

Sequence parameters (such as chromosome name and length) are obtained from the source SAM/BAM file itself, eliminating any additional sequence setup steps as are required in programs like IGV. Annotation data from GFF/GTF files are displayed if available (Fig. 1A). As a practical benchmark on human reference, a coverage BIP file can be generated from 54 M 120 bp mapped reads on a server with 96 GB of RAM in 12 min, or on an ordinary desktop with 6 GB of RAM in 19 min.

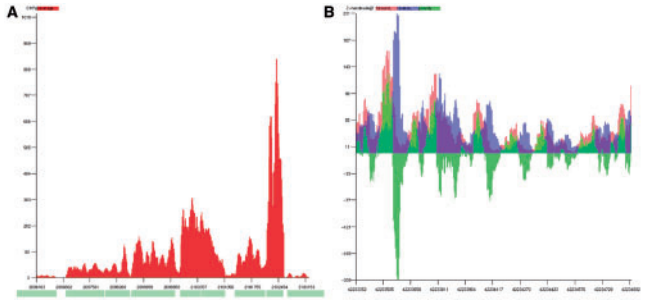

Fig. 1. (**A**) Shows coverage from an RNA-Seq experiment on a 7072 bp region of *Bacillus subtilis* subsp. natto BEST195. Gene annotations are shown as green bars below the *X*-axis. (**B**) Shows 'forward, reverse and polarity' layers from a ChIP-Seq analysis of CENPA focusing on a 500 bp region at in Z chromosome centromere of chicken (Shang *et al.*, 2010).

3 DISCUSSION

SAMSCOPE's BIP format is similar in its goal to the BigBed and BigWig (Kent *et al.*, 2010). However where BigBed/BigWig use B-trees and a complex data layout suitable for remote access over HTTP, SAMSCOPE uses a much simpler flat file format designed for local access through the POSIX *mmap* function. This design allows the entire process to run efficiently on commodity hardware. SAMSCOPE can be more efficient for its purpose because where BigBed is designed to contain any arbitrary annotation (e.g. scores from overlapping alignments etc.), SAMSCOPE focuses on per base aggregate statistics (e.g. coverage, polarity etc.). We find the BigBed and BIP formats to be complementary in this sense.

Funding: This work was supported by KAKENHI (Grant-in-Aid for Scientific Research) on Innovative Areas [No. 221S0002] from the Ministry of Education, Culture, Sports, Science and Technology of Japan.

Conflict of Interest: none declared.

REFERENCES

David,M. *et al.* (2011) SHRiMP2: sensitive yet practical SHort Read Mapping. *Bioinformatics*, **27**, 1011–1012.

Kent,W.J. *et al.* (2010) BigWig and BigBed: enabling browsing of large distributed datasets. *Bioinformatics*, **26**, 2204–2207.

Langmead,B. *et al.* (2009) Ultrafast and memory-efficient alignment of short DNA sequences to the human genome. *Genome Biol.*, **10**, R25.

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

Li,H. *et al.* (2009) The Sequence Alignment/Map format and SAMtools. *Bioinformatics*, **25**, 2078–2079.

Milne,I. *et al.* (2010) Tablet–next generation sequence assembly visualization. *Bioinformatics*, **26**, 401–402.

Nicol,J.W. *et al.* (2009) The Integrated Genome Browser: free software for distribution and exploration of genome-scale datasets. *Bioinformatics*, **25**, 2730–2731.

Pepke,S. *et al.* (2009) Computation for ChIP-seq and RNA-seq studies. *Nat. Methods*, **6**, 22–32.

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

Shang,W.H. *et al.* (2010) Chickens possess centromeres with both extended tandem repeats and short non-tandem-repetitive sequences. *Genome Res.*, **20**, 1219–1228.

Williams,L. (1983) Pyramidal parametrics. *Comp. Graph.*, **17**, 1–11.