alodar systems™

Solutions for Science R&D

WHITE PAPER

Genomic Pathway Storage and Documentation in a Graph-like Structure

Introduction

Biochemical pathway information is currently fragmentary. Argonne's WIT (What Is There) FAQ [1] lists metabolic reconstructions for 25 genomes, based on 2900 diagrams. KEGG (Kyoto Encyclopedia of Genes and Genomes) [2] has over 10,000 pathways, 132 organisms, and nearly 500,000 genes. They have, though, only 5,500 chemical reactions fully mapped out for those 10,000 pathways. The KEGG links page has separate destinations for protein-protein interactions, metabolic pathways, regulatory pathways, and reaction compounds, each with roughly a dozen links. While it is true that many projects have an entry in each category, there is relatively little information cross linked between these extensive resources. Thus, while these projects have accomplished a great deal of work, they have a long way to go before the gaps are filled in.

The staggering number of genes, sequences, and resulting enzymes produced by the National Human Genome Project will exacerbate this problem. Analysis of the results will add sequence and enzyme information to these databases without adding a corresponding amount of information on the role of those enzymes in human metabolism.

This white paper explores the use of advanced modeling concepts to cope with the challenges posed to scientists by the staggering size yet fragmentary nature of biochemical pathway information.

Modeling using MetaGraph

Since 1997 there has been a collaborative, open-source effort under way to develop a new kind of data storage representation for biological information, called MetaGraph [3,4]. MetaGraph is a kind of semantic network, and organizes data using a graph of nodes representing units of information, connected by edges representing relationships among them. Over eighty node types have been defined, and these include biological concepts, laboratory entities, evidence, and organizational groupings. Twelve edge types describe ways the nodes relate to one another, for example describing, composing, or succeeding. Each type of node and edge is depicted graphically with a distinct abbreviation, color, and style.

The MetaGraph system is designed to accommodate and embrace this inevitable drift in requirements, by storing only the information that is available and at the most appropriate level of detail. As understanding increases, representations become more detailed, but early information is still available. An investigator can add additional information to early data without needing to encode it again, and can make use of what information is there as more is discovered about it.

This approach is possible because MetaGraph entities are small and much of the critical knowledge is bound into the relationships between these simple entities. The relationships are fluid, and the entities may be free reorganized to accommodate new understandings or hypotheses. Relationships may also be created with information external to MetaGraph, freeing the scientist to choose whether to link directly to the external source or to construct and import the information into a MetaGraph representation.

Finally, MetaGraph was designed with hypothesis testing in mind. It is able to represent divergent and contradictory hypotheses in the database at the same time. Scientists annotate their hypotheses with the experimental data that validates it, and in so doing, store the needed information for future use.

Whereas MetaGraph holds the promise of helping to address the problems of evolving and incomplete information, it needs more work. Just as a schema must be defined before a database can be populated, so must a data representation be developed for a MetaGraph model.

A MetaGraph Schema for Reaction Pathways

Nodes and Edges

A MetaGraph graph is a reflection of our understanding of a biological concept, and talking about the graph is similar to describing the concept. Figure 1 is a simple graph describing the expression of a human protein.

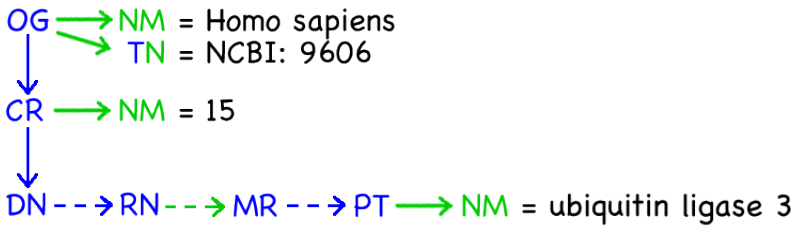

Figure 1: Protein expression

To verbally describe this expression using the node-edge nomenclature, one would say it begins with an Organism, which is DescribedBy Name Homo Sapiens, and a unique TaxNode 9606, in the NCBI classification. The Organism HasA Chromosome, DescribedBy Name 15, which is a TemplateFor RNA that TransformsInto mRNA which in turn is a TemplateFor the Protein DescribedBy Name ubiquitin ligase 3.

