
Tripal Network Documentation

Release 7.x-1.x

Stephen Ficklin

Mar 15, 2022

TRIPAL NETWORK EXTENSION MODULE

1	Installation	3
1.1	Step 1: Module Installation	3
1.2	Step 2: Set Permissions	4
2	Prepare Network Data	7
2.1	Preparing Genomic Data	7
2.2	Preparing Functional Data	8
2.3	Generating a GraphML file	9
2.3.1	Using Cytoscape	9
2.3.2	From a Tab Delimited File	9
2.3.3	From WGCNA Networks	9
2.3.4	From KINC Networks	10
3	Loading Network Data	11
3.1	Importing Functional Data	11
3.2	Importing a Network File	15
3.2.1	Step 1: Create a Network	15
3.2.2	Step 2: Create an Analysis Record	16
3.2.3	Step 3: Import the GraphML	18
3.2.3.1	Loader Stage 1	18
3.2.3.2	Loader Stage 2	19
3.2.3.3	Loader Stage 3	21
4	Visualization	23
4.1	Feature Neighborhood Viewer	23
4.2	The 3D Network Exploration App	25
4.2.1	Network Selection	26
4.2.2	Interactive Network Exploration	26

Warning: The Tripal Network module is currently in alpha release mode. If you find bugs please report them on the [GitHub issue queue](#)

The Tripal Network module supports inclusion of biological networks in a Tripal site. It provides:

- A variety of new_chado_tables for storing networks and because Chado does not currently have tables for storing compounds and pathways the module adds those as well. All of these new tables follow Chado design standards.
- A new Network content type. Network pages describe the networks that have been added to the site.
- Tripal fields linking existing genomic features pages to networks. The field provides a small interactive visualization of the first neighbor connections in the networks for a given feature.
- A full 3D application viewer for exploring the networks housed in the site.

Warning: The Tripal Network module is **compatible only with Tripal v3.6 or higher**.

INSTALLATION

1.1 Step 1: Module Installation

The Tripal Network module is available as a full Drupal module. Therefore, it can be installed following the typical Drupal module installation either via the GUI or via Drush. We'll do so here using a Drush command:

```
drush pm-enable tripal_network
```

If you do not have the module already downloaded and available in the *sites/all/modules* folder of your Drupal site, then you will be asked if you would like to download and then enable the module.

The module will create a new 'Network' content type. It will also create the a series of new tables in your Chado database for storing networks. Additionally, Chado does not have support for metabolic networks, so several new tables will be added for storing compounds, pathways and reactions. For a full listing of new Chado tables see the *new_chado_tables* section.

Upon installation you should see the following:

```
$ drush pm-enable tripal_network

The following extensions will be enabled: tripal_file
Do you really want to continue? (y/n): y
INFO (TRIPAL_ENTITIES): Done.
tripal_network was enabled successfully.
↳ [ok]
Custom table, 'compound' , created successfully.
↳ [status]
Custom table, 'compoundprop' , created successfully.
↳ [status]
Custom table, 'compound_dbxref' , created successfully.
↳ [status]
Custom table, 'compound_synonym' , created successfully.
↳ [status]
Custom table, 'reaction' , created successfully.
↳ [status]
Custom table, 'reactionprop' , created successfully.
↳ [status]
Custom table, 'reaction_substrate' , created successfully.
↳ [status]
Custom table, 'reaction_product' , created successfully.
↳ [status]
Custom table, 'pathway' , created successfully.
↳ [status]
```

(continues on next page)

(continued from previous page)

```
Custom table, 'pathwayprop' , created successfully.
↳ [status]
Custom table, 'pathway_dbxref' , created successfully.
↳ [status]
Custom table, 'pathway_feature' , created successfully.
↳ [status]
Custom table, 'network' , created successfully.
↳ [status]
Custom table, 'networkprop' , created successfully.
↳ [status]
Custom table, 'network_attr' , created successfully.
↳ [status]
Custom table, 'network_attrprop' , created successfully.
↳ [status]
Custom table, 'network_cvterm' , created successfully.
↳ [status]
Custom table, 'network_file' , created successfully.
↳ [status]
Custom table, 'network_node' , created successfully.
↳ [status]
Custom table, 'network_nodeprop' , created successfully.
↳ [status]
Custom table, 'network_feature' , created successfully.
↳ [status]
Custom table, 'network_compound' , created successfully.
↳ [status]
Custom table, 'network_edge' , created successfully.
↳ [status]
Custom table, 'network_edgeprop' , created successfully.
↳ [status]
Custom table, 'network_analysis' , created successfully.
↳ [status]
Custom table, 'network_analysisprop' , created successfully.
↳ [status]
Custom table, 'network_pub' , created successfully.
↳ [status]
Custom table, 'network_layout' , created successfully.
↳ [status]
```

1.2 Step 2: Set Permissions

Now that the Tripal Network module is installed, we must set permissions so that users can view and or create Network content types. To set permissions, navigate to **People** page and click the **Permissions** tab in the top right. Look for permissions that begin with prefix *Network:* and set the according to your needs.

