Analysis and Visualization of Biological Networks with Cytoscape

John "Scooter" Morris, Ph.D., UCSF (scooter@cgl.ucsf.edu)
Allan Kuchinsky, Agilent (allan_kuchinsky@agilent.com)
Alex Pico, Ph.D., Gladstone Institutes (apico@gladstone.ucsf.edu)

Table of Contents

Overview

Networks have long been used to represent important biological processes. Many of us remember memorizing the Krebs (TCA) cycle, which is usually shown as a directed graph, itself a type of network (Figure 1). Recently, however, the use of networks in biology has changed from purely illustrative and didactic to more analytic, even including hypothesis formulation. This shift has resulted, in part, from the confluence of advances in computation, informatics, and high-throughput techniques in systems biology. Today the analysis and visualization of biologically relevant networks has become commonplace, whether the networks represent metabolic, regulatory, or signaling pathways; protein-protein or genetic interactions; or more abstract connections between similar proteins or similar ligands. Networks are now routinely used to show relationships between biologically relevant molecules, and analysis of those networks is proving valuable for helping us understand those relationships and formulate hypotheses about biological function.

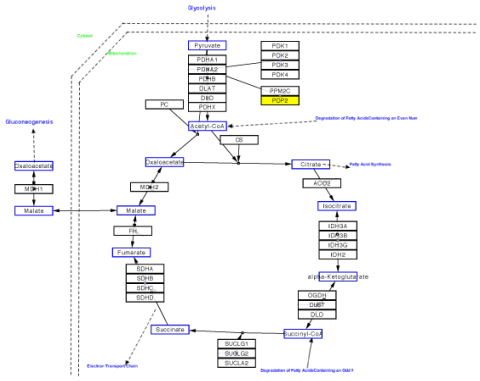


Figure 1. The TCA cycle from wikipathways

With the advent of high-throughput methods that generate vast amounts of data from diverse measurement sources – for example gene expression data from microarrays, protein or metabolite abundance from mass spectrometry – biological networks have become increasingly important as an integrating context for data. As a commonly understood diagrammatic representation for concepts and relationships, networks provide structure that helps reduce underlying complexity of the data. Network tools give us functionality for studying complex processes. We can analyze global characteristics of the data, via metrics such as degree, clustering coefficient, shortest paths, centrality, density. We can identify key elements (hubs) and 'interesting' subnets, which can help us to elucidate mechanisms of interaction. Also, visualization of data superimposed upon the network can help us understand how a process is modulated or attenuated by a stimulus.

Network tools have proven to be extremely useful in analyzing and visualizing important biological processes. Some general applications of networks in biology include:

- **Gene Function Prediction** – Examining genes (proteins) in a network context shows connections to sets of genes/proteins involved in same biological process that are likely to function in that process [1-4].

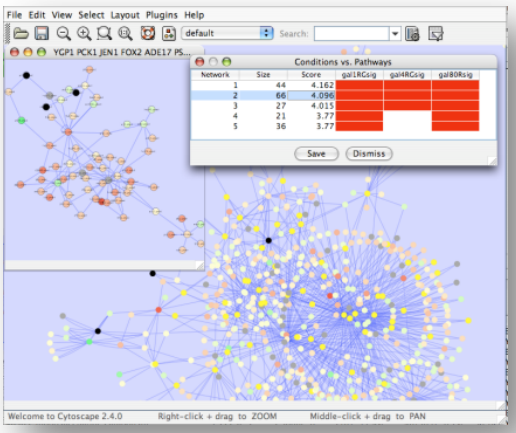


Figure 2. Gene Function Prediction using jActiveModules

- **Detection of protein complexes/other modular structures** – although interaction networks are based on pair-wise interactions, there is clear evidence for modularity & higher order organization (motifs, feedback loops) [5-9]

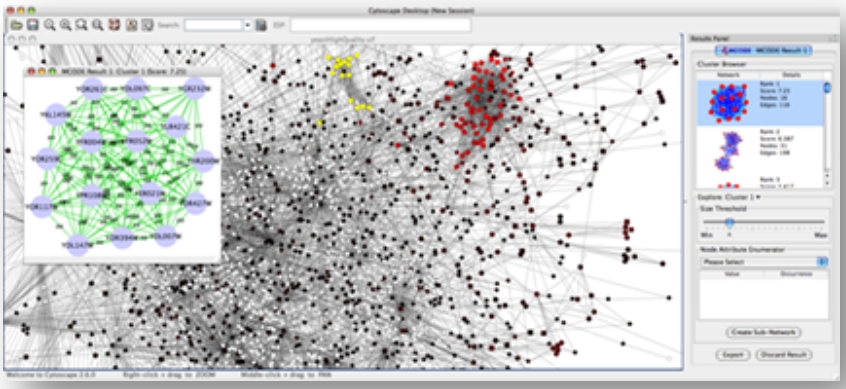


Figure 3. Identifying molecular complexes in large protein interaction networks using MCODE

- **Prediction of new interactions and functional associations** – There are several methods for predicting interactions and functional associations, based upon network structure and correlations amongst data. For example, orthology-based methods have been used to predict interactions for a species based upon orthology to interacting pairs of proteins in evolutionarily similar organisms[10]. Other researchers have used Bayesian network approaches to inferring gene regulatory networks from time course gene expression data[11]. In another approach, shown on the example below, statistically significant domain-domain correlations in protein interaction network suggest that certain domain (and domain pairs) mediate protein binding. Machine learning extends this to predict protein-protein or genetic interaction through integration of diverse types of evidence for interaction [12-14].

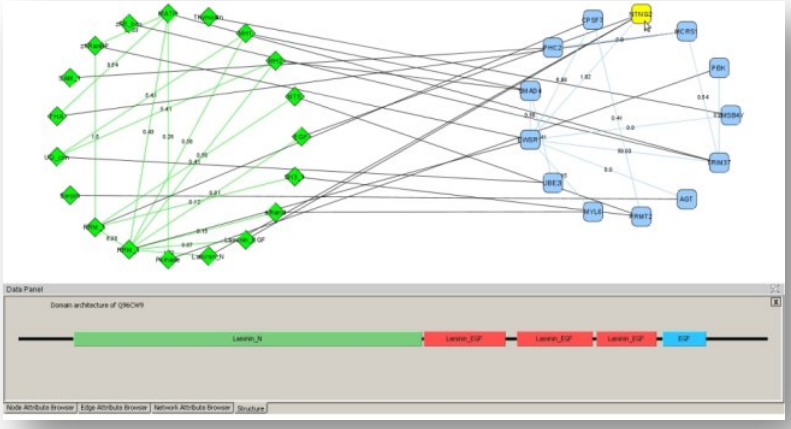


Figure 4. Visualiziing domain interactions and alternative splicing using DomainGraph

Moreover, these same tools and their associated analysis and visualization methods can provide key insights in the study of disease and in drug development. These include:

- **Identification of disease subnetworks** – identification of disease network subnetworks that are transcriptionally active in disease. These suggest key pathway components in disease progression and provide leads for further study and potential therapeutic targets [15-20].

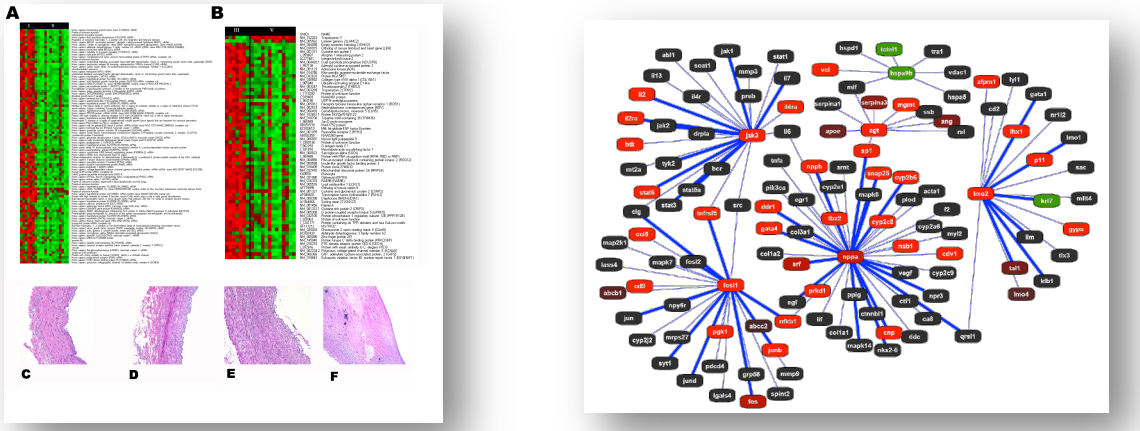


Figure 5. Gene expression profiles and American Heart Association (AHA) histological classification of atherosclerotic lesions (left panel). Differentiation scores were calculated for all genea across pairwise conditions (e.g. diabetic vs. non-diabetic patients). A large literature network was built for atherosclerosis. Connectivity analysis was used to extract a transcriptionally-active subnetwork for diabetic vs. non-diabetic conditions (right panel).

