Bioinformatics, 31(15), 2015, 2565–2567
doi: 10.1093/bioinformatics/btv178
Advance Access Publication Date: 29 March 2015
Applications Note

Genome analysis

bio-samtools 2: a package for analysis and visualization of sequence and alignment data with SAMtools in Ruby

Graham J. Etherington[1], **Ricardo H. Ramirez-Gonzalez**[2] **and Dan MacLean**[1,*]

[1]The Sainsbury Laboratory and [2]The Genome Analysis Centre, Norwich Research Park, Norwich NR4 7UH, UK

*To whom correspondence should be addressed.

Associate Editor: Gunnar Ratsch

Received on August 18, 2014; revised on February 17, 2015; accepted on March 23, 2015

Abstract

Motivation: bio-samtools is a Ruby language interface to SAMtools, the highly popular library that provides utilities for manipulating high-throughput sequence alignments in the Sequence Alignment/Map format. Advances in Ruby, now allow us to improve the analysis capabilities and increase bio-samtools utility, allowing users to accomplish a large amount of analysis using a very small amount of code. bio-samtools can also be easily developed to include additional SAMtools methods and hence stay current with the latest SAMtools releases.

Results: We have added new Ruby classes for the MPileup and Variant Call Format (VCF) data formats emitted by SAMtools and introduced more analysis methods for variant analysis, including alternative allele calculation and allele frequency calling for SNPs. Our new implementation of bio-samtools also ensures that all the functionality of the SAMtools library is now supported and that bio-samtools can be easily extended to include future changes in SAMtools. bio-samtools 2 also provides methods that allow the user to directly produce visualization of alignment data.

Availability and implementation: bio-samtools is available as a BioGem from http://www.biogems.info or as source code from https://github.com/helios/bioruby-samtools under the MIT License.

Contact: dan.maclean@tsl.ac.uk

1 Introduction

Many large scale genomics studies rely heavily on re-sequencing experiments in which genomes of individuals or populations are to be compared. Typically these experiments begin with shotgun sequencing of DNA using high-throughput methods followed by alignment of the sequences to a reference genome assembly. Over 100 different high-throughput aligners are now available (Wikipedia, 2014) (http://en.wikipedia.org/wiki/List_of_sequence_alignment_software), including BWA (Li and Durbin, 2010), Bowtie (Langmead *et al.*, 2009) and SOAP (Li *et al.*, 2008). The standard storage format for this data is Sequence Alignment/Map (SAM) format (Li *et al.*, 2009) which describes alignments in a read-wise fashion.

Analysis of the alignments and downstream inference requires high-throughput interrogation of numerous alignment files; hence, programmatic interfaces allowing easy access to the data in these files are widely used. Manipulation of SAM (and the compressed, indexed variant, BAM) can be done in many high-level languages with a variety of packages, including SAMtools (Li *et al.*, 2009), Picard (http://picard.sourceforge.net), GATK (McKenna *et al.*, 2010), bio-samtools (Ramirez-Gonzalez *et al.*, 2012) Bioperl (Stajich *et al.*, 2002), Rsamtools (Morgan *et al.*, 2013) and pysam (https://github.com/pysam-developers/pysam).

Here, we describe bio-samtools 2, a new Ruby language SAMtools interface that provides new classes for describing genomic regions and genetic variants, allows the easy addition of newly developed SAMtools features and can produce publication-quality visualizations of data with minimal effort by the coder.

We previously described bio-samtools (Ramirez-Gonzalez *et al.*, 2012) a Ruby language binding to the SAMtools library and a biogem of the BioRuby framework (Bonnal *et al.*, 2012). Since its release, bio-samtools has been downloaded >25 000 times. SAMtools is in constant development and hence there is a need to accommodate any future functionality of SAMtools by facilitating extension of our software by other programmers.

2 Approach

The original bio-samtools package was implemented around libbam.so (for Linux) and libbam.1.dylib (for Mac OS X), the core shared object library in the SAMtools package. bio-samtools used the Ruby Foreign Function Interface (https://rubygems.org/gems/ffi) package as a bridge between the two languages. We now use a straightforward but simple and flexible pipe interface that captures the standard output stream from SAMtools directly. This allows all functions to be executed and captured with ease by parsing and presenting data as it passes in from SAMtools standard output. A number of new Classes and methods have been added to bio-samtools 2 which allows users to carry out complex analysis with only a small amount of code. The major additions are described here.