This graph has been simplified for purposes of illustration, but it also illustrates an important feature of MetaGraph, which is that only the information that is known about an object is included. A more realistic graph describing protein expression would be annotated with additional information such as the gene location on the DNA strand, and would have links to publications or experiments

documenting the discovery. Some genes and proteins have been more fully explored and documented than others, and would be represented with correspondingly richer graphs. The same graphical flexibility allows MetaGraph to capture changes over time, where the graph evolves as more becomes known about a concept. Extending this principle, different but related graphs can describe alternate hypotheses about a concept, each annotated with a level of confidence.

This flexibility can be used to advantage to represent reaction pathways. The same MetaGraph schema encompasses reactions that have been extensively characterized and those that are merely hypothesized. The same MetaGraph schema holds multiple versions of the reaction pathway data.

Node Clusters

MetaGraph's flexibility gives it extraordinary descriptive power, but in the complete absence of constraints makes it very difficult to compare similar concepts. Constraints can be introduced by defining clusters, or subgraphs of nodes and edges, which represent templates or canonical organizations to which a particular concept should conform.

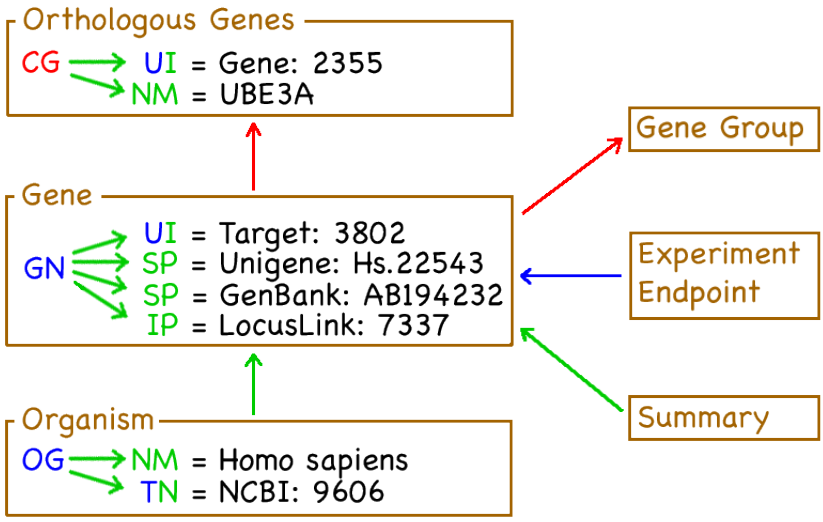

Figure 2: Gene and associated clusters

Six clusters are shown in Figure 2. The central Gene cluster is a graph with a Gene node described by several identifying parameters. Tightly connected to a Gene cluster are the Organism which contains it and the set of Orthologous Genes (COG) of which it is a member. Each is a graph with an appropriately typed central node, described by a name and a unique identifier. Loosely connected to the Gene are three related clusters, a more general Group of related Genes, an Experiment of which it is an Endpoint, and a Summary that references it. (The detailed graphs for these clusters are not depicted here.) A flexible language, called Intercalate, has been defined to describe the relationships between clusters as well as the organization and multiplicity of the nodes and edges within them [7].

Queries

A MetaGraph database is searched using a specially designed query language. There are two components to a query: the values of the attributes and the graphical organization. The first component, like a traditional database query, supplies values or wildcard patterns to be matched by attribute fields in the nodes. The second component is similar to a chemical substructure search, and succeeds by finding instances of subgraphs with types and connection geometry matching those of the query. In addition to exact structure matches, the graphical component also supports queries for unspecified node types and alternative connection geometries.

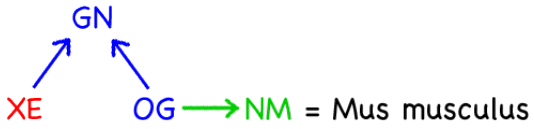

Figure 3: Query for mouse genes that are the endpoint of an experiment

The simple query illustrated in Figure 3 specifies we are looking for mouse genes that are the endpoint of an experiment. There are four nodes in the query, connected in a simple linear chain. The relationship structure to be matched is one where a Gene is part of an Experiment Endpoint, and at the same time part of an Organism described by a Name. The attribute constraint is that the value of the Name must be Mus musculus. The query returns a collection of Gene nodes satisfying those relationship and attribute constraints.

