BIOINFORMATICS ORIGINAL PAPER Vol. 26 no. 24 2010, pages 3119–3124
doi:10.1093/bioinformatics/btq599

Databases and ontologies Advance Access publication October 22, 2010

The GMOD Drupal Bioinformatic Server Framework

Alexie Papanicolaou[1,2,3,*] and David G. Heckel[2]

[1]Centre for Conservation and Ecology, University of Exeter in Cornwall, Penryn TR10 9EZ, UK, [2]Department of Entomology, Max Planck Institute for Chemical Ecology, Hans-Knöll Str 8, Jena D-07745, Germany and [3]CSIRO Ecosystem Sciences, Black Mountain Laboratories, Clunies Ross St, Acton 2601, Australia

Associate Editor: Alfonso Valencia

ABSTRACT

Motivation: Next-generation sequencing technologies have led to the widespread use of -omic applications. As a result, there is now a pronounced bioinformatic bottleneck. The general model organism database (GMOD) tool kit (http://gmod.org) has produced a number of resources aimed at addressing this issue. It lacks, however, a robust online solution that can deploy heterogeneous data and software within a Web content management system (CMS).

Results: We present a bioinformatic framework for the Drupal CMS. It consists of three modules. First, GMOD-DBSF is an application programming interface module for the Drupal CMS that simplifies the programming of bioinformatic Drupal modules. Second, the Drupal Bioinformatic Software Bench (biosoftware_bench) allows for a rapid and secure deployment of bioinformatic software. An innovative graphical user interface (GUI) guides both use and administration of the software, including the secure provision of pre-publication datasets. Third, we present genes4all_experiment, which exemplifies how our work supports the wider research community.

Conclusion: Given the infrastructure presented here, the Drupal CMS may become a powerful new tool set for bioinformaticians. The GMOD-DBSF base module is an expandable community resource that decreases development time of Drupal modules for bioinformatics. The biosoftware_bench module can already enhance biologists' ability to mine their own data. The genes4all_experiment module has already been responsible for archiving of more than 150 studies of RNAi from Lepidoptera, which were previously unpublished.

Availability and implementation: Implemented in PHP and Perl. Freely available under the GNU Public License 2 or later from http://gmod-dbsf.googlecode.com

Contact: alexie@butterflybase.org

Received on May 18, 2010; revised on September 28, 2010; accepted on October 19, 2010

1 INTRODUCTION

1.1 Emerging model species and bioinformatics

Next-generation sequencing (NGS) technologies have allowed an increasing number of biologists to utilize the -omic experimental strategy and support research programs by searching for statistically significant patterns in large-scale (LS) experiments (Collins *et al.*, 2003). Due to the limited number of bioinformaticians and resources, this rapid uptake of -omics is now causing a bioinformatic bottleneck. This bottleneck, which is more pronounced in the Ecological and Evolutionary Functional Genomics (EEFG) community (Beldade *et al.*, 2008), ought to be addressed without requiring custom-made and non-integrated solutions. The Generic Model Organism Database tool-kit (GMOD; http://gmod.org) is a consortium originally formed from functional genomics model organism communities to produce a standard set of open source software for handling, primarily, genomic data. Since its inception, the consortium has built or incorporated an impressive array of tools and standards. The uptake of GMOD tools and standards has been so successful that GMOD has expanded beyond the functional genomics community and is now been used by EEFG laboratories.