- **Subnetwork-based diagnosis** – subnetworks also provide a rich source of biomarkers for disease classification, based on mRNA profiling integrated with protein networks to identify subnetwork biomarkers (interconnected genes whose aggregate expression levels are predictive of disease state[21, 22]).

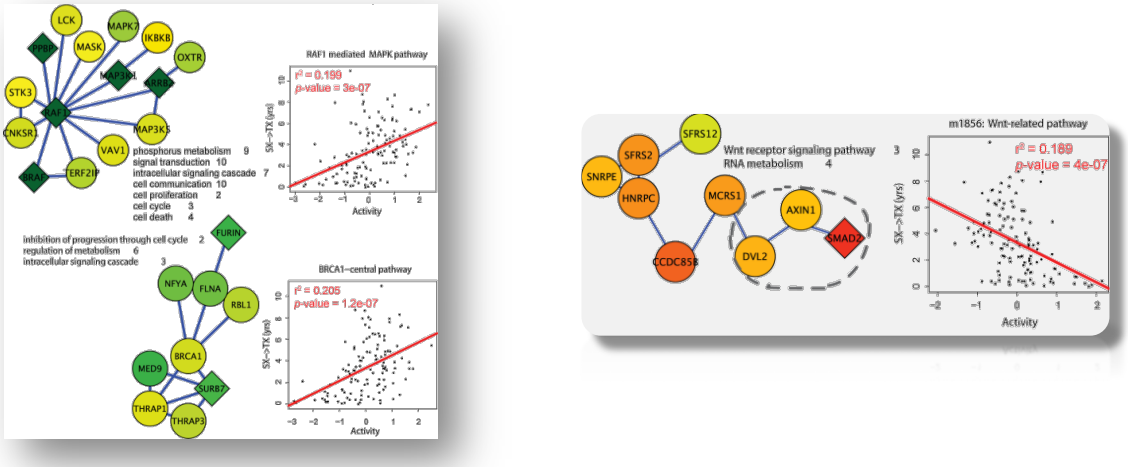


Figure 6. A network-based approach identified prognostic markers not as individual genes but as subnetworks extracted from molecular interaction databases. Gene expression profiles from Chronic Lymphocytoic Leukemia patients were mapped to a large human molecular interaction network. A search over this network was performed to identify prognostic subnetworks that could be used to predict treatment-free survival.

- **Subnetwork-based gene association** – molecular networks will provide a powerful framework for mapping common pathway mechanisms affected by collection of genotypes[23, 24].

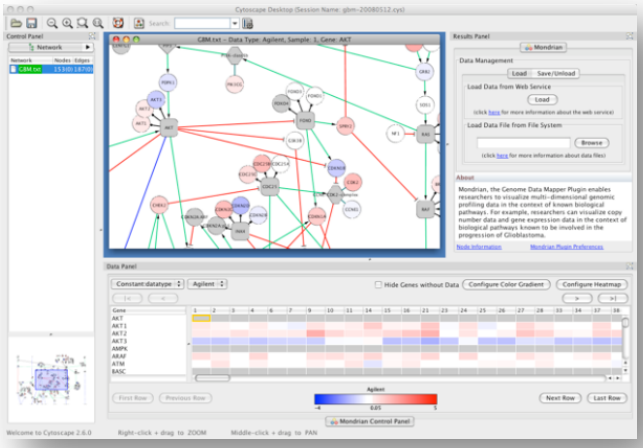


Figure 7. Cytoscape Mondian plugin with a dataset derived from the TCGA Glioblastoma Pilot Project. This dataset contains mutations, copy-number alterations, and expression data for 91 samples.

For the purposes of this tutorial, we will classify biological networks into three major categories: pathways, similarity networks, and interaction networks. Pathways include metabolic, regulatory, and signaling networks. Figure 2 shows a pathway containing genes involved in glioblastoma multiforme, a major form of brain cancer [25]. These genes were identified by a large-scale genetic analysis of copy number variation and genetic changes in 206 glioblastoma multiforme patients. The study was conducted as part of The Cancer Genome Atlas (TCGA) project. Notably, the study demonstrated that there was no single genetic defect responsible for glioblastoma multiforme, but that all of the cases showed significant pathway changes – strongly suggesting that this form of cancer is a "pathway disease." From a visualization standpoint, the real power is the ability to map expression, mutation, or copy number variation data onto pathways to reveal (or suggest) how the pathway and its components function under different sets of conditions, including disease states. Thus, the ability to analyze a variety of data sources and types and to map that data onto pathways is crucial. There are also

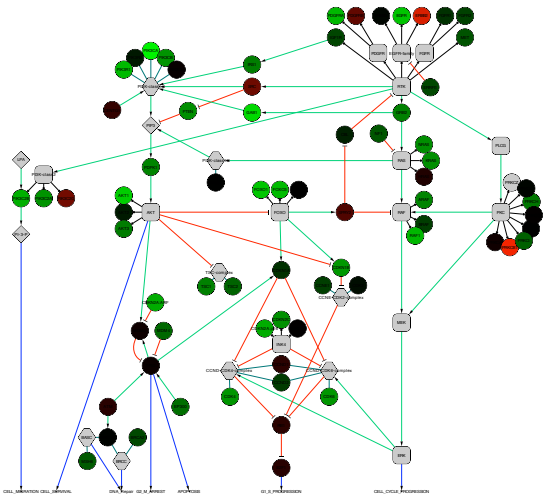


Figure 8. Partial pathway showing genes implicated in glioblastoma multiforme colored by the changes in copy number

techniques for deriving putative pathways from expression data[1] and for modeling the kinetics of biological processes [26] that are beyond the scope of this talk.

Interaction networks comprise the second category. In these networks, nodes represent biological entities and edges represent some form of interaction or relationship. A common example of this type is a protein-protein interaction (PPI) network. Figure 3 shows a yeast protein-protein interaction network generated by tandem affinity purification followed by mass spectrometry (TAP/MS) [27]. Analogous networks have been generated based on ligand similarities [28], protein similarities [29], and drug-target networks [30]. Generally, this class of biological networks can present as a "hair ball", where there is so much information that the meaningful relationships are difficult to discern. There is good evidence that analysis of a PPI network to find highly connected "hubs" can be used to predict protein complexes [8], and clustering of protein similarity networks can provide clues to protein family (and hence functional) assignments (Figure 4).

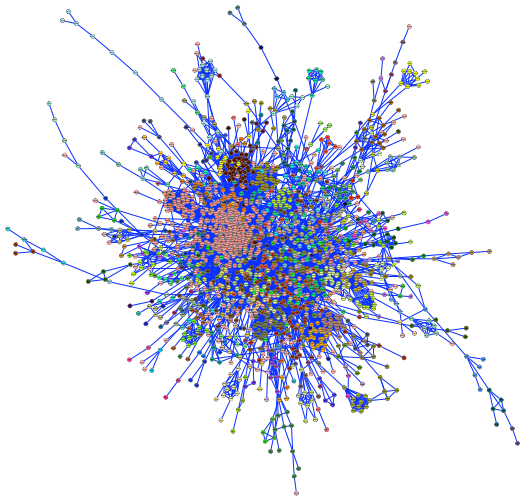


Figure 9. Partial protein-protein interaction network for *Saccharomyses cerevisiea* colored by predicted complexes.

A variety of analytical techniques can help to elucidate interaction networks. Clustering methods such as MCL [31] have proven valuable, although several algorithms more specific to various types of interaction networks have also been developed (c.f.[5]. In addition to clustering, a variety of metrics can be applied to an interaction network or nodes within the network. The average density (node degree) of the network, average shortest-path distance, number of connected components, measures of centrality, and the extent to which the network fits a scale-free model are all useful descriptors for the analysis of an interaction network. Altering the layout and visual attributes of the network can also be helpful.

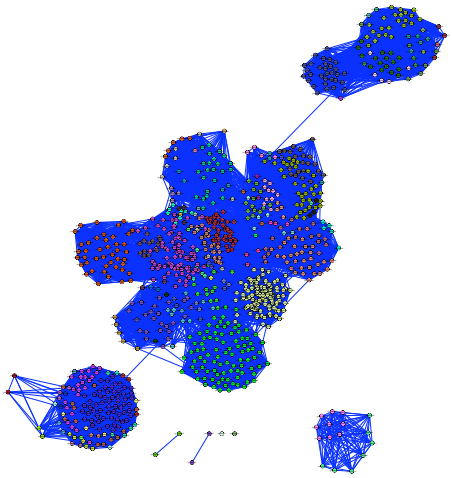


Figure 10. Protein similarity network of the amidohydrolase enzyme superfamily colored by subgroup.

Cytoscape is an open-source application for the visualization and analysis of (biological) networks. During my talk, I will use Cytoscape to demonstrate some of the techniques for visualizing and analyzing biological networks. In addition, I will demonstrate some ways that biological networks can be combined with other data to help elucidate function or the possible implications of changes in biological function due to perturbation, mutation, or infection.

