6.x-v0.2 Users Guide – January 22, 2010

©2009 Clemson University Genomics Institute

TABLE OF CONTENTS

License & Terms of Use 3

1 INTRODUCTION 4

1.1 WHY Use Tripal 4

1.2 AUDIENCE 4

1.3 DEPENDENCIES 5

1.3.1 DRUPAL 6 5

1.3.2 DRUPAL Terminology 5

1.3.3 DRUPAL Help 5

1.3.4 GMOD chado 1.0 6

1.4 VERSIONS 6

1.5 LIMITATIONS 6

2 Quick start 7

3 INSTALLATION & setup 8

3.1 DATABASES 8

3.2 SETUP of chado & Drupal 8

3.3 CONFIGURATION 9

3.4 DOWNLOAD and Installation of Tripal modules & Theme 9

3.5 PERMISSIONS 10

4 TRIPAL Modules 11

4.1 OVERVIEW 11

4.2 TRIPAL Core 12

4.3 TRIPAL Organism Module 12

4.4 TRIPAL Feature Module 18

4.5 TRIPAL Library Module 23

4.6 Tripal Analysis Module 23

4.7 TRIPAL Analysis Blast Module 24

4.8 TRIPAL Analysis Unigene Module 27
5 THEMING the site 28
5.1 Tripal Theme 28
5.2 ENABLING the Tripal Theme 28
5.4 CUSTOMIZING Content Pages 29
6 THEMING other GMOD or 3rd Party Tools 32
7 JOBS Management 34
7.1 Submitting and Viewing Jobs 34
7.2 Job STATUS 35
7.3 Running tripal jobs 35
7.3.1 Command-line 35
7.3.2 Cron 35
7.4 Debugging job errors 36
8 Taxonomy Management 36
9 CVTerm -- For specific module functionality 37
10 Indexing Content For Searching 38
11 Materialized Views 38
11.1 Managing Existing Views 39
11.2 Creating a New View 40
12 Re-indexing Content for Searching 41
13 Customizing Tripal 42
13.1 Writing Your Own Modules 42
13.2 Using CCK to Customize Tripal Content 42
13.3 Purchasing Customizations 42
13.4 Contributing Your modules 42

LICENSE & TERMS OF USE

Use of Tripal is governed by the open-source GNU Public License Version 3 (GPLv3) license. This license should always accompany the Tripal package and can be found in the LICENSE.txt file in the main directory of the package. Alternatively, more information about the GPL v3 license can be found here: http://www.gnu.org/copyleft/gpl.html.

This release of Tripal is copyrighted by Clemson University and developed by the Clemson University Genomics Institute (http://www.genome.clemson.edu/).

1 INTRODUCTION

Tripal provides a set of modules for construction and management of genomic databases. The goal for Tripal is to decrease development time of these websites, simplify management of the genomic data and other web content, facilitate construction of easy-to-use and attractive websites, and allow for expansion by community contribution.

1.1 WHY USE TRIPAL

With ever increasing volumes of genomic data, the need for an easy to install, customizable, highly tailored genomic website is increasing. Additionally, the need for a common framework to share genomic data is also increasing. Tripal attempts to address these issues by drawing on the strengths of Drupal and GMOD Chado. Both are actively developed, well documented and well supported.

The time required for development of any website using Drupal is greatly reduced. Drupal is easy to install, configurable and expandable. Users can and have written PHP modules using the Drupal application programmer interface (API) to extend the functionality of the software. Hundreds of user-contributed modules are available for download on the Drupal websites.

Chado provides a highly structured means to store genomic data. Its strength lies in its ability to accurately represent relationships between genomic features, analyses, genomic libraries, etc. It also provides a method for data sharing by providing a common framework for data storage.

Tripal combines the strengths of Drupal and Chado simplifying construction of genomic websites. Additionally, Tripal provides an API that allows anyone the ability to easily create modules that provide new or different presentations of data. When shared with the community, these modules can be "plugged" into any Tripal installation.

1.2 AUDIENCE

Tripal is intended for use by end-users with experience installing, loading and working with the Chado database who desire an easy to use web interface for their communities. Users who desire customizations beyond the default capabilities and who wish to perform their own customizations of Tripal will need a programmer with PHP and Drupal experience. Users who do not have the necessary resources or who would rather outsource

customizations can contact The Clemson University Genomics Institute which offers services for hosting and customization of Tripal-based websites: http://www.genome.clemson.edu/.

1.3 DEPENDENCIES

1.3.1 DRUPAL 6

Drupal is an open-source content management system popular with web designers. Drupal can be found online at http://drupal.org/. It allows for quick development of robust websites, easy manipulation of web content by non-technical users, and a large repository of contributed modules for adding common features. Drupal also provides management of user authentication and access permissions. Drupal is compatible with both MySQL and PostgreSQL. Both are popular open-source database applications.

The current version of Tripal requires Drupal 6.

1.3.2 DRUPAL TERMINOLOGY

The following is list of terminology commonly used with Drupal and used through this document:

- Node: a node is, in short, a web page. Tripal can create a node for each feature, organism, library and analysis.
- Taxonomy: categories or tags can be assigned to each node. These categories allow for advanced searching of content within the Drupal website. Taxon
- Module: Drupal modules consists of a set of PHP files that act together to provide a specific functionality. A list of user-contributed modules can be found on the Drupal website. Tripal is a suite of modules.
- Theme: the look-and-feel of the website can easily by changed by using "Themes". User-contributed themes can be found on the Drupal website. Each theme provides a different look-and-feel. Tripal should be used with another theme, but also provides some "theming" of its own.

The Drupal Menu

The menu tree in Drupal is represented in this document with menu items separated by '->' characters. For example, the Tripal management utilities are found under the Drupal Administration menu. This is represented as "Administration -> Tripal Management" in this document. As another example, the Blocks configuration utility menu item would be represented as "Administration -> Site Building -> Blocks".

1.3.3 DRUPAL HELP

Site administrators will interact with Tripal and Drupal as a content administrator, a programmer or both. The content administrator uses the Drupal interface for managing content on the website. The site programmer manipulates the underlying PHP code. Below is a set of references for working with and programming Drupal.

For the Content Administrator

Online Drupal documentation:

- Content admin documentation: http://drupal.org/handbooks
- Getting started guide: http://drupal.org/getting-started

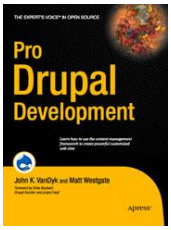
Books

- *Drupal 6*, by David Mercer (ISBN 1847192971)

For the Programmer

Online Drupal documentation:

1. Contributor Guide: http://drupal.org/contributors-guide
2. Drupal API: http://api.drupal.org/
3. PHP Reference: http://www.php.net/docs.php



Books

4. *Pro Drupal Development*, by John K. VanDyk, Matt Westgate (ISBN: 1590597559)

1.3.4 GMOD CHADO 1.0

GMOD, the Generic Model Organism Database, project provides a suite of open-source tools for working with genomic databases--large or small. GMOD can be found online at http://gmod.org/wiki/Main_Page. GMOD chado provides a community-supported database relational database schema for PostgreSQL called Chado which provides a platform for storing biological data including genomic features (e.g. genes, chromsomes, mRNA, etc), organisms, computational analysis, annotations, phenotypic data, etc.

1.4 VERSIONS

This document provides usage instructions for Tripal version 6.x-0.2. This is the second release of Tripal. Versioning consist of two sets of numbers separated by dashes. The first being the Drupal version and the second being the Tripal milestone number. The Drupal number indicates the Drupal version compatible with Tripal.

Tripal consist of a suite of Drupal modules, each with their own version numbers. Updates and new releases of modules may occur more frequently than milestone releases of the entire Tripal package. Therefore, Tripal modules consist of three sets of numbers separated by dashes: the first being the Drupal version for which the module is compatible, the second being the Tripal version for which the module is compatible and the third being the module release number (e.g. 6.x-0.2-m0.1)

As new versions of Tripal and its modules are released the respective version numbers are incremented.

***Note*:** A Drupal version of 6.x indicates that Tripal is compatible with all versions of Drupal 6.

1.5 LIMITATIONS

Version 6.x-v0.2 is the second public release of Tripal and as such provides a growing, yet still limited, set of tools for development and management of a Chado/Drupal based genomic website. There are many features which need adding. Every best effort has been made to construct Tripal for a generic audience. However, the software has the following limitations:

- Currently, Tripal does not provide an interface for bulk loading of data into the Chado database. Future releases of Tripal will most likely contain some interface for this, but for now all large-scale data entry must be performed using the existing command-line Chado tools.
- To allow for presentation and searching of Chado content in Drupal, each feature, library, organism, etc must by "synced" with Drupal as well as "indexed" for searching. For instances of Chado with a large number of features this process may take as long as several days depending on the amount of data and the speed of the server. For small sites these tasks will complete quickly.
- Currently, only some Chado tables are supported by Tripal. It is anticipated that modules will be developed for later releases of Tripal that will provide support for other tables. However, due to Drupal's and Tripal's open API any end-user can create new modules that provide missing functionality. These modules can "plug" in to Drupal and Tripal and may be contributed back to the Tripal user-base for others to use.
- Not every type of analysis or method of presentation of those analyses can be anticipated by Tripal developers. It is anticipated that this limitation can be alleviated as the Tripal core developers release new analysis modules, but also as users of Tripal create new analysis modules and share those back to the community.
- Bugs! The software is in beta form and therefore may have undesirable behavior. Caution should be taken to ensure proper backups and protection of the underlying Chado database.

2 QUICK START

For users generally familiar with Drupal, Chado and Tripal the following steps provide a general guide for quick setup of a Tripal-based website. The following steps provide the easiest method for site setup:

1. Install and populate the chado tables
2. Download and Install Drupal
3. Download the Tripal package
4. Copy and uncompress the Tripal modules into the ./sites/all/modules
5. Copy and uncompress the Tripal theme into the ./sites/all/themes directory
6. Setup the base theme:
    a. Option 1: Download and unpack the default Admire Gray theme
    b. Option 2: Choose your own base theme and update the tripal.info file
7. Enable the Tripal theme (no need to enable the base theme)
8. Use Drupal block administration to locate the Drupal and Tripal blocks as desired. Turn on the organism and library blocks if desired.
9. If desired, use the Drupal menu administration to remove Tripal menu items from the "primary links" menu if Drupal.
10. Enable the Tripal cron for automatic launch of queued jobs.
11. Use the Tripal administration to sync organisms.

12. Use the Tripal administration to set various options for the feature including the accession prefix, feature types for syncing, taxonomy, etc
13. Use the Tripal administration to sync features, libraries and analyses (if desired). Depending on the number of features this step may take a while to complete. Check the job status page periodically for sync status.
14. Check the Search page. Do all advanced search methods exist? If not try resetting the Taxonomy on the feature administration page.
15. If desired, use the Tripal administration pages to update the materialized views for sequence stats to appear on the organism and library pages.

Customizations:

- Write new analysis modules that integrate with Tripal to display analysis types not currently supported or to add different or new functionality as needed for your site. Remember to share your modules with the community (see http://www.genome.clemson.edu/tripal for further information)!
- Propose new features and/or new modules to the Tripal development team.
- Adjust the look-and-feel of your site by customizing the page templates, CSS and Javascript of your base theme and/or the Tripal sub theme.

3 INSTALLATION & SETUP

3.1 DATABASES

Drupal is compatible with both MySQL and PostgreSQL databases. Chado and therefore Tripal are compatible with PostgreSQL. Before installation of a Tripal-based genomic website choose an infrastructure and layout that best matches your needs. Your database administrator will need to create two databases (either both in Postgres or one in MySQL for Drupal and another in PostgreSQL for Chado) and provide the following connection information:

1. The database names for Drupal and for Chado.
2. The database server name for Drupal and for Chado.
3. The usernames to log on to the Drupal and Chado databases
4. The passwords to log on to the Drupal and Chado databases

Note: It is not recommended to install Chado and Drupal within the same database schema.

3.2 SETUP OF CHADO & DRUPAL

Setup and installation instructions for both Drupal and Chado can be found on their respective websites. Links and other reading material can be found in the Introduction section of this document. Tripal can be installed and enabled only after both databases are created and Drupal is installed.

Currently, Tripal does not support bulk loading of data into Chado. Therefore, it is recommended to use the existing bulk load tools that accompany the Chado package to load and prepare the Chado database. Tripal

provides support for adding single records such as a single feature, organism or library and allows editing of individual records, however bulk loading of data is preferred.

3.3 CONFIGURATION

Drupal must know how to connect to the Chado database. To do this, edit the settings.php file found in the <Drupal installation path>/sites/default directory, and change the following line:

```
$db_url = 'mysql://<username>:<pass>@<server>/<dbname>';
```

To this:

```
$db_url = array(
  'default' => 'pgsql://<dusername>:<dpass>@<dserver>/<ddbname>',
  'chado' => 'pgsql://<cusername>:<cpass>@<cserver>/<cdbname >',
);
```

Where the following tags in the code above should substituted accordingly:

<dusername>:	the username to log on to the Drupal database
<dpass>:	the password to log on to the Drupal database
<dserver>:	the Drupal database server name
<ddbname>:	the Drupal database name
<cusername>:	the username to log on to the Chado database
<cpass>:	the password to log on to the Chado database
<cserver>:	the Chado database server name
<cdbname>:	the Chado database name

This change creates an associative array with two entries that provides connection information for both the Drupal and Chado databases respectively.

Critical: It is essential that the connection name 'chado' be used for the second database line in the code above. Tripal requires that the connection name be 'chado'.