Network: View Content Allow the user to view <i>Network</i> content	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>
Network: Create Content Allow the user to create <i>Network</i> content <i>Warning: Give to trusted roles only; this permission has security implications.</i>	<input type="checkbox"/>	<input type="checkbox"/>	<input checked="" type="checkbox"/>
Network: Edit Content Allow the user to edit <i>Network</i> content <i>Warning: Give to trusted roles only; this permission has security implications.</i>	<input type="checkbox"/>	<input type="checkbox"/>	<input checked="" type="checkbox"/>
Network: Unpublish Content Allow the user to unpublish <i>Network</i> content. Unpublishing of content removes it from visibility on the site but does not delete the record in the underlying database.	<input type="checkbox"/>	<input type="checkbox"/>	<input checked="" type="checkbox"/>
Network: Delete Content Allow the user to delete <i>Network</i> content. When content is deleted it is first unpublished and then deleted from the database. <i>Warning: Give to trusted roles only; this permission has security implications.</i>	<input type="checkbox"/>	<input type="checkbox"/>	<input checked="" type="checkbox"/>

Warning: You should not give the anonymous user any permissions other than ‘view’ permission.

PREPARE NETWORK DATA

The Tripal Network module is designed to support gene co-expression networks (GCNs), protein-protein interaction networks (PPIs), and top-down correlative metabolic networks. To load a network you must have the following data:

- Gene Co-expression Networks or Protein-Protein Interaction Networks:
 - Required
 - * **The gene, mRNA (transcript) or protein features.** You can load genes, mRNA and proteins using the built-in Tripal GFF3 and FASTA loaders.
 - * **The network graph.** The network graph is imported into Tripal using a GraphML loader provided by the Tripal Network module. All networks must first be converted into a GraphML format.
 - Optional
 - * **Function annotations** about each gene or mRNA. Often users want to explore the functions of important nodes in the network. This often includes Gene Ontology (GO) annotations, Protein domains and families (perhaps from InterProScan) and perhaps KEGG pathways and ortholog assignments. The Tripal network module will make available any functional annotation in the 3D viewer. Sometimes annotations can be included in the GFF3 file but often they are not. The Tripal Network module provides a Chado Bulk Loader template for importing functional data if it is not already present in the database.

Warning: Support for top-down metabolic networks is still under development. Check back for updates with a future release of the module.

2.1 Preparing Genomic Data

Prior to importing a network you must first import the genes, mRNA or protein features that will be used as nodes in the network. This is often done using the GFF3 or FASTA loaders of Tripal. [Instructions for using these importers](#) are available in the [Tripal User's Guide](#).

2.2 Preparing Functional Data

If functional data in the form of controlled vocabulary terms, for example Gene Ontology terms, are already assigned to the genes, transcripts or proteins used in the networks then you can skip this step. However, if you need to add these data, you can do so using the Tripal [Chado Bulk Loader](#). This loader allows for import of any tab-delimited file provided a template for it exists. The Tripal Network module provides a template for importing functional annotations in a tab-delimited file with rows of data ordered in the following columns:

1. Gene, mRNA or protein or protein name. This corresponds to the value in the Chado feature table name column
2. Term accession (e.g. “GO:0004396”). This value should include the vocabulary short name (e.g. “GO”), separated by a colon and the accession number.
3. Term name (e.g. “hexokinase activity”).
4. Controlled Vocabulary (CV) name (e.g. “molecular_function”).
5. Database name (e.g. “GO”). The database name is often the same as the short name of the vocabulary, but, it can be similar to the CV name or it may be different. If you’re unsure what value to use here please ask for help on the [Tripal Slack workspace](#).
6. Term definition (e.g. “Catalysis of the reaction: ATP + D-hexose = ADP + D-hexose 6-phosphate. Source: EC:2.7.1.1”)

Some other requirements - The file must be tab-delimited. - Quotations are not needed to separate values in columns. - A one-line header line is supported. - The features in the file must all belong to the same organism.

Below is an example of a compatible file for a few *Oryza sativa* (rice) genes.

Feature	Term	Name	CV	DB	Definition
LOC_Os06g41050	GO:0000014			single-stranded DNA	endodeoxyribonuclease activity
	molecular_function		GO		Catalysis of the hydrolysis of ester linkages within a
	single-stranded deoxyribonucleic acid molecule by creating internal breaks.				
LOC_Os01g60660	GO:0000049			tRNA binding	molecular_function
			GO		Binding
					to a transfer RNA.
LOC_Os02g46130	GO:0000166			nucleotide binding	molecular_function
			GO		
					Binding to a nucleotide, any compound consisting of a nucleoside that is esterified
					with (ortho)phosphate or an oligophosphate at any hydroxyl group on the ribose or
					deoxyribose.

2.3 Generating a GraphML file

GraphML is an XML format for storing network graphs. The Tripal Network module currently only supports import of networks in this format. Below are a few instructions for generating GraphML files

2.3.1 Using Cytoscape

Cytoscape is one of the most popular network visualization tools. You can import networks and visualize them in a 2D space. You can create a GraphML file using Cytoscape by first loading a network and then [exporting it in GraphML format](#).

2.3.2 From a Tab Delimited File

Currently, there is not a loader for networks in a tab delimited format. Use Cytoscape to first import the network in the tab delimited file then export to GraphML.

2.3.3 From WGCNA Networks

The [Weighted Gene Co-expression Network Analysis \(WGCNA\)](#) software is one of the most popular tools for creating networks. WGCNA has been cited in thousands of publications that incorporate networks into their analysis. One of the major benefits of WGCNA is that it can circumscribe genes into modules of highly interacting genes. Those modules can then be associated with clinical or physiological traits and experimental conditions. Often it is these modules that are reported but the underlying network is not. It is possible to extract the network from WGCNA, rather than just the module list, but the exact threshold to use is not precise.