---

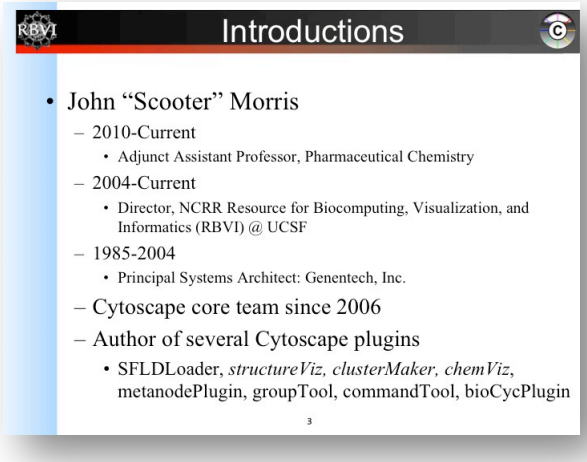
[1] c.f. the ExpressionCorrelation plugin from Gary Bader's lab: http://baderlab.org/Software/ExpressionCorrelation

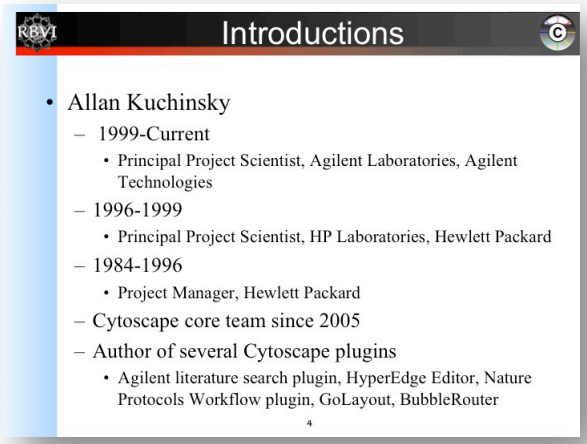
[2] It's approximately 2 because the shortest path between a non-hub node and all of the other nodes is 2

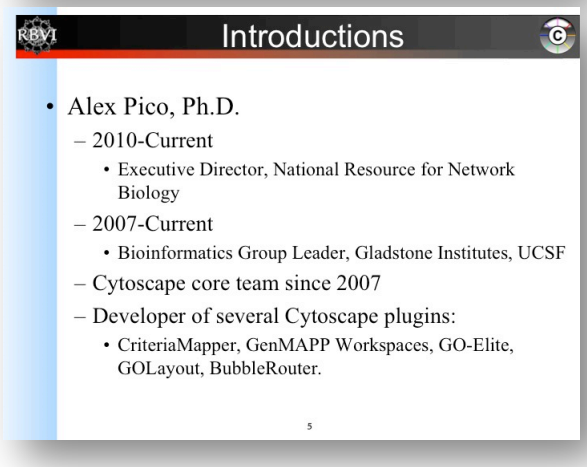
Introductions and setup

Introductions

The three instructors we have today are all experienced Cytoscape developers, with a cumulative of 15 years of participation in the Cytoscape core team. All three have a strong background in Cytoscape development, both from the perspective of core development, but also from the perspective of developing plugins that extend Cytoscape functionality. All three instructors have a long history of working in the biomedical field, both from the perspective of tool developers and the underlying science.





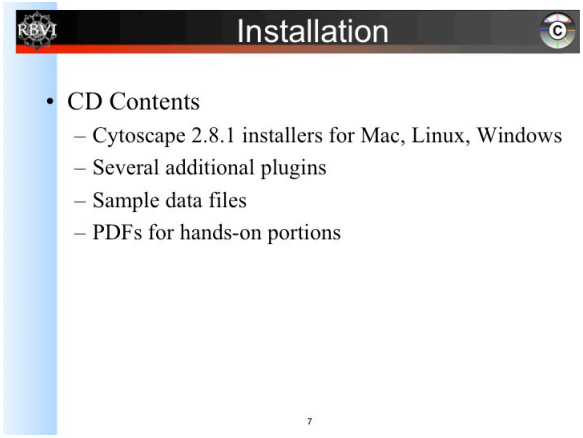


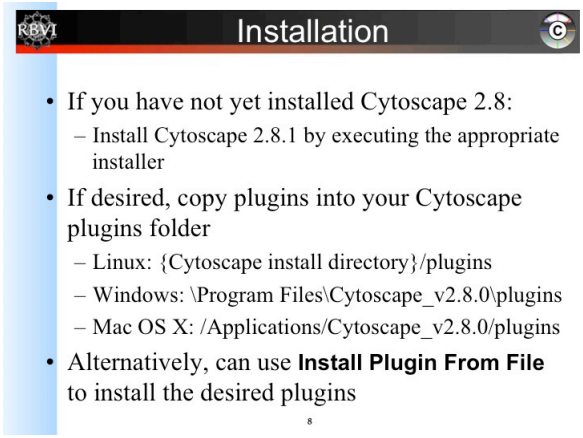
Notes

Setup

For the purposes of this tutorial, we will be using Cytoscape 2.8 (the current release is 2.8.1). The Cytoscape team is working on a new release, 3.0, which represents a significant architectural advance over the current release, but it has not been released as of this writing. Cytoscape has several components: the "core" of Cytoscape, which is distributed as a Java jar file, a set of "core" plugins that are always distributed with Cytoscape and provide some important functionality, a set of sample files, and a very large set of plugins the extend Cytoscape's functionality (in some ways significantly). Cytoscape 2.8 is available as installers for Mac, Windows, and Linux, which include the core, core plugins, and sample files. Plugins are generally available for download with Cytoscape's Plugin Manager.

It is also possible to load plugin jar files directly from hard disk or CD. To avoid potential network problems or contention, we have provided all of the plugins that we will use for today on the CD that we've distributed.





Notes

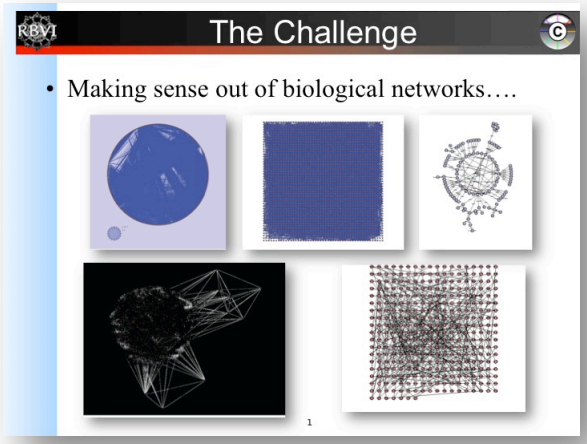
Biological Networks

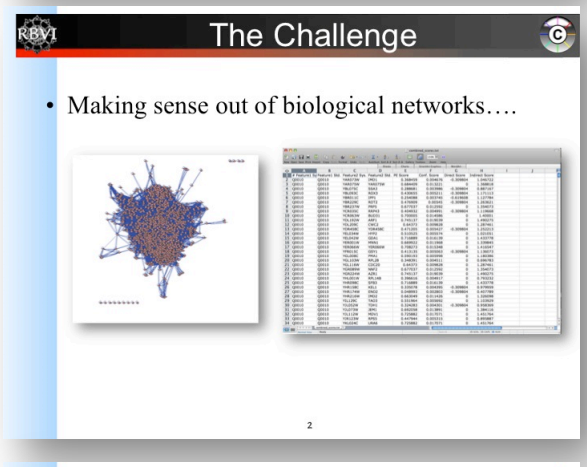
In this section, we will begin to explore the use of networks in biology. We begin by posing a challenge: how do we make sense of biological networks? We pose that challenge by providing a series of pictorial examples of networks in biology.

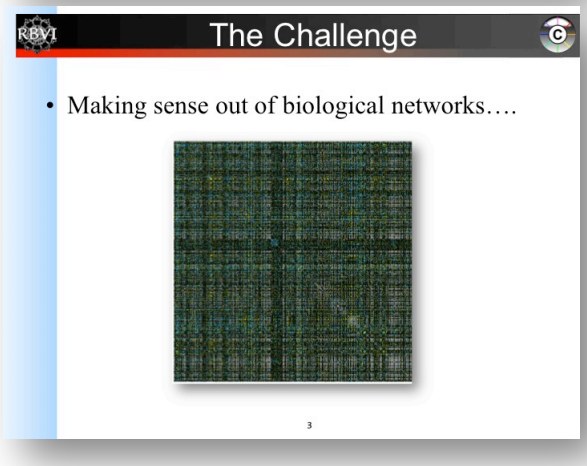
The Challenge

The images on this slide are all representations of biological networks. The challenge we are faced with is to extract the "meaning" behind these representations, which may be a purely visual challenge, but it might also involve analytical approaches.

All of the images at the right represent biological networks, including the Excel spreadsheet. Without more information about the content, these images don't tell us much. How can we extract this meaning? What are the analytical techniques? What are the common visualization approaches?

