Tripal DevSeed Documentation

Release 1.0

Bradford Condon, Meg Staton

Jan 03, 2019

Contents:

Welcome to the Tripal DevSeed documentation. This site provides instructions for loading DevSeed into Chado manually. The github repo is found here:

https://github.com/statonlab/tripal_dev_seed

Please note that the files referenced in this guide are available here: https://github.com/statonlab/tripal_dev_seed/tree/master/Fexcel_mini

CHAPTER 1

Quick Load: Seeders

Tripal DevSeed is supported by Tripal TestSuite's database seeders. A default seeder is provided that will load in the files hosted on this repo. To use it, uncomment the import statements for the data you would like to include, and run `./vendor/bin/tripaltest db:seed DevSeed`.

CHAPTER 2

Loading FASTA sequences

FASTA is a universal sequence format: when we talk about loading mRNA and polypeptide sequences, we're referring to FASTA and the FASTA loader. This step will create a bunch of mRNA features with which we can associate other data (i.e. BLAST, Interpro, etc).

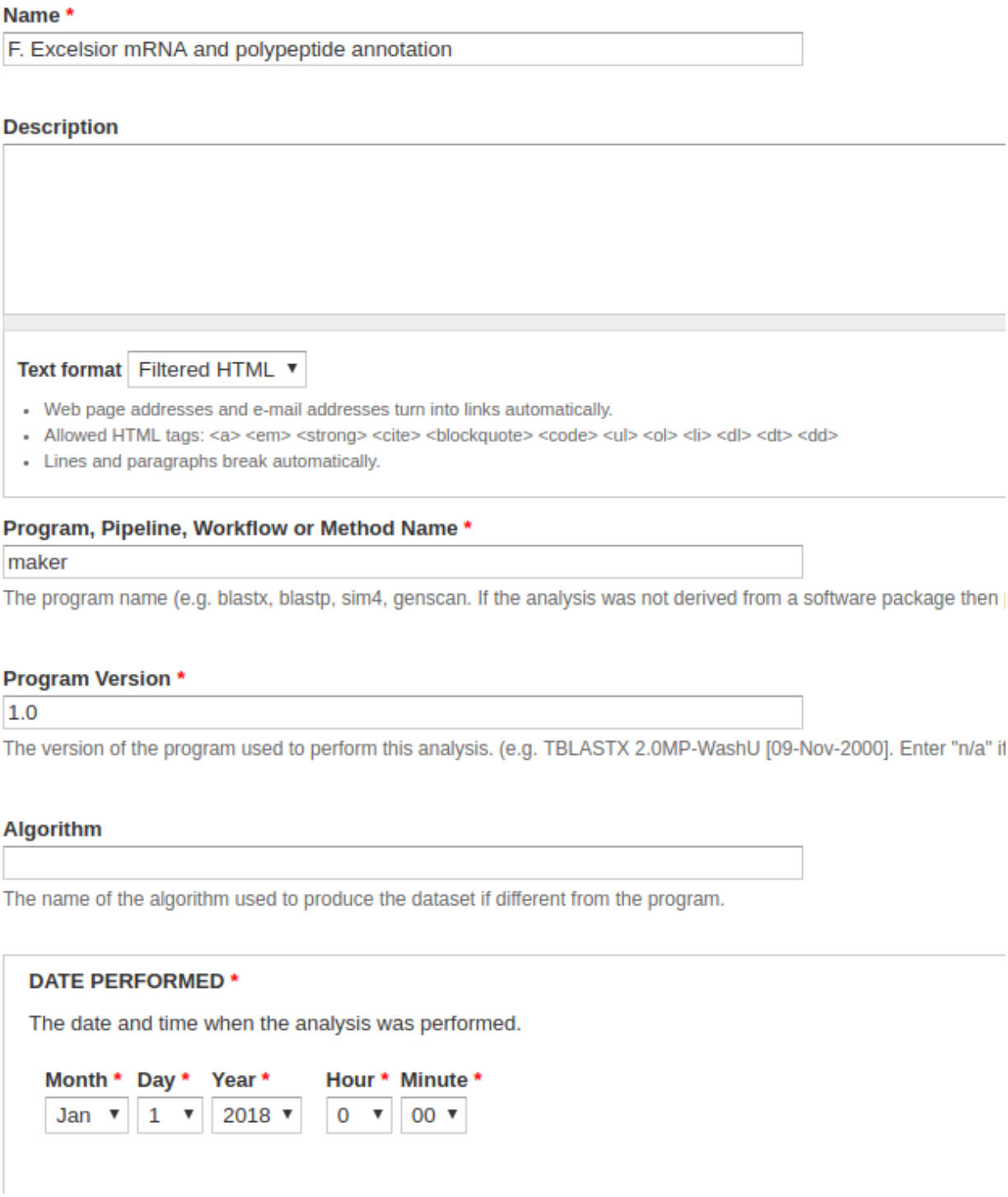
2.1 Create an Analysis

We need an analysis with which to associate both our CDS (mRNA) and proteins (polypeptides).

Navigate to **Content > Tripal Content** and click **Add Tripal Content** at the top of the page. Select **Analysis**. Because this is mostly just data to populate a test site, what we insert into these fields doesn't really matter. Naturally, however, if this were for a site we were releasing for public use, we would want this information to be accurate.

- **Name** - Something along the lines of, **F. Excelsior mRNA and polypeptide annotation**.
- **Program, Pipeline, Workflow or Method Name** - Something along the lines of, **maker**.
- **Program Version** - Something along the lines of, **1.0**.
- **Date Performed** - You can keep this default, but it's common to set this to an arbitrary date (e.g. January 1st, 1900) if you're unsure of the time when the analysis was performed.
- **Data Source Name**
  - For a new transcriptome, this should be labeled, **de novo assembly**.

All other fields can be left blank or at their default values. Click save.

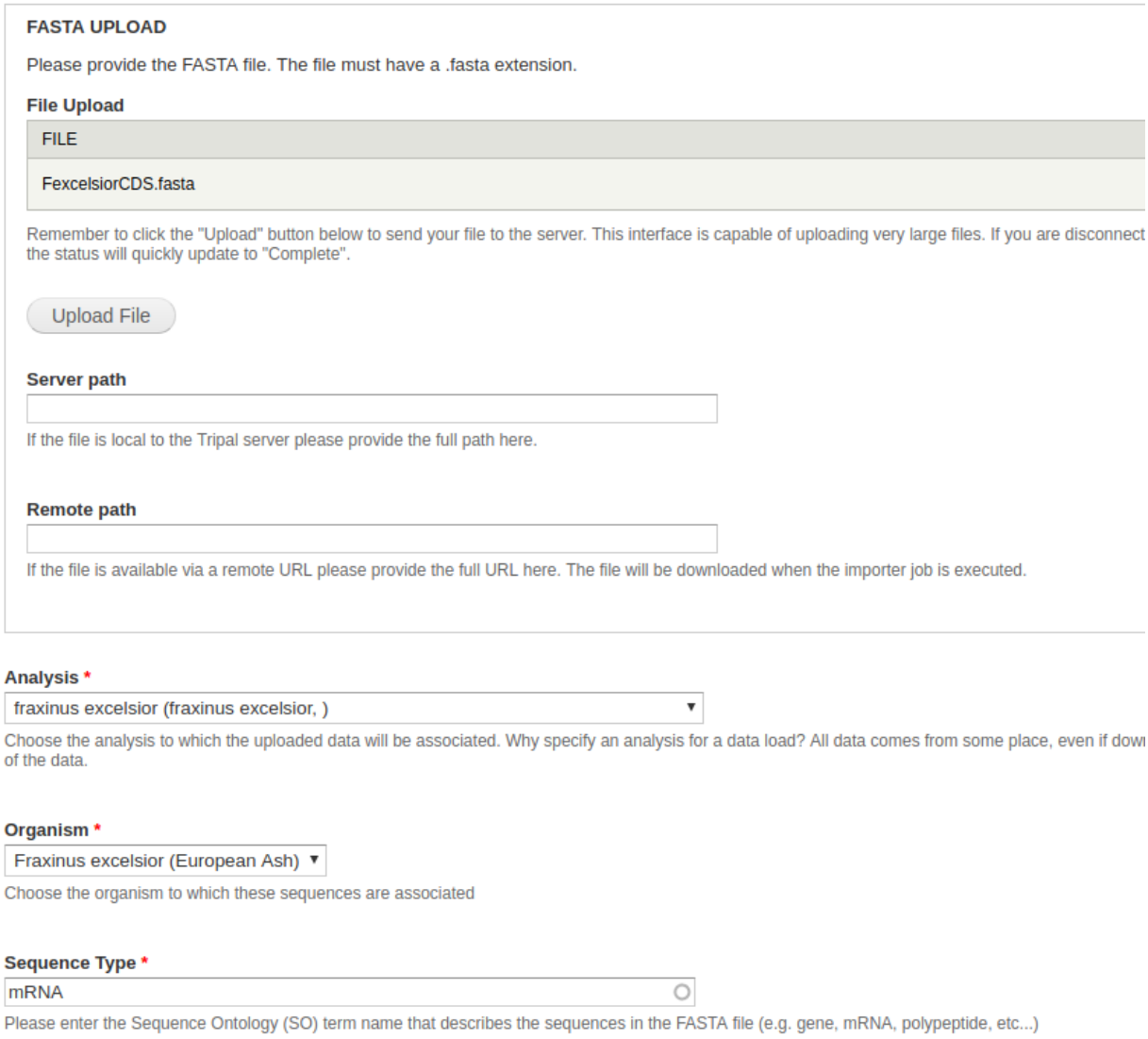


2.2 Loading the mRNA FASTA file

We load in our mRNA data first, then our proteins. Using the admin menu, navigate to **Tripal > Data Loaders > Chado FASTA Loader**.

- **File Upload** - From the dataset, this is the `mrna_mini.fasta` file.
- **Analysis** - Select the newly created analysis.
- **Organism** - Select *Fraxinus excelsior*.
- **Sequence Type** - Enter *mRNA*

All other fields can be left blank or at their default values. Click **Import FASTA file**. A green header should appear at the top of the page with a job for you to run. Once your CDS have uploaded successfully, you can move on to uploading the polypeptides.