MetaGraph queries manage structural relationships more cleanly and generally than SQL used in relational databases or Xquery used for XML data. SQL captures structural relationships between tables through the use of joins and foreign keys, but doesn't support filtering on the type and attributes of the relationship. Xquery relationships are artificially divided into those representing the element hierarchy and arbitrary id/idref connections. In MetaGraph, structural relationships and attributes are a direct reflection of the data organization. It is natural to extend simple queries by appending more nodes to bring additional focus to the results.

Reaction Pathways

Biological reactions are complex processes, but at the level of a block diagram they can be modeled as catalyzed transformations of one or more reactants into one or more products. The catalyst in a biological reaction is an enzyme. The use of the terms reactant and product suggests a temporal order, however under suitable conditions a *reversible* reaction may transform its products into reactants.

A reaction pathway is a grouping of reactions where the products of one reaction become the reactants of the next in a temporal sequence. Which reactions are part of a given pathway, and the place one pathway ends and another begins, are generally determined by convention to reflect the activity of the products in higher-level cellular function. Biological reaction pathways are frequently nonlinear, with loops and branches and cross-links to other pathways.

From the beginning the design of MetaGraph has included nodes and edges designed to represent reactions, although the initial applications developed in its framework have not included this kind of data. MetaGraph provides the ability to model, in a single framework, both the inferred existence of a reaction, as well as details of its operation as they are discovered experimentally.

MetaGraph explicitly represents the connections between reactions, pathways, and components, and encourages queries about the organization that may be more difficult to pose it a relational framework. Scientific questions often have direct translations to MetaGraph queries. For example:

- What compounds are products of the most reactions? These may prove poor drug targets because production is too difficult to inhibit.

- What compounds are reactants of the most reactions? These may cause too many side effects if inhibited.

- Which reaction pathways have longer non-branching sequences? These promise flexibility in the choice of location for inhibition.

- What reactions share no reactants or products with other reactions? These need more study because they may be miscategorized as part of a pathway, or a result of data entry error, or simply have not been experimentally verified.

To pose such questions, we need to define nodes edges, and clusters that can be used to construct MetaGraph representations of reaction pathways.

Nodes used in Reaction Pathways

To use MetaGraph to organize and present maps of metabolic pathways one first must construct the Metagraph schema. A MetaGraph schema is a semantic network and represents concepts embodied in the data. One builds a schema by examining the specific body of data and capturing its unique representational characteristics. There are several repositories of pathway information published and accessible on the Internet. For this white paper we chose to use data from the Kyoto Encyclopedia of Genes and Genomes (KEGG) [2].

KEGG identifies two unique components of reaction pathways, the compounds and the reactions. MetaGraph already contains a basic node type representing a pathway (abbreviated PY), but because this is a domain that hasn't been fully explored in the past, MetaGraph does not contain node types sufficiently specific to represent the components of a pathway. For this purpose we defined two new basic node types, the KEGG Entry (KE) and the reaction (RX).

We wish to represent and query the relationships and interconnections between the pathway components but avoid storing detailed information about enzymes, products, and reactions. Consequently, the clusters we define contain only the basic information needed to uniquely identify a component. This is augmented with a new type of string parameter (SP) containing an URL

reference back to the original KEGG data.

KEGG data also contains information about how best to display the pathway information. This display information has been captured in a Graphics (GX) node, but is not otherwise used for the purposes of this project.

Clusters used in Reaction Pathways

Each unique concept in KEGG is represented by a corresponding cluster, consisting of the appropriate root node attached to additional descriptive nodes. We have defined three cluster types, Entry, Reaction, and Pathway. An Entry cluster holds a generic entity specified by the KEGG ontology. A Reaction cluster represents a single reaction, that will usually be connected to substrate, product, and catalyzing KEGG entities. A Pathway cluster is a collection of metabolic paths from reactants to products, each path competing with other reactions using similar chemicals in metabolic function.

Figure 4: Entry cluster

Figure 4 shows an Entry cluster consisting of a central KEGG Entry node, annotated with name, link, and number attributes. The name is the KEGG identifier including the data type, the link is an URL referencing the complete entry definition in the KEGG database, and the number distinguishes this use of the component from other reactions where the same enzyme may be active.

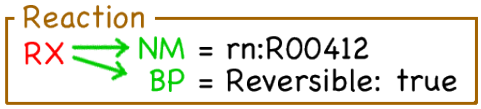

Figure 5: Reaction cluster

Figure 5 shows a Reaction cluster having a central Reaction node annotated with a name, and a flag indicating whether the reaction is reversible.