3.4 DOWNLOAD AND INSTALLATION OF TRIPAL MODULES & THEME

Tripal can be installed after Drupal is functional and the Chado database is available. Tripal can be installed following the typical method for installation of Drupal modules. These steps include:

1. Download the desired modules from the Tripal source repository.
2. Copy the Tripal module files into the <Drupal installation path>/sites/all/modules directory of your Drupal installation. Create the directory if it does not exist.
3. Unpack each Tripal module using the 'tar' command: tar -zxvf tripal-6.x-0.2b.tar.gz.
4. Move the theme_tripal directory into the <Drupal installation path>/sites/all/themes directory

5. Log into the Drupal site as an administrator, navigate to the modules administration (Administer -> Modules) page and enable the Tripal modules. Figure 1 below shows an example list of modules available.
6. Enable the modules in the following order:

   Search, Path, Tripal Core
   Tripal CV
   Tripal DB
   Tripal Organism
   Tripal Feature
   Tripal Search
   Tripal Library
   Tripal Analysis
   Tripal Blast, KEGG, Interpro & Unigene

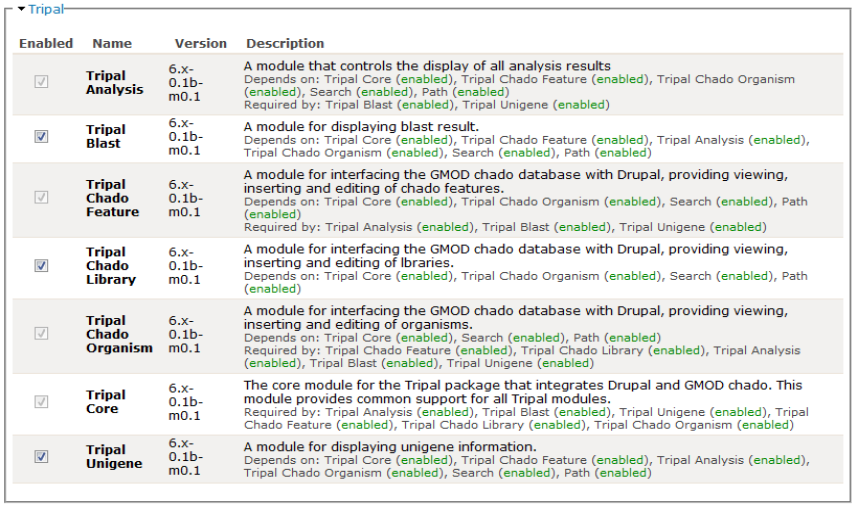


Figure 1

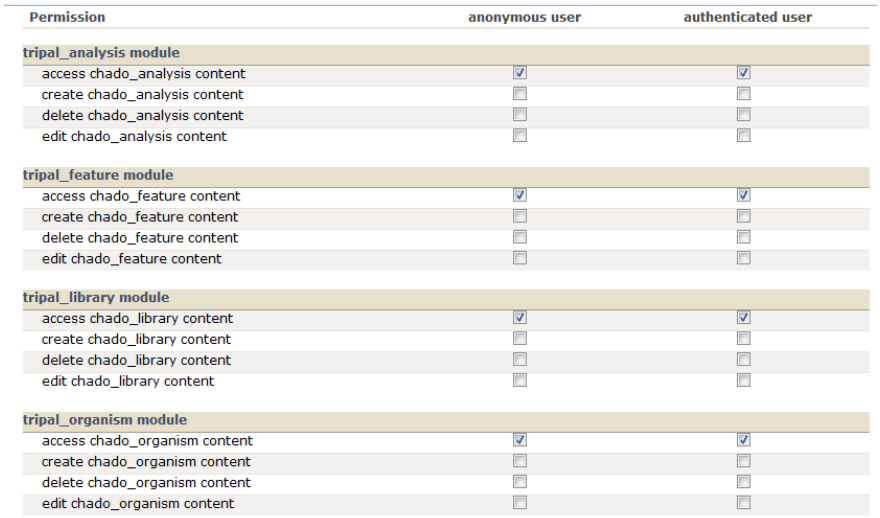
It is not required to install all Tripal modules. Only those desired and those that serve as dependencies should be enabled. In all cases, the Tripal core module must be enabled because it serves as a dependency for all other modules. Follow the instructions in section 4 for configuring each module.

Additionally, Tripal is accompanied with a Drupal theme that provides a default look-and-feel for the display of the Chado data. It is recommended to use this theme the first time Tripal is used, and customizations can be made at a later time. The Tripal theme cannot function properly on its own. It must be used in conjunction with another theme of your choice. See section 5 titled 'Tripal Theme' for further instructions. To install the Tripal theme follow these steps:

3.5 PERMISSIONS

Drupal manages access to content by providing a set of permissions that can be enabled for user "roles". Users can then be assigned to specific roles that provide access to view content and perform certain tasks. Tripal modules provide their own set of permissions. Permissions can be viewed and set by selecting the "Administer ->

User Management -> Permissions" menu item. Figure 2 below shows an example set of permissions provided by the Tripal modules.



Permission	anonymous user	authenticated user
tripal_analysis module		
access chado_analysis content	☑	☑
create chado_analysis content	☐	☐
delete chado_analysis content	☐	☐
edit chado_analysis content	☐	☐
tripal_feature module		
access chado_feature content	☑	☑
create chado_feature content	☐	☐
delete chado_feature content	☐	☐
edit chado_feature content	☐	☐
tripal_library module		
access chado_library content	☑	☑
create chado_library content	☐	☐
delete chado_library content	☐	☐
edit chado_library content	☐	☐
tripal_organism module		
access chado_organism content	☑	☑
create chado_organism content	☐	☐
delete chado_organism content	☐	☐
edit chado_organism content	☐	☐

Figure 2

Two roles exist by default: anonymous user and authenticated user. To allow permissions, simply check the boxes for the respective permission to be granted and click the "Save Permissions" button at the bottom of the page. It is recommended that site administrators establish additional roles using the "Administer -> User Management -> Roles" menu item to allow for fine-grained control of management of Tripal content.

4 TRIPAL MODULES

4.1 OVERVIEW

Tripal currently consists of a set of modules for integrating organism, features, libraries and analysis data. More modules may be available in the future as they are developed. Table 1 below lists the modules available with Tripal v0.2, indicating the Chado tables from which they extract data and a brief description

Table 1. Modules provided by Tripal

Tripal Module Name	Chado Tables	Description
Tripal Core	None	The Tripal Core module is required before using any of the other Tripal modules. It provides management for jobs, materialized views and an Application Program Interface (API) for common functionality needed by all other modules.
Tripal Organism	organsim	The Tripal Organism module provides an interface for creating, editing and viewing pages for organisms.
Tripal Feature	feature, dbxref, feature_synonym, feature_dbxref	The Tripal Feature module provides an interface for creating, editing and viewing pages for genomic features.
Tripal Library	library,	The Tripal Library module provides an interface for creating,

	library_feature,	editing and viewing pages for genomic libraries.
Tripal Analysis Core	analysis, analysis_feature	The Tripal analysis module does not generate content but provides "core" functionality for all analysis modules. Developers of new analysis modules would depend on this module to integrate their new analysis modules.
Tripal DB	db	The Tripal DB module provides a simple interface for adding and editing external databases.
Tripal CV	cv, cvterm, cvtermpath	The Tripal CV module provides an API for constructing browseable trees and pie charts for controlled vocabularies. Additionally, it comes with a tree browser for viewing any ontology.
Tripal Search	feature	Provides a link on the search results page that allows users to download a FASTA file of results.
Tripal Blast		
Tripal Interpro		
Tripal KEGG		
Tripal Unigene		

4.2 TRIPAL CORE

The core module is required before use of any other Tripal module. It provides behind-the-scenes functionality for all other modules as well as an API for the construction of new Tripal modules. This module provides management of Jobs, taxonomy, controlled vocabulary terms and materialized views (all discussed later in this guide). See sections 7-9 for further information functionality provided by the core module.

4.3 TRIPAL ORGANISM MODULE

The organism module is responsible for the display and management of data from the Chado organism table. Tripal provides viewing and editing of organisms as well as an administrative page that allows for “syncing” and “indexing” of all or some of the organisms in Chado. Only those organisms that are synced and indexed will appear on the website. This allows for administrators to hide some organisms from view. Instructions for syncing and indexing organisms are found below. When enabled, the Tripal organism module adds organism pages for each synced organism, a Drupal block, an “Organism” menu items, an organism summary page, and a configuration page.

The Organism Page

The organism page provides organism details, an image. Optionally, it can contain an overview of the types and number of features (including a pie chart) associated with the organism. The organism page can also optionally include an interface for browsing features assigned to the organism. While this may not be the most effective way for find content of interest, it does provide an "introduction" for new users that allow them to browse the site and

become familiar with how features are display and the information available to them. Both the feature summary and feature browser can be disabled. Figure 5 below provides an example organism page.

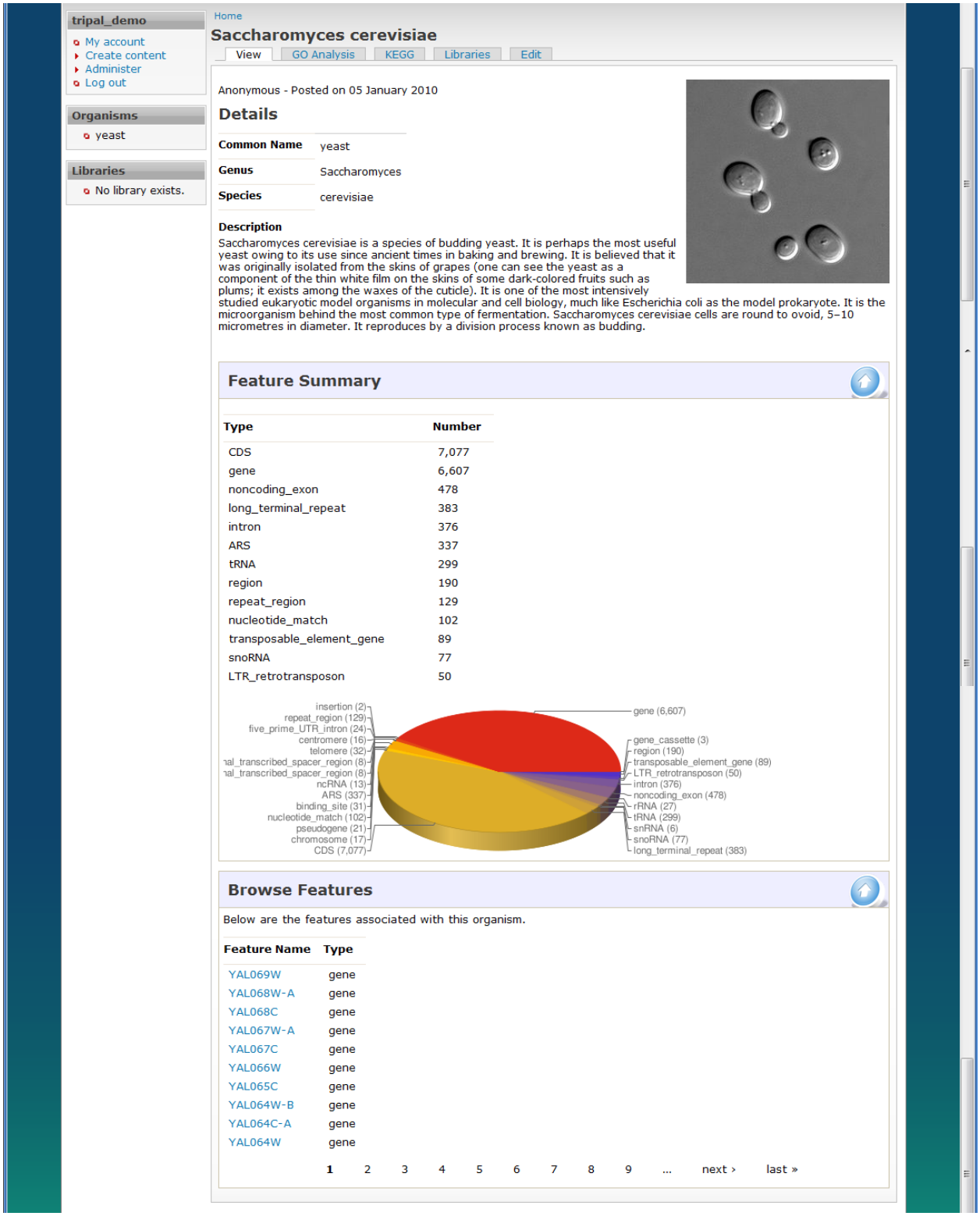


Figure 3

The Organism Block, Menu and Summary Page

In Drupal, blocks are used to summarize content by providing links to pages of high interest. Typically blocks are found along the periphery of the site. An example of an organism "block" can be seen on the left hand side in Figure 4 below. Additionally, an "Organisms" menu item is located within the primary menu of the site. When clicked, the organism summary page appears. The menu item and an example of the summary page can be seen in figure 4 below. Both the block and the menu item can be relocated or removed using the Drupal administrative interface. Additional content may be added to this page by other modules. For example, the library module will add a new section showing which libraries are associated with an organism. The "Organisms" link, available just below the Tripal header on the image, will take users to a page listing all organisms available on the website.