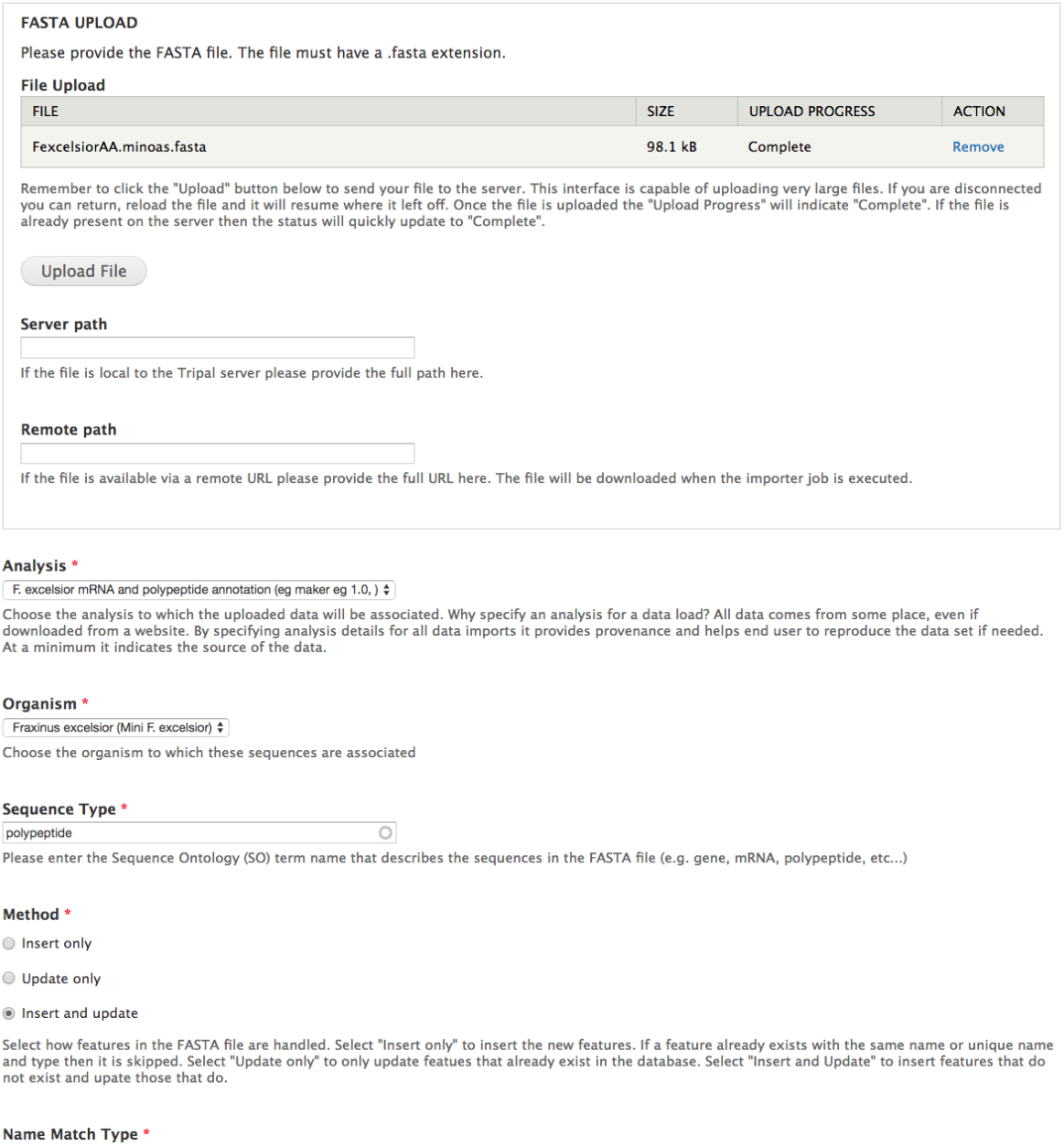


2.3 Loading the Amino Acid (polypeptide) file

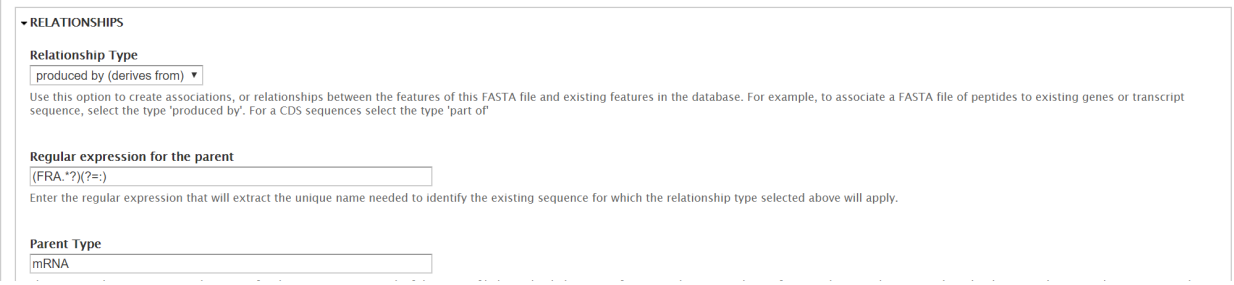
The process for uploading the polypeptides is similar to above, but with some slight differences to the fields.

- **File Upload** - From the dataset, this is the `FexcelsiorAA.minoas.fasta` file.
- **Analysis** - Select the newly created analysis.

- **Organism** - Select Fraxinus excelsior.
- **Sequence Type** - Enter *polypeptide*.



In the additional options section, you have the option to extract the feature name with a regexp, link your sequences to an external database using a regexp, and to define relationships. Because our polypeptides are derived from our mRNA CDS, we'll set the **relationship type** to *produced by*, and provide a regexp to link the terms. If you're following this guide with the *F. excelsior* miniature dataset, then the proteins and mRNA have the same name, and you can use this regexp: `>(.*)`.



All other fields can be left blank or at their default values. Click **Import FASTA file**. A green header should appear at the top of the page with a job for you to run. Once your CDS have uploaded successfully.

2.4 Viewing Results

For now, you won't be able to actually see your results through the user interface until we publish them. This is fine; assuming you have followed the guide, you shouldn't have any issues and can safely move on to the next steps.

However, if you really need to check, you can see your results through the database. Features can be found in the `chado.feature` table. If it's populated with the names of your features, you should be good to go.

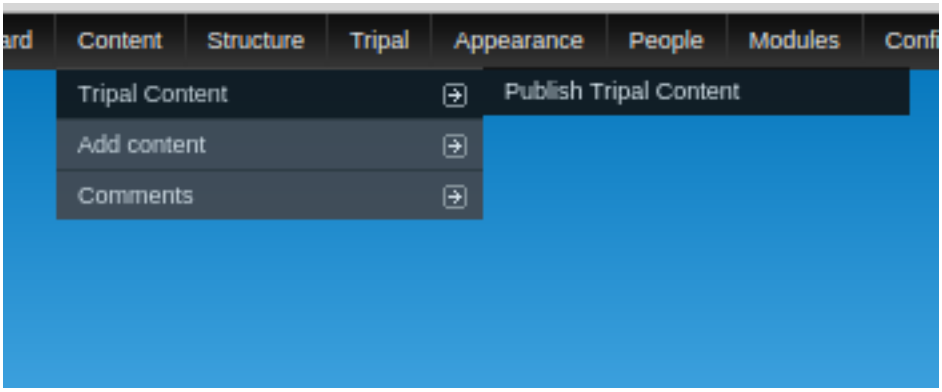
It might also be worth checking the `chado.feature_relationship` table, as this is what determines whether the amino acids were successfully linked to the proteins. If it's populated, you should be good to go.

CHAPTER 3

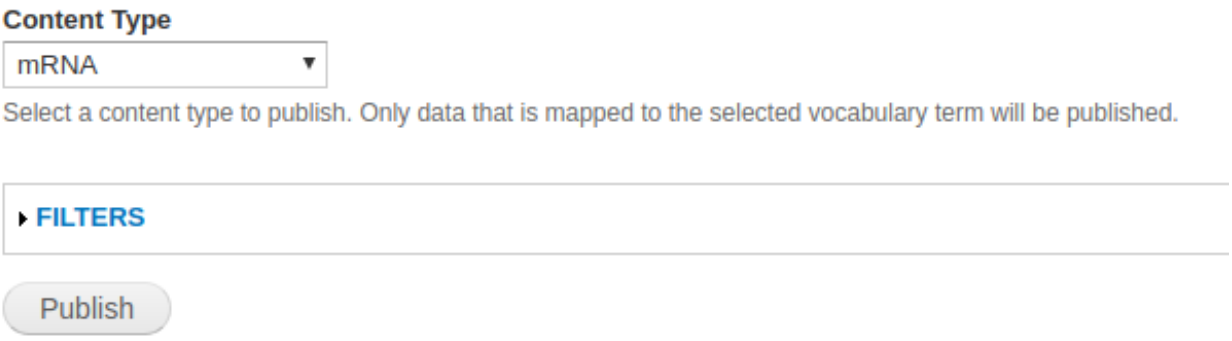
Publishing mRNA

When we publish data in Tripal, we are creating entities for records in the chado database. The process is relatively simple.

From the admin menu, navigate to **Content > Tripal Content > Publish Tripal Content.**