Indeed, 'MOD' databases are now commonplace in the -omics field (http://gmod.org/wiki/GMOD_Users). Until recently, GMOD software had focused on whole-genome sequencing. As researchers from other fields make use of -omic, bioinformatic and artificial intelligence (AI) approaches, GMOD has expanded into other fields such as phylogenetics (Heinicke *et al.*, 2007), microarray research (Day *et al.*, 2007), molecular ecology (a Chado extension from the National Synthesis Center for Evolution http://www.nescent.org/informatics/software.php), transcriptomics without a reference genome (Papanicolaou *et al.*, 2009) and others. Further, the cost-effectiveness of NGS does not apply to the downstream cost associated with computational analysis of the data; quite the opposite, in fact. Therefore, there is an ever-growing need for cost-effective and integrated solutions that improve the capabilities of wet-lab biologists to mine their own data before publication. Even though a number of commercial tools exists, some are not affordable. Others are closed-source software and thus cannot be adapted. Moreover, most are not integrated into the larger GMOD framework. Individual attempts within the GMOD consortium have yet to provide a generic visualization front end. The web site creation tool, GMODWeb (O'Connor *et al.*, 2008) is of interest but it has limited scope, but it is useful in rapidly generating a web-based front-end for a Chado database (Arnaiz *et al.*, 2006; Mungall *et al.*, 2007). Tripal (http://gmod.org/Tripal) offers an efficient front-end for Chado but no generic framework. InterMine (Lyne *et al.*, 2007) is a more powerful graphical user interface (GUI) for a database, driven by lightweight JavaScript but it is a complicated framework to use for development. The Ensembl system (Hubbard *et al.*, 2002) is an example of a complete platform for processing genomic data, but it was custom built for the needs of the Sanger Institute rather than a community software. Indeed, most of above software are open source but not necessarily developed for open-development. In order to minimize reliance on continued funding, the community could orientate toward more generic frameworks

*To whom correspondence should be addressed.

explicitly designed for open-development. Bioinformatic work-flow visualizations, such as Taverna and Galaxy (Giardine, 2005; Oinn *et al*., 2004), are both geared toward data analysis, even though the latter allows for custom plugins. Although the Galaxy team is working toward a more general framework for data dissemination, for many bioinfomaticians, the Ensembl solution seems more robust. Ensembl is an entire bioinformatic framework with both analysis and dissemination tools, but it has a very steep learning curve and is not a GMOD component. It would be of interest, however, for the entire GMOD community to develop a generic 'plumbing' framework so that (i) laboratories can rapidly deploy web sites with data analysis/dissemination tools (such as Taverna or a BLAST server); (ii) bioinformaticians can rapidly program new applications (such as custom front-ends on par with InterMine).

1.2 The Drupal content management system

One solution is to use a content management system (CMS) such as Wordpress, Drupal or others. CMSs are platforms for storing, managing, disseminating data of any type. Often they have been used to drive web sites, including 'blogs', but research projects such as Scratchpads (Smith *et al*., 2009) have also been successful. Some researchers use CMSs for building their laboratory web sites. Some CMSs support a number of useful concepts such as the RDF, XML and similar protocols, ontologies, controlled vocabularies and the other concepts relating to the Semantic Web. Further, CMSs are often a complete software package with tools for managing community-based data, such as users, roles and fine-grained permissions. Some CMSs are modular, allowing for users to program their own plugins and extend functionalities. Drupal is such a CMS. It is open source and can be downloaded freely from http://drupal.org. It is written in PHP, a language that is straightforward for nascent bioinformaticians to learn. It supports a number of database engines, including mySQL, Oracle and the GMOD supported PostgreSQL. Further, Drupal is built with security in mind, has powerful user-management tools and is highly modular, allowing for plugins to be developed and deployed in a standardized and streamlined fashion. Importantly, Drupal is popular and well documented. The widespread use has resulted in a large active community of users and developers (e.g. see http://egressive.com/article/who-uses-drupal).

This article initiates a long-term effort in creating a bioinformatic framework for the Drupal CMS within the specifications of GMOD. We developed three modules for three categories of users. First, GMOD-DBSF is a generic function framework for Drupal developers of bioinformatic tools. Then we built two modules for end users: (i) a powerful similarity-search software (e.g. BLAST) server for wet-lab biologists and system administrators benefiting from a friendly GUI and (ii) an RNAi experiment databasing platform. The latter can be easily modified for other experimental data, but it was developed and used in a community-wide review on failed and unpublished RNAi experiments in Lepidoptera (butterflies and moths; the only taxon where RNAi experiments are often unsuccessful).