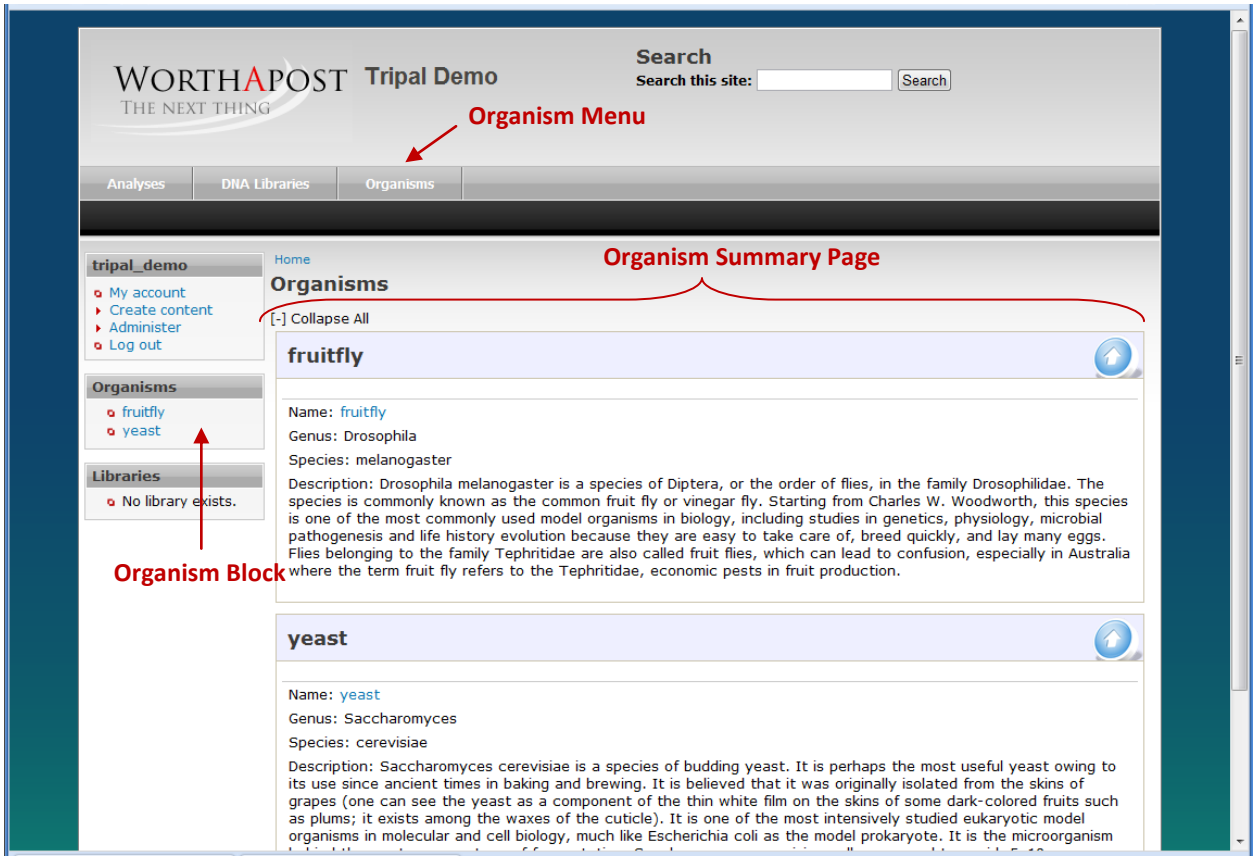


Figure 4

Adding a New Organism

An organism can be added to the site by an authenticated user with proper permissions. To do so, click the Drupal menu item titled 'Create Content'. On the page that appears, click the "Organisms" link. A form will appear requesting information such as the genus, species, abbreviation, common name, description and an option image of the organism. Fill out the required fields and click 'Save'. Figure 5 shows the "Create Organism" page. Once saved, the new organism is added to Chado and a new page is added to Drupal.

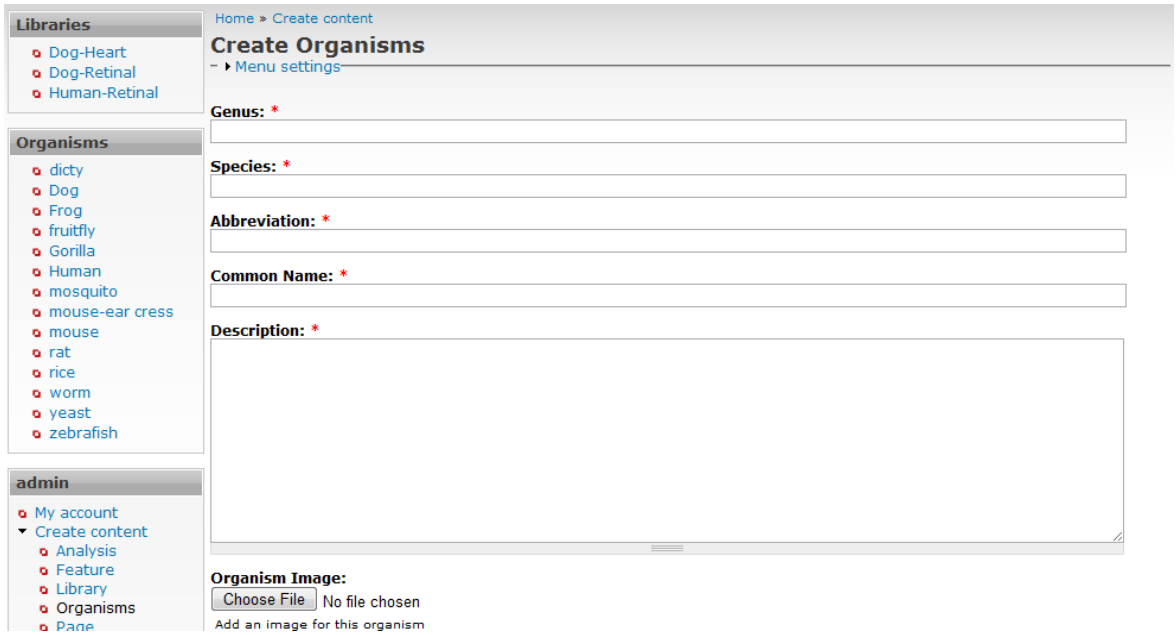


Figure 5

Editing an Organism

Authenticated users with proper permissions can edit the organism. Users can change the common name, genus, species, abbreviation and description as well as add a picture to the page. When saved, these edits are saved in the Chado database. To edit a page, click the 'Edit' tab that appears just above organism description as seen in Figure 6. This tab is available only to authenticated users with proper permissions. Changes to the organism are saved to Chado database and the organism page reflects those changes.

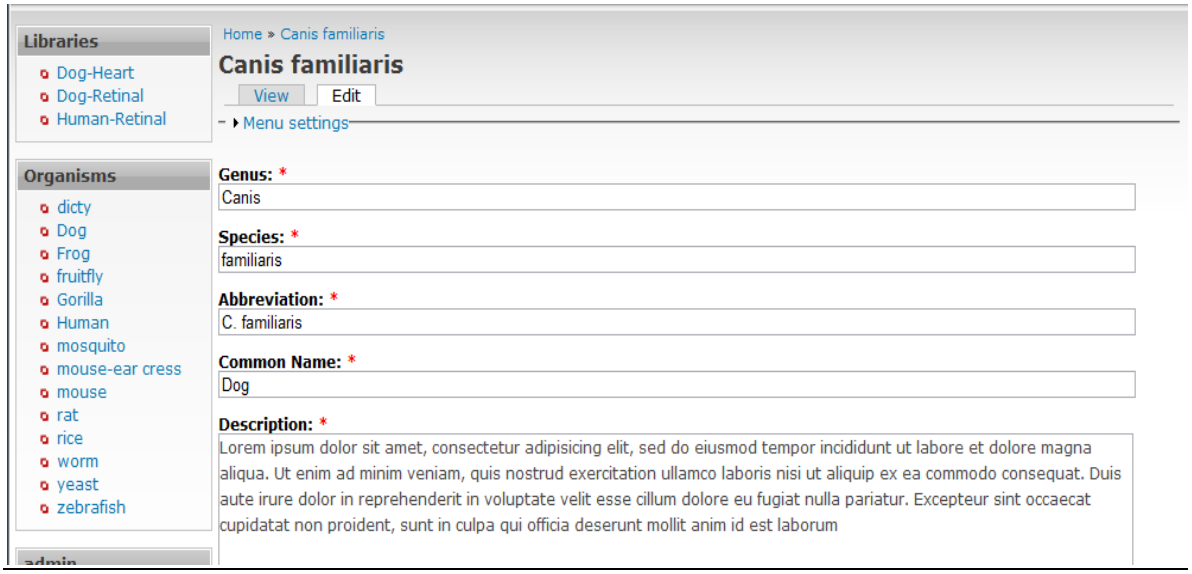


Figure 6

Organism Module Administration Page

The organism module provides an administrative/configuration page that provides functionality for syncing organisms, indexing organism features, setting taxonomy for organism features and database cleanup. This administrative page can be found by selecting the 'Administer' -> 'Tripal Management' -> 'Organisms' menu item.

Syncing an Organism

Organisms that exist in the Chado database prior to installation of Tripal can be visible through Tripal through a "syncing" process. When an organism is synced a new web page is created in Drupal and a link is made between the new page and the Organism in Chado. This Drupal page is then available for access online and searching. Figure 7 shows the section on the administrative page where this can be performed. Select the check boxes for each organism to be synced and clicking the 'Sync' button. Organisms that are not synced will remain in the Chado database but will not have web content created in Drupal. Additionally, Drupal does not store information about the organism. Information remains in the Chado database.

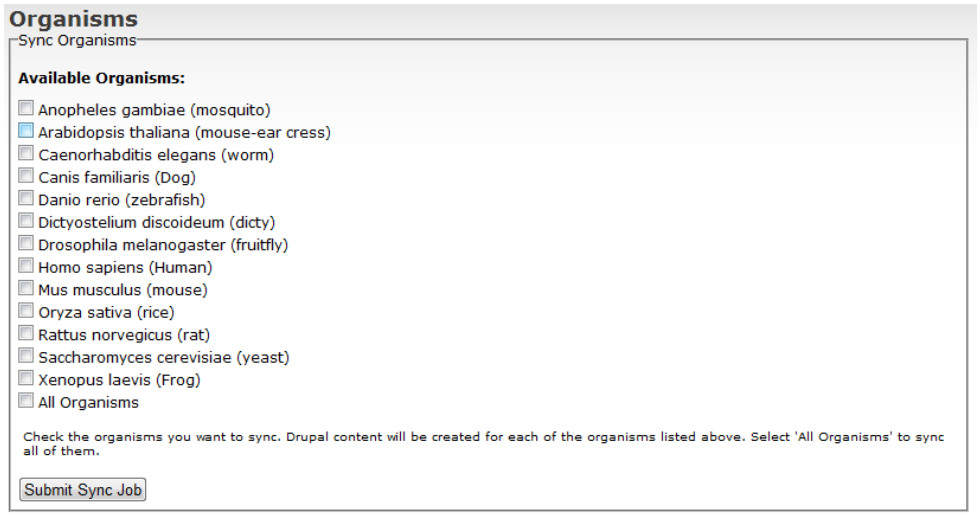


Figure 7

Indexing an Organism's Features

Features associated with organisms (e.g. genes, CDS, mRNA, etc.) may have their own web pages as well. More information regarding syncing of features can be found in the next section. When content is "indexed" the page content is extracted and ordered for searching. Drupal provides a full-text searching capability that allows a user to search for any word on the page. Indexing is the process by which Drupal ranks and stores those words for searching. Pages that are not indexed will not be available for searching. The Tripal feature module has an administrative page very similar to the organism administrative page that allows the user to index all features on the site for searching. However, the same functionality is provided on the organism administrative page to allow for indexing of features by organism rather than all features on the site. Figure 8 shows the section of the page

where indexing can be performed. Select only those features that should be indexed. Only features that have been previously synced using the Tripal Feature module administrative page will be indexed.

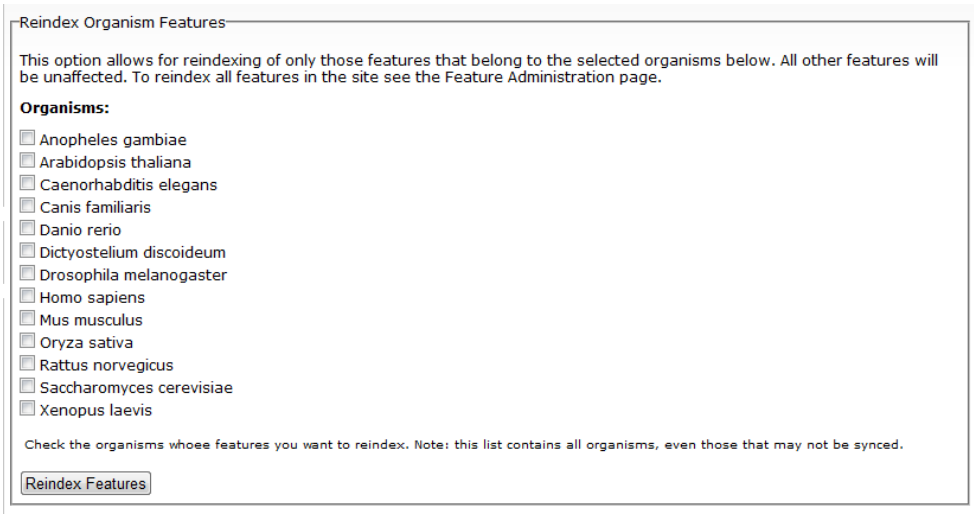


Figure 8

Note: Drupal will not use the new indexing for searching if 100% of the site content is not indexed. Once the site is 100% indexed all content is searchable.

Organism Feature Taxonomy

Cleanup

Because Drupal and Chado share different database schemas no data integrity constraints exist between the two. It is therefore possible that an organism in the Chado database can be deleted without Tripal recognizing the deletion. The side-effect would result in an organism page with no content. This could happen if someone deleted the organism from Chado without using Tripal. In this case the Drupal page would still exist for the organism that no longer exists. The organism administration page provides a "cleanup" utility that corrects this type of discrepancy. Figure 9 below shows the section of the organism administration page where cleanup can be performed. To perform a cleanup operation simply clicks the 'Clean up orphaned organisms' button. All Drupal content for deleted organisms will be removed from the Drupal database.

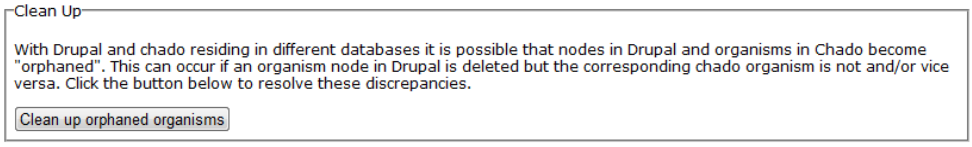


Figure 9

4.4 TRIPAL FEATURE MODULE

The feature module is responsible for the display and management of data from the Chado feature table. Tripal provides viewing and editing of features as well as an administrative page that allows for syncing and indexing of all or some of the features in Chado. A view of an example feature page is shown in Figure 10 below.