Select mRNA from the **Content Type** dropdown and click **Publish.**

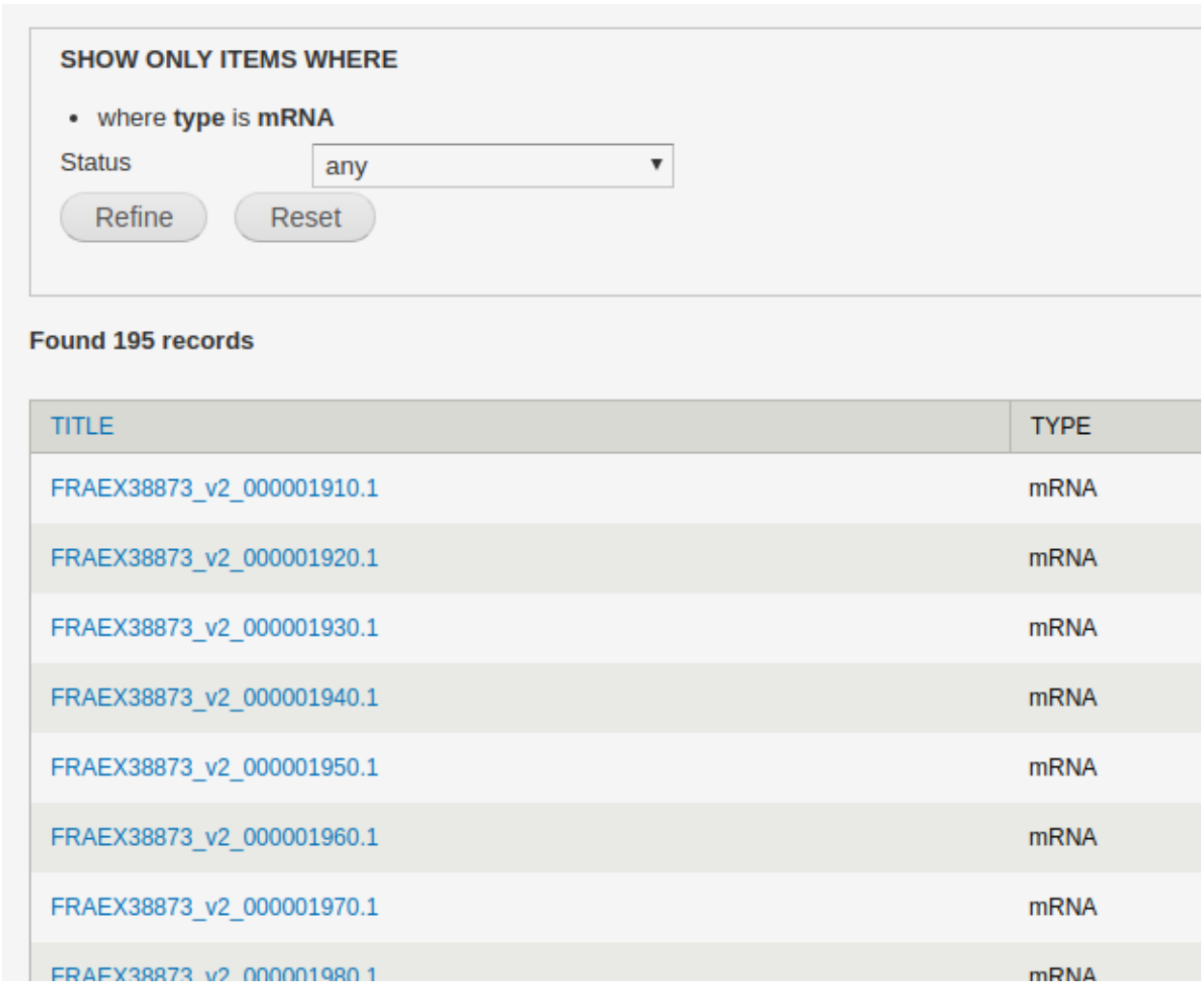


A green header should appear with a job for you to run. Run the job and you're done.

CHAPTER 4

Viewing Published Data

You can check to make sure that publishing was successful by navigating to **Content > Tripal Content.** You can sort by Content Type > mRNA to display only the published mRNA results.


SHOW ONLY ITEMS WHERE
where type is mRNA
Status
any
Refine
Reset
Found 195 records
TITLE
TYPE
FRAEX38873_v2_000001910.1
mRNA
FRAEX38873_v2_000001920.1
mRNA
FRAEX38873_v2_000001930.1
mRNA
FRAEX38873_v2_000001940.1
mRNA
FRAEX38873_v2_000001950.1
mRNA
FRAEX38873_v2_000001960.1
mRNA
FRAEX38873_v2_000001970.1
mRNA

CHAPTER 5

Loading GFF3

5.1 Load the landmarks

First, load the landmark scaffolds. The repo includes a FASTA file with scaffold names only, no sequences, for this purpose. Use the FASTA loader as described in the mRNA section: you do not need to define a parent relationship. You can use the SO term contig for the type.

5.2 Load the GFF file

Consider the example GFF file below.

```
##gff-version 3
Contig0 FRAEX38873_v2   gene    16315   44054   .   +   .   ID=FRAEX38873_v2_
↪000000010;Name=FRAEX38873_v2_000000010;biotype=protein_coding
Contig0 FRAEX38873_v2   mRNA    16315   44054   .   +   .   ID=FRAEX38873_v2_
↪000000010.1;Parent=FRAEX38873_v2_000000010;Name=FRAEX38873_v2_000000010.1;
↪biotype=protein_coding;AED=0.05
Contig0 FRAEX38873_v2   five_prime_UTR  16315   16557   .   +   .   ID=FRAEX38873_v2_
↪000000010.1.5utr1;Parent=FRAEX38873_v2_000000010.1
Contig0 FRAEX38873_v2   exon    16315   16967   .   +   .   ID=FRAEX38873_v2_
↪000000010.1.exon1;Parent=FRAEX38873_v2_000000010.1
Contig0 FRAEX38873_v2   CDS 16558   16967   .   +   0   ID=FRAEX38873_v2_000000010.1.
↪cds1;Parent=FRAEX38873_v2_
```

The below table explains each column.

5.2.1 Preprocessing

Every line of the GFF file will result in a **new feature**. The above example will create gene, mRNA, five_prime_UTR, exon, CDS, **and protein** features (see below for how to skip protein creation). If you'd like to not load five_prime_UTR features, for example, delete them from the file beforehand.

5.2.2 The GFF Importer

First, upload the file. In order to use the GUI uploader, the file extension should be `.gff` or `.gff3`. See below for information on GFF types.

Landmark Type

The **landmark** is the Chado feature on which the individual features are being mapped. This is typically a scaffold, contig, or chromosome (we chose contig above). If your landmarks are not uniquely named for this organism, you can specify the type here.

Protein names

As before, you may need to specify a regexp so that your proteins are correctly linked to your mRNA. Note that if you dont specify a protein regexp, it will look for proteins that are [mrna_name]-protein. This could result in new proteins being inserted accidentally! I've submitted a change that will allow you to **skip creating proteins** in this manner, look for it soon.

5.3 A note on GFF versions

GFF files are not the most uniform files around. There are GFF, GFF2, GTF, and GFF3. The Tripal GFF loader does its best, but it was designed to work withGFF3.

CHAPTER 6

Loading BLAST Annotations

6.1 Creating An Analysis

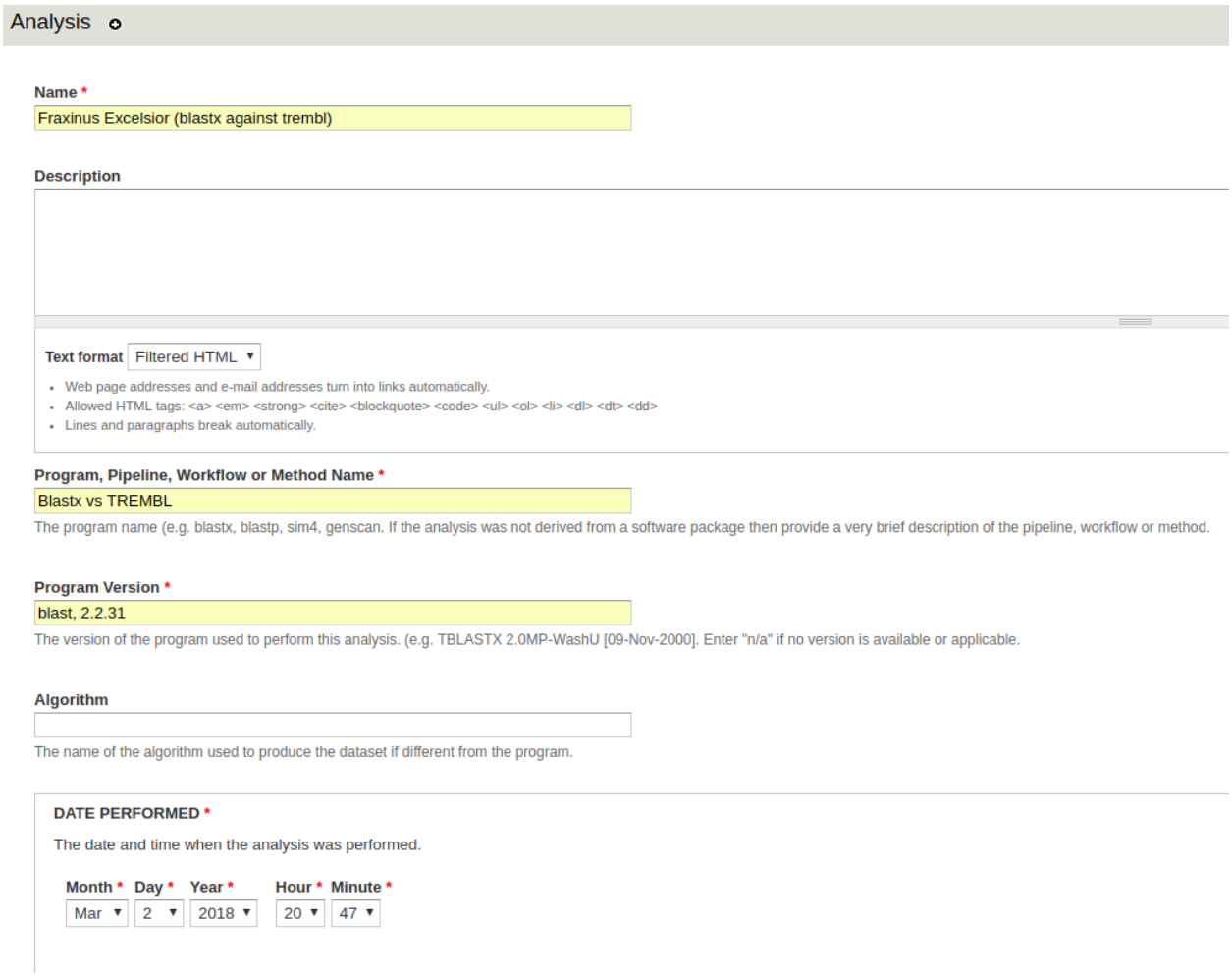
To load a blast analysis, navigate to **Content > Tripal Content**. At the top of the page, click **Add Tripal Content** and select **Analysis** from the list of content types. Some sites may have custom analysis types for each type of analysis performed. For our dataset, we need to make two analyses: one for TrEMBL and one for Swiss-Prot.

(note: the above step is optional, but recommended).

Enter data into the following fields:

- **Analysis Name** - The name should be **<organism common name> (<blast version> against <database>)**. For example **American Chestnut (blastx against sprot)**.
- **Program, Pipeline Name or Method Name** - Note that as of the time that this is being written, an analysis will not be saved if this field matches the Program, Pipeline Name or Method Name of an existing analysis. For this reason, it's recommend that you use **Blastx vs Swiss-Prot** for sprot and **Blastx vs TREMBL** for TREMBL.
- **Program, Pipeline or Method version** - Something along the lines of **blast, 2.2.31**.
- **Date Performed** - This should be the date the blast was run. If the blast process of scripts took several days, use the first day the job was created. If no date can be ascertained, then use 01, 01, 1900.
- **Data Source Name** - This will be the name of the unigene. There is not really a standard for a source that is a whole genome (like Chinese chestnut).
- **Data Source Version** - This is the version of the unigene or assembly. This field is optional and may be left blank.