2 METHODS

We used the Drupal 6 CMS. As the Chado package uses PostgreSQL, this database engine is required. The GMOD-DBSF module is a base module and thus required for all other modules in this framework; the other modules are optional. The GMOD-DBSF base module can utilize an installation of the Chado package but installing it is not necessary as it is not required for the biosoftware_bench module. The BioPerl (http://bioperl.org; Stajich *et al*., 2002) package and freely available Perl libraries (from CPAN) are needed along with certain Drupal modules: the Tabs module (http://drupal.org/project/tabs) is used to deploy tabular web content; the JQuery module (http://drupal.org/project/jquery) to deploy and seamlessly maintain the JQuery JavaScript library. Further, an external JQuery-utilizing library, dynatree (http://code.google.com/p/dynatree), is used to produce 'check-box trees'. Commonly used annotation software, such as BLAST, annot8r, InterProScan and SSAHA2 (Altschul *et al*., 1997; Ning *et al*., 2001; Schmid and Blaxter, 2008; Zdobnov and Apweiler, 2001), were integrated into biosoftware_bench, but using them requires that they are installed on the server (not all software needs to be installed: administrators can select which ones they wish to make available). For sequence retrieval, the fastacmd and Bio::Seq::Index approaches were used for the BLAST and SSAHA2 databases, respectively. To enable job management, Condor (Thain *et al*., 2005) was used as it is simple and can perform well on both a PC-farm and a single multi-core host. Future versions of this framework aim to make use of the Sun Grid Engine.

2.1 Specifications

The framework adheres to certain criteria. It (i) is open source under a non-restrictive license and thus can be customized and expanded; (ii) can be integrated with other widely used bioinformatic applications and implement the GMOD standards; (iii) is secure to both the user and the server; (iv) provides GUIs to both end users and administrators; (v) is developer friendly by extending Drupal's application programming interface (API) according to the Drupal community specification. Drupal itself has a powerful API: e.g. the deployment of the third party modules, such as the ones presented here, requires no more than a single line of code. This complements their installation, which is usually 'point-and-click'.

2.2 Aims

The work presented here focused on producing three Drupal modules. The first is GMOD-DBSF, which provides a framework for developing new bioinformatic Drupal modules. It is responsible for (i) importing a subset of the Chado tables to Drupal, (ii) creating new tables in Drupal using Chado conventions; (iii) providing functions to communicate with Chado and Drupal database schemas; and (iv) providing other, generic, functions useful for bioinformatic module development.

We also built two example applications. First, a software server with the BLAST, InterProScan, annot8r and SSAHA2 software deployed by default; additional plugins can be generated by the community. Second, a web-based database for storing experimental information from RNAi experiments. We used the Minimum Information Criteria for RNAi experiments (MIARE) as provided by the MIARE working group (http://miare.sourceforge.net) and the Lepidoptera RNAi Working Group, an international group composed of 70 scientists from 42 institutions in 21 countries (Terenius *et al*., under review).

2.3 Schema

We opted not to use Chado for public data entry and manipulation; in our work, Chado is used as a long-term and secure data warehouse. We prefer not to allow the public to commit changes to the Chado database but still wish to provide a bidirectional user-interface. We, therefore, use Chado for read operations of data residing in the data warehouse, but opted to create a Chado schema within Drupal for read/write operations of user-contributed data. With Chado being a highly generic schema, there are a number of tables unused in this instance of GMOD-DBSF (e.g. the MAGE module). Therefore, we imported only the basic Chado tables in the Drupal database (the feature, organism, cv, dbxref, pub tables and their dependencies). Drupal is then extended with additional tables created using

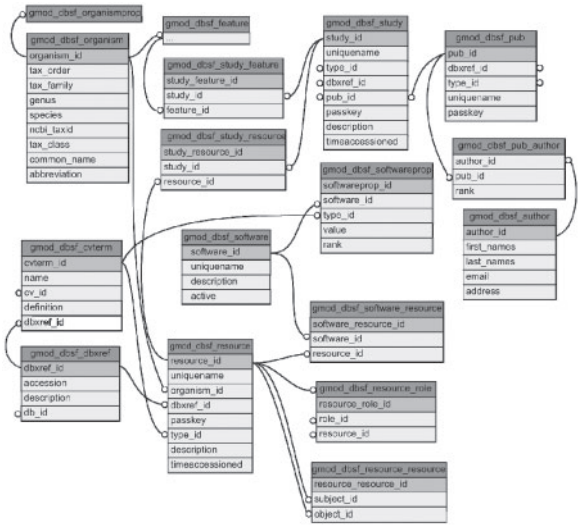

Fig. 1. Part of the database schema built by GMOD-DBSF. Chado conventions ensure that this schema can interact with an installed Chado database. Some tables and links omitted for clarity.