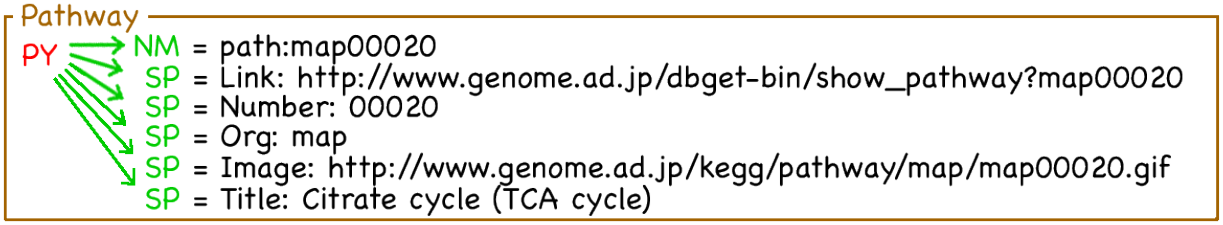

Figure 6: Pathway cluster

Figure 6 shows a Pathway cluster with a Pathways node connected to a number of attributes: a name, link, and number like an Entry, but also the origin of the pathway, a descriptive title, and an URL referencing an image of the pathway on the KEGG web site.

Relationships among Pathway Clusters

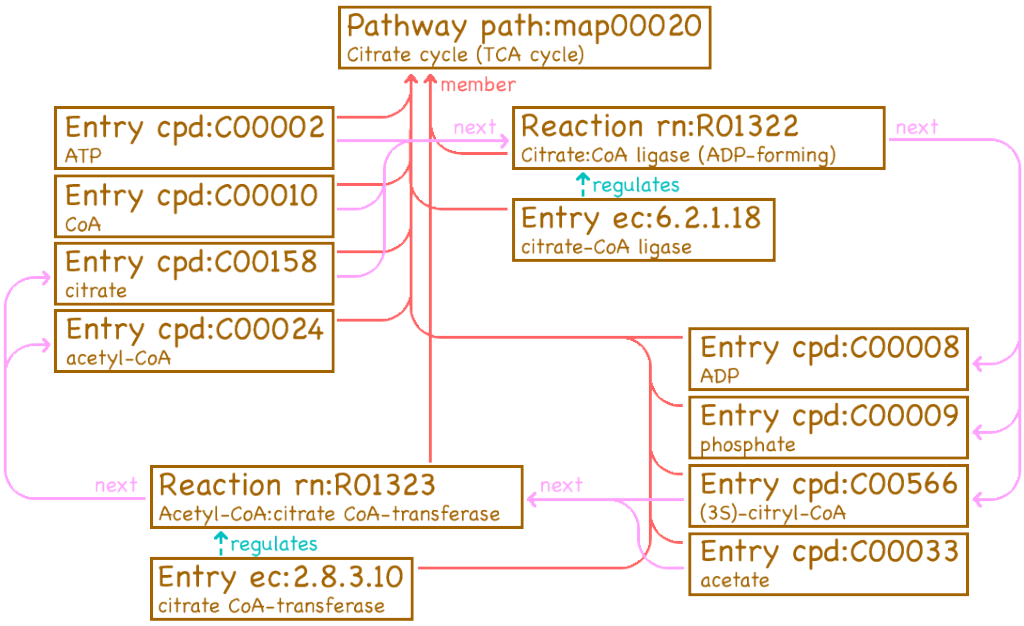

Figure 7: Relationships among clusters

Relationships among the clusters are defined to reflect the sequence of reaction events in a metabolic pathway. Figure 7 shows a portion of the Citrate or TCA cycle pathway, with two of its member reactions. Reaction R01322 is regulated by enzyme 6.2.1.18 and creates three product compounds from three substrates. Reaction R01323 is regulated by enzyme 2.8.3.10 and creates two product compounds from two substrates. The citrate and (3S)-citryl-CoA compounds are shared by both reactions. The two reactions given here happen to be reversible, so in this case the distinction between substrates and products is unimportant and the direction of those links can be ignored. Both reactions and all associated entries are linked as members of the pathway group. Additional relationships, not illustrated here, are constructed between each enzyme and precursors and successors in the pathway.