Below is example R code that outputs a graphML file using functions from WGCNA and the [igraph R library](#):

```
# Import igraph
library(igraph)
library(WGCNA)

# Set the network threshold. This will affect the size of the network.
# You may need to adjust this value to generate the desired size network.
hard_threshold = 0.082

# Example code for running WGCNA (fill in appropriate arguments in place of the ...)
net = blockwiseModules(gemt, ..., saveTOMs = TRUE, saveTOMFileBase = "TOM")
blocks = sort(unique(net$blocks))

# Stores the network edges in an edges array.
edges = data.frame(fromNode= c(), toNode=c(), weight=c(), direction=c(), fromAltName=c(),
  ↪ toAltName=c())

# iterate through each block, load the TOM file and convert the
# network
for (i in blocks) {
  # Load the TOM from a file.
  load(net$TOMFiles[i])
  TOM_size = length(which(net$blocks == i))
  TOM = as.matrix(TOM, nrow=TOM_size, ncol=TOM_size)
  colnames(TOM) = colnames(gemt)[net$blockGenes[[i]]]
```

(continues on next page)

(continued from previous page)

```
row.names(TOM) = colnames(gemt)[net$blockGenes[[i]]]

cydata = exportNetworkToCytoscape(TOM, threshold = hard_threshold)
edges = rbind(edges, cydata$edgeData)
}

# Set the interaction to 'co' for co-expression or correlation and
# rename the columns to be more intuitive.
edges$Interaction = 'co'
output = edges[,c('fromNode','toNode','Interaction', 'weight')]
colnames(output) = c('Source', 'Target', 'Interaction', 'WGCNA_weight')

# Now save the edges as a graphML file.
g = graph_from_data_frame(edges, directed = FALSE)
write_graph(g, opt$network_graphml_file, 'graphml')
```

2.3.4 From KINC Networks

The Knowledge Independent Network Construction (KINC) toolkit was developed by the Ficklin and Feltus Labs at Washington State University and Clemson University respectively. KINC is C++, Phyton and R suite of tools used to construct Condition-Specific Gene Co-expression Networks (csGCNs). However, it can also be used for top-down metabolic network construction as well. One of the tools that it provides is a script named `kinc-net2graphml.py` Python script which can be used for exporting a KINC network into GraphML format compatible with the Tripal Network module.

LOADING NETWORK DATA

3.1 Importing Functional Data

This step is optional and only required if you do not already have functional annotation data for the nodes (i.e., genes, mRNA, proteins) of your network. Once the tab-delimited file of functional data is created as per the instructions in the *Prepare Network Data* section, you can import it using the **Chado Bulk Loader**. The Tripal Network module provides a template that supports the tab-delimited format described in the previous section. To use it, navigate to **Tripal >> Data Loaders >> Chado Bulk Loader**. You should see the bulk loader page:

The screenshot shows the 'Chado Bulk Loader' web interface. At the top, there's a dark blue header with the title 'Chado Bulk Loader' and a navigation menu with tabs: 'JOBS', 'TEMPLATES', 'CONFIGURE', and 'HELP'. Below the header, a breadcrumb trail reads 'Home » Administration » Tripal » Chado Data Loaders » Bulk Loader'. A link '+ Add Bulk Loading Job' is visible. The main content area contains three input fields: 'Template Name Contains', 'Job Status', and 'Job Name Contains', each with a corresponding text box and a small icon. An 'Apply' button is positioned to the right of these fields. Below the input area, a message states 'No Tripal Bulk Loading Jobs have been created.' At the bottom of the interface, there are links for 'Comments' and 'Add content'.

Click the **Add Bulk Loading Job** link. The following page appears.

Create Bulk Loading Job

[Home](#) » [Add content](#)

BASIC DETAILS

Job Name *

Template *

Tripal Network: Functional Annotations Importer ▼

Please specify a template for this loader

Data File

Please specify the data file to be loaded. This must be a tab-delimited text file with UNIX line endings.

File has a Header

☒ Yes
 ☐ No

Keep track of inserted record IDs

☐ Yes
 ☒ No

This enables the ability to revert an entire loading job even if it completed successfully. Furthermore, it displays the number of records successfully inserted into each table.

Menu settings

☐ Provide a menu link

Revision Information

Be sure to give the job a name that makes sense to you, select the Template named **Tripal Network: Functional Annotation Importer**, and then set the full path to your Data File. The Chado Bulk Loaders requires that the file exists on the web server and does not have an upload function. Once saved you will see the job submission page.

Bulk Loading Job: Import GO terms for rice genes

View

Edit

Manage display

Devel

Overview

Comments

Data Fields

Overview

Job Name	Import GO terms for rice genes
Submitted By	1
Job Creation Date	January 7, 2022, 11:50 am
Last Updated	January 7, 2022, 11:50 am
Template Name	Tripal Network: Functional Annotations Importer
Data File	/local/research/osativa_heat_drought/03-preprocessing/01-functional_data/oryza_sativa.MSU_v7_0.genes2GO.txt
Job Status	Initialized

Submit Job

Constant Values

New set of Constants

The following fields are constants in the selected template that you need to set values for.

Genus

Enter the case-sensitive value of this constant for your data file

Species

Enter the case-sensitive value of this constant for your data file

Feature Term (e.g. gene, mRNA, protein)

Enter the case-sensitive value of this constant for your data file

Add Constant Set

Here you can set the constant values that the template requires. These are the genus and species name to which the features in the file belong, and the feature type. If these annotations belong to genes then enter 'gene'. If they belong to transcripts, enter 'mRNA' or 'protein' for protein features. Click the **Add Constant Set** button when complete. You will then see the constant set you just added in a table:

▼ Constant Values

You have already added constants to this bulk loading job. Each row in the following table represents a set of constants. Each set will be used to load your data file with the specified template resulting in the each record in the template to be loaded x number of times where there are x sets of constants (rows in the following table).

