BIOINFORMATICS APPLICATIONS NOTE Vol. 27 no. 19 2011, pages 2754–2755
doi:10.1093/bioinformatics/btr442

Genome analysis

Advance Access publication July 29, 2011

Gee Fu: a sequence version and web-services database tool for genomic assembly, genome feature and NGS data

Ricardo Ramirez-Gonzalez[1], Mario Caccamo[1] and Daniel MacLean[2,*]

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

Associate Editor: Alex Bateman

ABSTRACT

Summary: Scientists now use high-throughput sequencing technologies and short-read assembly methods to create draft genome assemblies in just days. Tools and pipelines like the assembler, and the workflow management environments make it easy for a non-specialist to implement complicated pipelines to produce genome assemblies and annotations very quickly. Such accessibility results in a proliferation of assemblies and associated files, often for many organisms. These assemblies get used as a working reference by lots of different workers, from a bioinformatician doing gene prediction or a bench scientist designing primers for PCR. Here we describe Gee Fu, a database tool for genomic assembly and feature data, including next-generation sequence alignments. Gee Fu is an instance of a Ruby-On-Rails web application on a feature database that provides web and console interfaces for input, visualization of feature data via AnnoJ, access to data through a web-service interface, an API for direct data access by Ruby scripts and access to feature data stored in BAM files. Gee Fu provides a platform for storing and sharing different versions of an assembly and associated features that can be accessed and updated by bench biologists and bioinformaticians in ways that are easy and useful for each.

Availability: http://tinyurl.com/geefu

Contact: dan.maclean@tsl.ac.uk

Received on April 12, 2011; revised on July 11, 2011; accepted on July 23, 2011

1 INTRODUCTION

NGS technologies, easy to use bioinformatics pipelining and workflow management tools like Galaxy (Goecks *et al*., 2010) that lower barriers to access to powerful tools for genome assembly such as Velvet (Zerbino and Birney, 2008) have made it possible for individual laboratories to create draft quality genome assemblies in very short timescales. Projects describing fragmented but valuable assemblies (Farrer *et al*., 2009; Kemen *et al*., 2011; Raffaele *et al*., 2010) have provided many new insights. Laboratories whose main expertise is molecular biology can have difficulty managing the large datasets that they have created. Bioinformatic and bench based analyses generate new sequence and feature annotations and laboratories can easily lose track of the many changes to their sequence and annotated features.Thus, the amount of data generated in NGS experiments is driving a need for automation and sustainable storage solutions for assembly and feature data. We have created an application called Gee Fu that stores genomic sequence and feature information and provides a platform to allow groups with diverse levels of bioinformatics and computing skills to easily store, track versions, edit and visualize feature data and keep track of versions, share and visualize basic sequence data. There are many genome browsers and annotation editors, including GBrowse and Apollo (Lewis *et al*., 2002). Gee Fu extends and integrates into this eco-system by providing a light-weight database version system that can import from and export to other annotation systems and genome browsers if needed. It provides access and utility to bioinformaticians who want to access data programatically and others who prefer a graphical interface in one central repository that allows changes made by one worker to be reflected to others instantly. Gee Fu has been useful in our next-generation sequence based assembly and annotation efforts (Kemen *et al*., 2011).

2 IMPLEMENTATION

Gee Fu is an application built on Ruby-on-Rails, a rapid, open-source web application development framework based on the Ruby language (http://rubyonrails.org) and is centred around a straightforward relational database schema for feature and sequence data. In this schema, the organism is the root object and may have many genome versions and reference sequences. Each sequence object has many feature objects which are described as in GFF3 and parsed into the database using the BioRuby GFF3 class. Sequence read feature information from high-throughput alignments to reference sequence is stored in BAM format (Li *et al*., 2009) accessed using samtools-ruby. Features also belong to experiment objects which describe the study in which features are created. Features may be edited and old versions are stored in a predecessors table, providing a feature history. As with other systems, features are tied to a reference sequence and do not automatically transfer to new sequence, allowing for complex rearrangements of the sequence between versions. Even for large assemblies from plants or mammals, system requirements are light, enough storage for the sequence and features in the RDBMS and for any BAM files is required. Memory requirements are low and 4 GB RAM will suffice.

3 DATA ACCESS

Input: organism, genome (including sequence), experiment and feature data may each be added to the database from FASTA and GFF3 format files. This is easily accomplished using the forms in the web-interface, with Rails rake tasks that can be executed from

*To whom correspondence should be addressed.

the command-line or through custom Ruby scripts that can access the object relational mapping layer in Rails directly. All genomes are viewable at all times and Rails allows extension to allow restriction of certain data to specified users.

Web analyses: web forms allow feature retrieval by genomic interval and database ID. Identified features are returned to the user in the browser, grouped features such as the CDS and UTR features that make up genes are rendered together for ease of comprehension and edits can be made to features directly in the browser window, and these are automatically versioned (Fig. 1). Sequence in a genomic interval can be retrieved in FASTA format and features exported to text in GFF, EMBL or GenBank formats.