the Chado conventions (Fig. 1). New tables in the 'resource' group were created to allow better representation of sequence-less features. Likewise, a software table group is utilized specifically for software variables and is linked with the resource data using the software_resource. Further, a new study group of tables has been created to allow for generalized databasing of wet-lab experimental data. Publications are supported via the Chado pub schema. We used new tables to better integrate authorships using the author and pub_author tables. This implementation allows seamless integration with core Drupal data: e.g. a resource_roles allows linking of the resources with specific Drupal username groupings (roles). All of these tables are installed automatically during the point-and-click installation of GMOD-DBSF. It was expected that certain applications would require the synchronization of data between the Drupal and Chado databases. For example, InsectaCentral requires it for its Community Annotation module. For the security of Chado as a data warehouse, developers should be cautious but secure protocols can be developed using Drupal's features. GMOD-DBSF offers such feature/resource-specific synchronization. In InsectaCentral's implementation, a special administrator user-group is allowed to use these functions and synchronization changes with Chado.

2.4 RNAi experiment

In order to efficiently provide a cataloguing platform for the RNAi experiments, the Lepidoptera RNAi working group used the MIARE ontologies. MIARE is a set of reporting guidelines that describes the minimum information that should be reported about an RNAi experiment to enable the unambiguous interpretation and reproduction of the results. We then built the genes4all_experiment module using GMOD-DBSF and enhanced it via community feedback. Three different datatypes are used: sequence features, sequence-less features and publications. To distinguish between the first two, the latter is called a resource and has a separate set of tables in our schema. Two types of sequence data are used: (i) the target gene, which may be derived from a species other than the one targeted (due to lack of sequence information), and (ii) the RNAi construct. Three types of resources are used: (i) experimental animals, (ii) delivery protocol and (iii) assay protocol. Considering that the genes4all_experiment caters primarily to unpublished research, the publication GUI requests only the communicating author but, as we mentioned above, the schema can handle multiple authors and their details via the author and pub_author tables.

3 RESULTS

3.1 Drupal for bioinformatics using GMOD-DBSF

The core Drupal program has limited capabilities for bioinformatics. As a CMS, it is most capable in storing, displaying and organizing data as stored in the so-called 'nodes': authored web-pages linked with ancillary data. Extensions, called 'modules', extend its functionality. For example, the Tabs module that we use allows for multiple web-pages to appear as tabs. Such modules provide their own API and thus allow other modules to make use of a complicated functionality using only a line of code. The GMOD-DBSF module is one such module. Bioinformaticians can use it to perform an increasing number of operations (see http://gmod-dbsf.googlecode.com/files/GMOD-DBSF_dev_manual_1.0.pdf). Indeed, we hope that as the bioinformatics community embraces Drupal, GMOD-DBSF will also expand.

Currently, GMOD-DBSF offers a number of functionalities not available in the Drupal core. A set of functions allows a generic interaction with Chado tables. The function *gmod_dbsf_add_cv()*, for example, allows for one to add a new Controlled Vocabulary (CV) by providing the name of the CV and an array with the CV terms they wish to add. This function can connect to a Chado database via the *gmod_dbsf_db_execute()* function or operate on the local Drupal database (or make use of the *gmod_dbsf_is_chado()* auto-detect function). Similar functions operate to add, delete and populate the feature, db, pub and other Chado tables. Ancillary Chado tables, such as the featureprop and feature_cvterm tables, often require complicated SQL commands with multiple joins. A number of gmod_dbsf functions cater to simplify manipulating these tables by simply passing the desired variables. For example, a featureprop table can be populated with a single line of code which passes the feature ID or feature name, the CV term and properties one wishes to associate. This approach is the *raison d'etre* of GMOD-DBSF: to allow other modules to query and manipulate Chado in a standardized fashion, and also to accelerates the development of other modules. Other convenience functions allow a developer to install a materialized view, a new table or PostgreSQL function. A few functions aim to provide secure approaches for oft-used tasks. The *gmod_dbsf_create_uid()* function (all non-core functions in Drupal begin with the module's name) creates a unique MD5 identifier, based on a user's session ID, time and optionally a text string, which can be used for file uploads. The *gmod_dbsf_batch_upload_fasta()* function allows users to upload a FASTA file to the server even if it is many megabytes or takes a considerable amount of time. It is used, for example, by the biosoftware_bench software server to allow users to upload datasets for use as query or subject databases. Finally, a few functions have been created to make use of BioPerl functions. For example, one function is responsible for creating and parsing GFF3 files, another, the *gmod_dbsf_get_taxonomy_from_ncbi()*, uses Bio::DB::Taxon to query NCBI (via Entrez or via a local NCBI Taxonomy database flatfile) for the taxonomy of a species. In InsectaCentral, this function is used in conjunction with the *gmod_dbsf_get_add_organism()* function to build a GUI for InsectaCentral curators to add new

organisms and ancillary phylogenetic information into the Chado database.