Group	Organism		Feature Type CVTerm	Operations
	Genus	Species	Feature Term (e.g. gene, mRNA, protein)	
1	Oryza	sativa	gene	Edit Delete

Now, you can click the **Submit Job** button to import the data. The job will be executed in the way your Tripal site is setup for job execution.

If you run the job manually on the command-line you will see output similar to the following as the job executes:

```

Tripal Job Launcher
Running as user 'administrator'
-----
2022-01-07 11:58:10: There are 1 jobs queued.
2022-01-07 11:58:10: Job ID 936.
2022-01-07 11:58:10: Calling: tripal_bulk_loader_load_data(8)
Template: Tripal Network: Functional Annotations Importer (10)
File: /local/research/osativa_heat_drought/03-preprocessing/01-functional_data/oryza_
↳sativa.MSU_v7_0.genes2G0.txt (38000 lines with data)
Preparing to load...
Loading...
Constants:
  - organism.genus = Oryza
  - organism.species = sativa
  - cvterm.name = gene
Preparing to load the current constant set...
  Open File...
  Start Transaction...
  Defer Constraints...
  Acquiring Table Locks...
    ROW EXCLUSIVE for db
    ROW EXCLUSIVE for cv
    ROW EXCLUSIVE for dbxref
    ROW EXCLUSIVE for cvterm
    ROW EXCLUSIVE for feature_cvterm
  Loading the current constant set...
Progress:
[|||||||-----] 20.89%. (7938 of 38000) Memory:↳
↳39418608

```

3.2 Importing a Network File

3.2.1 Step 1: Create a Network

At this point we should have the “node” data for the network loaded (i.e., genes, mRNA, proteins) as well as any functional annotations. Before importing the GraphML file we must first create the network page that describes how the network was created. To do this navigate to **Add Tripal Content** and click the **Network** content type that appears at the bottom in the **Other** section. The following page appears:

Here you will provide a unique identifier for the network, a name, specify the organisms and specify the type. Examples of network types include:

- Context-specific gene co-expression network (csGCN)
- Gene co-expression network. (GCN)
- Gene regulatory network prediction (GRN)
- Metabolite correlation network. (MCN)
- Protein interaction network prediction (PPI)
- Weighted gene co-expression network analysis (WGCNA)

Note: Chado typically expects a unique computer readable identifier for an entity. There are no rules what those identifiers should be but often the identifier is a computer readable unique name and the name is a human readable

name.

The example screenshot below adds a gene co-expression network created using WGCNA of a [rice dataset](#) exposed to [heat and drought](#). For this example the network page appears as:

Oryza sativa GEO:GSE74793 WGCNA Co-expression Network

[View](#) [Edit](#) [Reload](#)

[Annotations](#)
[Publication](#)
[Summary](#)

Summary	
Resource Type	Network
Identifier	Oryza sativa GEO:GSE74793 WGCNA Co-expression Network
Name	Oryza sativa GEO:GSE74793 WGCNA Co-expression Network
Organism	Oryza sativa
Network Type	Weighted gene co-expression network analysis
Attribute	<p>A network may have one or more attributes that are used to describe it and can be assigned to the graph, its nodes and/or edges. Use the drop-down below to explore the attributes in this network.</p> <div> -- Select -- </div> <p>Select an attribute of this network for more details about it.</p>

Currently, the **Attribute** row in the **Summary** table has an empty drop-down. We have not yet loaded the network nodes and edges and until then there will be no attributes.

3.2.2 Step 2: Create an Analysis Record

To help ensure provenance most data that is imported into Tripal is associated with an Analysis record. It is on this record that we will provide the details about how the network was constructed. To create an analysis record navigate to **Add Tripal Content** and click **Analysis**.

The following example screenshot describes how the network was created.

Oryza sativa GEO:GSE74793 WGCNA Co-expression Network

[View](#)[Edit](#)[Reload](#)

[Summary](#)
[Annotations](#)
[Publication](#)
[Relationship](#)

Summary

Resource Type	Analysis
Name	Oryza sativa GEO:GSE74793 WGCNA Co-expression Network
Description	<p>Samples Selection</p> <p>RNA-seq data used to construct the network was selected from the NCBI Project, PRJNA301554. This included 475 samples. The SRA IDs for each sample was collected.</p> <p>Data Source Citation:</p> <p>Wilkins O, Hafemeister C, Plessis A, Holloway-Phillips MM, Pham GM, Nicotra AB, Gregorio GB, Jagadish SV, Septiningsih EM, Bonneau R, Purugganan M. EGRINs (Environmental Gene Regulatory Influence Networks) in Rice That Function in the Response to Water Deficit, High Temperature, and Agricultural Environments.. The Plant cell. 2016 10; 28(10):2365-2384.</p> <p>Gene Expression Quantification</p> <p>The nf-core compatible Nextflow workflow GEMmaker was used to quantify gene expression. All 475 SRA IDs were provided. GEMmaker automatically retrieved the samples and used Kallisto to quantify expression. Results from all samples were combined into a single gene expression matrix (GEM) using TMM normalized values. Reads were aligned to the MSU v7.0 <i>Oryza sativa</i> genome reference.</p>