Other fields may be left at their default values or empty. Click save.



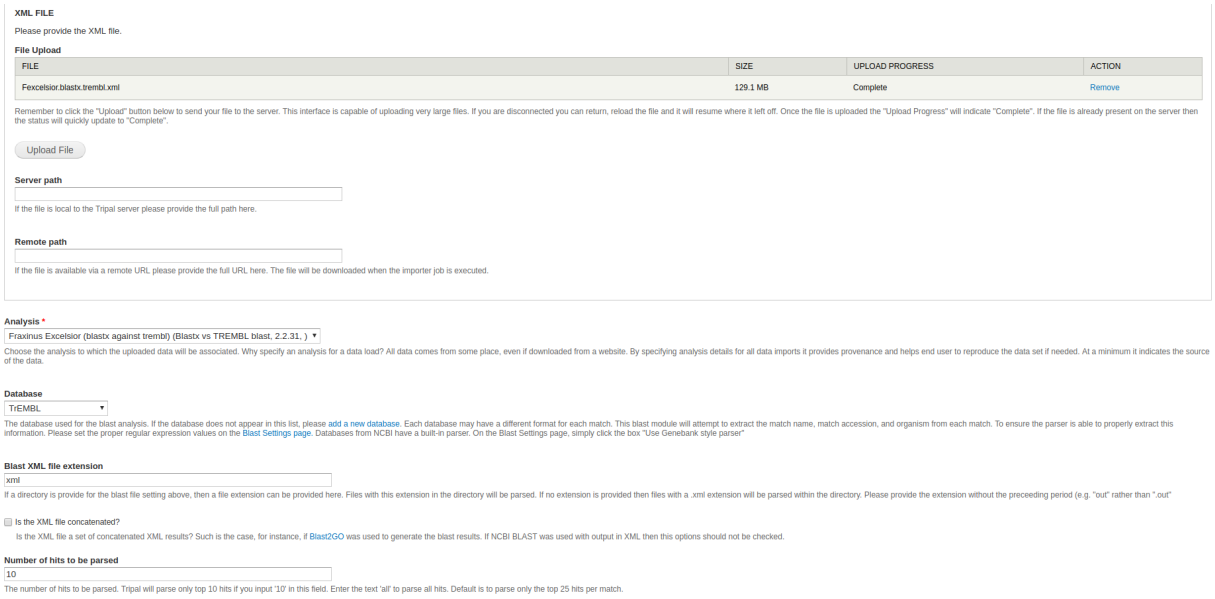
6.2 Loading BLAST Results

The BLAST loader is handled by the `tripal_analysis_blast` module. The BLAST loader can only load data from blast results in XML format.

Locate the BLAST loader from the menu through **Tripal > Data Loaders > Chado BLAST XML Results Loader.**

- **XML File** - Select and upload a blast xml file or provide a path to the blast xml file. If you are using a path, do not provide the file extension. If you input a directory without the tailing slash, all xml files in the directory will be loaded.
- **Analysis** - Select the newly created blast analysis.
- **Database** - You will need to create database entries for Swiss-prot and TrEMBL. Select the database that corresponds to the XML file you're loading (i.e. Swiss-prot for sprot and trembl for trembl).
- **Blast XML File Extension** - If you provided a path to the xml file instead of uploading a file directly, this would be the time to specify the file extension. This would typically be set to **xml.**
- **Number of hits to be parsed** - Set this value to **10.**

Other fields may be left at their default values or empty. Click **Import File** at the bottom of the page. You will need to run the job provided.



6.3 Viewing BLAST Results

Most fields are not enabled by default: this includes the BLAST results field. In order for the BLAST results to show up on mRNA entities, we must enable the field.

From the menu, navigate to **Structure > Tripal Content Types.** If the field `format__blast_display` is not listed, you should press the "Check for new fields" button in the upper left, and the field should be automatically added (but disabled by default). In the new window, select **manage display** in the table next to the content type **mRNA.**

At the bottom of this window is a field of **disabled** content types, under which **Blast Results** should be located. Drag Blast Results out of the disabled field.

Blast results should now be viewable in any mRNA content.

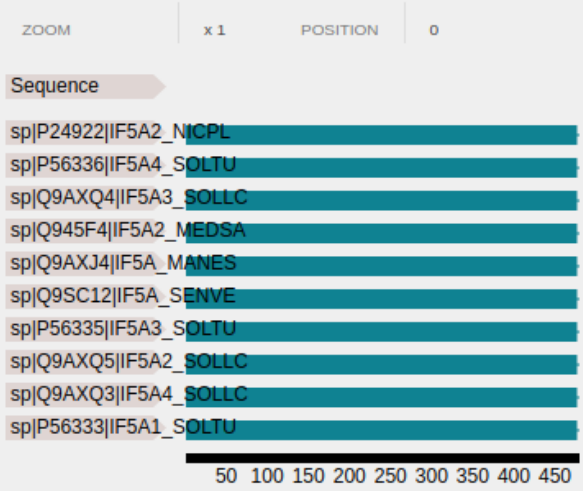
FRAEX38873_v2_000001910.1

View | Edit

Sequences
Summary

Summary

Resource Type	mRNA
Organism	Fraxinus excelsior
Name	FRAEX38873_v2_000001910.1
Identifier	FRAEX38873_v2_000001910.1
Time Accessioned	Friday, March 2, 2018 - 20:54
Time Last Modified	Friday, March 2, 2018 - 20:54
Blast Results	The following BLAST results are available for this feature: • BLAST of FRAEX38873_v2_000001910.1 vs. Swiss-Prot • BLAST of FRAEX38873_v2_000001910.1 vs. TrEMBL **BLAST of FRAEX38873_v2_000001910.1 vs. Swiss-Prot** Analysis Date: 2018-01-01 Analysis Name: Fraxinus Excelsior (blastx against sprot) Total hits: 10



Match Name	Stats	Description
sp|P24922|IF5A2_NICPL	E-Value: 3.821e-110, PID: 94.34	Eukaryotic translation initiation factor 5A-2 OS=N... [more]

CHAPTER 7

Loading InterProScan Annotations

7.1 Creating An Analysis

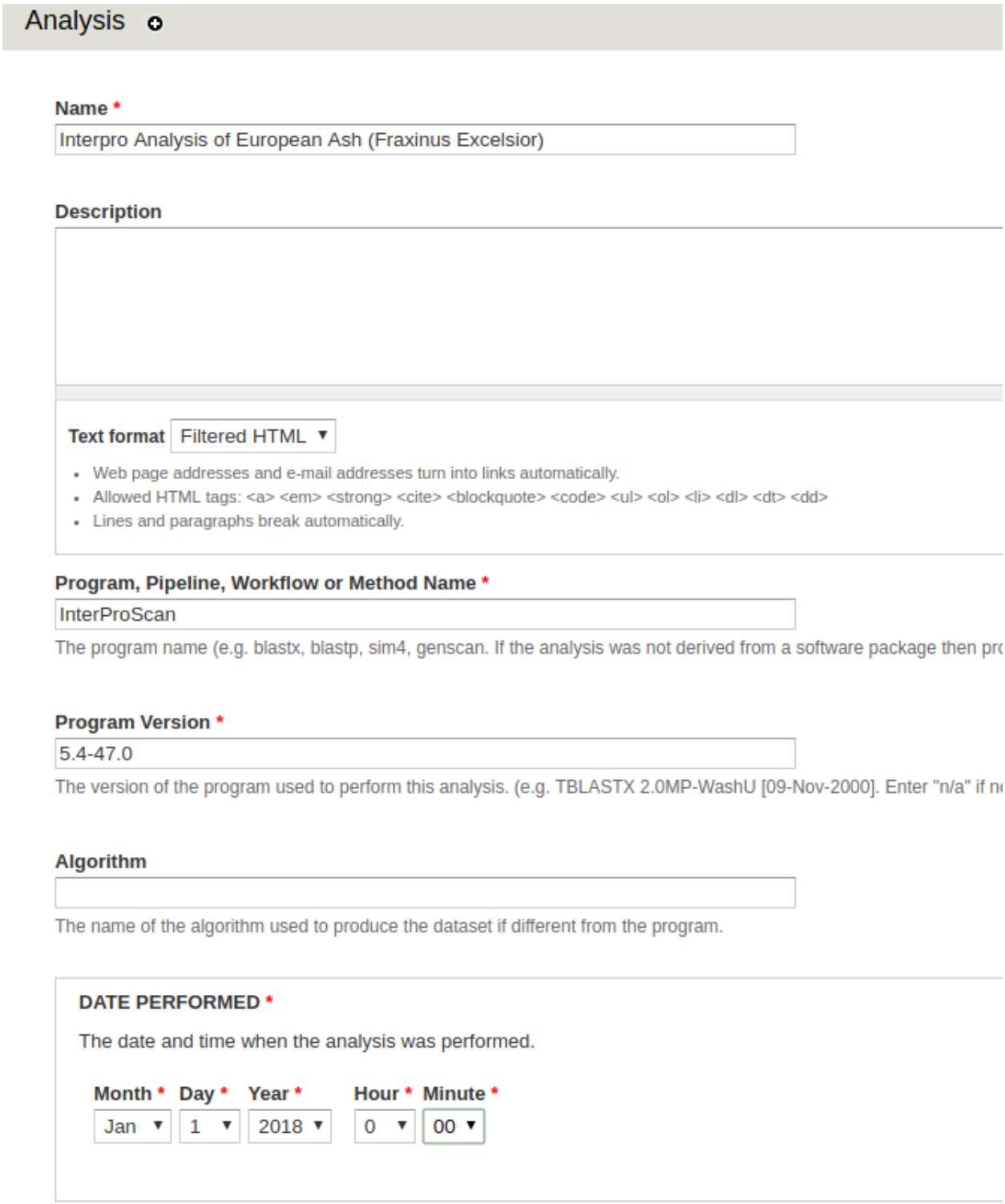
To load an interpro analysis, we first need an analysis to associate it withs. Navigate to **Content > Tripal Content**. At the top of the page, click **Add Tripal Content** and select **Analysis** from the list of content types. Some sites may have custom analysis types for each type of analysis performed.

(note: the above step is optional, but recommended).