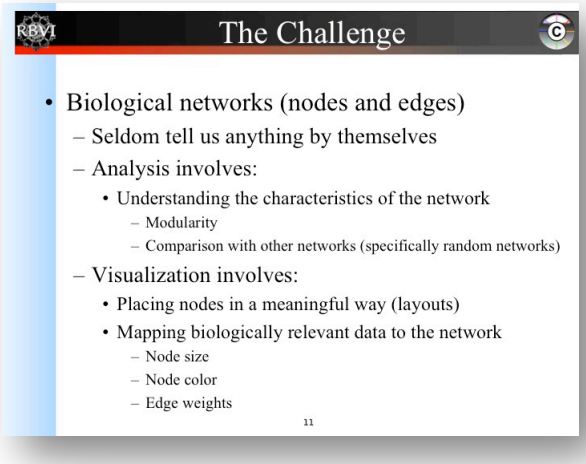




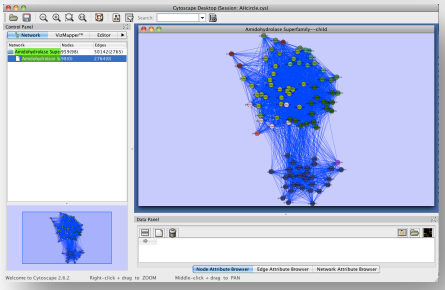


Notes

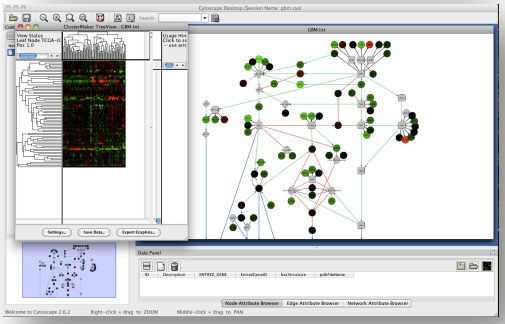
If we simply think of a biological network as a list of nodes and the edges that connect them, we're not going to be able to gain much information. However, if we add information to those nodes and edges to that we can analyze the interactions (or similarities) in more depth, or we use that additional information to visualize the nodes in some meaningful manner, we will find it easier to gain (or communicate) insight about aspects of the network. There are a number of analytical and visualization approaches that can help us, which are described below.



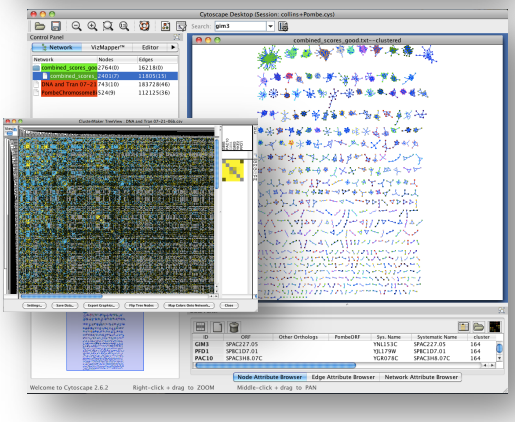
Taking the networks that we showed before, we can begin to analyze or visualize additional data. In the image at the right, we've colored the nodes in the network by protein family membership (members of protein families share functional characteristics), and then performed an edge-weighted layout where the edge weights represent the BLAST similarity between the proteins. As you can see pretty quickly that similar proteins tend to group together.



In this example, we've combined a network representation with an analysis of some of the associated data. The image at the left is a hierarchical clustering of all of the genes in the TCGA glioblastoma study vs. all of the patients in the study. This allows us to look for patterns in the heat map and associate those patterns with specific genes or groups of genes in the pathway.



In the final example on the right, we have combined two different visualizations with two different analyses. The heat map on the left represents a hierarchical cluster of genetic interactions and the network shows the results of an MCL cluster of a set of physical interactions. These views are linked, allowing users to select groups in one view and determine if the same groups exist in the other view. This allows researchers to explore areas where there are tight protein-protein physical interactions as well as genetic interactions, providing pretty strong evidence for the existence of a complex.

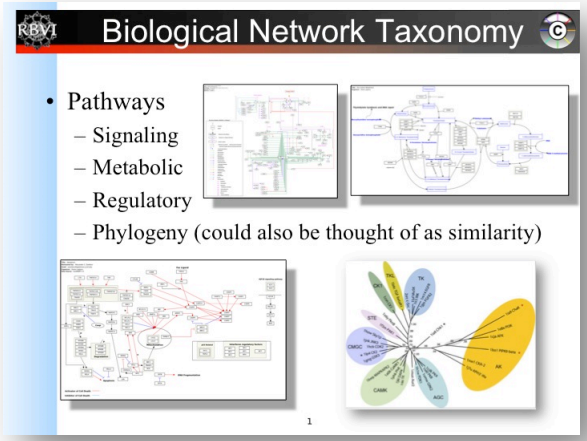


But, how do we know what kinds of analyses make sense, and what kinds of visualizations are appropriate?

Biological Network Taxonomy

Before tackling this question, we need to understand that not all "biological networks" are the same. In particular, there is a sort of taxonomy of networks – each visualization or analytical technique can be either more or less appropriate for the different network types. For our purposes, we can divide these biological networks into 3 main groups: pathways, interaction networks, and similarity networks.

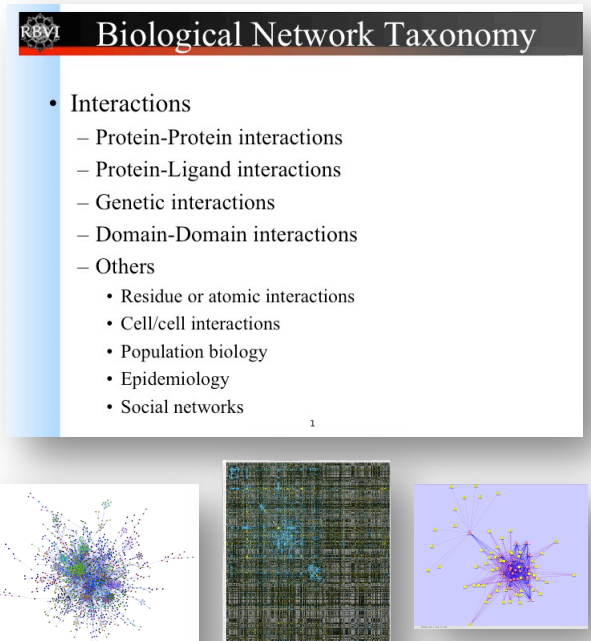
Pathways



The first type of network in our taxonomy is probably the most familiar. We've all seen pathway diagrams, whether those pathways represent signaling pathways, metabolic pathways, or regulatory pathways. These networks are often hand-curated diagrams that have been constructed for didactic purposes. However, even though the positions and graphical annotations associated with these networks does not lend them to the normal types of network analysis techniques, they can be extremely useful as templates on which we can paint expression profile information, or any other kinds of annotation that we want to use to show information associated with the curated pathway. Phylogenetic trees can also be thought of in a similar fashion of those trees have been hand curated like the kinase phylogenetic tree [32] shown in the slide.

Notes

Interactions

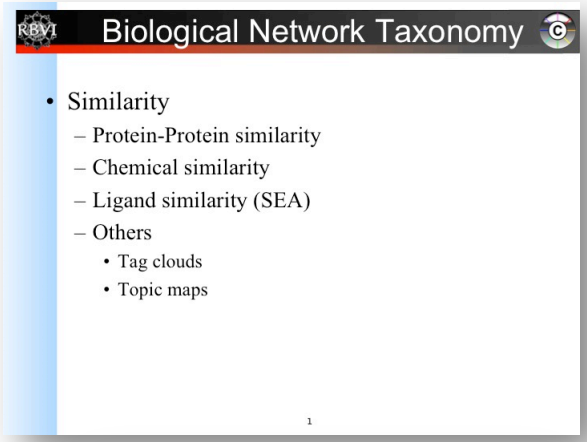


The second type of network in our taxonomy are interaction networks. While pathways are probably familiar to most because of their use for educational purposes, interaction networks are what most people of when we think of "network biology". Basically, these networks reflect the interactions between biological entities. The entities might all be proteins, giving us the canonical protein-protein interaction network shown to the right in the first frame. The interacting entities might also be genes, in which case, the network could be a genetic interaction network. The middle panel at the right shows a particular representation of an epistatic miniarray profile (EMAP). These networks are formed by recording the differential results of double-delete mutants when compared to the expected combination of single-delete mutants. The last network shows a protein-ligand interaction network. Interaction networks don't necessarily need to have only one interacting entity, and as we are rediscovering the importance of metabolic pathways, the "metabolome", which combines metabolites with the enzymes and regulatory proteins which control metabolism. There are also efforts underway to understand how the interactomes of pathogens interact with the interactomes of their hosts – yet another kind of "mixed" interaction network.