	Network Construction The WGCNA R package was used to construct the network with the following parameter settings: <ul style="list-style-type: none"> Correlation method: Pearson's Cut height: determined automatically Power: determined automatically Minimum cluster size: 30 Minimum missing values allowed per gene: 0.2 percent Block Size: 5000 Network hard threshold: 0.269
Program, Pipeline, Workflow or Method Name	WGCNA
Program Version	1.69
Algorithm	
Date Performed	Friday, January 1, 2021 - 18:02
Data Source	Source Name: NCBI SRA Source Version: NCBI Project PRJNA301554 Source URI: https://www.ncbi.nlm.nih.gov/bioproject/PRJNA301554/

3.2.3 Step 3: Import the GraphML

3.2.3.1 Loader Stage 1

After creation of the Network page, the GraphML file can be imported at **Tripal >> Data Loaders >> Chado GraphML Loader**. This loader uses a multi-stage form. In the first stage you can upload the GraphML file, set the organism to which the network belongs,

Chado GraphML Loader

Home » Administration » Tripal » Data Loaders

GRAPHML FILE

Please provide the GraphML file

Existing Files

--Select a file--

You may select a file that is already uploaded.

File Upload

FILE	SIZE	UPLOAD PROGRESS	ACTION
network_edges.graphml	23.8 MB	Complete	Remove

Remember to click the "Upload" button below to send your file to the server. This interface is capable of uploading very large files. If you are disconnected you can return, reload the file and it will resume where it left off. Once the file is uploaded the "Upload Progress" will indicate "Complete". If the file is already present on the server then the status will quickly update to "Complete".

Upload File

Server path

If the file is local to the Tripal server please provide the full path here.

Organism *

Oryza sativa

Choose the organism to which this network belongs.

Analysis *

Oryza sativa GEO:GSE74793 WGCNA Co-expression Network

Choose the analysis to which the uploaded data will be associated. Why specify an analysis for a data load? All data comes from some place, even if downloaded from a website. By specifying analysis details for all data imports it provides provenance and helps end user to reproduce the data set if needed.

3.2.3.2 Loader Stage 2

The second stage of the loader provides a form to associate controlled vocabulary (CV) terms to each graph, node and edge attribute. Tripal strives to provide data in a **FAIR** manner. This requires that all attributes be distinguishable using CV terms. The loader will extract each attribute from the GraphML file and provide a form that you can use to specific a vocabulary term for each attribute.

Instructions

All attributes, that you want loaded, for the graph, nodes and edges in the GraphML file must be associated with a controlled vocabulary term. Use the form below to make the association with attributes that were found in the file provided. If an appropriate term is not present you can add one [here](#). You can find terms at the [EBI Ontology Lookup Service](#). If you cannot find an appropriate term, you can add one to the "local" vocabulary. However, best practice is to use existing terms if possible. If you leave any attribute without an association, then the property will be excluded and will not be visible to end-users.

Ranking Edges in the Network

The network viewer works best with at least one term that end-users can use for ranking edges. The following are examples of possible terms that could be used for ranking:

- Similarity Score (SWO:0000425)
- Importance Weight (NCIT:C48192)
- Rank (NCIT:C48904)
- WGCNA edge weight (local:WGCNA_edge_weight)

► **NODE V_NAME**

► **EDGE E_INTERACTION**

► **EDGE E_WGCNA_WEIGHT**

Next Step

For networks created using [KINC](#) or which use the example code in the [Prepare Network Data](#) section to export a WGCNA network to GraphML then attributes will be recognized automatically and assigned controlled vocabulary terms. For example, in the screenshot below the term for the **EDGE E_WGCNA_Weight** attribute has already been set by the loader:

EDGE E_WGCNA_WEIGHT

Type

WGCNA edge weight

Enter the name of the term that specifies the type. The type must be the name of a term in a controlled vocabulary and the controlled vocabulary should already be loaded into this site.

Lookup Term

MATCHING TERMS

Please select the best matching term. If the same term exists in multiple vocabularies you will see more than one option below.

☒ WGCNA edge weight

Vocabulary: local (local) Locally created terms.
Term ID: local:WGCNA_edge_weight.
Definition: The edge weight from a WGCNA network is the distance (e.g., euclidean) between the vector of topological overlap measures of each node with every other node.

Numerical

When viewing network relationships, the end-user can perform layering and filtering of nodes and edges using the attributes assigned to them. To support filtering, we need to know the type of data for each attribute because elements with numerical or categorical attributes can be filtered differently. The GraphML loader will try to guess the data type, however, you can specify the value here to prevent the loader from guessing incorrectly or to refine the type. For example, the loader does not know if a numerical attribute is also ordinal or if a text value is categorical. Text attributes that are not specified as categorical will not be filterable.

Any attributes without an assigned controlled vocabulary term will not be imported with the network.

3.2.3.3 Loader Stage 3

The final stage is naming the graph. Here you provide a human readable name and a unique name. The two can be the same but no two networks can have the same unique name.

Graph Name *

Oryza sativa GEO:GSE74793 WGCNA Co-expression Network

The human readable name for this graph. This name was extracted from the GraphML file provided.

Graph Unique Name *

Oryza sativa GEO:GSE74793 WGCNA Co-expression Network

A unique name for this graph. It need not be human readable, but it must be unique from any other existing graph. This name was extracted from the GraphML file provided.

Previous Step

Submit

After clicking the **Submit** button, the job for importing the file will be submitted to Tripal. The job will be executed in the way your Tripal site is [setup for job execution](#).