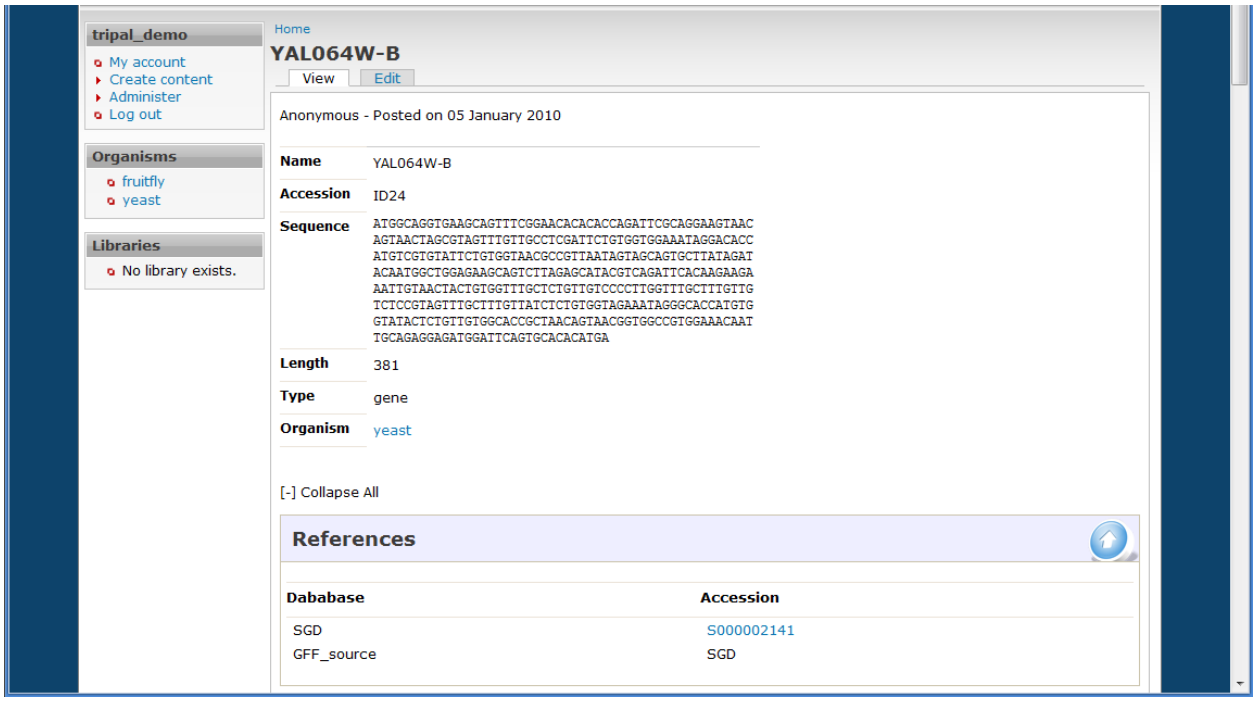


Figure 10

The feature page provides by default the feature name, type, accession number within this site, the residues, length, organism and if applicable the library to which the sequence belongs. Below this section will appear additional information about the feature as available. For instance, the Tripal unigene analysis module will automatically place a box below the details if this feature is part of an alignment to a contig.

Adding a New Feature

In most cases a bulk load of feature data will be required to add new features to the Tripal website. At the current time, Tripal does not support bulk data loads and site-administrators should use the bulk load tools that come with Chado. It may be necessary at times to add a single feature. A feature can be added to the site by an authenticated user with proper permissions. To do so, use the Drupal menu item 'Create Content -> Features'. A form will appear requesting information such as the feature name, unique name, residues, etc. Fill in the required fields and click 'Save'. Figure 11 below shows the page for adding a new feature. Once saved, the new feature is added to Chado and a new page is added to Drupal.

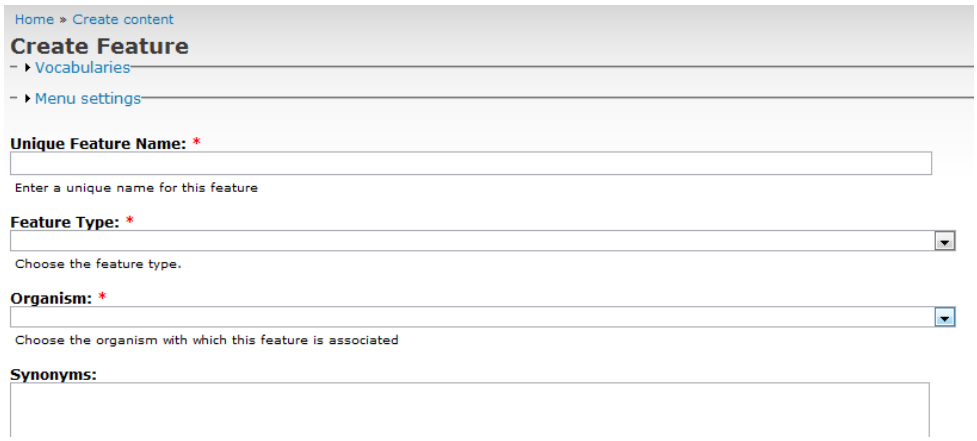


Figure 11

Editing a Feature

Authenticated users with proper permissions can edit any feature. When saved, these edits are saved in the chado database. To edit a feature, click the 'Edit' tab that appears above the feature details as seen in Figure 12. This tab is available only to authenticated users with proper permissions. Changes to the features are saved to the chado database and the feature page reflects those changes.

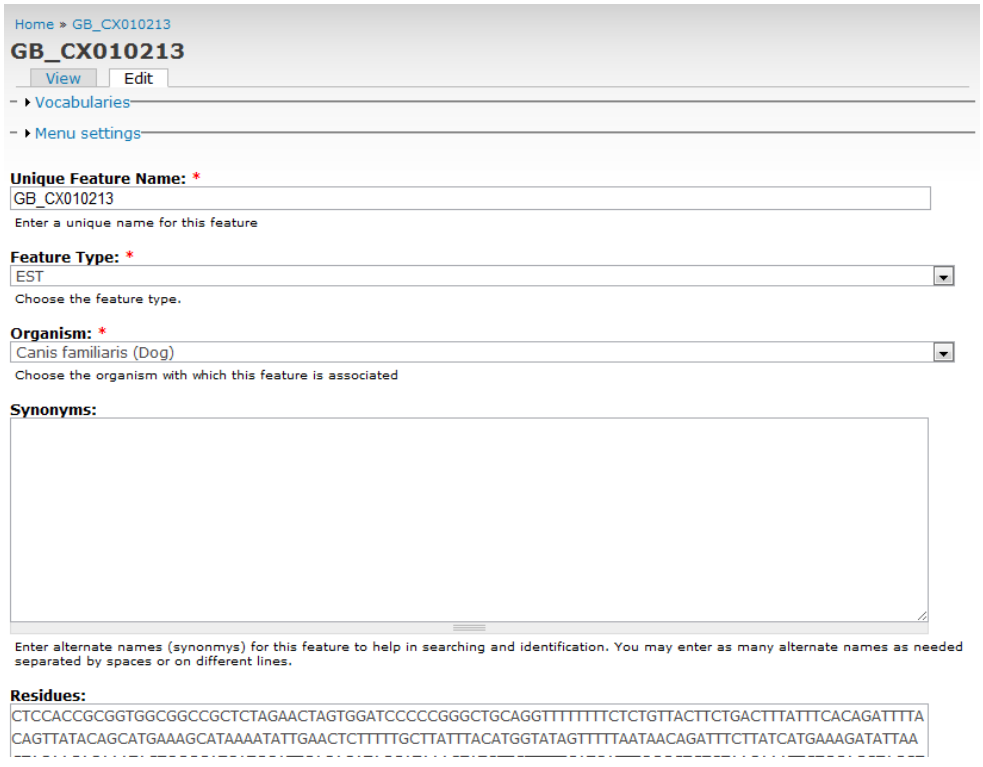


Figure 12

Feature Accession Numbers

Currently, Tripal assigns unique accession numbers to all synced features. These accession numbers are used for easy access to features within the web site. The accession numbers use a custom prefix followed by the chado feature_id number. For example, the default prefix for a Tripal site is simply 'ID'. For a feature with a feature_id of '29348', the accession number assigned by Tripal will be 'ID29348'. This accession number can be used to easily access the feature from the web. For instance, for an imaginary website URL http://www.mygenomesite.org/, the feature would be accessible by adding the accession to the URL: http://www.mygenomesite.org/ID29348.

The accession prefix can be modified by using the Tripal feature administration page found by clicking 'Administer -> Tripal Management -> Features". Figure 13 below shows the form element that allows for setting the accession prefix and the Feature administration page.

Figure 13

Important: once the accession prefix is set, it should not be changed. Please set the prefix before syncing any features.

Setting Feature "Taxonomy"

Drupal provides a method for categorizing content by allowing users to assign "taxonomy" or categorical terms to Drupal nodes. These categories are then available in the advanced search for filtering of search results. Tripal provides an interface for automatically assigning taxonomy (or categories) to all features. The categories that may be assigned to features are: organism, feature type, analysis name and library name. For instance if the organism category is enabled, all features will be assigned a category with the genus and species of the organism to which they belong. This will allow filtering of features by organism when searching. Figure 14 below shows the section of the feature administration page where automatic assignment of taxonomy can be enabled for features. To enable taxonomy for features, click the checkboxes for the categories desired, and click the 'Save Configuration' button at the bottom of the page (not shown in the figure below). This will save the settings without setting taxonomy. To save the settings and set taxonomy for all features click the "Set/Reset Taxonomy for all feature nodes" button.

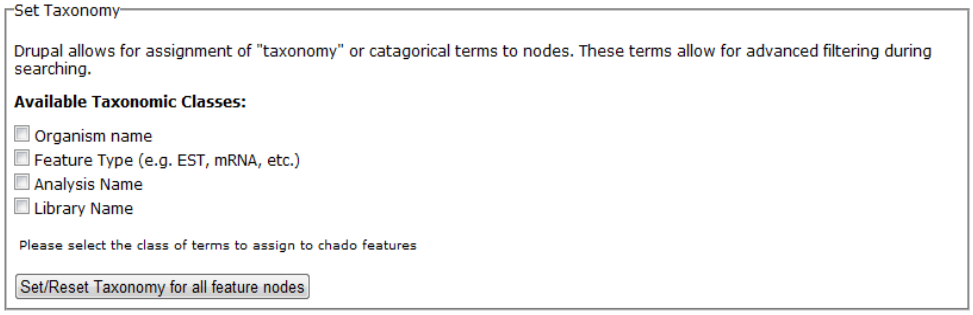


Figure 14

Syncing Features

Features that exist in the chado database prior to installation of Tripal can be added to Drupal through a "syncing" process. When a feature is synced the information about the feature, such as the name, residues, feature type, etc are pulled from the Chado database and a new Drupal page is created. This Drupal page is then available for access online, searching, etc. just like any other Drupal "node". Features can be synced on the feature administration page of Tripal, available by clicking 'Administer -> Tripal Management -> Features'.

Important: before any features are synced the user should first set the access accession prefix and the taxonomy categories as described in the previous sections.

Chado is capable of housing many types of features, however, content pages for all features may not need to be created. For instance, using the Sequence Ontology (SO), EST alignments to a contig are represented in chado using an intermediary EST_match feature. The site-administrator may want to show the contigs and ESTs but not the EST_match features. To create content for specific feature types, enter the SO terms in the 'Feature Types' text box on the feature administration page. Only features of the specific type will be "synced" with Drupal. Finally, click the 'Sync all features' button to sync the features. Only features that belong to organisms that are already synced will be synced. A feature of the desired types, that belongs to an organism that is not synced will not be synced. Figure 15 below shows the section of the administration page where setting the feature types and syncing can be performed.

Figure 15

Clicking the 'Sync all Features' button will generate a Tripal Job that can be monitored on the Jobs administration page (see the Tripal Jobs section below for more details). For sites with a large number of features it may take quite a long time to complete syncing (potentially several days). However, once features have been synced they remain synced.

In the event that new features are added after an initial sync, only those features that have not been synced which are new, will be synced the next time the button is clicked. Taxonomy will automatically be assigned to each feature (if enabled) when a feature is synced. In the event that features are synced before the taxonomy is

enabled, the site-administrator may reset all taxonomies by clicking the "Set/Reset Taxonomy for all feature nodes" after the sync has completed.

Indexing Features

When a feature is "indexed" the feature page content is extracted and content is ordered for searching. Drupal provides a full-text searching capability that allows a user to search for any word on the page. Indexing is the process by which Drupal ranks and stores those words for searching. Features in chado that are not indexed will not be available for searching.

When a feature is synced the indexing is automatically performed. However, feature pages can be "reindexed" at any time using the "reindex" facility on the feature administration page. This is useful in the event that the content on the feature pages change. For instance if content in the chado database is changed on the command-line or through directly editing the database, or if an update is released for the Tripal feature module that changes the presentation of the features, Drupal would not be aware of the changes and hence any new text on the page would not be searchable. Reindexing the organisms will correct this. Figure 16 below shows the section of the feature administration page where reindexing can be performed.

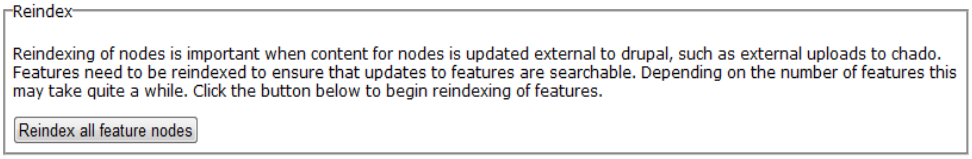


Figure 16

Note: After the initial syncing or reindexing of features, the content becomes available for searching. However, Drupal will not use the new indexing if 100% of the site content is not indexed. The Drupal cron will automatically handle indexing of new pages. But be aware that if after syncing or reindexing that the chado content cannot be found when searching, check to ensure that the entire site is 100% indexed. This can be determined under the Search Settings of the Site Configuration menu of Drupal. If not, try to manually run the Drupal cron. Once the site is 100% indexed all content is searchable.

Cleanup

Because Drupal and chado share different database schemas no data integrity constraints exist between the two. It is therefore possible that a feature in the chado database can be deleted without Tripal recognizing the deletion. The side-effect would be a feature page with no content. In this case the Drupal page would still exist for the feature that no longer exists. The feature administration page provides a "cleanup" utility that corrects this type of discrepancy. Figure 17 below shows the section of the feature administration page where cleanup can be performed. To perform a cleanup operation simply click the 'Clean up orphaned features' button. All Drupal content for deleted features will be removed from the Drupal database.