Enter data into the following fields:

- **Analysis Name** - The name should be something like **Interpro Analysis of ( )**. For example: **Interpro Analysis of Honey Locust (Gleditsia triacanthos).**
- **Program, Pipeline Name or Method Name** - This should be InterProScan.
- **Program, Pipeline or Method version** - The version of interproscan. For example, 5.4-47.0.
- **Date Performed** - This should be the date the blast was run. If the blast process of scripts took several days, use the first day the job was created. If no date can be ascertained, then use 01, 01, 1900.

Other fields may be left at their default values or empty. Click save.



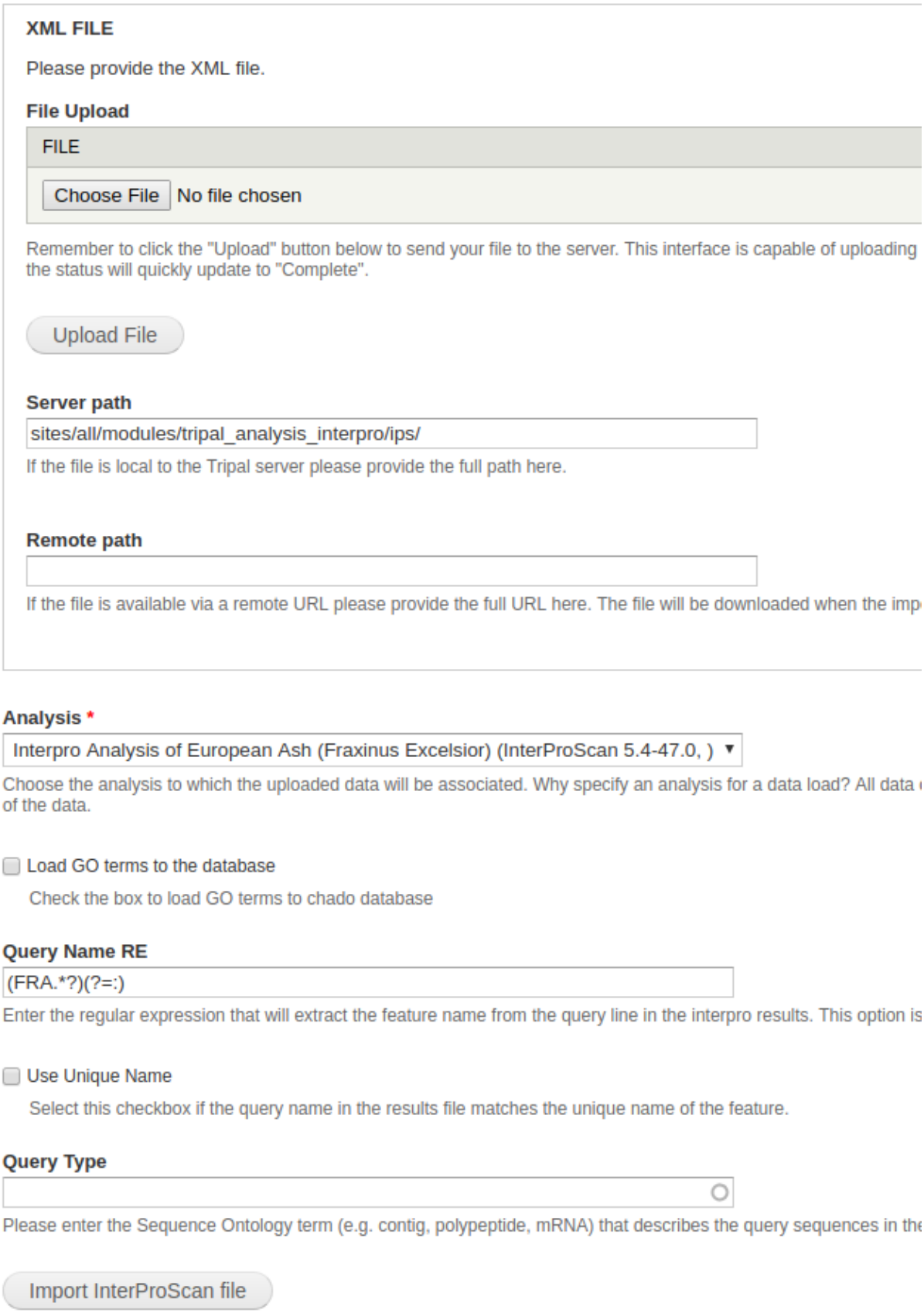
7.2 Loading InterProScan Results

The InterProScan loader is handled by the `tripal_analysis_interpro` module. The InterProScan loader can only load data from InterProScan results in the xml format.

Locate the InterProScan loader from the menu through **Tripal > Data Loaders > Chado InterproScan XML Results Loader.**

- **XML File** - You will need to upload an entire directory of xml files, so enter a server path that will locate the directory containing the xml files of the InterProScan results.
- **Analysis** - Select the newly created interpro analysis.
- **Query Name RE** - You will need to use the same regexp you used to load in the polypeptides. For this dataset, no regexp is needed.

Other fields may be left at their default values or empty. Click **Import File** at the bottom of the page. You will need to run the job provided.


XML FILE
Please provide the XML file.
File Upload
FILE
Choose File
No file chosen
Remember to click the "Upload" button below to send your file to the server. This interface is capable of uploading
the status will quickly update to "Complete".
Upload File
Server path
sites/all/modules/tripal_analysis_interpro/ips/
If the file is local to the Tripal server please provide the full path here.
Remote path
If the file is available via a remote URL please provide the full URL here. The file will be downloaded when the imp
Analysis *
Interpro Analysis of European Ash (Fraxinus Excelsior) (InterProScan 5.4-47.0,)
Choose the analysis to which the uploaded data will be associated. Why specify an analysis for a data load? All data
of the data.
Load GO terms to the database
Check the box to load GO terms to chado database
Query Name RE
(FRA.*?)(?=:)
Enter the regular expression that will extract the feature name from the query line in the interpro results. This option is
Use Unique Name
Select this checkbox if the query name in the results file matches the unique name of the feature.
Query Type
Please enter the Sequence Ontology term (e.g. contig, polypeptide, mRNA) that describes the query sequences in the
Import InterProScan file

7.3 Viewing InterProScan Results

Unless specified otherwise, InterProScan results are hidden by default.

From the menu, navigate to **Structure > Tripal Content Types.** In the new window, select **manage display** in the table next to the content type **mRNA.**

At the bottom of this window is a field of **disabled** content types, under which **InterPro results** should be located. Drag InterPro results out of the disabled field.

Blast results should now be viewable in any mRNA content.

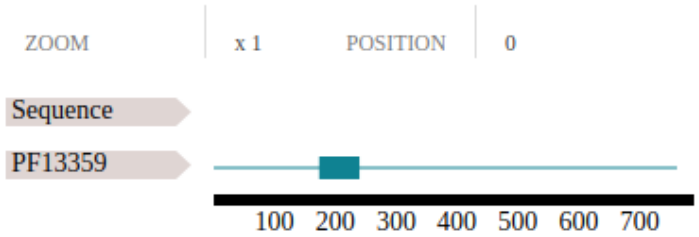
NOTE: If InterPro results does not appear as a field, navigate to **manage fields** and click **Check for new fields.**

InterPro results:

The following InterPro results are available for this feature:

Analysis Name: Interpro Analysis of European Ash (Fraxinus Excelsior)

Date Performed: 2018-01-01



IPR Term	IPR Description	Source	Source Term	Source Description	Alignment
IPR027806	Harbinger transposase-derived nuclease domain	PFAM	PF13359	DDE_Tnp_4	coord: 174..240 e-value: 4.9E-6 score: 26.1