Console: the object relational mapping (ORM) layer in Rails allows the database to be interacted with directly from a console and scripts. Database records are returned as objects, each with methods appropriate to their class allowing easy object-oriented scripting on the database without the need to develop additional database interaction code. The ORM layer also allows Gee Fu to be database management system (DBMS) agnostic.

Web-service: Gee Fu implements a straightforward but useful API for retrieval of data via GET requests. Gee Fu can return data on all the model classes in the database in XML or JSON format. Feature data can be returned as lists of objects for local analysis or as a summary, e.g. of sequence read coverage in a specified genomic interval.

Integration with other Annotation Editors and Genome Browsers: Gee Fu is able to import and export features in GFF3, a common feature format used by many other annotation editing programs, like Apollo. It is possible to export features from an experiment and edit them externally, then load them back in with edits being automatically updated and versioned. GFF3 is compatible with popular browsers and databases such as GBrowse. Gee Fu has methods to respond directly to requests made by the freely available AnnoJ viewport, making it suitable for creating a fast genome browser with a few simple configuration steps. The web-service API of Gee Fu can easily be extended to send data to other services such as JBrowse for feature rendering.

4 DISCUSSION

Gee Fu helps a growing problem in genomics: the storage and sharing of in-development assemblies and feature data within small groups of collaborators. Lack of specialist data management skills in groups using NGS technologies mean that versions of an assembly can proliferate, and become out of sync with each other causing at best confusion and time spent remedying the situation, at worst loss of data or inaccuracy. Gee Fu helps to centralize assembly and feature storage and provides easy data access to workers with varying degrees of experience with databases and computational methods. As a Rails application Gee Fu benefits from the functionality of ActiveRecord, the ORM layer in Rails, and all database information can be accessed programatically in an object-oriented fashion facilitating high-throughput analyses with custom scripts. Gee Fu uses our samtools-ruby gem, a Ruby wrapper around SAMtools (Li *et al*., 2009) that wraps all the API functions allowing BAM files to be used alongside the conventional database. Gee Fu allows for data sharing over the web with a light web-service API

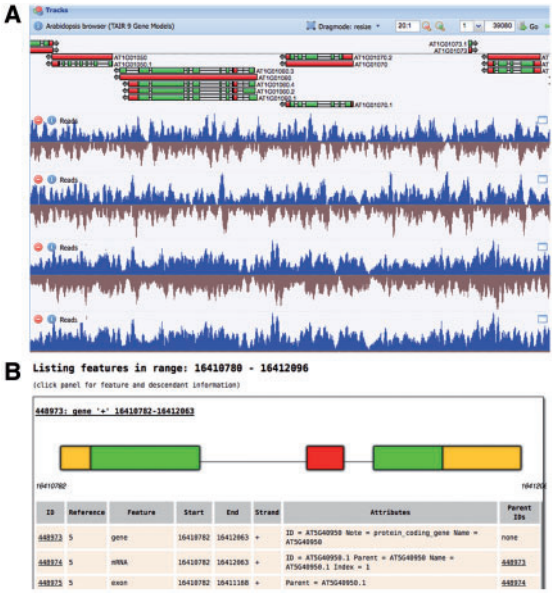

Fig. 1. Two visualizations showing different aspects of Gee Fu. (**A**) AnnoJ view showing gene model and four BAM file read tracks. (**B**) Web-browser view of features for direct editing.

ideal for scripts and websites avoiding bulk data exchange, making collaboration on a growing project simple. Gee Fu's database schema is open and lightweight, allowing for extension via the Rails framework which enjoys a large, supportive expert user-base, meaning the application is customisable to the needs of individual groups. Additional analysis tools can easily be integrated. Freely available viewport software such as AnnoJ and JBrowse mean that Gee Fu can be the data store for a easily set up and fast genome browser.

ACKNOWLEDGEMENT

Thanks to Raoul Bonnal for gem help and Julian Tonti-Fillipini for AnnoJ help.

Funding: Gatsby Charitable Foundation (to D.M.); Biotechnology and Biological Sciences Research Council (to R.R.G. and M.C.).

Conflict of Interest: none declared.

REFERENCES

Farrer,R.A. *et al*. (2009) De novo assembly of the Pseudomonas syringae pv. syringae B728a genome using Illumina/Solexa short sequence reads. *FEMS Lett.*, **291**, 103–111.

Goecks,J. *et al*. (2010) Galaxy: a comprehensive approach for supporting accessible, reproducible, and transparent computational research in the life sciences.*Genome Biol.*, **11**, R86.

Kemen,E. *et al*. (2011) Gene gain and loss during evolution of obligate parasitism in the white rust pathogen of Arabidopsis. *PLoS Biol.*, **9**, e1001094.

Lewis,S.E. *et al*. (2002) Apollo: a sequence annotation editor. *Genome Res.*, **3**, 82–85.

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

Raffaele,S. *et al*. (2010) Genome evolution following host jumps in the Irish potato famine pathogen lineage. *Science*, **330**, 1540–1543.

Zerbino,D.R. and Birney,E. (2008) Velvet: algorithms for de novo short read assembly using de Bruijn graphs.*Genome Res.*, **18**, 821–829.