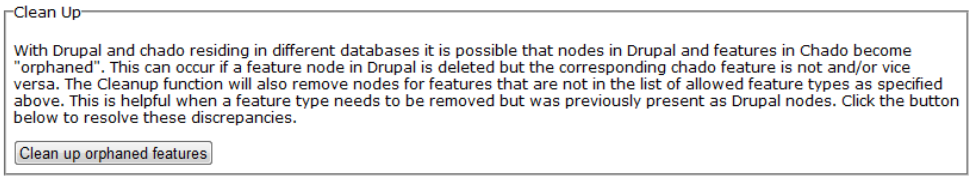


Figure 17

4.5 TRIPAL LIBRARY MODULE

The library module is responsible for the display and management of data from the Chado library table. Tripal provides viewing and editing of libraries as well as an administrative page that allows for syncing and indexing of all or some of the libraries in Chado. Editing, and administration of libraries is similar to organisms and features. See the Tripal Feature Module and Tripal Organism Module sections for more detail information about syncing, indexing assigning taxonomy and cleanup operations which are also available with the library module.

The library module generates a library "block" which provides a brief listing of all libraries and can be located anywhere on any page. Figure 18 below shows this block on the left hand side of the page. The module also adds a "DNA Libraries" menu item to the primary menu of Drupal which links to a page showing a complete listing of all synced libraries. Both the block and the menu item can be relocated or removed using the Drupal administrative interface.

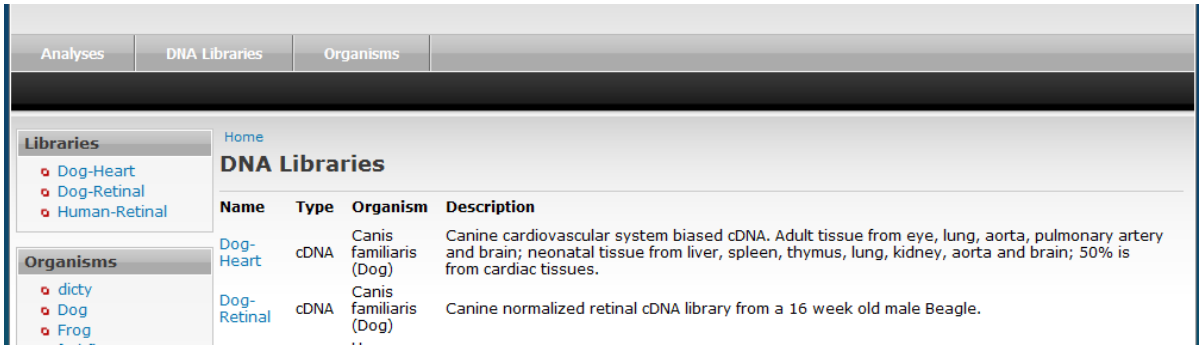


Figure 18

4.6 TRIPAL ANALYSIS MODULE

The analysis module provides core functionality for all other analysis modules. The two analysis modules that accompany the 6.x-0.1b release of Tripal (the blast and unigene analysis modules) are dependent on the analysis core module. It is not possible for the core developers to anticipate every type of analysis implemented nor the desired method to display the results. It is anticipated that community developers will create their own analysis modules as needed to support the needs of their communities. The two modules that accompany Tripal were created by the CUGI developers for their specific projects but are shared with this release of Tripal as example and in the hope they may be useful to others.

In general, the analysis module is responsible for the display and management of data from the Chado analysis table. Tripal provides viewing and editing of analyses as well as an administrative page that allows for syncing and indexing of all or some of the analyses in Chado. Editing, and administration of analyses is similar to organisms and features. See the Tripal Feature Module and Tripal Organism Module sections for more detail information about syncing, indexing assigning taxonomy and cleanup operations which are also available with the analysis module.

The analysis modules will add an "Analyses" link to the primary menu of the website. This menu item links to a page that provides a list of all synced analyses. This menu item can be removed using the Drupal menu administration page.

One major feature of the analysis module is that it serves as a foundation for all other analysis modules. Analysis modules often add content to other Tripal pages such as the feature page. The analysis administration page provides an interface for enabling and disabling automatic addition of content. Figure 19 below shows this interface. To disable addition of content to the feature pages, uncheck the boxes for each analysis module.

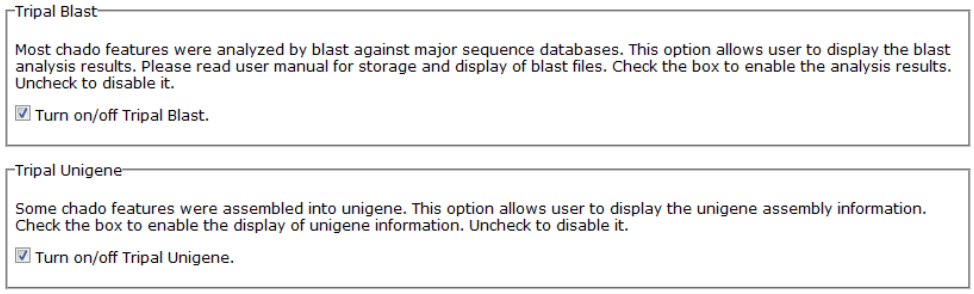


Figure 19

4.7 TRIPAL ANALYSIS BLAST MODULE

The goal of the Tripal Blast Analysis module is to present as much blast information as possible to the user for manually verifying putative function of a particular feature. It is not intended to present blast results that have been manually added through a curation process to show evidence for a particular function or characterization. Blast results that might need to appear in GBrowse, for instance, should be stored normally in the feature and featureloc tables of Chado. Figure 20 shows an example of the box that appears on a feature page that has associated blast results. When the Blast module is enabled, this box will appear on all feature pages for which the module finds blast results.

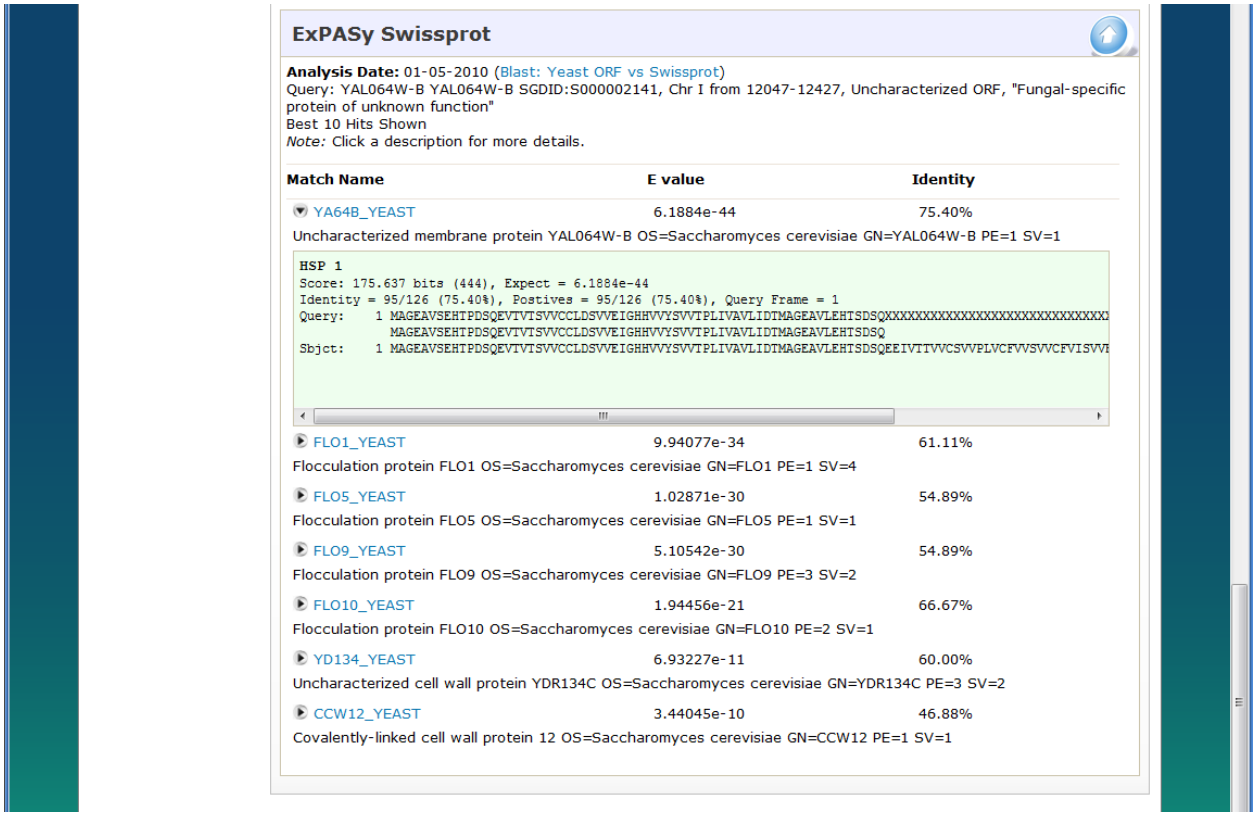


Figure 20

The blast results box displays the top 10 blast hits by default. Users who wish to see the top 25 or all of the blast hits can click the links towards the top of the box. The module uses an AJAX interface for quickly updating the results. When available, the match name becomes a clickable link that takes end-users to the source database for the match. Clicking on the description of the blast hit will show an alignment of the blast match (as seen in the green box above).

Adding New Blast Results

In the previous version of Tripal, blast results were dissected and saved on the file system and a special script was used to parse the results. Now, blast results can be added using Tripal. The 'Create Content' -> 'Analysis: Blast' menu item can be used to upload a new Blast results. Figure 21 shows the page that can be used to create the new analysis. The information entered here is saved in the analysis table of Chado when saved. Blast results are not uploaded as a file through the web interface, but rather must first be stored in a directory accessible to the web server. As seen in Figure 21, a text box is provided where the full path of the Blast results file can be specified. To upload results to Chado, enter the full path in the text box and select the checkbox that reads 'Submit a job to parse the XML output into the analysisfeatureprop table. Select the database from the drop down against which the query sequences were blasted, and add any parameters used when the blast was execute. Finally, click 'Submit' at the bottom of the page. Once the page is submitted a new analysis entry is added to Chado and a web page is created for the analysis. The parsing of the blast results will occur later using the Tripal Jobs system. Periodically Tripal will check for pending jobs and run them. Blast results will be parsed at that time and results will be added to the corresponding feature pages. See section 7 for further information about the Jobs

management system. When multiple blast files are present from the same analysis (e.g. when a query file is split and blasted against the same database) the directory where all blast files are stored can be provided. Tripal will import all files with a .xml extension as blast results for the analysis.

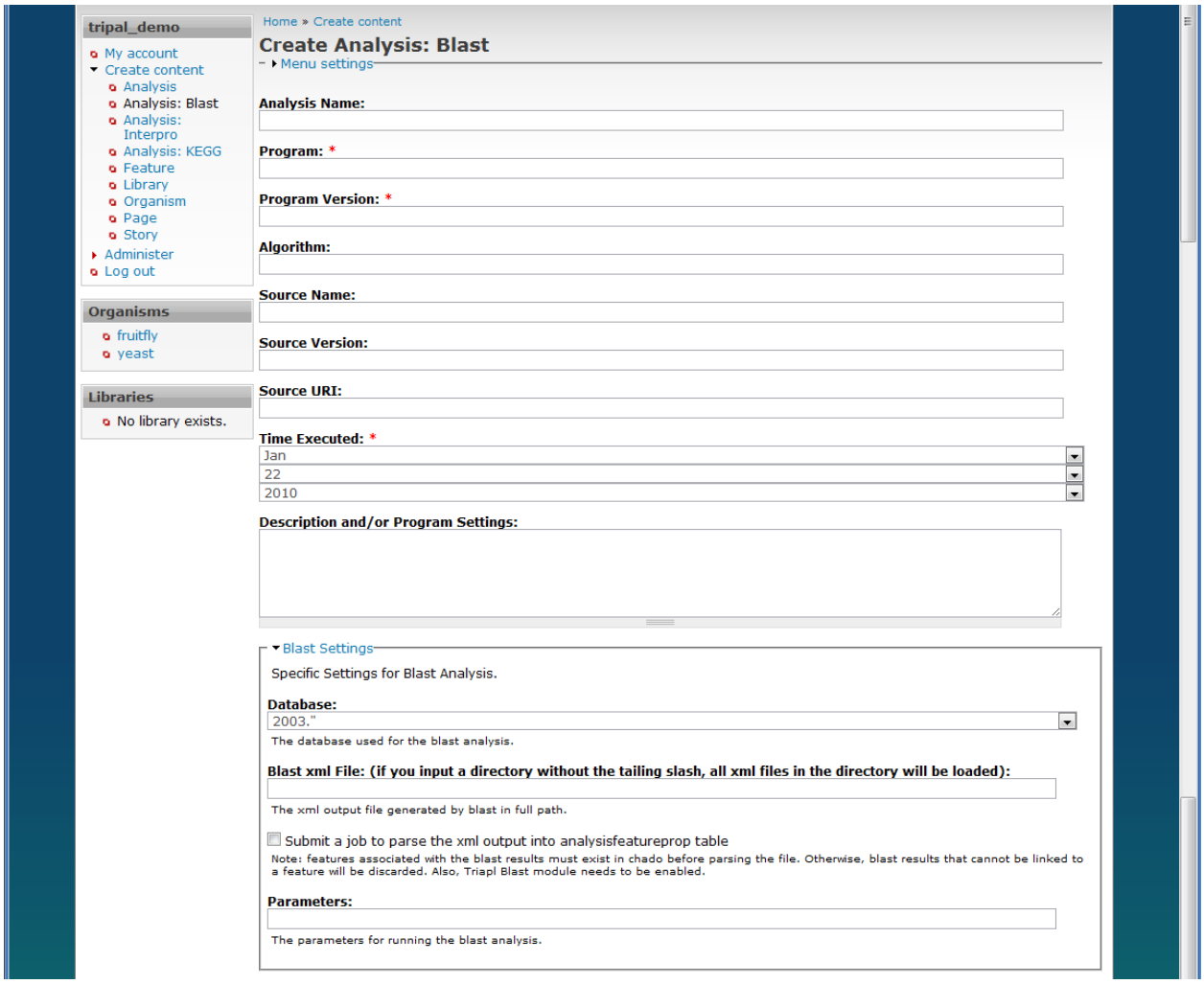


Figure 21

Parsing Match Names

When parsing blast results, Tripal will extract the match name as the first set of characters up to the first occurrence of a space. This may not be the desired behavior since in many databases, such as NCBI, feature accessions are not the only text found before the first space. The module supports specialized parsing of blast matches through the use of perl-style regular expressions. These regular expressions indicate the exact position of the match name within the blast hit results. In Figure 22 the user can select a database from the drop down menu and then provide three regular expressions, one each to extract the hit name, hit description and hit accession. When properly extracted, the blast results will contain match names with links to the originating database. When

the blast analysis is created (see Figure 21) the user selects the database blasted against which allows the parser to use the regular expressions provided. In the event that the database does not exist, the user can create a new database entry using the Tripal DB module administrative page.