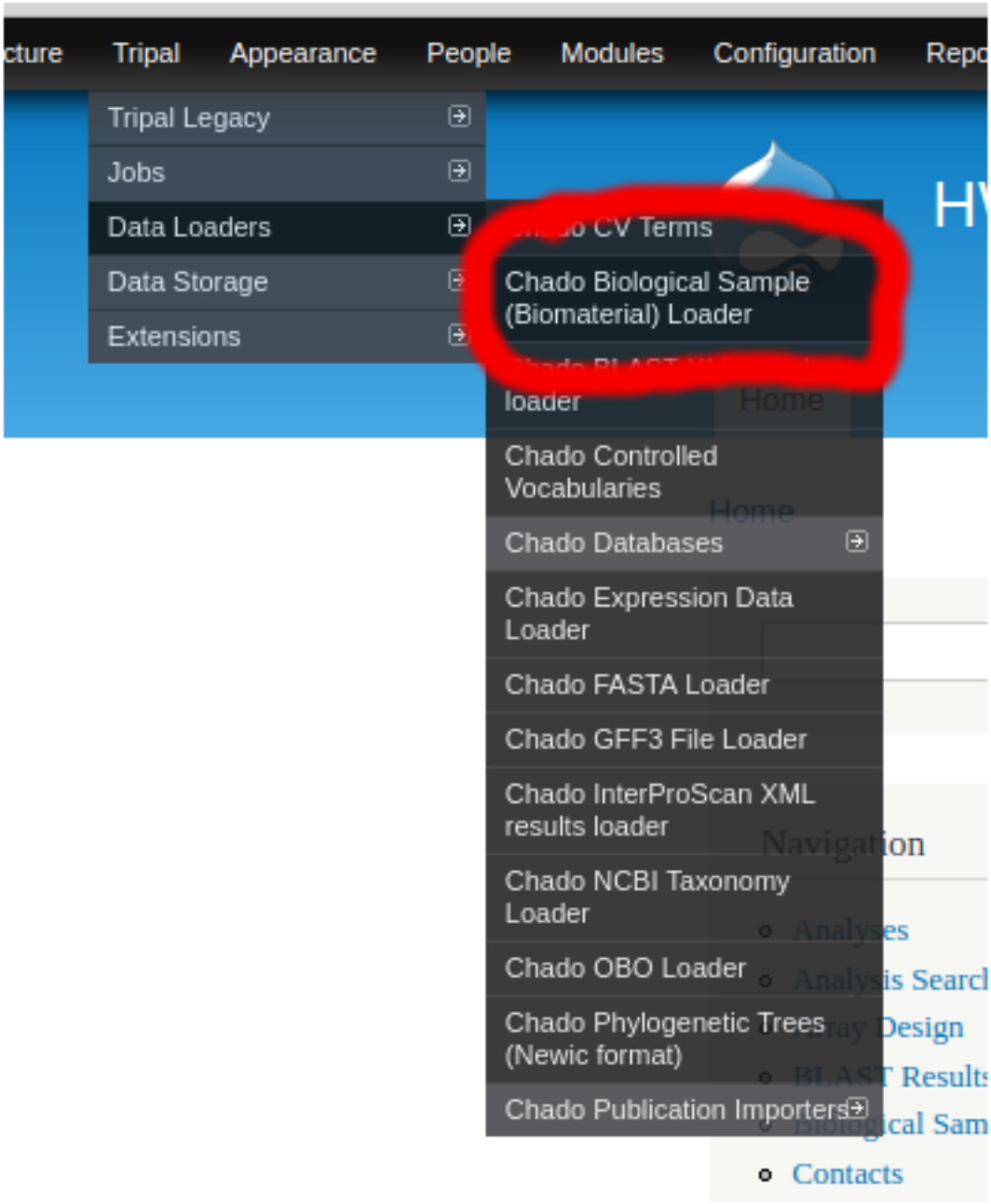
CHAPTER 8

Load Biosamples

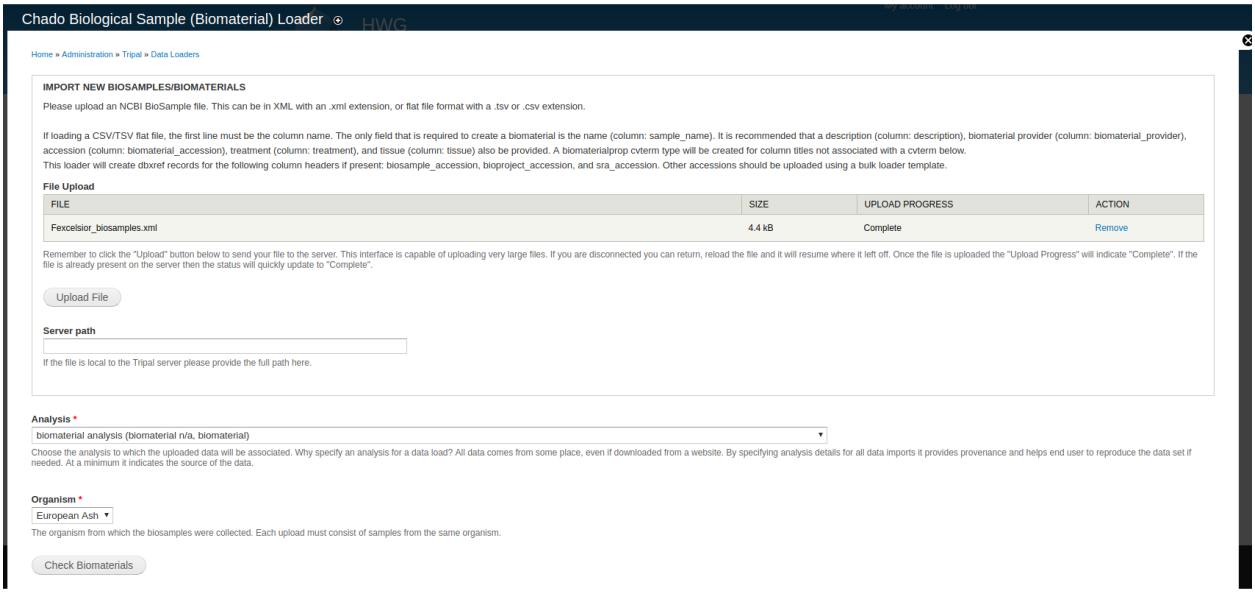
The biosample importer allows you to specify an analysis: for this pipeline, we won't.

8.1 Load the samples

The Biosample loader is provided by the `tripal_biomaterial` module (distributed with the `tripal_analysis_expression` module), and is located at `admin/tripal/loaders/chado_biosample_loader`. Biosamples can be loaded as either an `xml` file, or a set of `csv/tsv` files. `xml` is preffered, and can be optained from NCBI. `csv/tsv` format requires that the first line is the column names for the biosample properties.



Select the organism. Note that loading biomaterials from multiple species at a time is not supported. Split up your files to load one organism at a time.

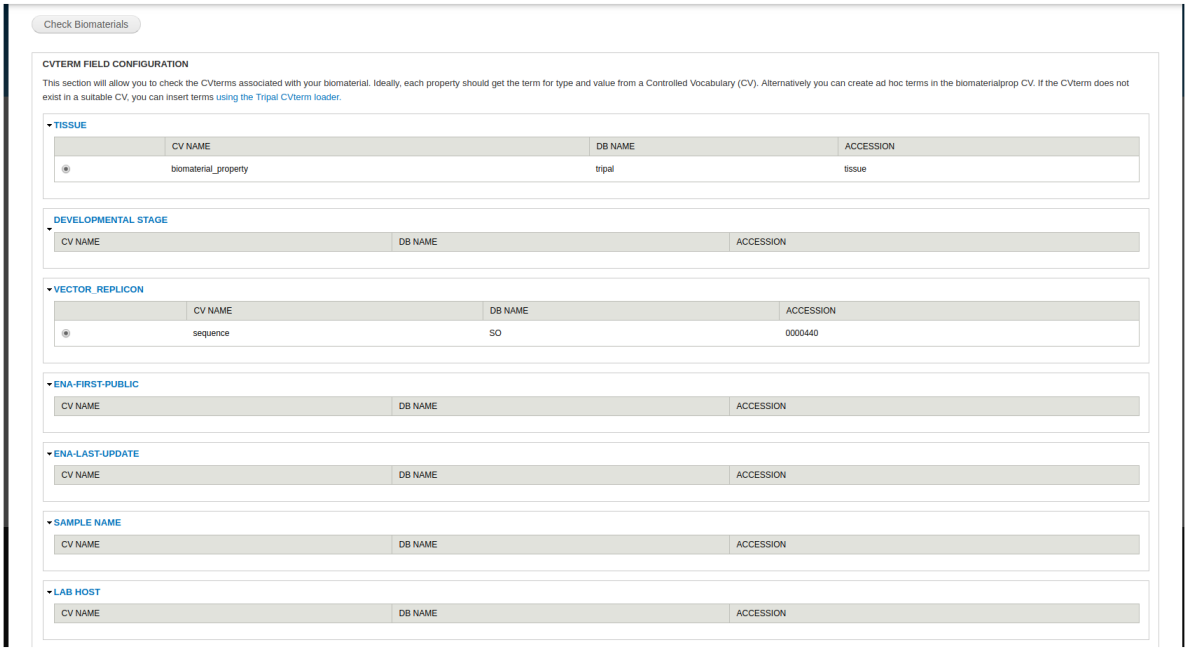


After your file is uploaded, press the **Check Biomaterials** button to access the *CVTERM FIELD CONFIGURATION* section. The section will list each property associated with your biosamples. If a term exists in the CVterm database matching the property, it will appear in this section. For **every biosample property**, associate the property with a CVterm. In a perfect world, all terms will map to an established CV (sequence ontology, plant trait ontology, etc). If no term is listed, or if the only terms listed are biomaterial_property terms, you should

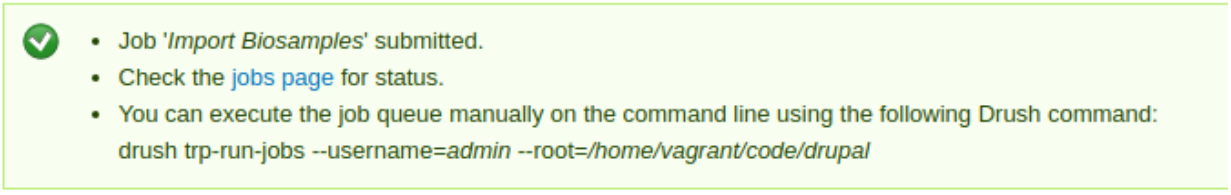
- Load appropriate CVterms for each property. You can load an entire CV, or individual CVterms using the CVterm loader located at `admin/tripal/loaders/chado_cvterms`.
- Rename the properties in your source file so that they match existing CVterms. You can look up available CVterms at `admin/tripal/loaders/chado_cvterms`.
- Re-upload the biosample file, and rerun **Check Biomaterials**.
- Repeat this process until you have suitable CVterms associated with all biosample properties.

New feature: the above process can now also be applied to the **property values**. Please see the github documentation for more information.

That said, you can import your biosamples without assigning CVterms. In this case, the generic biomaterial_property CV will be used.



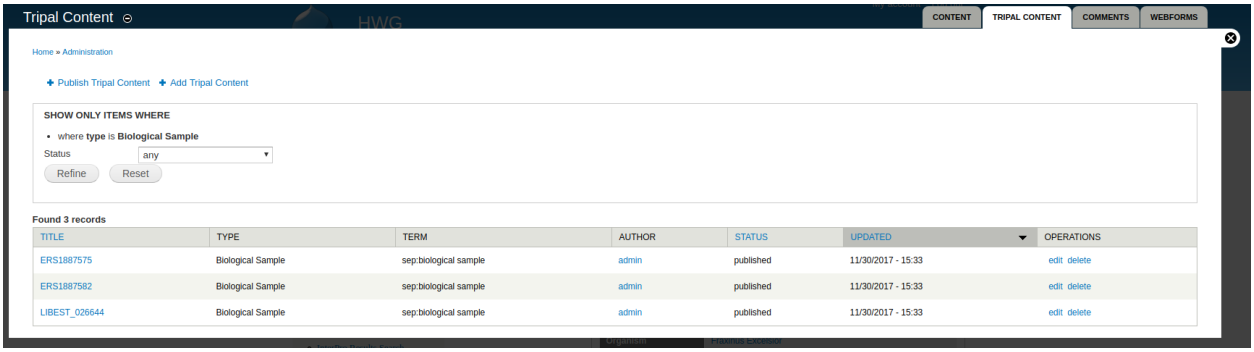
After clicking Submit, you will need to run the job for the samples to be processed.



8.2 Publish the biosamples

Once the samples are loaded, they must be published to appear as entities. To do so, go to `Content -> Tripal Content -> Publish Tripal Content` and select the **Biological Sample** content type.

Once published, the biomaterial data can be located through the menu under **Content > Tripal Content.** Filter results by **Type > Biological Sample.**



Below is an example of successfully uploaded biomaterial data.

CHAPTER 9

Loading Expression Data

9.1 Create Analysis

You will first need an analysis to associate the expression data. To do so, navigate to **Tripal_content -> Add Tripal Content**. Select **Analysis**.