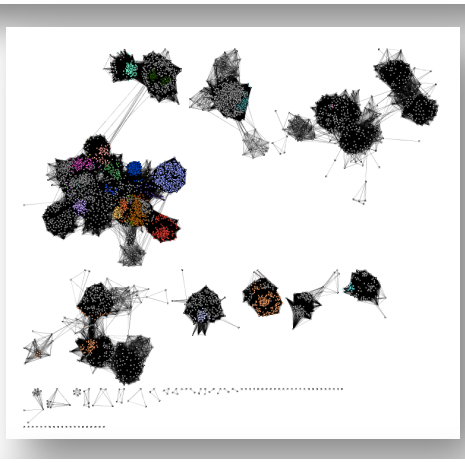
Of course, there are many kinds of biological interactions we might be interested in, up to and including how people interact with each other. Such social networks are beyond our scope, but social network analysis is very similar to biological networks analysis and provide a fruitful source of algorithms and visualization techniques.

Notes

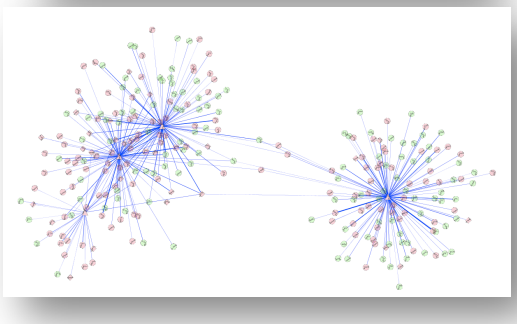
Similarity



The final type of networks we want to discuss are similarity networks. In similarity networks, the nodes represent biological entities and the edges represent some measure of the similarity between them. There are several types of similarity networks that are commonly used in biology today. One common similarity metric is the Tanimoto coefficient[33-35], which represents the similarity between two small molecules based on the chemical fingerprints of each of them[36]. Other similarity metrics include sequence similarity as measured by BLAST[29, 37], PSI-BLAST[38], or Smith-Waterman[39], structural similarity as measured by RMSD or other structural similarity measures[40-45], or the ligand similarity as measure by the similarity ensemble approach (SEA) method[28].



There are other types of non-biological networks that use various kinds of similarity measures. Tag clouds[46] and topic maps[47], which is one of the semantic web technologies.



The images at the right show two examples of similarity networks. The network on top is a protein-protein similarity network showing the Amidohydrolase enzyme superfamily from the Structure-Function Linkage Database (SFLD)[48]. The colors on the network represent proteins of similar function. Note that these proteins tend to group together based on their BLAST similarity[29].

The network on the bottom shows a network of small-molecules where the edges represent the Tanimoto similarity between them. These networks can be useful to find molecules with similar structural characteristics

Notes

Analytical Approaches

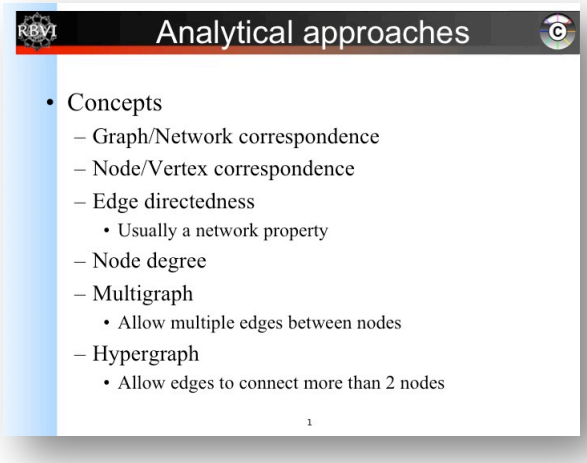
The analysis of networks is a large and complex topic that we can't do justice in a single tutorial (even less a tutorial handout). In general, network analysis is part of the mathematics known as graph theory, and there are entire conferences (and many textbooks) devoted to the area. A good starting point might the Wikipedia article[49] or the online book "Graph Theory with Applications"[50]. Our goal here is to provide a brief introduction and touch on some of the main approaches used with biological networks.

Concepts

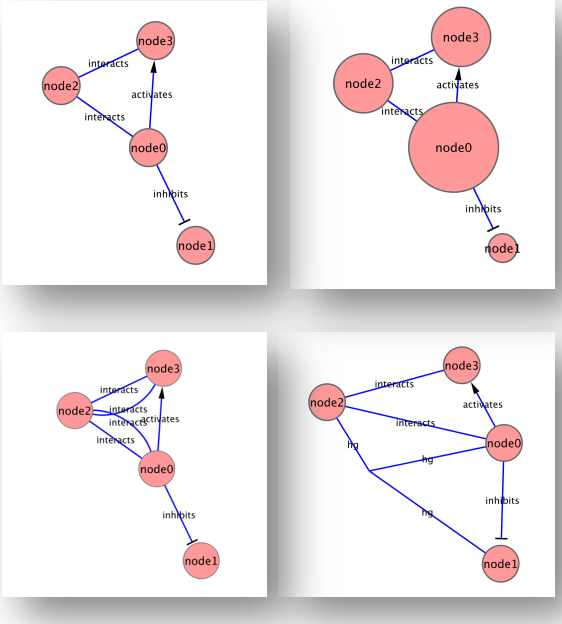
In mathematical terms, a biological network (any network for that matter) is a graph, often written:

$$G = (V(G), E(G), \psi_G)$$

where $V(G)$ are the set of vertices (nodes) in the graph and $E(G)$ are the set of edges. In this particular notation, ψ_G is the set of incidence functions that define which edge goes with which vertices.



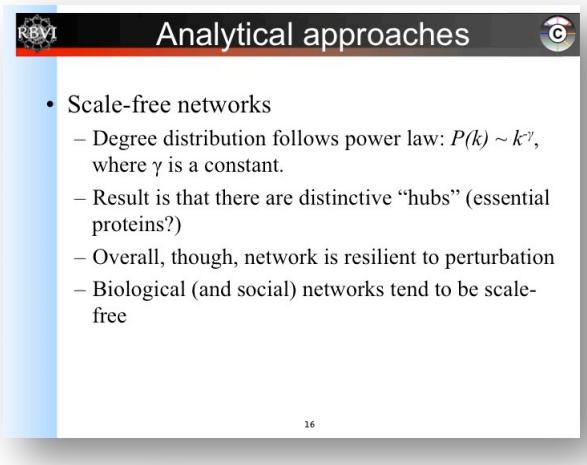
The edges between nodes can either be directed or undirected. This is easiest to understand when considering the *degree* of a node. In an undirected network, the degree of a node is simply the number of edges connected to it. In the first simple network at the right, the node (**node0**) has three edges connected to it, so it has a degree of 3. In a network with directed edges, we need to expand our concept of degree to include *in-degree*, the number of edges that connect *to* this node, and *out-degree*, the number of edges that originate *from* this node. In the second network at the right, the size of the nodes reflects the node degree.



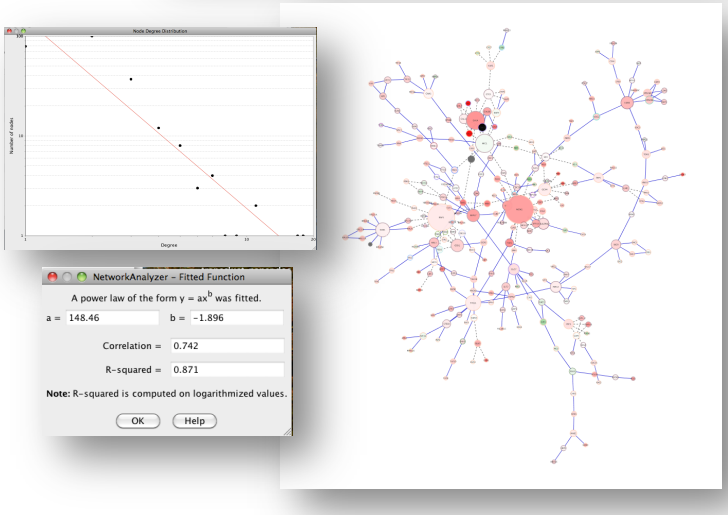
There are also differences between the types of networks. The first network at the right is a *multigraph*. In a multigraph, there can be multiple edges between nodes. The network at the far right on the other hand, is a *hypergraph*. In a hypergraph, an edge can be connected to more than two edges.

Notes

Scale-free networks



One property of network topology that is of interest is the *degree distribution* – that is, the distribution of how many edges each node has (also referred to as the *connectivity distribution*)[51]. A network is said to be *scale-free* if the degree distribution fits a power law. It has been reported that many types of biological networks are scale free[52-62]. The characteristics of scale-free networks are that there is a short path from any node to another node (*small world property*), there are many nodes with few connections and a few nodes with many connections (*hubs*), and the hubs are enriched with essential/legal nodes (*centrality and lethality principal*)[52, 63].