Remote References in MetaGraph

The concept of clusters in MetaGraph serves to abstract the details of data storage away from the information being stored. The clusters and the relationships between them represent data that is encoded and stored entirely within MetaGraph. Two mechanisms have been defined to abstract data residing outside MetaGraph:

- Cross Reference nodes contain all the information needed to reference data in a relational database.

- URL parameter nodes contain a descriptive reference to data identified by an URL.

Both types of nodes can be incorporated into a MetaGraph graph or cluster to identify and store the access information.

Cross Reference Nodes

Certain data is suited to a relational model, and already exists in that form. Rather than extending the MetaGraph schema and replicating that data in it, a better approach is to create a level of abstraction that unifies both relational and MetaGraph storage. We have defined and implemented a cluster-like encoding for relational data so that it is possible to navigate from nodes in MetaGraph to relational tables, and vice versa.

When expressing a relational table as a cluster, a central requirement is the ability to reference a primary key that is unique across all the data in the relational schema. To accomplish this, we have implemented a new kind of node that serves as a proxy to the row in the remote table. This type of node is called a CrossReference, abbreviated XR. A cross reference node is a kind of descriptor containing fields to represent the remote database name, table name, and a string representation of the primary key. The implementation also defines functions to retrieve the attribute names and values from fields in the referenced row of the remote table.

The remote database name is a key into a connection table maintained by the persistence layer used with MetaGraph. The connection table contains all the information needed to connect to the remote database, including driver, host, user, and password. The remote table name can optionally have a schema name prepended. The encoding of the remote primary key is designed to work with both simple and compound keys by organizing them into a custom XML format. For example, a row referenced by the unique value 99 in the field “primary_key_field” would be identified with the following values.

```
<?xml version="1.0" encoding="UTF-8"?>
 <ref>
  <key name="primary_key_field">99</key>
 </ref>
```

The “key” element is repeated, with appropriate changes to the name and value, for each component of a compound key.

URL Parameter Nodes

Some data exists in Internet-accessible form and can be uniquely identified by an URL, or uniform resource locator, a string starting with http:// and a host identifier, followed by location of the data on that host. An URL is stored in a StringParameter node, abbreviated SP. A string parameter is a kind of descriptor containing fields to hold the value of the parameter, in this case the URL, and a name identifying its use.

URL nodes appear in the definitions given above for the Entry and Pathway clusters. The Entry example in Figure 4 contains a parameter named Link, with the value of an URL referencing compound information on the KEGG web site. The Pathway example in Figure 6 contains similar Link and Image parameters.

5. Benefits realized from representing pathways in MetaGraph

With the MetaGraph schema for reaction pathways defined as describe above, one can then browse the database constructed according to that schema and pose queries on the data.

The MetaGraph Viewer

The MetaGraph Viewer shows the actual nodes and edges in a graph when the Node display mode is selected. The node at the center of the display is at the focus, and is shown surrounded by nodes to which it has edges. At the user's option, the display may be extended to show further layers of edges and their destination nodes, although interactive response may suffer if the depth is set to a large value. The attributes of the node at the focus of the display are displayed in a panel on the side. A new node may be moved to the focus by clicking on it with the mouse.