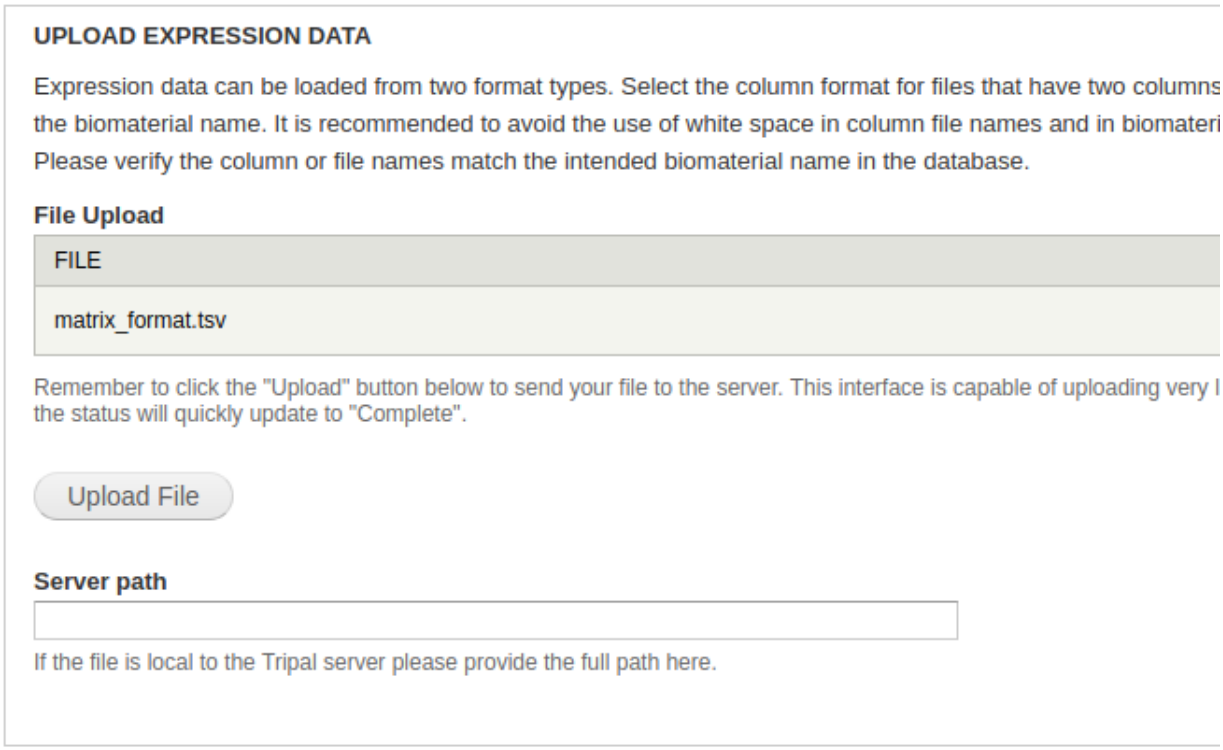
- **Name** - Something along the lines of **Fraxinus Excelsior Expression**.
- **Program, Pipeline, Workflow or Method Name** - Something along the lines of **e.g DESESQ2**.
- **Program Version** - Something along the lines of **e.g 1.0**.
- **Date Performed** - Leave at default.

9.2 Load the expression data

Expression data is loaded by the `tripal_analysis_expression` module using the Chado Expression loader, located at `admin/tripal/loaders/Chado_Expression_Data_Loader`. Expression data should be in column or matrix format.

- **File Upload** - For this dataset, the simplest method is uploading the **.tsv** file.
- **Analysis** - Select the same analysis specified for the biosamples.
- **Organism** - Select an organism. In this case, the organism is **European Ash**.
- **Source File Type** - If you're uploading the .tsv file, select **Matrix**. If you're uploaded the .txt files, select **Column**. Keep in mind that if you upload the .tsv file, you do not need to upload the .txt files.
- **Name Match Type** - Select unique name.

All other fields can be ignored. Click **Import expression data**. A green header should appear at the top of the page with a job for you to run. Run it and you're done.



Analysis *

Fraxinus Excelsior Expression (expression n/a,)

Choose the analysis to which the uploaded data will be associated. Why specify an analysis for a data load? All data come
of the data.

Organism *

European Ash

The organism from which the biosamples were collected.

Source File Type *

- ○ Column Format
- ◉ Matrix Format

Data can be loaded from two format types. Select the column format for files that have two columns - transcript id and expr
avoid the use of white space in column file names and in biomaterial names.

Name Match Type *

- ○ Name
- ◉ Unique name

Expression data can be associated with features via the feature name or the feature unique name.

9.2.1 Publishing

Publishing is not necessary for expression data, as we don't create any new Tripal Entities.

9.3 Viewing Expression Results

The easiest way to check to see if your expression results were successfully uploaded is by referring to the `chado_elementresult` table. If the table has contents, you know the results were uploaded successfully.

If you can't access the database, the alternative is to display the expression results directly on a feature page. Results are hidden by default, so we have to enable them in order to view them. This can be done with the admin menu by navigating to **Structure > Tripal Content Types**. In the row **mRNA**, click **Manage Fields** and towards the top of the window, click **Check for new fields**. This will take a moment, but a new field should be found called `data__gene_expression_data`.

At the top of the window, click **Manage Display**. Scroll all the way to the bottom of the window and look for a **Disabled** field, in which an **Expression** field should be contained. Move this out of the disabled table.

Now our results should be available to view. Navigate to any feature page (from the admin menu, Content > Tripal Content, click any record with type mRNA) and you should see your expression results.

Select an Expression Analysis

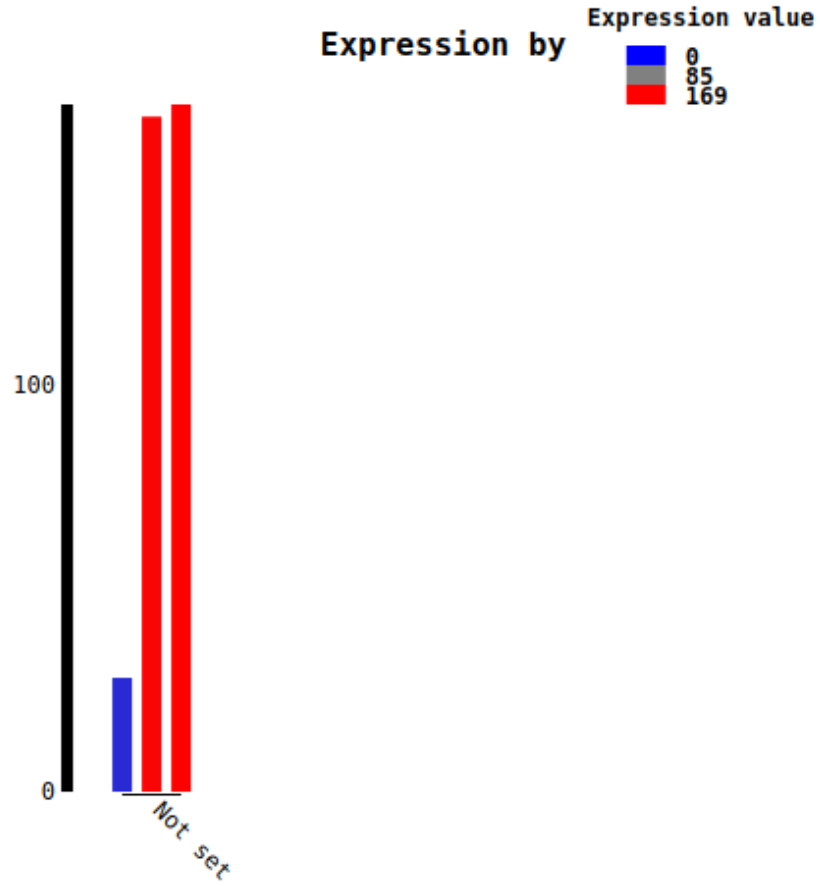
Fraxinus Excelsior Expression ▾

Select a property to group and sort biological samples ENA-FIRST-PUBLIC ▾

Select a property to color biological samples Expression value ▾

Hover the mouse over a column in the graph to view more information about that biological sample. You can click and drag to rearrange groups along the x-axis. You can also click and drag to move the legend.

Only Non-Zero Values | Reset



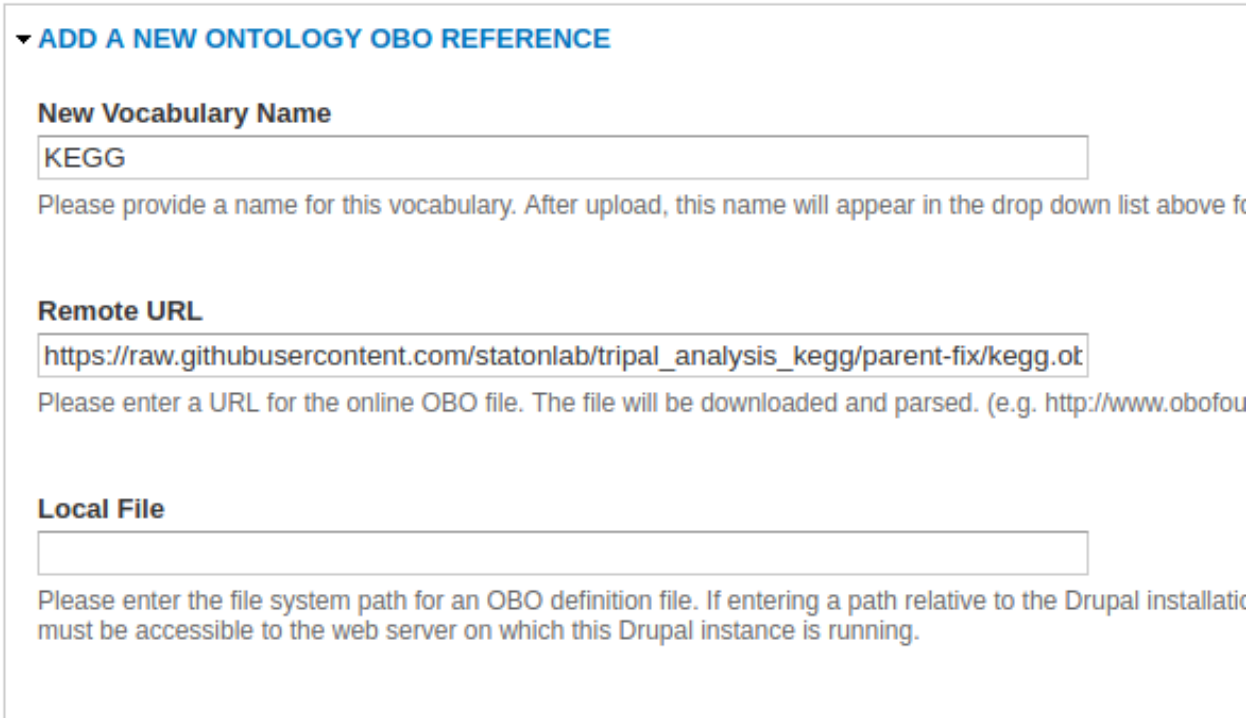
CHAPTER 10

Loading KEGG Annotations

10.1 Loading the KEGG Ontology

You will need to load the KEGG terms in before you can begin loading data. In the admin menu, navigate to **Tripal > Data Loaders > Chado Vocabularies > OBO Vocabulary Loader**. Click **Add a New Ontology OBO Reference**.