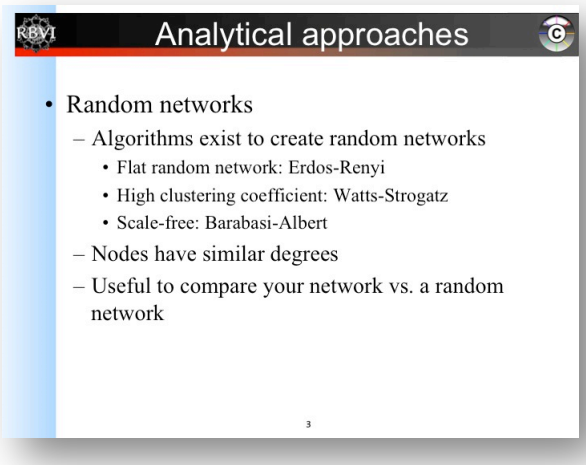


Scale-free networks have interesting properties for biological systems – in particular, they are robust to random breakdowns[64]. They are also (as the name implies) invariant to changes in scale. On the other hand, recent analysis of several data sources have begun to throw into question exactly how well many biological networks fit the scale-free power law distribution[63, 65-67]. So, while none of the authors have suggested that biological networks don't exhibit some scale-free characteristics, they don't fit the power-law degree distribution well enough to be considered scale-free.

It should also be noted that biological networks aren't the only network type that tends to be scale-free. For example, both social networks and the Internet tend to be scale-free[68, 69]. In both cases the overall topology tends to be one with a few hubs of high degree and lots of lower-degree nodes.

Notes

Random networks



Random networks (random graphs) are important tools for determining the extent to which a computationally derived network differs from a similar "random network". This is, in principal, the same idea behind the BLAST expectation value or the *p* value that you might get from a statistical test.

Networks, however, are complicated, and developing an appropriate probability model is non-trivial. There are several algorithms commonly used to generate random networks. In the simplest case, you can just generate a graph, *G(n,p)*, where for any two nodes N_1 and N_2, there is a probability *p* that there is an edge between them[70]. This is similar to the Erdős-Rényi model([71, 72] as cited in [73]), but in the Erdős-Rényi model, the number of edges is restricted to a fixed number, *M*. Thus, the graph, *G(n,M)*, is a graph where all of the *M* edges appear with equal probability.

The problem with both of these "flat" models is that neither of the models are likely to result in graphs that exhibit the characteristics of biological networks (small world, scale-free) discussed above. One approach to this is to explicitly model the random graph such that it exhibits small-world properties (short average path lengths and high clustering). The is the approach proposed by Watts and Strogatz[74]. In the Watts and Strogatz model, there are three key parameters: the number of nodes, *N*, the mean degree of the nodes, *K*, and a tuning parameter β, which is between 0 and 1. The algorithms begins by generating a network with *N* nodes, each connected to *K* neighbors, *K*/2 on each side. Then for every edge (n_i,n_j) rewire that edge with probability β such that there are no loops and there is no duplicate edges. The result depends on the value of β. If β is near zero, the result is a regular lattice. If β is one, this approaches the random graph similar to the Erdős-Rényi model with $p = \frac{NK}{2\binom{N}{2}}$.

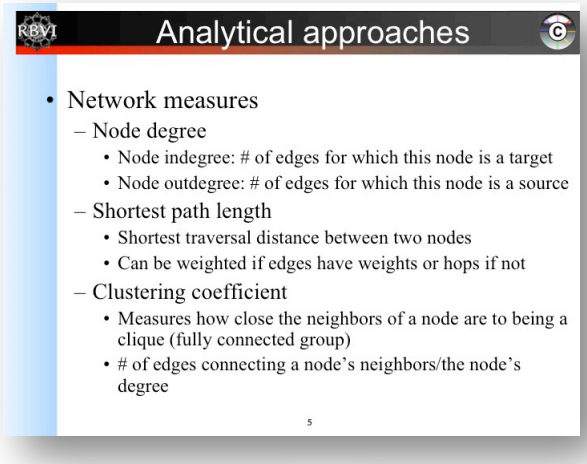
Another approach is to implement a random graph that is scale-free. The Barabási-Albert model is an approach to generating random scale-free graphs[68]. This approach starts with a small network *G(n,m)*, where *n* is the number of nodes (≥2) and *m* is the number of edges. The requirement is that all nodes have degree of at least 1. Then new nodes are added according to a probability p_i:

$$p_i = \frac{k_i}{\sum_j k_j},$$

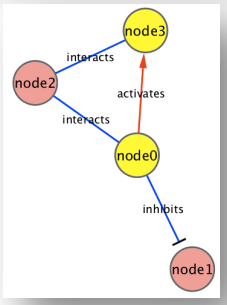
where k_i is the degree of the node *i*. This results in hubs (nodes with more edges) continuing to get more edges and nodes with fewer edges being less likely to get new edges. This results in a degree distribution that fits the scale-free model quite well, but is still random in nature.

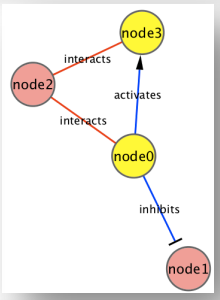
Network measures

We've the three most common network measures already: node degree, path length, and clustering coefficient. The first two of these are intuitively understandable. The third is a little more difficult to conceptualize since it doesn't fit our concept of clusters (i.e. groupings of nodes or modularity) very well.

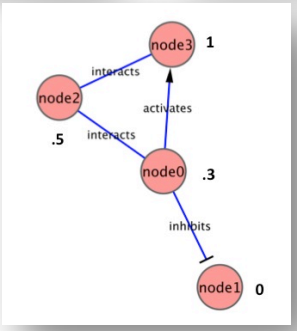


Node degree is, as we've already mentioned, the number of edges connected to this node. In a directed network, the node *indegree* is the number of edges directed towards this node, and the node *outdegree* is the number of edges directed away from this node. In the network at the right, for example, **node3** has an *indegree* of 2 and an *outdegree* of 1 (assuming we count the undirected edge as both in and out).





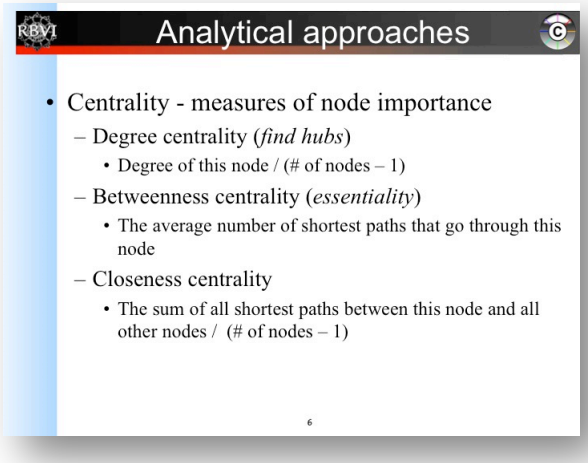
Path length is also relatively easy to imagine. If we look for the shortest path from **node0** to **node3** (the first network at the right) it's the edge between them. On the other hand, the shortest path from **node3** to **node0** goes through **node2** (because the edge between **node0** and **node3** is directed). The length of the path is often just a hop count (1 in the first example, 2 in the second), but can also be weighted, which might mean the shortest path is not the path that traverses the fewest nodes.



The clustering coefficient is a measure of the degree to which nodes form a complete graph. It was originally defined to measure the degree to which a network exhibits small-world properties[74]. For undirected graphs, the local clustering coefficient is given as: $C_i = \frac{2|\{e_{jk}\}|}{k_i(k_i-1)}$

In the network example at the right (assuming it's undirected), **node3** has two neighbors (degree 2), **node2** and **node0** share an edge, so we have (2*1)/2(2-1) = 1. On the other hand, **node0** is degree 3, but only **node2** and **node3** are connected, so we have (2*1)/3(3-1) = .3. The network average clustering coefficient can be used to express the degree to which a graph exhibits small-world properties. The average is simply: $\overline{C} = \frac{1}{n}\sum_{i=1}^{n} C_i$.

Another important set of networks measures that has important properties are the various centrality measures. These approaches (in general) attempt to provide a measure of the importance of a given node. There are many centrality measures, but we'll just discuss three of them here.



The first centrality measure we'll discuss is *degree centrality*. Nodes with high degree centrality are the hubs in scale-free networks, for example. This is an easy measure to compute the degree centrality (C_D) of node v: $C_D(v) = \frac{\deg(v)}{n-1}$, where n is the number of nodes in the network.