2.1 Bio::DB::Pileup

SAMtools MPileup utility provides data in a reference-position-wise format. bio-samtools abstracts this in the Bio::DB::Pileup class and returns simple objects that provide easy access to further methods, such as calculating allele frequencies, calling a consensus and creating a VCF (Danecek *et al.*, 2011) object. The code below demonstrates how easily pileup data can be used to extract the consensus bases from a BAM file.

```
bam = Bio::DB::Sam.new(:fasta => ’ref.fasta’,
    :bam => ’my_bam.bam’)
bam.mpileup do |pileup|
    pileup.consensus
end
```

2.2 Bio::DB::VCF

The Bio::DB::VCF class represents variant information in an informative, concise and compact manner, allowing only variant positions to be maintained. Using the Bio::DB::VCF class, bio-samtools allows the analysis of VCF-formatted data, including the comparison of variants across any number of samples. It allows the user to easily carry out analysis of consensus calls, genotypes and allele frequencies, which can then be used for further downstream analyses.

bio-samtools 2 also has a number of extra analysis methods not available in the SAMtools library. For example, with just a few lines of code a user could easily count the reference and non-reference bases at each position and then go on to calculate the allele frequencies of those bases. bio-samtools also allows users to calculate the consensus sequence for any site or identify the genotype for any indel or SNP. bio-samtools 2 also has the advantage of working within other Ruby code to complement bespoke analysis methods and can also be used in the development of web applications.

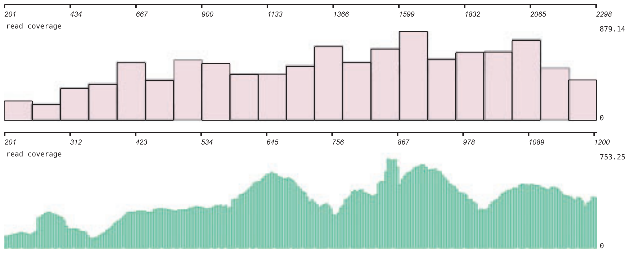

Fig. 1. Coverage plots demonstrating the formatting available to bio-samtools 2. The height of the plot represents coverage over a given region. The numbered scale above each plot represents the genome position. The width of each bar on the plot is set by the bin parameter when calling the plot_coverage method

2.3 Visualization

We have also added the functionality of creating publication-quality visualizations of genome read coverage plots for regions of sequence alignments (Fig. 1). This is achieved by taking a Bio::DB::Sam object (from the class representing the SAM/BAM file), creating a pileup object and extracting the resulting read coverage. This is then plotted into SVG which can be written to a file or loaded straight into a webpage. We have provided an example of some bio-samtools 2 code to demonstrate the simple but effective code to create rich plots direct from bio-samtools 2.

```
bam = Bio::DB::Sam.new(:fasta => ’ref.fasta’,
    :bam =>’my_bam.bam’)
bam.plot_coverage(”chr_1”, 201, 2000,:bin=>20, :
    svg => ”out.svg”,:fill_color => ’red’)
```

3 Conclusion

bio-samtools 2 is a useful and flexible Ruby library for programmatic access to SAMtools. It contains extra analysis and visualization methods not present in the SAMtools library, which allows the coding of complex analysis methods with a small amount of code. The flexibility of bio-samtools 2 also allows new SAMtools methods to be implemented easily and quickly. bio-samtools 2 can also be integrated into other Ruby software, including web applications.

Acknowledgement

We thank Raoul J.P. Bonnal for maintaining the bio-samtools source code and gem repository and Jirapong Nanta, Ben J Woodcroft, Francesco Strozzi, Bradford Powell, Fedor Gusev and Yuri Pirola for contributions and useful suggestions.

Funding

RHRG is funded by a Norwich Research Park PhD Studentship and a Funding and Maintenance Grant from The Genome Analysis Centre. G.J.E. and D.M. are supported by The Gatsby Charitable Foundation.

Conflict of Interest: none declared.

References

Bonnal,R.J. *et al.* (2012) Biogem: an effective tool-based approach for scaling up open source software development in bioinformatics. *Bioinformatics*, **28**, 1035–1037.

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

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

Li,H. and Durbin,R. (2010) Fast and accurate long-read alignment with burrows-wheeler transform. *Bioinformatics,* **26**, 589–595.

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

Li,R. *et al.* (2008) Soap: short oligonucleotide alignment program. *Bioinformatics,* **24**, 713–714.

McKenna,A. *et al.* (2010) The genome analysis toolkit: a mapreduce framework for analyzing next-generation dna sequencing data. *Genome Res.*, **20**, 1297–1303.

Morgan,M. *et al.* (2013) *Rsamtools: Binary alignment (BAM), FASTA variant call (BCF), and tabix file import*, R package version 1.18.3, http://bioconductor.org/packages/release/bioc/html/Rsamtools.htm.

Ramirez-Gonzalez,R. *et al.* (2012) Bio-samtools: Ruby bindings for samtools, a library for accessing bam files containing high-throughput sequence alignments. *Source Code Biol. Med.*, **7**, 6.

Stajich,J.E. *et al.* (2002) The bioperl toolkit: Perl modules for the life sciences. *Genome Res.*, **12**, 1611–1618.

Wikipedia (2014) List of sequence alignment software.