3.2 Bioinformatic software bench

3.2.1 Innovations When a laboratory generates multiple pre-publication datasets, a local solution for mining, searching and manipulating the data must be deployed. This leads to cumbersome administration and maintenance and the need for constant bioinformatic support. There are a number of main innovations of biosoftware_bench: (i) graphical administration; (ii) deployment of command-line software; (iii) use of a secure daemon to handle job submissions with the option to use the Condor job management system; and iv) linking datasets with phylogenetic information. Further, the ability to deploy datasets only available to certain users or groups allows for the existence of a single server to handle both public and pre-publication data. As the system is integrated with a laboratory's web site and user authorization is handled by Drupal, the entire process appears seamless to the user. Moreover, the deployed software can also be used by other modules, i.e. without a GUI. By reusing the same biosoftware_bench functions, another module can utilize them to prepare and process software results. For example, a module currently under development allows for community members to submit an open reading frame, which is then automatically processed and annotated with the BLAST, annot8r and InterProScan software, with the resulting data stored first into Drupal and then transferred into Chado by a curator.

3.2.2 Installation of software plugins The module comes with plugins for BLAST, annot8r, InterProScan and SSAHA2, but others can be coded by the community. Bioinformatic software can be installed through the biosoftware_bench module. Two 'include' (.inc) files are needed for each software. One file guides the installation, including the use of CV terms to define options. The second file is responsible for the interface and batch jobs. New software can be deployed within a few hours by creating two such files and providing Perl routines to handle any output graphs. Once deployed, administrators have access to a set of options that allows them to select which software they wish to install and if they wish to make use of the Condor job management system. This latter feature allows administrators to utilize a PC-farm or a multi-core server to control job submissions. In both cases, a Perl daemon containing the aforementioned Perl routines, processes the jobs as an unprivileged user. For the software servers, it also post-processes the output of the software search in order to provide the output as a number of file formats. A Bio::Graphics-driven image of an alignment of the hits to the query is also produced and colored according to the significance statistic.

3.2.3 Administration The biosoftware_bench module provides an administrator's GUI to minimize user-errors and reduce the time required to setup and maintain a software server. In Drupal, administrative rights are decoupled for each module and each action. Users with specific administrative rights have a GUI where they can specify the location of datasets, see which ones are available and choose which to deploy. The administrator can provide friendly names and group memberships (e.g. 'Genomes', 'Transcriptomes', 'UniProt', etc.) to assist users selecting an appropriate dataset. System and security checks prevent errors with typing or dataset formatting, and thus ensure that the database is populated only with functional datasets. Further, linking them to a species through the NCBI Taxonomy database can be used to enable the phylogeny-driven dataset selection. One security feature allows the administrator to decide if a dataset is to be made restricted to a specific set of users. This allows for the secure deployment of both public and pre-publication datasets from the same server and interface. In the future, for large web sites biosoftware_bench ought to load the dataset in the database rather than use flatfiles. The current method of providing formatted flatfiles is, however, the most straightforward approach and will suit the bulk of biosoftware_bench administrators, in particular bioinformaticists with limited programming or databasing skills.

3.2.4 End-user capabilities The privacy mechanism allows end users to see only the datasets that their username and role memberships allow. In the BLAST server, they can choose to run multiple BLAST algorithms simultaneously, expand their subject dataset by uploading a multi-FASTA file and use a phylogeny to select species- or taxon-specific subject datasets. Once the search is submitted, a self-refreshing page with a unique submission identifier (SUID) appears and can be used to bookmark the page. The system uses 'cron' jobs to purge old files, and administrators can decide when result files are flagged as being old and ready to be deleted. For the BLAST software, the results are first produced as XML but BioPerl modules provide an additional choice of text and HTML output. For other software, when possible, an XML is also provided as well as GFF and/or HTML and text. A Bio::Graphics-driven alignment graph provides an overview of the queries, any hits and their respective scores. The tabular presentation of significant hits allows users to download hits of interest as a FASTA file.