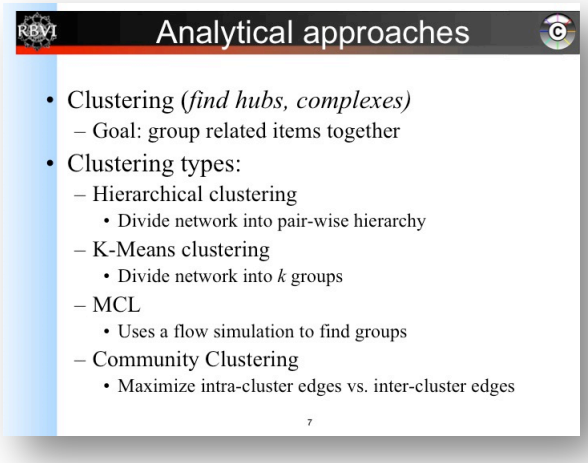
Betweenness centrality is another centrality measure than tends to reflect the essentiality of a node in the network. Essentially it measures to extent to which "all roads lead through" this node. The betweenness centrality for a node v is calculated as: $C_B(v) = \sum_{s \neq v \neq t \in V} \frac{\sigma_{st}(v)}{\sigma_{st}}$, where σ_{st} is the number of shortest paths from s to t and $\sigma_{st}(v)$ is the number of shortest paths from s to t that go through v. Intuitively, this makes sense – if a large percentage of the shortest paths between two nodes go through a given node, removal of that node will have a significant effect on the network topology (from the perspective of those two nodes).

Closeness centrality is the degree to which this node is close to all other nodes. It is again calculated based on shortest paths: $C_C(v) = \frac{\sum_{t \neq v \in V} S(v,t)}{n-1}$, where $S(v,t)$ is the shortest path between v and t. So, in a star topology, where all nodes are connected to a single hub, the closeness centrality measure for the hub is 1 and ~2 for all other nodes[2].

Notes

[2] It's approximately 2 because the shortest path between a non-hub node and all of the other nodes is 2 except for the hub node, in which case the shortest path is 1.

Clustering



Clustering is a heavily used technique for analyzing networks, both biological and otherwise. The overall goal of clustering is to group items together that are related based on some measure. Clustering is an active area of research and there are many clustering algorithms that have long been used for biological applications, and even more algorithms that are being developed for specialized purposes.

Before we talk about specific clustering approaches, it is important to understand that all of the clustering approaches depend on some metric for determining the similarity of the items being clustered. This similarity metric is termed a *distance* metric in clustering terms, and there are a number of ways to calculate the distance in feature space (that is, the terms or values you are using to determine the similarity between objects). A common measure is the *Euclidean* distance, which is simply the distance between two points in n-dimensional space:

$$d(p,q) = \sqrt{(q_1 - p_1)^2 + (q_2 - p_2)^2 + \cdots + (q_n - p_n)^2}$$

Other common techniques are based on the *Pearson correlation*, *r*, between any two series of numbers $x = (x_1, x_2, \ldots, x_n)$ and $y = (y_1, y_2, \ldots, y_n)$, which is defined as:

$$r = \frac{1}{n}\sum_{i=1}^{n}\left(\frac{x_i - \bar{x}}{\sigma_x}\right)\left(\frac{y_i - \bar{y}}{\sigma_y}\right)$$

where σ_x is the standard deviation of the *x* series, and σ_y is the standard deviation of the *y* series.

This term can be either *centered* (as above), or *uncentered*, which assumes a mean of zero (even if it's not). There are many other approaches to calculating the distance, from taking the negative log of the BLAST e-value to much more complicated approaches designed to account for specific characteristics of the data.

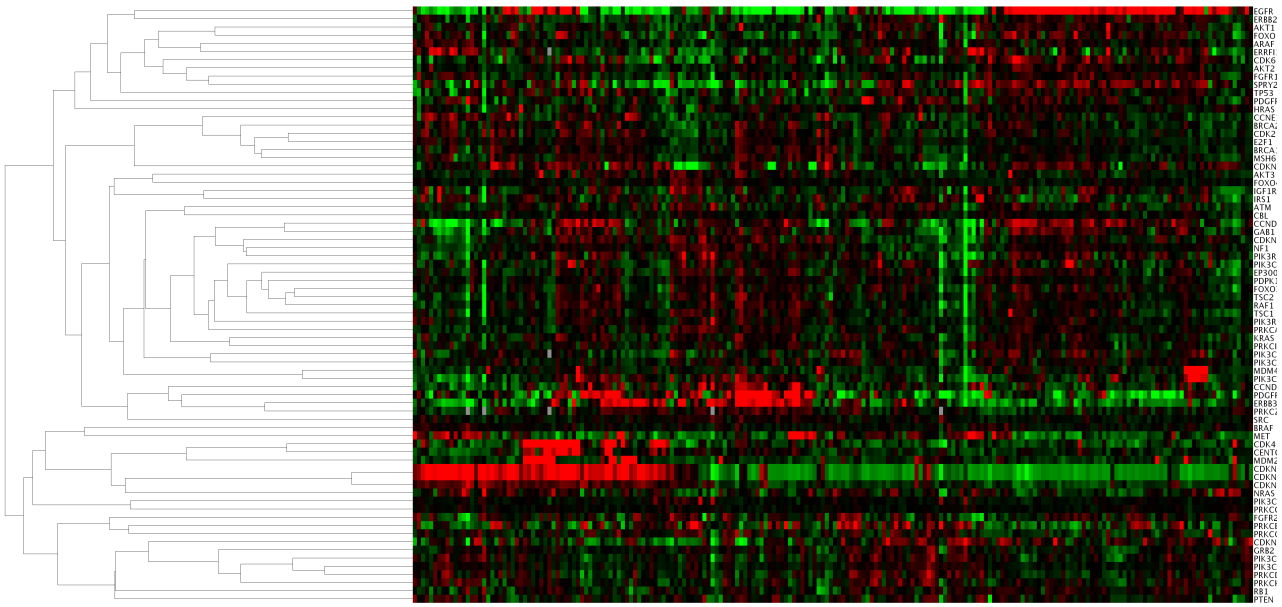
Hierarchical Clustering

A very common clustering approach is *hierarchical* clustering[75]. As the name implies, this approach divides the objects into a pairwise hierarchy. Hierarchical clustering has been used for many years as one of the major approaches to analyzing and visualizing microarray data[76]. An important first step in performing hierarchical clustering is to determine the distance metric (above). The second step is to determine how to link the pairwise distances[3]:

- *Single linkage* clustering takes the minimum pairwise distance,
- *Complete linkage* clustering takes the maximum pairwise distance,
- *Average linkage* clustering (UPGMA) takes the average of all of the pairwise distances,
- *Centroid linkage* clustering takes the distance between the centroids of all pairs of elements.

[3] This list is taken from the clustering approaches used in the original Cluster program from Eisen and colleagues, which has been inherited by clusterMaker and other Cluster-clones.

Once the metrics and linkages have been selected, clustering may be accomplished by either an *agglomerative* (bottom-up) or *divisive* (top-down) method. In either case, the result is tree (hierarchy) where the nodes closer together in the tree are more similar. For microarray data, this is often shown as a dendrogram associated with the heatmap that reflects the fold changes in the expression data (see the example below).



k-Means Clustering

Another common clustering technique is *k-means[77, 78]*. In k-means clustering the algorithm divides the data set up into *k* groups in such a way that the value of the item gets assigned to the cluster with the nearest mean. The approach is relatively simple: given a set of **n** data items $x = \langle x_1, x_2, \ldots, x_n \rangle$ the idea is to partition the **n** items into *k* sets so as to minimize the within-cluster sum of squares (WCSS):

$$\arg\min_{\mathbf{S}} \sum_{i=1}^{k} \sum_{x_j \in S_i} \|x_j - \boldsymbol{\mu}_i\|^2$$

where $\mathbf{S} = (S_1, S_2, \ldots, S_n)$ are the clusters and $\boldsymbol{\mu}_i$ are the mean of the points in each cluster S_i. k-means has been used in a number of applications, and has been incorporated in to a number of other algorithms.

There are many other clustering algorithms and combinations of algorithms used in network applications – far too many for us to cover here. Often these algorithms are general algorithms (e.g. Community clustering[79], MCL[31, 80, 81], Spectral Clustering[82-86], and Affinity Propagation[87, 88]) and often they designed for special purposes (e.g. SCPS[89], MCODE[5], FORCE[90], TransClust[91]). Some algorithms are actually combinations of algorithms (e.g.

AutoSOME[92]). We're going to cover only three of these algorithms (MCL, Spectral, and Affinity Propagation), but the interested reader is encouraged to explore the references below.

MCL Clustering

MCL clustering (MCL is short for Markov Clustering) is a clustering approach that simulates a weighted random walk through a network. The idea behind the algorithm is that because edges within the natural groupings will most likely stay within the group, the vast majority of the steps in a random walk will be within the natural group. The other way to think about it is by imagining edges as flows – most of the flow through a network with natural clusters will stay within the clusters – very little will flow between the clusters. The simulation of the random walk is by alternate application of two operations: *expansion* and *inflation*. First, the distance matrix is converted to a stochastic matrix (a non-negative matrix where each of the columns sums to 1). In the expansion step, the stochastic matrix is squared using the normal matrix product. In the inflation step, the Hadamard product of the matrix (entry-wise multiplication by an inflation parameter, I) is taken. After the inflation step, a scaling step is added which returns the matrix to a stochastic matrix. Repeated expansion and inflation will have the result of removing cells in the distance matrix (i.e. edges) that represent inter-cluster edges.