VISUALIZATION

There are two ways to visualize network connectivity:

- The neighborhood viewer on a feature page
- The full browser 3D Viewer application

4.1 Feature Neighborhood Viewer

The Tripal Network module provides a field that can be added to feature pages (e.g. genes, mRNA, protein pages). You must first enable the viewer for each feature type on which you want it to appear. To do this navigate to the **Structure >> Tripal Content Types** and select the content type of the nodes in your network. If the nodes are genes, then click the **manage fields** link for the **Gene** content type. When the page appears, click the **Check for new fields** link at the top of the page:

Gene
EDIT
MANAGE FIELDS
MANAGE DISPLAY

+ Check for new fields

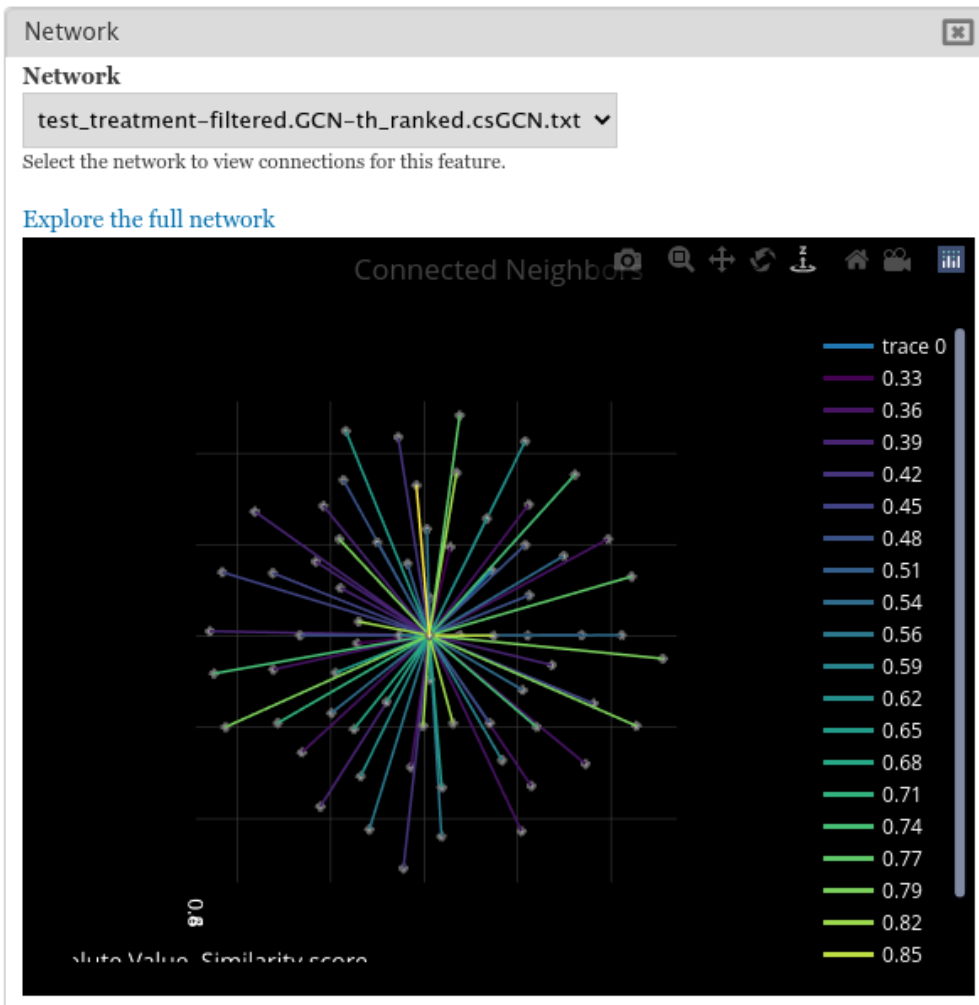
Show row weights

LABEL	MACHINE NAME	FIELD TYPE	WIDGET	OPERATION
+ Resource Type	rdfs__type	Content Type	Content Type	edit delete
+ URL redirects	redirect	Redirect module form elements		
+ Accession	data__accession	Site Accession	Site Accession	edit delete

Next, click the **Manage Display** tab to place the field where you would like it to appear on the page. See the [Configuring Page Layout](#) documentation in the Tripal User's Guide for instructions on organizing fields on a page.

Once the field has been organized and placed, you can then open any feature page that serves as a node in your network and see an interactive 3D viewer of the neighborhood of connections for that feature.