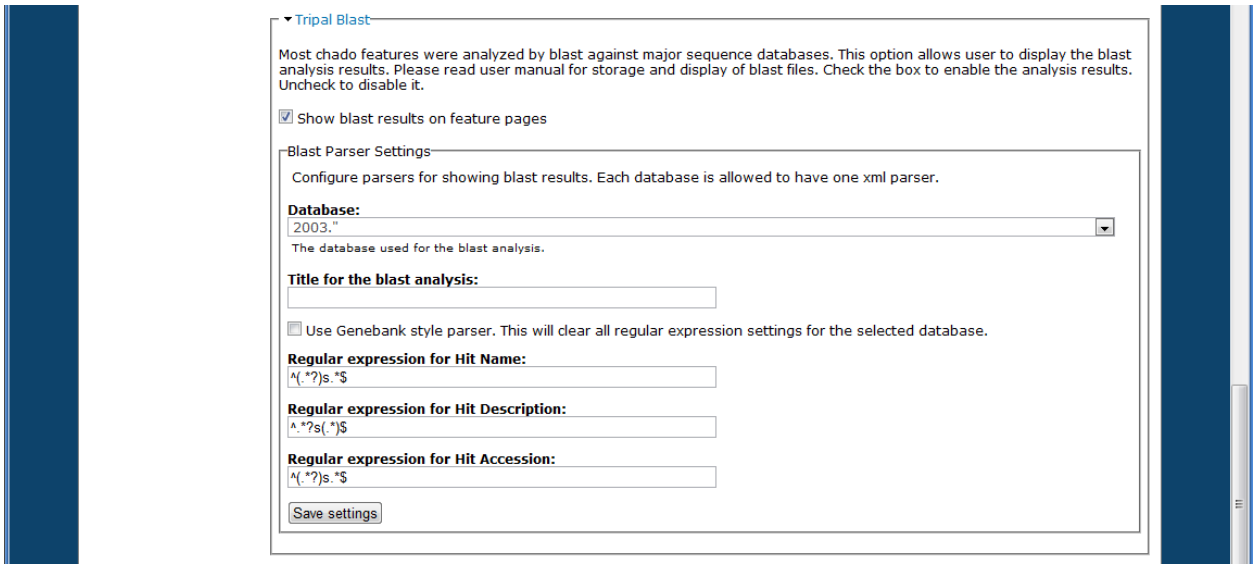


Figure 22

Important: After updating blast results it is important to re-index the feature content so that any new additions are searchable.

4.8 TRIPAL ANALYSIS UNIGENE MODULE

The Tripal Analysis Unigene module presents information about features involved in EST unigene assemblies. It locates these relationship by querying the featureloc table of chado for features assigned the Sequence Ontology (SO) term of 'EST_match'. These EST_match features correlate the ESTs that form the consensus 'contig' sequence. The ESTs and contigs must be assigned the SO terms 'EST' and 'contig' respectively. Figures 23 and 24 below show an example of the boxes added to the feature pages of the 'contig' and 'EST' features. For a 'contig' feature the box titled 'ESTs in this contig' appears with a listing of the member ESTs that form the consensus, as well as the positions of those ESTs in the alignment.

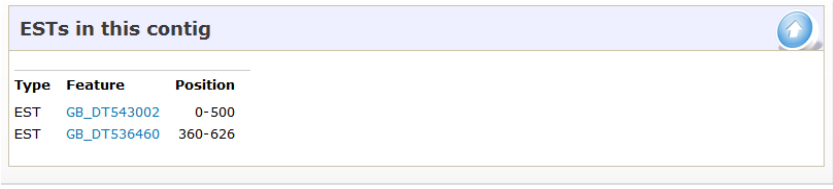


Figure 23

For an 'EST' feature that is part of an alignment that forms a contig, the box 'Alignments' appears listing all of the contigs to which the feature belongs.

Figure 24

CUGI provides a simple Perl script for parsing CAP3 and other compatible ACE format files. The script, named ace2gff3 generates a GFF file for uploading CAP3 assemblies to chado. The script can be found via the Tripal page of the CUGI website: http://www.genome.clemson.edu/tripal.

5 THEMING THE SITE

5.1 TRIPAL THEME

The Tripal theme is designed to allow site administrators to use their own themes to create a custom look-and-feel for their Tripal-based sites. The theme provides CSS style sheets and JavaScript for display of Tripal specific content and is designed to not interfere with the principle theme (as much as possible). An example of style provided by the Tripal them are the expandable boxes used heavily in Tripal. Styling of these boxes is handled by the Tripal theme. Figure 25 shows examples of these boxes on a feature page.

Figure 25

5.2 ENABLING THE TRIPAL THEME

The Tripal theme can work in conjunction with any other theme. Drupal has several themes that accompany the distribution. The site administrator can select any of these default themes or choose a 3rd party theme. Drupal themes are found in the <Drupal installation path>/themes directory and 3rd party themes downloaded separately should be found in the <Drupal installation path>/sites/all/themes directory. Each theme has its own .info file that

describes the theme for Drupal. For, instance, the Garland theme that comes with Drupal is found in the <Drupal installation path>/themes/garland directory and has an info file named garland.info.

To enable the Tripal theme, first ensure that the Tripal theme is located in the <Drupal installation path>/sites/all/themes directory. Then, edit the .info file for the site's current theme and add the following line anywhere in the file:

```
base theme = tripal
```

For example, if the site's default theme is 'Garland', then edit the <Drupal installation path>/themes/garland.info file and add the line above.

The Tripal theme will appear in the Administer -> Site Configuration -> Themes page of Drupal. Figure 26 shows how it appears. The Tripal theme should be enabled through this interface (the checkbox should be checked), but should not be set as default (the circular box) should not be selected.

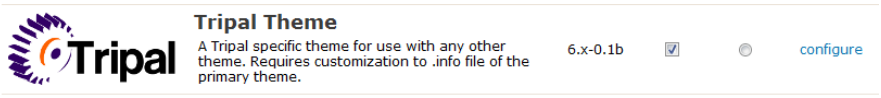


Figure 26

Note: While every best effort has been made to ensure that the Tripal theme will not interfere with the site's existing theme, it may occur that some CSS settings are not compatible with the look-and-feel of the primary theme. In this case, site administrators would need to alter the CSS of either the Tripal theme or the primary theme.

5.4 CUSTOMIZING CONTENT PAGES

Beyond the overall look and feel, each Tripal node, or content type (e.g. organism, feature, analysis and library) has a default layout and set of content. Default templates exist for each content type, however, these templates can be customized, and other items can be added. These templates are provided in PHP files that can be found in the <Drupal install path>/sites/all/themes/theme_tripal directory of the Drupal installation. The following PHP templates are provided for Tripal content:

- node-chado_analysis.tpl.php (Tripal analysis template)
- node-chado_feature.tpl.php (Tripal feature template)
- node-chado_library.tpl.php (Tripal library template)
- node-chado_organism.tpl.php (Tripal organism template)

Additionally, templates may also exist for other content types provided through user-contributed modules, such as the blast, KEGG, and InterPro modules.

The naming of these files is consistent with the Drupal naming standards and should not be changed. Each file generates a custom layout for the content of the Tripal analysis, feature, library and organism modules respectively. These files can be changed to alter the data displayed or the order of content on the page. Below is an example template for Tripal organisms:

```
<?php if($teaser){ ?>
    <a href="<?php print $node_url?>"><?php print $title?></a>
<?php } else { ?>
   <?php $features  = $node->features ?>
   <?php $libraries = $node->libraries ?>
   <div class="node<?php if ($sticky) { print " sticky"; } ?>
                  <?php if (!$status) { print " node-unpublished"; } ?>">
   <?php if ($picture) { print $picture; }?>
   <?php if ($page == 0) { ?>
     <h2 class="title"><a href="<?php print $node_url?>"><?php print $title?></a></h2>
   <?php }; ?>
   <span class="taxonomy"><?php print $terms?></span>
   <div class="content">
     <div class="tripal_organism-image">
        <img src=<?php print file_create_url(file_directory_path() .
"/tripal/tripal_organism/images/".$node->genus."_".$node->species.".jpg")?>>
     </div>
     <?php if ($submitted): ?>
        <div class="metanode"><p><?php print t('') .'<span class="author">'. theme('username',
$node).'</span>' . t(' - Posted on ') . '<span class="date">'.format_date($node->created,
'custom', "d F Y").'</span>'; ?></p></div>
     <?php endif; ?>
     <div class="tripal_organism-details">
        <h3>Details</h3>
           <table class="tripal_table_vert">
              <tr>
                 <th nowrap>Common Name</th>
                 <td><?php print $node->common_name?></td>
              </tr>
              <tr>
                 <th>Genus</th>
                 <td><?php print $node->genus?></td>
              </tr>
              <tr>
                 <th>Species</th>
                 <td><?php print $node->species?></td>
              </tr>
           </table>
           <?php if($node->description){ ?>
              <b>Description</b>
              <p class="organism"><?php print $node->description?></p>
           <?php }; ?>
        </div>
     </div>
     <div class="content"><?php print $content ?></div>
     <?php if ($links) { ?>
       <div class="links"> <?php print $links?></div>
     <?php }; ?>
  </div>
<?php }; ?>
```

This code provides a template for formatting any organism. For all templates, the variable `$node` is populated with the necessary data variables for accessing the Chado content. For example, in the code example above the common name of the organism is accessible with the variable: `$node->common_name`. Other variables such as `$teaser`, `$title`, and `$taxonomy` are Drupal related variables.

The following table provides a list of data variables available for use in the template files. These may be used for changing the templates

Table 2

Template	Variable	Description
Tripal Analysis	`$node->organism_id`	The organism_id from the chado organism table.
	`$node->genus`	The genus name for the organism
	`$node->species`	The species name for the organism
	`$node->common_name`	The common name for the organism
	`$node->abbreviation`	The abbreviation for the organism
	`$node->description`	The comment from the chado organism table

	`$node->features`	The results from the organism_feature_count materialized view. This is a PHP array of values containing the column names of the chado feature table for features that belong to the organism.
Tripal Feature	`$node->feature->feature_id`	The feature_id from the Chado feature table
	`$node->feature->featurename`	The name of the feature
	`$node->feature->uniquename`	The unique name of the feature
	`$node->feature->residues`	The feature sequence
	`$node->feature->seqlen`	The sequence length
	`$node->feature->genus`	The genus of the organism to which this feature belongs
	`$node->feature->species`	The species of the organism to which this feature belongs
	`$node->feature->common_name`	The common name of the organism to which this feature belongs
	`$node->feature->organism_id`	The organism_id from the Chado organism table for the organism to which this feature belongs
	`$node->feature->cvname`	The type name assigned to this feature (e.g. EST, mRNA, etc).
	`$node->feature->type_id`	The type_id value from the Chado feature table
	`$node->feature->is_obsolete`	The is_obsolete value from the Chado feature table.
	`$node->synonyms`	An array of synonymous names for this feature
	`$node->synonyms[$x]->name`	The name of the synonym. $x provides the index into this array. Use $x to loop through the array.
	`$node->references`	The array of external references associated with this feature.
	`$node->references[$x]->accession`	The accession number for the external reference. The $x provides the index into this array. Use $x to loop through the array.
	`$node->references[$x]->dbdesc`	The database description for this reference. This is the 'description' column from the chado 'db' table.
	`$node->references[$x]->db_id`	The database id from the chado 'db' table.
	`$node->references[$x]->db_name`	The database name from the chado 'db' table
	`$node->references[$x]->urlprefix`	The 'urlprefix' from the chado 'db' table
Tripal Library	`$node->uniquename`	The library uniquename
	`$node->library_type`	The type of library
	`$node->organism_id`	The organism_id for the organism to which this library belongs
	`$node->genus`	The genus of the organism
	`$node->species`	The species of the organism
	`$node->common_name`	The common name of the organism
	`$node->library_description`	The library descriptoin
	`$node->features`	An array of feature types and their numbers for this library.
	`$node->features[$x]->num_features`	The number of features for the given type. The

		$x is used to loop through the array.
	$node->features[$x]->feature_type	The feature type.
Tripal Organism	$node->organism_id	The organism_id from the chado organism table
	$node->abbreviation	The abbreviation for the organism
	$node->genus	The genus
	$node->species	The species
	$node->common_name	The common name
	$node->description	A description of the organism
	$node->features	An array of feature types and their numbers for this organism.
	$node->features[$x]->num_features	The number of features for the given type. The $x is used to loop through the array.
	$node->features[$x]->feature_type	The feature type.

6 THEMING OTHER GMOD OR 3RD PARTY TOOLS

In many cases it is desirable to integrate existing non-Drupal, web-based tools into the website. These tools often have their own style sheets and functionality that make direct integration with the Drupal theme difficult. One solution is to install these tools independent and "alongside" Drupal. Users can access the tools via links from the website, but these tools do not share a common visual theme with the site. One advantage of this approach is the ease of installation because it requires no Drupal integration. One disadvantage is the inconsistency of the look-and-feel between different parts of the website.

IFrames provides a suitable method for integration of GMOD and other 3rd party tools. For example, the NCBI wwwblast tool, can be integrated in this way. First, download the wwwblast version best suited for your hardware platform from the NCBI FTP server: ftp://ftp.ncbi.nih.gov/blast/executables/LATEST/. Second, install the wwwblast by following the installation instructions that accompany the package. Run the software standalone through a web browser to ensure the tool is functioning correctly before attempting to integrate with Drupal. Third, create a new Drupal page for the blast server. A page can be created in Drupal by clicking the 'Create Content' menu item and then select the 'Page' content type. Figure 27 below shows the Create Page form.