3.3 Experiment module

The experiment module was custom built for the International Lepidoptera RNAi Working Group (Terenius *et al*., under review). It utilizes functions provided by GMOD-DBSF. A number of core functions exist and adapting them for other types of experiments is straightforward. The GUI was built to provide a good balance between user-friendliness and data security. Each fully completed submission is live in real time, partial submissions can be continued later and even completed submissions may be edited by authorized individuals. The date and time of the last submissions/edits is stored in the database. The user is requested to first provide a unique name for their submission, their email address (which is not made public) and a non-unique passkey. The passkey can be used by multiple submissions and its purpose is to prevent unauthorized edits. Further, data linked with a passkey can be reused by subsequent submissions and allows for continuing incomplete ones. After providing these credentials, the user is presented with a panel of six tabs. The first tab relates to the sequence-based features: the target gene and RNAi construct. The second tab handles non-sequence data (resources) such as experimental animals and protocols. The third tab contains the publication data, including external database cross-references. Such references are also available for the resources and features, allowing users to identify experimental animal stocks or GenBank gene identifiers. Once all required information is provided, a finalize tabs becomes available and users are able to review their submission prior to storing the study as 'complete'. At any time, users can stop

and continue their submission at a later day by making use of their passkey credentials.

In order to reduce work load to curators, much of the data are driven by controlled vocabularies (as provided by the community). Building new modules might be considered time consuming. It might be of interest, therefore, to note that the design and deployment of this module required weeks of full-time equivalent work, excluding a week of responding to community feedback. Using the existing module as a template, however, other types of experiments can be supported in a matter of hours.

4 DISCUSSION

4.1 Utility as a community resource

Unlike other software, the work presented here aims to integrate well with an existing laboratory website. This system allows laboratories to deploy software locally. This is especially useful for software that can take advantage of clusters of computers [e.g. the RaxML phylogenetics or PAML molecular evolution software (Stamatakis *et al.*, 2008; Yang, 2007)]. Further, by utilizing a CMS, laboratories can deploy the biosoftware_bench module via the point-and-click approach. They can, likewise, create their entire web content including feed aggregators (e.g. Atom), blogs and file servers. Indeed, a user-friendly system can be the key to allow a specific -omics community (such as one centered around a taxon or a genome-sequencing project) to develop and interact with a central resource such as a large database supporting that community. Drupal modules offer a straightforward installation and also allow for customization within a variety of existing 'themes'. It is possible, then, to provide the feeling that the -omic data, BLAST servers and standard web-pages are part of one package.

4.2 Utility as a bioinformatic framework

With the explosion of information and the paucity of expertise, Drupal is already being applied across biological disciplines: recent work funded by the European Union Framework 6 has produced Scratchpads, a Drupal project for Natural History collections (Smith *et al.*, 2009). With advances in information technology and increased interest in semantic integration, the genomics community will benefit from choosing a diverse and robust system, such as Drupal, for integrating, analyzing and displaying information. With more genome-sequencing project coming to fruition, there will be laboratories focusing on datatypes such as ecological and population data which, thus far, are not part of genome databases. GMOD-DBSF is a step toward addressing these emerging needs without worsening the bioinformatic bottleneck.

This new API for Drupal makes the co-existence of Chado and Drupal seamless to the end user and reduces the learning curve for the bioinformatic community. Additionally, a large number of core functions or third party modules are available to be used by the bioinformatic community. One example is Drupal's abilities for data federation. A single settings file (settings.ini) defines the database names and access credentials, allowing for a federated database system in the sense that a single web-page can be served by multiple databases which may reside on multiple hosts. This may be of special interest as such an approach would allow us to build to a heterogeneous system of database engines or gain remote access to other database servers. In InsectaCentral's implementation, for example, we deploy Drupal and Chado as core databases and then a SeqFeature::Store database for each of the 200 hosted species. In future versions of InsectaCentral, a laboratory will be able to deploy a local copy of InsectaCentral, a local copy of a Drupal database and connect to the public Chado and SeqFeature::Store databases. They can then deploy their private data as local Chado and SeqFeature::Store databases so that a mix of private data and data from the up-to-date InsectaCentral is seamlessly served to the end user. Further, the Services module (http://drupal.org/project/services) provides the means for integrating multiple interfaces such as XMLRPC, JSON, REST, SOAP, etc., avoiding, thus, the need to set up a separate BioMart instance (Smedley *et al.*, 2009). This allows a Drupal site to provide web services to other software via multiple interfaces while using the same callback code. Even though Chado was built to be generic and therefore easy to exchange data between groups, different genome-sequencing teams have implemented it in a slightly different way so that cross-communication is not straightforward and adaptors have to be written. The Drupal CMS can become a solution to this compatibility issue between Chado databases.

