Badger—an accessible genome exploration environment

Citation for published version:
Elsworth, B, Jones, M & Blaxter, M 2013, 'Badger—an accessible genome exploration environment'
Bioinformatics, vol 29, no. 21, pp. 2788-9. DOI: 10.1093/bioinformatics/btt466

Digital Object Identifier (DOI):
10.1093/bioinformatics/btt466

Link:
Link to publication record in Edinburgh Research Explorer

Document Version:
Publisher's PDF, also known as Version of record

Published In:
Bioinformatics

Publisher Rights Statement:
This is an Open Access article distributed under the terms of the Creative Commons Attribution License (http://creativecommons.org/licenses/by/3.0/), which permits unrestricted reuse, distribution, and reproduction in any medium, provided the original work is properly cited.

General rights
Copyright for the publications made accessible via the Edinburgh Research Explorer is retained by the author(s) and / or other copyright owners and it is a condition of accessing these publications that users recognise and abide by the legal requirements associated with these rights.

Take down policy
The University of Edinburgh has made every reasonable effort to ensure that Edinburgh Research Explorer content complies with UK legislation. If you believe that the public display of this file breaches copyright please contact openaccess@ed.ac.uk providing details, and we will remove access to the work immediately and investigate your claim.
Badger—an accessible genome exploration environment

Ben Elsworth, Martin Jones and Mark Blaxter*
Institute of Evolutionary Biology, University of Edinburgh, West Mains Road, Edinburgh EH9 3JT, UK
Associate Editor: Alfonso Valencia

ABSTRACT
Summary: High-quality draft genomes are now easy to generate, as sequencing and assembly costs have dropped dramatically. However, building a user-friendly searchable Web site and database for a newly annotated genome is not straightforward. Here we present Badger, a lightweight and easy-to-install genome exploration environment designed for next generation non-model organism genomes.
Availability: Badger is released under the GPL and is available at http://badger.bio.ed.ac.uk/. We show two working examples: (i) a test dataset included with the source code, and (ii) a collection of four filarial nematode genomes.
Contact: mark.blaxter@ed.ac.uk

Received on June 7, 2013; revised on July 24, 2013; accepted on August 7, 2013

1 INTRODUCTION
High-throughput sequencing has revolutionized genome sequencing. New sequencing technologies and improved computational tools mean that a high quality draft eukaryotic genome can be sequenced and assembled in days, on a budget accessible to most researchers. This has produced an explosion of genome projects, an increasing number of which involve multiple species or strains.

Genome data and annotation must be made accessible to collaborators via a restricted environment and to the wider research community following publication. Several genome exploration environments (GEEs) exist, including ENSEMBL (Flick et al., 2013) and National Center for Biotechnology Information (NCBI) Genomes (http://www.ncbi.nlm.nih.gov), but these do not allow real-time updates or restrictions on data access. ENSEMBL can be used as a stand-alone GEE, as can projects like GeneDB (Logan-Klumpler et al., 2012), but these require skills not available in groups new to genome informatics. GEEs that use the Generic Model Organism Database (GMOD) Chado database model (Mungall et al., 2007), including Tripal (Ficklin et al., 2011) and Chado on Rails (http://gmod.org/wiki/Chado_on_Rails), are similar to ENSEMBL in scope. Tripal is the most mature GMOD GEE and is widely used, but requires a dedicated software engineer to develop and design each genome instance.

We here present Badger, an easy-to-install GEE for genome scientists who need to produce a web-accessible portal for new data. A single installation of Badger can contain data from multiple species, and each species can have multiple genome drafts and gene sets.

*To whom correspondence should be addressed.
access control. Genome and annotation data are stored in only 15 custom domain classes, making the code base easy to understand and extend. Badger uses PostgreSQL (http://www.postgresql.org/) as a persistent data store.

Badger was developed on Linux and has been deployed on both Ubuntu (12.04) and Centos (6.3) distributions. Hardware requirements will vary with the size of the dataset; we recommend a minimum of 4 GB random access memory and 2 central processing unit cores. Badger requires Grails (v2.1.6) and Java Development Kit (v1.6+). PostGreSQL (v8.4+) and BLAST+ (v2.2.26+) Detailed installation instructions are available at https://badger.bio.ed.ac.uk. Optionally, OrthoMCL (v2.0.4+) is required for clustering. Muscle (v3.8.31+) (Edgar, 2004) for ortholog group alignment, InterProScan (v4+) (Quevillon et al., 2005) for domain identification and GBrowse for genome browser delivery.

Loading data into a Badger instance is straightforward (Fig. 1). First, metadata are added, including species information and sequence and annotation file paths. The minimum data requirement is a collection of contigs in FASTA format, which will generate a basic BLAST server, a genome overview page and a sequence download tool. The full potential of the GEE is achieved with annotated gene sets. Badger accepts gene set data (GFF3 format) with corresponding transcript and protein sequences (FASTA format), and also data in BLAST XML output, InterProScan raw output and Tab Separated Value (TSV) custom annotation files. TSV import allows custom annotation types to be added without editing underlying code. Ortholog group information should be provided in OrthoMCL format. New species can be added rapidly by taking advantage of existing annotation. Badger can import data from external resources and FASTA format genome and gene sequences. External data within a GBrowse instance can also be embedded.

Data files can be uploaded, edited and updated individually through the administrator interface or en masse using a script (an example is provided in the source code). Once data files are uploaded and metadata entered, Badger parses input files, loads data into the PostgreSQL database, sources publications for the species from NCBI PubMed, creates BLAST databases, compresses FASTA files for download, generates a phylogenetic tree and catalogs ortholog data. Publication data are updated weekly. The whole process can be trivially rerun at any time to allow the inclusion of new data. Badger data upload is fast. For computationally expensive data in overview pages, Badger makes extensive use of caching to ensure that the interface remains responsive. For full-text searching of annotations, Badger uses PostgreSQLs full-text indexing, allowing it to search around 0.5 million annotations in a few seconds. The test dataset (100 scaffolds, 289 genes and 6615 annotations) takes <5 min, whereas the four-species filarial nematode dataset (50 000 scaffolds, 50 000 genes and 700 000 annotations) takes <24 h, to recreate on a minimum specification machine.

4 CONCLUSION

Badger is a lightweight GEE with a gene- and annotation-centric approach capable of storing, searching and visualizing diverse genomic data. Installation is simple, and a fully functional instance can be created quickly, even by novice users. Although not a substitute for data submission to databases of record, Badger is a customizable environment for public and collaborative display of gene-centred genomic information, and will aid in preparation for publication and submission.

ACKNOWLEDGEMENT

Thanks to the Blaxter laboratory for the feedback.

Funding: BBSRC BBR AfterParty (BB/J023585/1) (to M.J.), in collaboration with Prof Paul Brakefield funded by the ERC grant, EMARES (250325) (to B.E).

Conflict of Interest: none declared.

REFERENCES