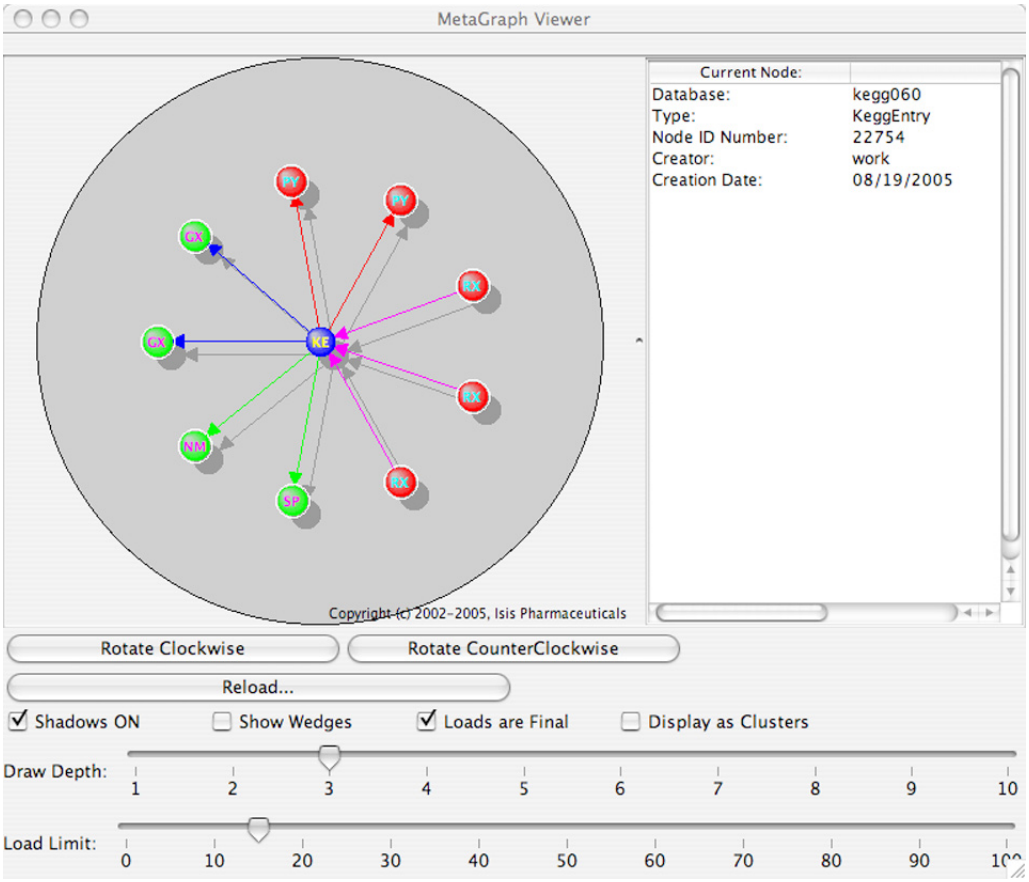

Figure 9: MetaGraph node viewer

Figure 9 shows a central KeggEntry node representing a compound that is the product of three different reactions and part of two different pathways. The compound is described by Name and URL parameter nodes. The attached Graphics nodes represent information about how best to display the compound in each pathway.

When the MetaGraph viewer is set to Cluster display mode, the circular nodes are replaced with squares symbolizing clusters of nodes. These are shown with edges to related clusters, as are the nodes. The panel on the side summarizes the information represented by the nodes in the focus cluster.