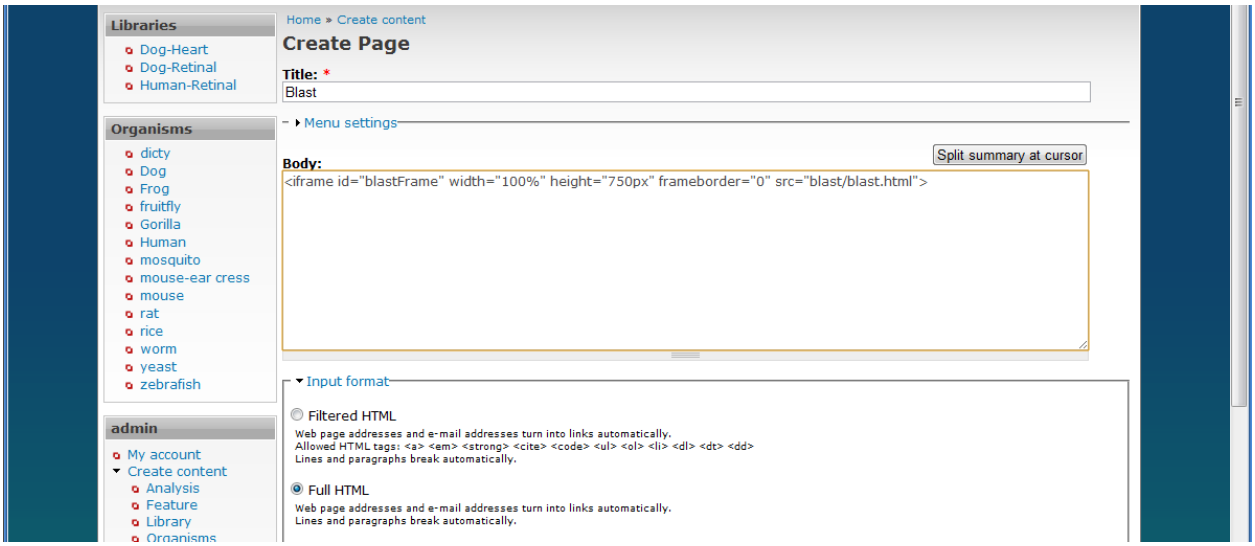


Figure 27

Add a title such as 'Blast' for the page and add the following text in the "Body" section to create the IFrame:

```
<iframe id="blastFrame" width="100%" height="1100px" frameborder="0" src="blast/blast.html">
```

Set the width and height to allow the tool to fit within the space provided on the page. The 'src' argument of the IFrame should be the URL for the standalone blast server. Finally, before saving the page, be sure to open the input format section below the Body text area and select the item "Full HTML". Figure 28 shows the Blast tool embedded within the site's theme, thus providing a more consistent look and feel to for end-users. Finally, use the Drupal menu utility to add this new blast page as a menu item if desired.

One disadvantage to IFrames is the scroll-bar that appears along the right side and bottom of the IFrame if the content extends beyond the specified height and width. Additionally, any content within an IFrame will not be indexed for searching by the Drupal search engine.

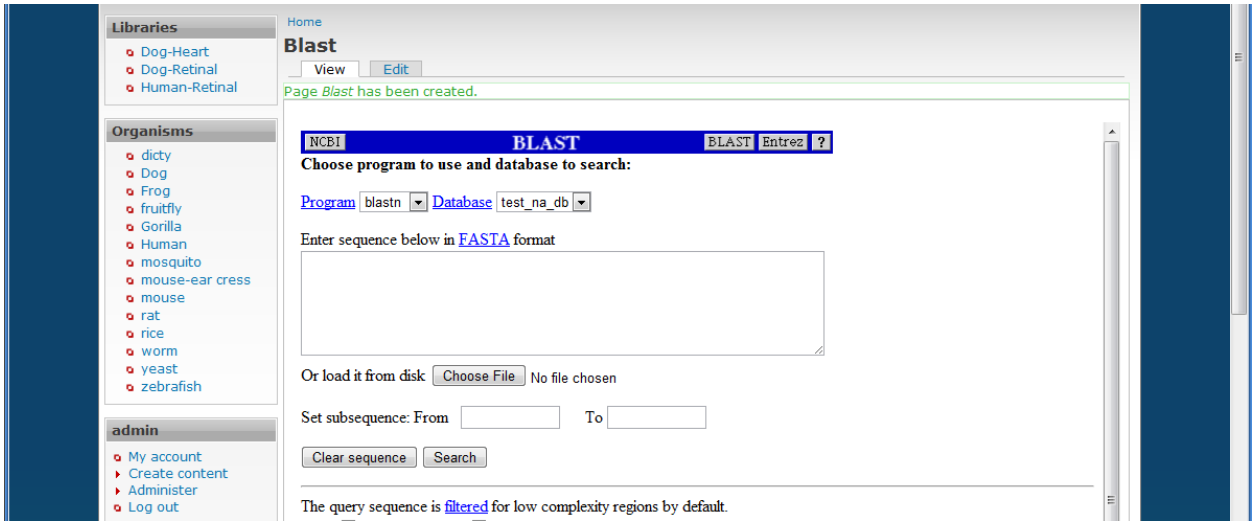


Figure 28

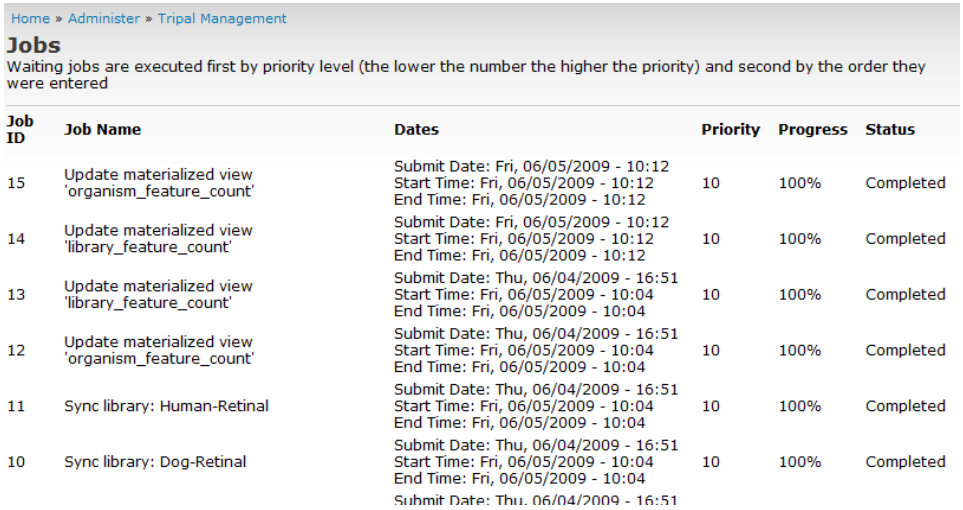
7 JOBS MANAGEMENT

Tripal offers a job management subsystem for managing tasks that may require an extended period of time for completion. Drupal uses a UNIX-based cron job to handle tasks such as checking the availability of updates, indexing new nodes for searching, etc. Drupal's cron uses the web interface for launching these tasks, however, Tripal provides several administrative tasks that may time out and not complete due to limitations of the web server. Examples including syncing of a large number of features between chado and Drupal. To circumvent this, as well as provide more fine-grained control and monitoring, Tripal uses a jobs management sub-system built into the Tripal Core module. It is anticipated that this functionality will be used for managing analysis jobs provided by future tools, with eventual support for distributed computing.

The Tripal jobs management system allows administrators to submit tasks to be performed which can then be launched through a UNIX command-line PHP script or cron job. This command-line script can be added to a cron entry along-side the Drupal cron entry for automatic, regular launching of Tripal jobs. The order of execution of waiting jobs is determined first by priority and second by the order the jobs were entered.

7.1 SUBMITTING AND VIEWING JOBS

Jobs are submitted through the 'Tripal Administration' menu group. Tripal modules provides an administrative interface for tasks such syncing content, re-indexing content, setting or resetting taxonomy terms, and database cleanup. The site administrator selects appropriate options as provided by the module which in turn submits a job on-behalf of the administrator. Jobs can be viewed on the 'Jobs' page which is accessible through the 'Jobs' link in the 'Tripal Administration' menu group. Figure 29 below shows the Jobs. The Jobs page shows all jobs which have been submitted and provides the job name, the date of submission, start and end times, a progress indicator and a status.



Home » Administer » Tripal Management

Jobs

Waiting jobs are executed first by priority level (the lower the number the higher the priority) and second by the order they were entered

Job ID	Job Name	Dates	Priority	Progress	Status
15	Update materialized view 'organism_feature_count'	Submit Date: Fri, 06/05/2009 - 10:12 Start Time: Fri, 06/05/2009 - 10:12 End Time: Fri, 06/05/2009 - 10:12	10	100%	Completed
14	Update materialized view 'library_feature_count'	Submit Date: Fri, 06/05/2009 - 10:12 Start Time: Fri, 06/05/2009 - 10:12 End Time: Fri, 06/05/2009 - 10:12	10	100%	Completed
13	Update materialized view 'library_feature_count'	Submit Date: Thu, 06/04/2009 - 16:51 Start Time: Fri, 06/05/2009 - 10:04 End Time: Fri, 06/05/2009 - 10:04	10	100%	Completed
12	Update materialized view 'organism_feature_count'	Submit Date: Thu, 06/04/2009 - 16:51 Start Time: Fri, 06/05/2009 - 10:04 End Time: Fri, 06/05/2009 - 10:04	10	100%	Completed
11	Sync library: Human-Retinal	Submit Date: Thu, 06/04/2009 - 16:51 Start Time: Fri, 06/05/2009 - 10:04 End Time: Fri, 06/05/2009 - 10:04	10	100%	Completed
10	Sync library: Dog-Retinal	Submit Date: Thu, 06/04/2009 - 16:51 Start Time: Fri, 06/05/2009 - 10:04 End Time: Fri, 06/05/2009 - 10:04	10	100%	Completed

Submit Date: Thu, 06/04/2009 - 16:51

Figure 29

7.2 JOB STATUS

A job state is provided when viewing jobs on the "Jobs" administration page. This state indicates the status of the job and helps the user measure progress of each job. The following table provides a listing of job states and a description of their meaning.

Table 3. Job Status Descriptions

Status	Description
Waiting	The job has been submitted but not yet run. A job will remain in this state until the job is launched. Jobs may be launched manually or automatically (see next section).
Running	The job has been submitted, launched and is currently running but has not yet completed its work. A progress indicator (in terms of percent complete) is shown for each job.
Completed	Once a job has reached 100% complete the status is changed from 'Running' to 'Completed'. This job is kept in the view for historical purposes.
Error	The job has been submitted, launched, began running but failed to terminate properly. For version 0.1b Tripal does not provide diagnostic information for debugging of problems with jobs. Please see the section below titled "Debugging Job Errors" for more information to correct errors that prevent jobs from completing.

7.3 RUNNING TRIPAL JOBS

Tripal jobs may be executed either manually or automatically using the UNIX cron. It is recommended to use a cron job on a regular interval to launch waiting jobs, but the command-line may be used when more time critical jobs are needed.

7.3.1 COMMAND-LINE

Tripal jobs can be launched manually from the command-line. First change to the directory where Drupal is installed, and execute the following command:

```
php ./sites/all/modules/tripal_core/tripal_launch_jobs.php <username>
```

Where <username> is the name of a Drupal user that has administrative privilege.

7.3.2 CRON

Typically with Drupal sites a cron entry is added to the UNIX crontab by the site administer to perform periodic tasks relating to Drupal management. The UNIX cron is a task management system that allows users to automatically launch recurring tasks at specified time intervals. Without a cron entry to automatically launch Tripal jobs the site administrator would need to manually run jobs on the command-line as described above. Follow these steps to add a cron entry for the Tripal jobs system:

- First, open the crontab editor by issuing this command on the command-line:

```
crontab -e
```

- Second, add a new "cron entry" similar to the following:

```
0,15,30,45    *    *    *    *         (cd    <Drupal    Path>;    php
./sites/all/modules/tripal_core/tripal_launch_jobs.php <username> )
```

  Where <Drupal Path> is the location of your Drupal installation and <username> is the name of a Drupal user that has administrative privileges. The cron entry is split across two lines above to fit within this document, however it should only be a single line when adding as a new cron entry.

In the cron entry example above, the Tripal jobs system would be launched every fifteen minutes of every hour of every day of every week of every month. Please review the crontab manual pages for further information about managing schedules for tasks. The cron manual pages can be viewed by issue the 'man crontab' command on the UNIX command-line.

If a Tripal job is executing when the cron attempts to run the script, the script will terminate to allow the first instance to complete execution. This prevents the Tripal job scheduler from over subscribing resources on the server when many long running jobs may be queued.

Important: The 'php' command must be in your path for the cron entry to work. If not, specify the full path to php in the cron entry above.

7.4 DEBUGGING JOB ERRORS

Currently, Tripal does not provide a mechanism for assisting with debugging of jobs that may fail. It simply reports when a job has not completed successfully by setting the status to 'Error'. The best means for finding the failure is to perform the following:

1. Check Drupal report logs for errors: sometimes Tripal modules may report notices, warnings and errors to the Drupal report log. These reports can be found under the Reports menu of the Drupal Administration pages and may be useful to determine the cause of an error.
2. Recreate the job and rerun the job manually on the command-line watching for an error message. If the job requires a very long time to execute, redirect the standard error (STDERR) output to a file for later inspection.
3. Post a message to the Tripal forum found on the CUGI website requesting assistance.

8 TAXONOMY MANAGEMENT

Drupal provides a method for categorizing content by allowing users to assign "taxonomy" or categorical terms to Drupal nodes. In Drupal, taxonomies are grouped by "vocabularies". Taxonomies and vocabularies can be managed by clicking the Drupal "Administration -> Content -> Taxonomy" menu item. Taxonomy terms are available in the "Advanced Search" form for filtering of search results.

The Tripal Feature module currently supports automatic assignment of taxonomy to chado feature pages. Tripal supports the following types of taxonomic terms:

Vocabulary	Terms
Organism	This vocabulary is automatically added by Tripal when enabled on the Feature Administration page. Each synced feature is automatically assigned the genus and species as a taxonomic term. The terms for this vocabulary become the unique set of genus and species for all synced features.
Library	This vocabulary is automatically added by Tripal when enabled on the Feature Administration page. Each synced feature is automatically assigned the library name to which it is associated as a taxonomic term. The terms for this vocabulary become the unique set of library names associated with all synced features.
Analysis	This vocabulary is automatically added by Tripal when enabled on the Feature Administration page. Each synced feature is automatically assigned the name of the analysis method to which it is associated. The terms for this vocabulary become the unique set of analysis names associated with all synced features.