4.3 Integrating with other software

This generic framework could tap into the concept of bioinformatic work flows, such as those offered by Taverna and Galaxy. This is an interesting possibility to consider and may inspire the EEFG community to use these tools. Meta-servers and software to run bioinformatic applications are constantly being developed. A number of command-line software packages now have their own web servers and a dedicated journal now exists (the annual *Nucleic Acids Research* Web Server issue). The most robust and widely used meta-application amongst these is the Galaxy framework. Even though originally developed for genomic data, it has now expanded to other types of data through an active developer community. Galaxy does not offer the main benefits of a CMS (i.e. ease of customization and a rich API). Further, administration of a multi-lab server can be a daunting task for the often over-worked bioinformatician. The biosoftware_bench approach provides full control of the visualization and processing routines. As Drupal is taken up by the GMOD consortium, bioinformaticians who provide new tools would benefit from preparing a biosoftware_bench.inc file (i.e. their software can be easily deployed and laboratories readily can manage and administer it without requiring access to a dedicated bioinformatician).

An increasing number of applications exist for displaying genome data to web-users [e.g. the FlyBase database (Drysdale and Crosby, 2005), the UCSC Genome Browser (Kent *et al.*, 2002), the Ensembl system and the ubiquitous GBrowse (Stein *et al.*, 2002)]. As more laboratory groups generate -omic data, there will be a pressing need to develop more such software. One example is the GMODWeb, which builds a web site for a Chado database using the Turnkey application (http://turnkey.sourceforge.net). Like GMODWeb, GMOD-DBSF utilizes an external application to drive content deployment but, instead of Turnkey, it uses Drupal, another open source software. Drupal has the advantages of a broader end-user base and hundreds of developers, and is built to be robust and secure for users and the host server. Because of the large number of functions provided by the core and contributed modules, the Drupal solution will become a powerful tool for bioinformaticians.

GMOD-DBSF is the only Drupal application built to a generic GMOD API. Another implementation, Tripal, also a GMOD tool (http://gmod.org/wiki/Tripal), is available and in active development. It provides a direct interface with Chado, allowing users to edit the contents of a Chado database. The two modules are not mutually exclusive as GMOD-DBSF is aimed as a base module to facilitate development of other modules. With Tripal and the software presented here, the 'Drupal solution' provides a feature set unavailable in any of the other software. Indeed, we could envision multiple Drupal sites linking and sharing their data in a seamless manner.

5 CONCLUSION

The software presented here was built specifically for the research communities that are only now emerging into the -omics era. For example, NGS transcriptome data are widely used to address central biological questions in non-model species but many laboratories do not yet have the means to make the best use of these data. Due to funding constraints, these communities also have a paucity of bioinformaticians. Developed tools must, therefore, be general enough so that they can be used between laboratories and also straightforward to customize so that wet-lab biologists with a bit of training in programming can deploy and maintain the software. Supporting this new cadre of 'bioinformaticists' is vital in order for the communities of emerging model species to reap the rewards that NGS technologies have to offer. We have shown that our software can assist with development of bioinformatic web services. Because Drupal modules are licensed under the GNU Public license and our software was built to be generic and expandable, it would be of interest to the bioinformatic community to expand it. To assist users and developers, we have provided screencast tutorials via another Drupal project, SciVee (http://scivee.tv), which can be accessed via http://gmod.org/gmod-dbsf. We anticipate that the uptake of the Drupal CMS by the bioinformatic community will result in a powerful new set of tools.