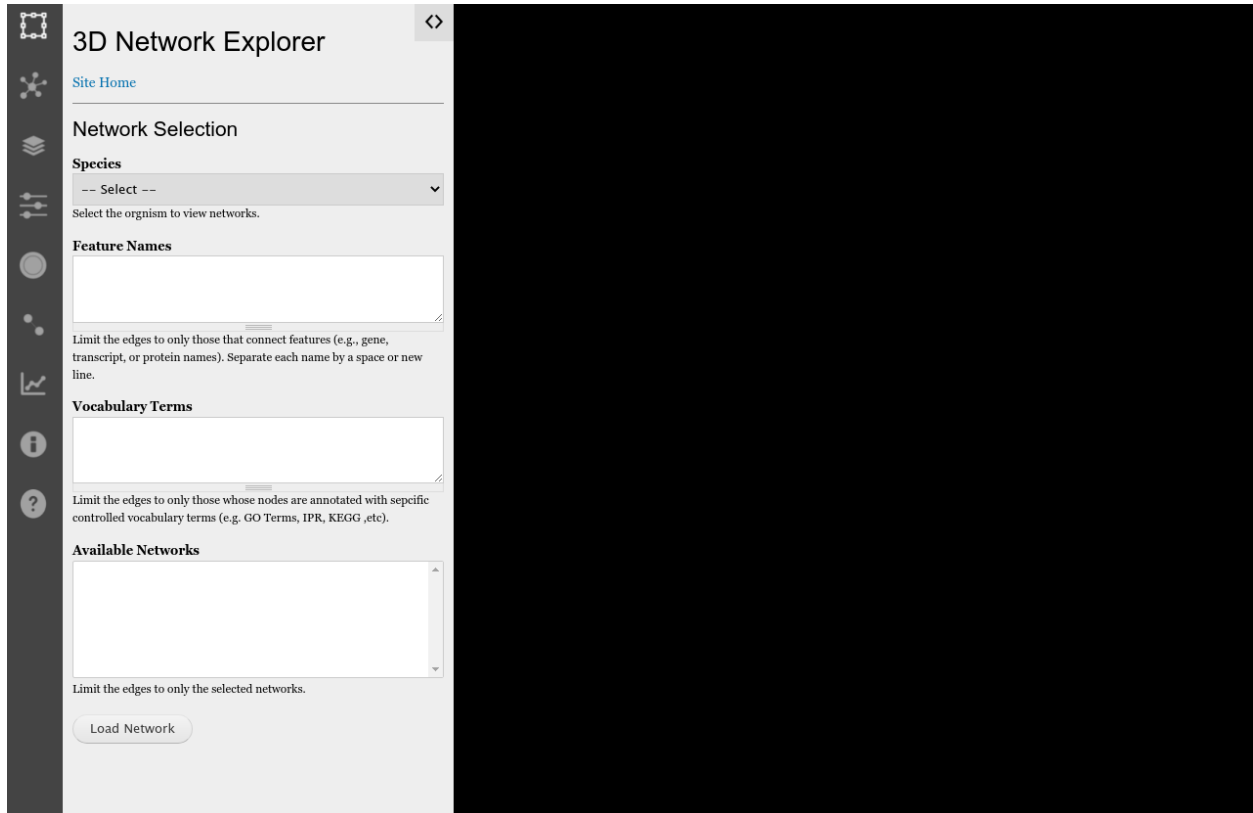
LOC_Os04g28420

[View](#) [Edit](#) [Reload](#)[Annotations](#)
[Publication](#)
[Sequences](#)
[Summary](#)
[Synonyms](#)
[Transcripts](#)
[Network](#)

Site visitors will be able to view the connections for each network that is loaded. The features are limited for this view, but a link titled **Explore the full network** is available that, when clicked, will take the user to the full browser 3D Network Exploration App.

4.2 The 3D Network Exploration App

A Tripal site that has the Tripal Network module installed will have available the 3D Network Explorer app. It is available at the relative URL `/networks/viewer` of your website. When the user first arrives the 3D viewer has a left-sidebar with controls and a blank exploration window to the right:



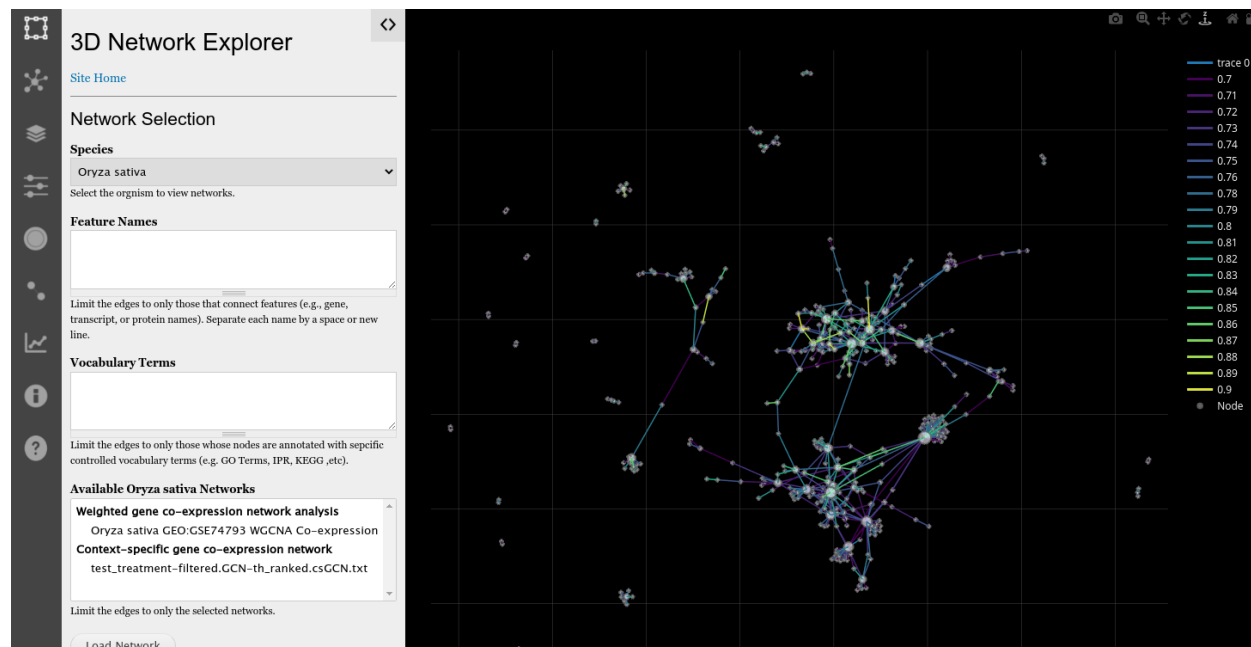
The icon bar on the far left provides the following functionality in order of appearance from top to bottom:

- Network Selection
- Network Details
- 3D Layers
- Property Filters
- Node Details
- Edge Details
- Analyses
- Information
- Help

4.2.1 Network Selection

By default, the network selection sidebar is open when the 3D viewer first appears. Here the user selects a network by selecting the species and clicking the **Load Network** button. Only species that have imported networks will be available in the list. The user can limit the network to a specific set of features (nodes) or controlled vocabulary terms such as Gene Ontology terms that are assigned to the nodes.

By default the network loaded will only include 500 of the top edges. This number can be changed in the property filters sidebar.



4.2.2 Interactive Network Exploration

The exploration window provides a control panel in the top right corner:



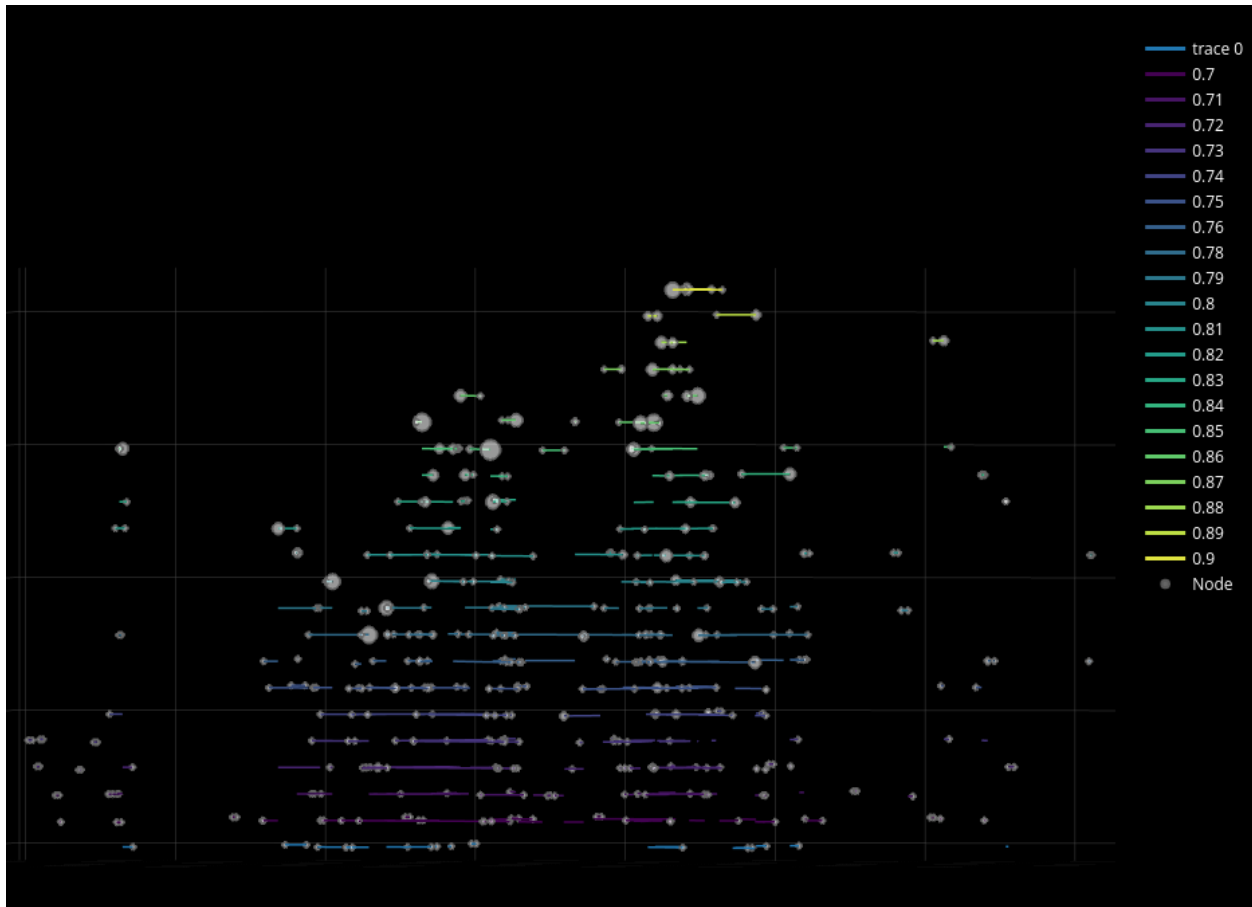
The control panel supports the following in order of the icons that appear:

- Download screenshot of the network
- Zoom in or out
- Pan right, left up or down
- Orbital rotate
- Turntable rotation
- Reset camera

Zooming, panning and rotating are also available with the mouse:

- zooming: mouse wheel
- rotation: left-click and drag
- panning: right-click and drag

By rotating, the end user can explore in the Z-axis (3rd dimension) a stacked layout of edges for an attribute. The following screenshot shows the network stacked by correlation values, where those colored more brightly and near the top are more highly correlated.



Users can explore the attributes of nodes and edges by clicking on either. If a node is clicked then the Node Details sidebar is opened. If an edge is clicked then the Edge Details sidebar is opened.