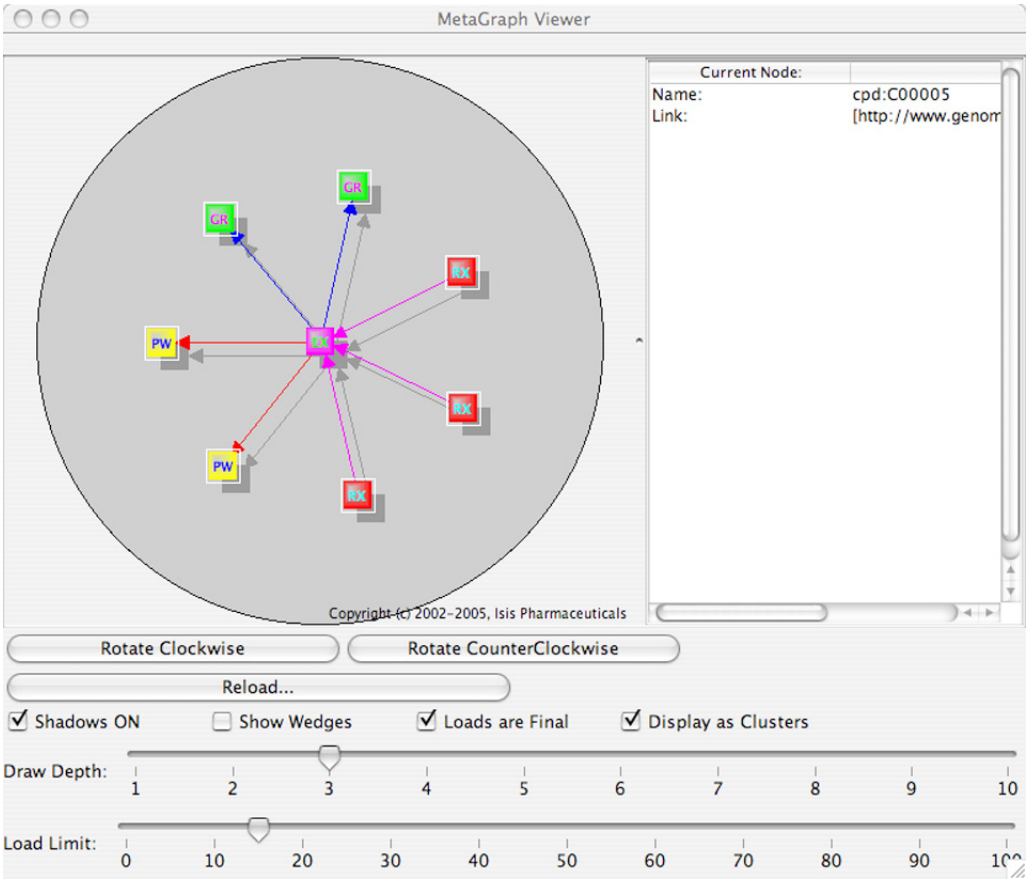

Figure 10: MetaGraph cluster viewer

Figure 10 shows the same compound in cluster format. The KeggEntry cluster, abbreviated EX, is still central, but its Name and URL descriptors have been collapsed under the root KeggEntry node. Their values are displayed in the detail panel on the right. The reactions, pathways, and graphics associated with the compound have been similarly reduced to clusters.

The MetaGraph viewer is an interactive tool for exploring the graphical structure of the data. Clicking on any of the displayed clusters causes it to move to the center, its details displayed to the right, and all its linked information drawn in an array around it. The buttons and sliders at the bottom of the screen control the characteristics and complexity of the display, as well as the amount of data preloading done to improve interactive performance.

Query results

When made explicit in MetaGraph, the relationships between compounds, reactions, and pathways provide a basis for discovering and expressing structural similarities and commonalities in metabolic pathways that may not be immediately apparent from the raw data. These relationships enable interesting scientific questions to be posed and answered. Here are answers to some of the example queries presented above..

Q: What compounds are products of the most reactions? These may prove poor drug targets because production is too difficult to inhibit.

Compound	no. of reactions producing
Pyruvate	41
Ammonia	36
Glyoxylate	16
Acetyl coenzyme A	15
Oxaloacetate	15
Carbon dioxide	15
L-Glutamate	15
Acetic acid	14
D-Fructose	14
Formic acid	14

Q: What compounds are reactants of the most reactions? These may cause too many side effects if inhibited.

Compound	no. of reactions consuming
Acetyl coenzyme A	29
Glycine	17
L-Aspartate	15
Glutathione	15
L-Glutamate	14
Pyruvate	13
L-Tyrosine	13
UDPglucose	12
L-Arginine	12
Cytidine	12

Q: Which reaction pathways have longer non-branching sequences? These promise flexibility in the choice of location for inhibition.

This question is still under investigation. The queries take a long time to complete due to the large number of reaction linkages.

Q: What reactions share no reactants, products, or enzyme catalysts with other reactions? These need more study because they may be miscategorized as part of a pathway, or a result of data entry error, or simply have not been experimentally verified.

Reaction	Name
R00019	Reduced ferredoxin:H+ oxidoreductase
R00074	2-Aminophenol:oxygen oxidoreductase
R00749	Ethanolamine ammonia-lyase
R00846	sn-Glycerol-3-phosphate:oxygen 2-oxidoreductase
R00848	sn-Glycerol-3-phosphate:(acceptor) 2-oxidoreductase
R00855	CDPglycerol phosphoglycerohydrolase
R00856	CTP:sn-glycerol-3-phosphate cytidylyltransferase
R00932	2-Hydroxyglutarate glyoxylate-lyase (CoA-propanoylating)
R01013	Acyl-CoA:glycerone-phosphate O-acyltransferase

References

[1] WIT (What Is There): http://wit.mcs.anl.gov/WIT2/

[2] KEGG: http://www.genome.ad.jp/kegg/kegg.html

[3] Metagraph: http://www.metagraph.org/

[4] John McNeil, Alan Goates. "MetaGraph Framework: A Knowledge Storage Platform That Breaks Out of the Relational Paradigm," Proceedings of the O'Reilly Bioinformatics Technology Conference, Feb. 3, 2003. http://conferences.oreillynet.com/cs/bio2003/view/e_sess/3437

[5] Trinity College: http://www.tcd.ie/Biochemistry/IUBMB-Nicholson/

[6] Access Excellence: http://www.accessexcellence.org/AB/GG/citric_Cycle_a.html

[7] Intercalate: http://alodar.com/intercalate/