ACKNOWLEDGEMENTS

We would like to thank the University of Exeter for computational support, drupal.org, InsectaCentral.org users and the Lepidoptera RNAi team for guidance and end-user testing. We would also like to thank Dr Lars Jermiin and three anonymous reviewers for enhancing the quality of the manuscript. No conflicting interests exist. Author contributions: A.P. conceived, designed and programmed the software, co-ordinated and drafted the manuscript. D.H. tested the software, advised on design and drafted the manuscript.

Funding: Max Planck Gesellschaft (to A.P. and D.G.H.); the European Union Research Network GAMEXP (to A.P.); Office of the Chief Executive fellowship by the Australian Commonwealth Scientific and Research Organization (CSIRO) (to A.P.).

Conflict of Interest: none declared.

REFERENCES

Altschul,S.F. *et al*. (1997) Gapped BLAST and PSI-BLAST: a new generation of protein database search programs. *Nucleic Acids Res.*, **25**, 3389–3402.

Arnaiz,O. *et al*. (2006) ParameciumDB: a community resource that integrates the Paramecium tetraurelia genome sequence with genetic data. *Nucleic Acids Res.*, **35**, D439–D444.

Beldade,P. *et al*. (2008) Butterfly genomics eclosing. *Heredity*, **100**, 150–157.

Collins,F.S. *et al*. (2003) The Human Genome Project: lessons from large-scale biology. *Science*, **300**, 286–290.

Day,A. *et al.* (2007) Celsius: a community resource for Affymetrix microarray data. *Genome Biol.*, **8**, R112.

Drysdale,R.A. and Crosby,M.A. (2005) FlyBase: genes and gene models. *Nucleic Acids Res.*, **33**, D390–D395.

Giardine,B. (2005) Galaxy: A platform for interactive large-scale genome analysis. *Genome Res.*, **15**, 1451–1455.

Heinicke,S. *et al*. (2007) The Princeton Protein Orthology Database (P-POD): a comparative genomics analysis tool for biologists. *PLoS One*, **8**, e766.

Hubbard,T. *et al*. (2002) The Ensembl genome database project. *Nucleic Acids Res.*, **30**, 38.

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

Lyne,R. *et al*. (2007) FlyMine: an integrated database for *Drosophila* and *Anopheles* genomics. *Genome Biol.*, **8**, R129.

Mungall,C.J. *et al*. (2007) A Chado case study: an ontology-based modular schema for representing genome-associated biological information. *Bioinformatics*, **23**, i337–i346.

Ning,Z. *et al*. (2001) SSAHA: a fast search method for large dna databases. *Genome Res.*, **11**, 1725–1729.

O'Connor,B. *et al*. (2008) GMODWeb: a web framework for the generic model organism database. *Genome Biol.*, **9**, R102.

Oinn,T. *et al*. (2004) Taverna: a tool for the composition and enactment of bioinformatics work flows. *Bioinformatics* **20**, 3045–3054.

Papanicolaou,A. *et al*. (2009) Next generation transcriptomes for next generation genomes using est2assembly. *BMC Bioinformatics*, **10**, 447.

Schmid,R. and Blaxter,M.L. (2008) annot8r: GO, EC and KEGG annotation of EST datasets. *BMC Bioinformatics*, **9**, 180.

Smedley,D. *et al*. (2009) BioMart–biological queries made easy. *BMC Genomics*, **10**, 22.

Smith,V. *et al*. (2009) Scratchpads: a data-publishing framework to build, share and manage information on the diversity of life. *BMC Bioinformatics*, **10**, S6.

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

Stamatakis,A. *et al*. (2008) A rapid bootstrap algorithm for the RAxML Web servers. *Syst. Biol.*, **57**, 758–771.

Stein,L.D. *et al*. (2002) The generic genome browser: a building block for a model organism system database. *Genome Res.*, **12**, 1599–1610.

Thain,D. *et al*. (2005) Distributed computing in practice: the condor experience. *Concurr. Comput.*, **17**, 323–356.

Yang,Z. (2007) PAML 4: phylogenetic analysis by maximum likelihood. *Mol. Biol. Evol.*, **24**, 1586–1591.

Zdobnov,E.M. and Apweiler,R. (2001) InterProScan-an integration platform for the signature-recognition methods in InterPro. *Bioinformatics*, **17**, 847–848.