Tripal provides an interface in the "Administration -> Tripal Management -> Features" menu for automatic assignment of categories to chado feature pages.

<SCREEN SHOT OF ADVANCED SEARCH MENU>

Taxonomic terms will only be set for features when the checkboxes are set on the feature administration page. Features that are synced before the taxonomy settings are selected will not have taxonomy assigned. Use the reset taxonomy button to reset taxonomy for all features on the site.

9 CVTERM -- FOR SPECIFIC MODULE FUNCTIONALITY

Tripal does not yet provide a facility for creating and managing the controlled vocabulary (CV) terms in the chado database. This is a module that will need to be added later. However, Tripal requires a few cvterms to function properly. These terms are added automatically by Tripal on installation.

To keep Tripal cvterms separate from other vocabularies, a new vocabulary named 'tripal' is created automatically in the cv table. Also, a 'tripal' database is added to the db table as well. The cvterm is then added to the cvterm table and an accession for this term is added to the dbxref table.

The following table shows cvterms, the intended usage, and the modules in Tripal that creates them. Currently, Tripal only uses a single cvterm, however new cvterms may be added to Tripal in new modules using the Core API. See the "Writing Your Own Modules" in section 13 below for further information.

Table 4

Cvterm	Tripal Module	Usage
library_description	tripal_library	The chado library table does not have a field for the description of the library. However, the tripal_library module does require a

	library description when creating or updating a library. This information is stored as a library property in the libraryprop table. The cvterm is intended to qualify the property as a library description for Tripal.

10 INDEXING CONTENT FOR SEARCHING

Drupal provides out-of-the-box a full text searching capability. Essentially all words on the page are extracted and indexed for quick searching by end-users. The pages generated by Tripal for viewing chado content can be indexed for searching. Pages that are not indexed are not available for searching. The "Tripal Modules" section above provides details for how to index content. However, when an organism, feature, library or analysis is first "synced" the contents are automatically indexed for searching, and only need to be "re-indexed" when updates are made to the content. Search indexing in Drupal has a few caveats that should be remembered:

- Before any Chado or Tripal content can be searched, the entire site must be listed as 100% indexed. The indexing status can be seen on the 'Search Settings' page of the 'Site Configuration' page of Drupal administration menu. Figure 30 shows a site that is only 98% indexed. To ensure the site is 100% indexed be sure that the Drupal cron is properly setup and running.

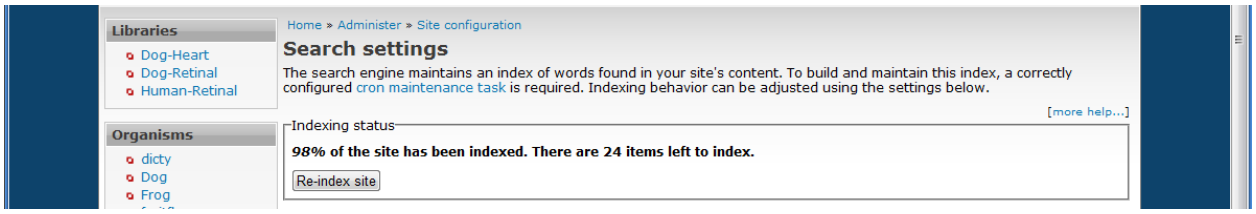


Figure 30

- Not all content on a Tripal page is indexed. For instance, the DNA or amino acid sequence on the feature page is not indexed for searching. This would unnecessarily bloat the database with "words" that cannot effectively be found. Additionally, for the tripal_analysis_blast module the blast alignments are not indexed but the blast hit names are.
- Changes to features, organisms, libraries or features that existed prior to the change will not have those changes reflected in the search indexing. This means that changes cannot be found by a user who performs a search. Re-index the content on the respective Tripal administration page (e.g. Feature Administration, Organism Administration, etc) for those changes to be represented in the search index.

11 MATERIALIZED VIEWS

Chado is excellent because of its integrity constraints and the breadth of information that can accurately be stored. It provides a consistent means for representing complex relationships and a platform that facilitates information sharing. However, one major complaint of Chado is the slowness of queries when thousands of records are present. For this purpose, materialized views have been introduced by the chado developers to allow for quick

lookup of specific data sets. Unlike typical Views supported by PostgreSQL, materialized views are persistent, meaning they are not populated on the fly when requested, but rather are populated and remain populated. While materialized views are quite useful they require a bit of database development expertise to create a set of materialized views that can represent data sets most needed for quick look up and web presentation. They also must be updated periodically to reflect changes to the underlying database. Despite these trade-offs, materialized views can be created and managed quite effectively in Tripal.

Important: A Perl script for creating, and managing materialized views is present in the 1.0 version of Chado. However, Tripal implements its own management tool for creating and managing materialized views. Therefore, any materialized views created through Tripal cannot be managed by the Chado Perl script and the materialized views created with the Perl script cannot be managed by Tripal.

11.1 MANAGING EXISTING VIEWS

Tripal modules come pre-populated with two materialized views. These views are automatically installed when the tripal_organism and tripal_library modules are installed, and provide a count of feature types available for each organism and library respectively. For example, Figure 31 shows a the number and types of features on an organism page in a section titled "Features for this organism". The content for this section is queried from the organism_feature_count materialized view which is provided by default by the tripal_organism module.

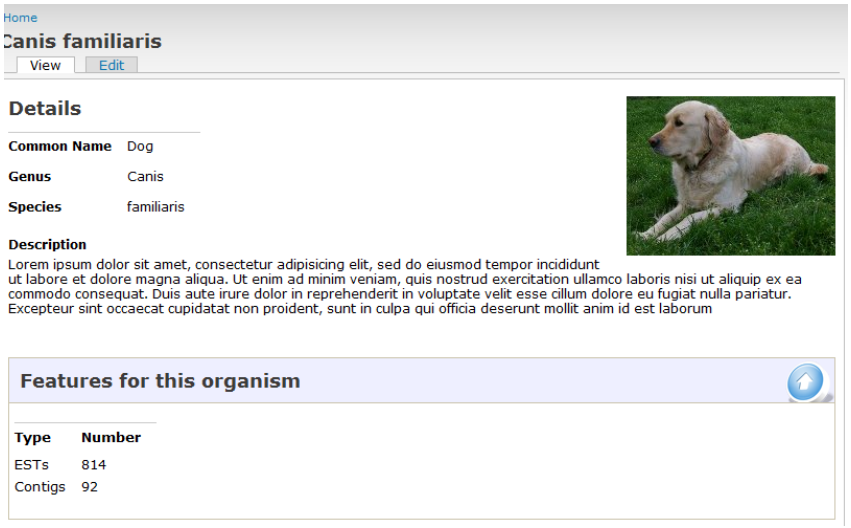


Figure 31

By default the contents for both materialized views do not appear on the organism nor library pages. The materialized views must first be updated before the content appears.

Note: The views that come with Tripal by default may be deleted or simply not updated if this type of information is not required or desired.

To update a view, navigate to the Materialized Views link under the Tripal Administration section. Figure 32 shows how this page appears.

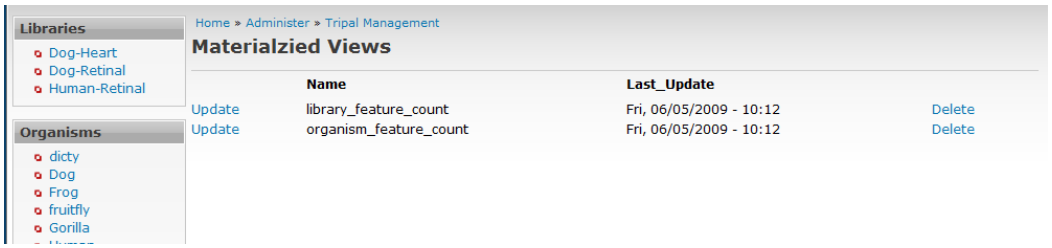


Figure 32

To update any view simply click the 'Update' link to the left of the materialized view. This adds a job to the job management to update the view. Once the view has been updated, a date appears under the "Last Update" column that specifies when the update was last performed. Currently, materialized views must be updated manually through this interface. A scheduling mechanism will be available in future releases of Tripal to allow for automated updates of these views. To delete a view simply click the 'Delete' link in the right-most column.

11.2 CREATING A NEW VIEW

To create a new materialized view in Tripal, navigate to the Materialized Views link under the Tripal Administration section. Click the 'Create View' sub menu link. The Figure 33 shows the Create View page

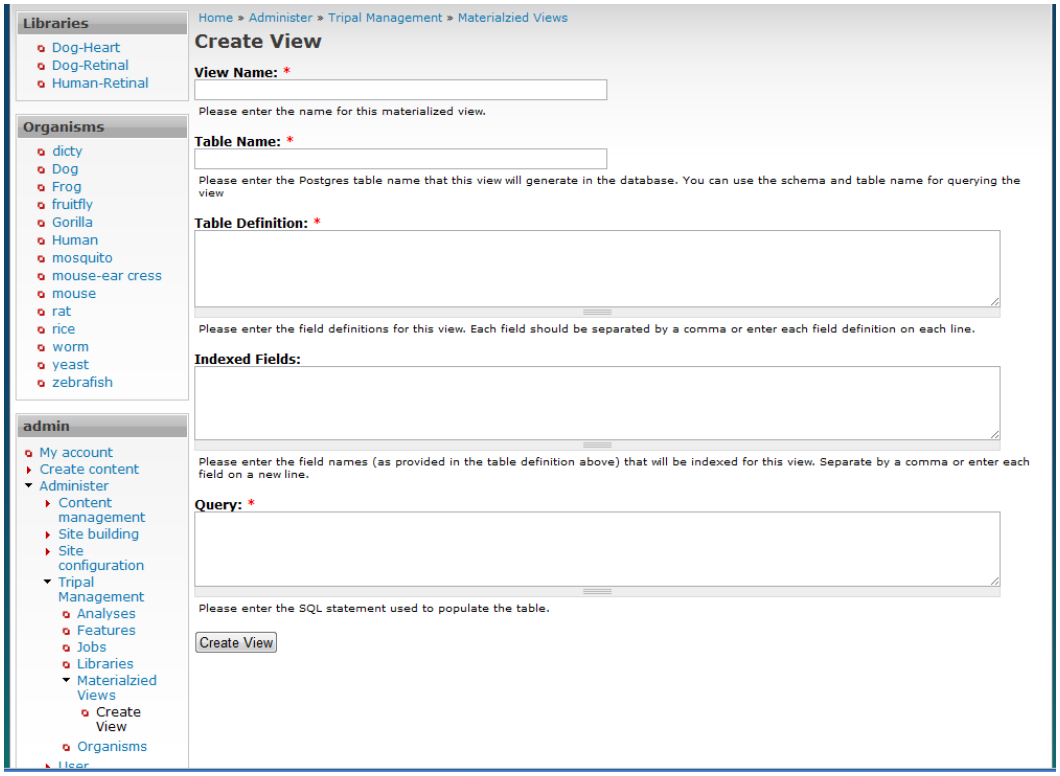


Figure 33

The Create View provides five fields for defining a new view. The format of the text for each field is quite specific and requires careful attention. It is recommended that a database engineer assist with the development and entry

of materialized views. The following table explains each field in the form as well as provides example inputs to recreate the organsim_feature_count materialized view:

Table 5

Field	Explanation	Example Entry
View Name	A name for the view.	`organism_feature_count`
Table Name	A materialized view is in reality a database table. This field provides the name for that table. It must follow naming standards for a PostgreSQL table.	`organism_feature_count`

Table Definition	The table is created using the definition provided here. Each column in the new table must be provided separated by a comma. For each entry, the column name is specified followed by the column type. These types must be valid PostgreSQL column types.	`organism_id integer,`
		`genus character varying(255),`
		`species character varying(255),`
		`common_name character varying(255),`
		`num_features integer,`
		`feature_type character varying(255)`

Indexed Fields	To improve query speeds of the view it is recommended to provide indexes on columns that are used in the WHERE or JOIN clauses of SQL statements that will query this view.	`feature_id, organism_id`

Query	An SQL query is needed to extract data from the chado tables for populating the materialized view. This field provides that query.	`SELECT O.organism_id, O.genus,`
		`  O.species, O.common_name,`
		`  count(F.feature_id) as num_features,`
		`  CVT.name as feature_type`
		`FROM Organism O`
		`  INNER JOIN Feature F`
		`  ON O.Organism_id = F.organism_id`
		`  INNER JOIN Cvterm CVT`
		`  ON F.type_id = CVT.cvterm_id`
		`GROUP BY O.Organism_id, O.genus,`
		`  O.species, O.common_name,`
		`  CVT.name`

Note: Tripal will create the table for the materialized view within the Chado database.

12 RE-INDEXING CONTENT FOR SEARCHING

Initially, when data is first synced between Chado and Drupal using the 'Sync' functions on the various modules, all Chado content will be indexed for searching. The Tripal organism, feature, library and analysis modules provide a sync function for generating pages in Drupal for the chado data. When sync'ed the pages are automatically indexed for searching.

With time, content may change. These changes should be represented in the search index as well. Each module also provides a re-indexing functionality which iterates through all of the Tripal content for the module and recreate the search index. To perform this action, see the respective sections above for each module related to re-indexing content. Currently, Tripal does not have functionality to re-index a single Tripal node. That functionality is coming soon.

13 CUSTOMIZING TRIPAL

One powerful feature of Drupal is its open API which allows users to easily create custom functionality. Tripal expands on the Drupal API by providing its own API for easily adding jobs, creating cvterms, and materliazed views. Creating Drupal modules to integrate with Tripal requires a "learning curve" period, but once understood any custom content can be easily created. This section describes the Tripal API as well as training for creating a Drupal module that integrates with Tripal.

13.1 WRITING YOUR OWN MODULES

13.2 USING CCK TO CUSTOMIZE TRIPAL CONTENT

13.3 PURCHASING CUSTOMIZATIONS

The Clemson University Genomics Institute (CUGI, http://www.genome.clemson.edu/) provides customization and support services. Pricing is provided on a per project basis. Please contact CUGI bioinformatics at bioinfo-info@genome.clemson.edu for information regarding pricing.

13.4 CONTRIBUTING YOUR MODULES