- **New Vocabulary Name** - You can just call this vocabulary KEGG.
- **Remote URL** - You can use the OBO from the Staton Lab repo or the Tripal repo by copy and pasting the URL into this field.
- **Local File** - This field is an alternative to the Remote URL field. If you don't want to use a link, you can download the KEGG Ontology instead and simply specify the file path relative to your Drupal installation instead (e.g. sites/default/files/kegg.obo).

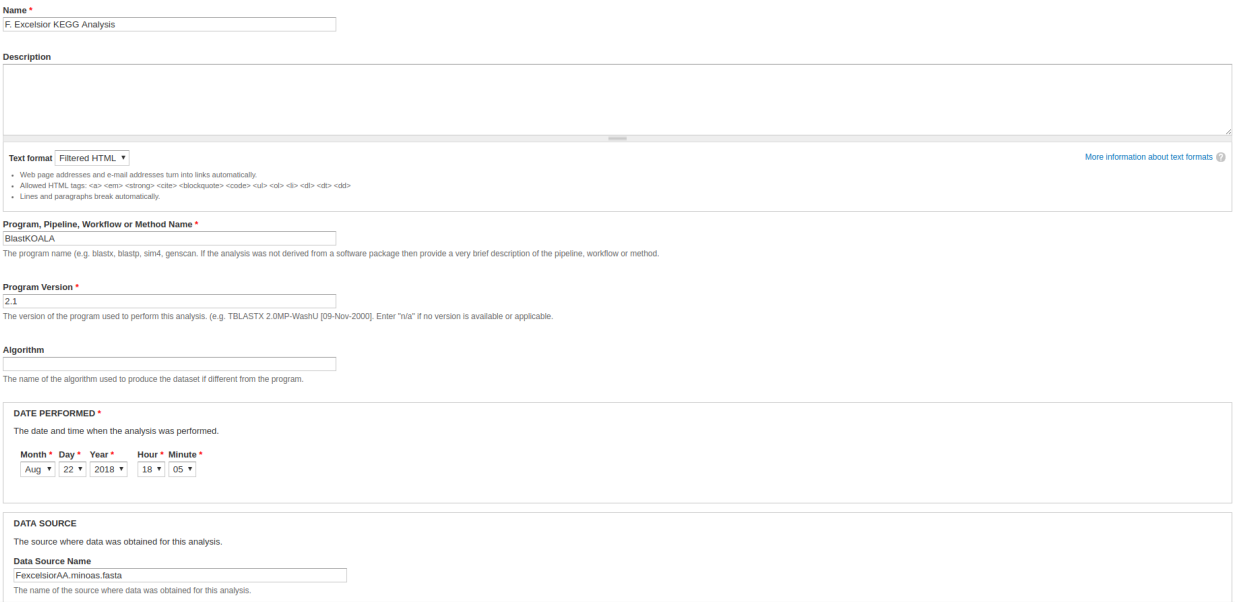


10.2 Create an Analysis

We will need to create an analysis with which to associate our KEGG data. Navigate to Content > Tripal Content. At the top of the page, click Add Tripal Content and select Analysis from the list of content types. Some sites may have custom analysis types for each type of analysis performed.

- **Name** - Something along the lines of, **F. Excelsior KEGG annotation**.
- **Program, Pipeline, Workflow or Method Name** - Something along the lines of, **BlastKOALA**.
- **Program Version** - Something along the lines of, **2.1**.
- **Date Performed** - You can keep this default, but it's common to set this to an arbitrary date (e.g. January 1st, 1900) if you're unsure of the time when the analysis was performed.
- **Data Source Name** - This should be named after the protein file from which the KEGG data was obtained (e.g. FexcelsiorAA.minoas.fasta).

All other fields can be left blank or at their default values. Click save.

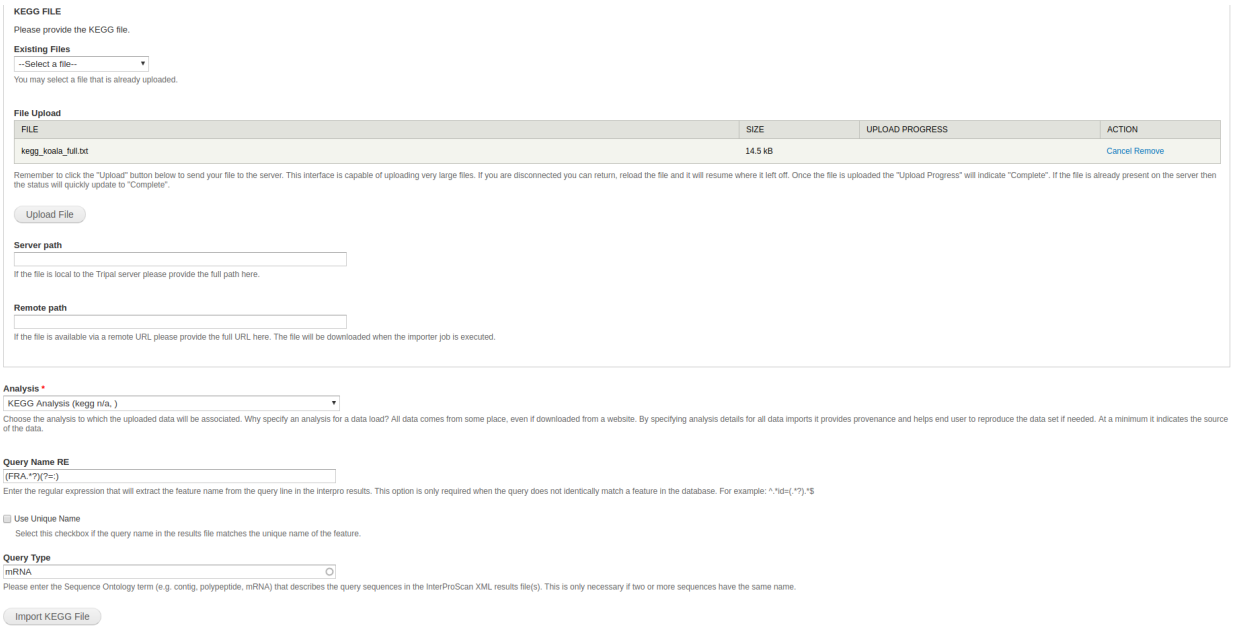


10.3 Loading KEGG Data

Now that we have the ontology and an analysis that we can associate our data with, we can begin loading the KEGG data. Navigate to **Tripal > Data Loaders > Chado KEGG Loader** in the admin menu.

- **File Upload** - The KEGG file in the dataset is `f_excelsior_ko.txt`.
- **Analysis** - Select the KEGG analysis created for this data (i.e. the one created above).
- **Query Name RE** - A regular expression to match the names in the kegg output to features in the database.
- **Query Type** - The feature type you'd like to associate the annotations with. Can be left blank if the name is unique and that is the desired feature type.

Users may choose to associate the KEGG annotations with the polypeptides themselves, **or** with the parent mRNA features in which case specifying a regular expression and/or type is necessary.



Once the fields are filled out, you can click `Import KEGG File`. Run the job provided and you should be good to go.

CHAPTER 11

Annotating on Galaxy

Coming soon

CHAPTER 12

Understanding linking records

This section is provided to help users understand why we specify which records data is associated with when loading.

Many of the load steps require you to specify **which Chado record** to associate something with, or **how to find a parent record**. A polypeptide feature is derived from an mRNA (the "central dogma" in biology): the mRNA record in chado.feature is the **parent** record of the polypeptide record in chado.feature.

12.1 Who is the Entity?

Note that this guide is written assuming that your **entity records** (the records that Tripal creates pages for) will be mRNA **only**. Other features are still created in Chado, they just dont have their own dedicated page. This is to prevent a user from ahving to click through from a scaffold to a gene to an mRNA to a protein just to see the protein sequence. This means, however, that when you load in annotations for other features, you have to take care of what the annotation is associated with. Interproscan annotations, for example, we associate with the mRNA, despite running them on the protein, because we want them to show up on the mRNA page. This is the purpose of the regular expressions and specifying the type when running these loaders.

12.2 Load Order and Regular Expressions

We (HardwoodGenomics, the developers fro this module) have a history of loading the FASTA files to create feature records for mRNA, then loading proteins, and finally the GFF. This means we link the proteins to the mRNA at the FASTA loader step. In order for this to work, you need a regexp that can link protein to mRNA. In subsequent loading steps, where annotations were done using the protein, we associate the annotations with the **parent mRNA** instead (see above for why).

For most genbank entries, this wont work. The numbers assigned to the protein XP_0000 record and the XM_0000 record might be different! In these situations, you must load the GFF first, which hopefully manually designates which proteins belong to which mRNA. However, when you load the annotations done with the proteins (such as the interproscan annotations) they will associate with the protein.

Without a regular expression to link these, you may instead opt to create entity records for the proteins as well as the mRNA. Alternatively, custom fields would need to be created to display, for example, Interproscan annotations associated with proteins on the parent MRNA page (this is what is currently done with the protein sequence field, for example).

CHAPTER 13

License

This project is open source and provided under the GPL-3.0 license: please see the GitHub repo for more information at https://github.com/statonlab/tripal_alchemist/blob/master/LICENSE.

It was created by Bradford Condon and Meg Staton from the University of Tennessee Knoxville. If you would like to make a contribution, simply fork the repo and make a pull request from there.

The project "logo" is derived from the collectible card game Hearthstone, copyright © Blizzard Entertainment, Inc. Hearthstone® is a registered trademark of Blizzard Entertainment, Inc. Tripal Alchemist is not affiliated or associated with or endorsed by Hearthstone® or Blizzard Entertainment, Inc.