MCL clustering has been used for a large number of biological applications, including the finding of protein complexes in protein-protein interaction networks and the grouping of proteins in protein similarity networks. MCL has proven to be very fast and robust with then number of edges is reasonably low, but can have problems resolving dense networks necessitating some form of algorithm or user-chosen cut-off value to reduce the edge density[93]. MCL has the nice characteristic that it does not necessitate the user to select the number of clusters in advance, although the inflation parameter I does have to be specified.

Spectral Clustering

Spectral clustering takes in name from the use of spectral properties of the similarity (or distance) matrix constructed from the network. Given a set of data points A, the similarity matrix may be defined as a matrix S where S_{ij} represents a measure of the similarity between points i and j which are members of the set A. Spectral clustering techniques make use of the spectrum of this matrix of the data to perform dimensionality reduction for clustering in fewer dimensions.

One such technique is the Normalized Cuts algorithm[94, 95], commonly used for image segmentation. It partitions points into two sets (S_1,S_2) based on the eigenvector v corresponding to the second-smallest eigenvalue of the Laplacian matrix

$$L = I - D^{-\frac{1}{2}}SD^{-\frac{1}{2}}$$

of S, where D is the diagonal matrix

$$D_{ij} = \sum_j S_{ij}$$

This partitioning may be done in various ways, such as by taking the median m of the components in v, and placing all points whose component in v is greater than m in S_1, and the rest in S_2. The algorithm can be used for hierarchical clustering by repeatedly partitioning the subsets in this fashion.

A related algorithm is the Meila-Shi algorithm[82], which takes the eigenvectors corresponding to the k largest eigenvalues of the matrix $P = SD^{-1}$ for some k, and then invokes another (e.g. k-means) to cluster points by their respective k components in these eigenvectors.

Spectral clustering techniques are very useful in biology, but they have the disadvantage that since they essentially divide the data into two sets, you must either combine them with something like k-means or use a hierarchical decomposition to arrive at a more refined clustering.

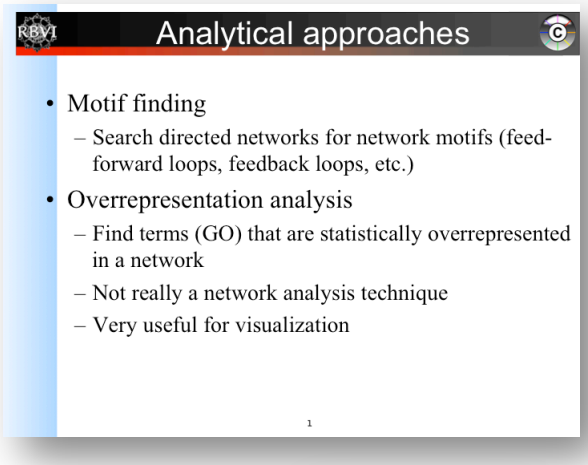
Affinity Propagation

Affinity propagation[87] is a newer algorithm that takes a message passing approach rather than a mathematical approach to clustering. Basically, as with the other approaches, affinity propagation takes a similarity matrix s(i,j), which represents the starting point of the algorithm. In addition, each point is given a preference value s(k,k) which is used to seed the likelihood of this point being an exemplar for the formation of a cluster (this is often just set to a flat value to allow the algorithm to learn the number of clusters). Then, the points exchange messages of two types: responsibilities (r(i,k)) are sent from point i to point k and reflects the degree to which k is a good exemplar for point i; and availability (a(i,k)) is sent from point k to point i to reflect the evidence for i to choose k for its exemplar. The algorithm runs until some stopping point usually based on the degree to which r(i,k) and a(i,k) change during each pass. See their web site (http://www.psi.toronto.edu/index.php?q=affinity propagation) for more information about the algorithm and its application.

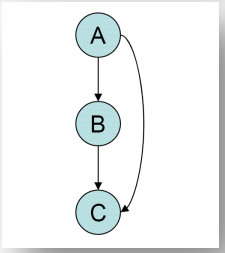
Affinity propagation has numerous applications in biology and seems to perform well in the datasets provided by the authors. Some comparative analysis by others[96] suggests that other algorithms might be less susceptible to noise and more robust for some applications.

Notes

Network motifs



A network motif is a pattern of connectivity that occurs more frequently than might be expected by a random connection of nodes[97]. As might be expected by the reuse we often see in biology, biological networks tend to have a small set of network motifs that act like components in a larger circuit[98, 99]. Network motifs have been identified in the gene regulation network of *E. coli*[100] as well as a larger set of networks[101]. There are a number of network motifs that have been identified in biology, including feed forward loops[102-106] (like the one shown at the right), feedback loops[107-109], positive and negative auto-regulation loops[110]. These biological circuits are critical to regulatory processes in the cell, so identifying them in protein-protein interaction networks can provide important clues to the pathway which the protein participate in[111-113].

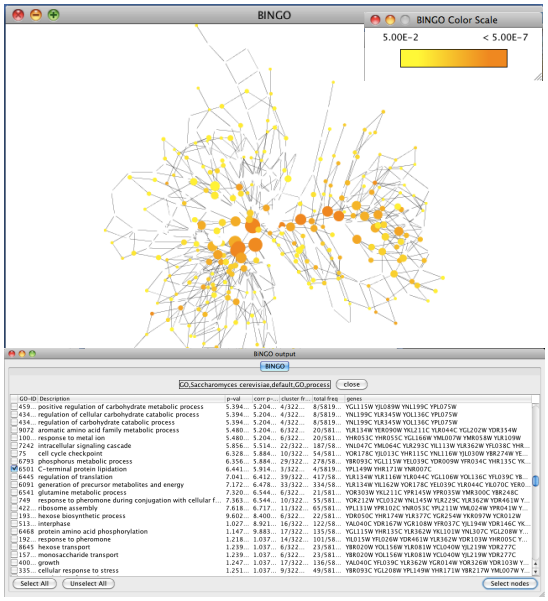


Simple Feed Forward Loop

Overrepresentation analysis

Overrepresentation analysis (ORA) is an important tool used to identify aspects or attributes of a subset of nodes that are statistically more common in those nodes than in the full set. The most common approach is to cluster a group of genes based on expression data and look for overrepresentation of various gene ontology (GO)[114] terms in the groups to determine if a particular expression pattern suggests a particular biological process[115-118].

One of the things to keep in mind when doing ORA is that the resultant *p*-values may need to be adjusted since multiple tests are conducted. This makes sense – if we're performing multiple tests we increase the possibility that we'll get a false positive based on random chance. Two methods for correcting for multiple tests are the Dunn-Bonferroni Familywise Error Rate (FWER)[119] correction and the Benjamini & Hochberg False Discovery Rate (FDR)[120] correction.



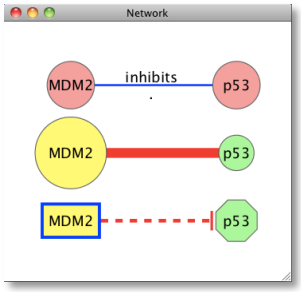
The image at the right shows the results of an overrepresentation analysis of a yeast expression data set using the Cytoscape BiNGO plugin[121].

Visualization

In the previous sections, we made use of a number of visualization techniques that are easily taken for granted. In this section, we will detail the techniques and key decisions involved in producing network and pathway visualizations. Using biological networks to visualize data is a critical aspect of exploratory analysis: facilitating interpretation, new insights and new hypotheses (PMID: 20824171). We are visual creatures, after all.

Depiction

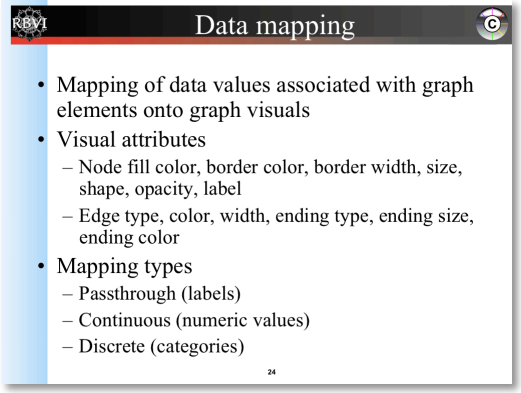
The basic visual motif of networks in Cytoscape is that of nodes and edges. In biological networks, the nodes often represent genes, proteins or small molecules, while the edges (or lines) represent interactions and relationships between connected nodes (see figure at the right). Beyond this core motif, all other visual features (e.g., shape, size, color, thickness, label, transparency, etc) are flexible and can be used to represent practically any data value, annotation or attribute.



Node and edge motif in various visual styles

Data Mapping

The first thing most users want to do in Cytoscape is to map their data onto networks for visualization. The variety of data and network types has already been explored in previous sections. Here, we will focus on the mechanics of data mapping using the VizMapper interface.

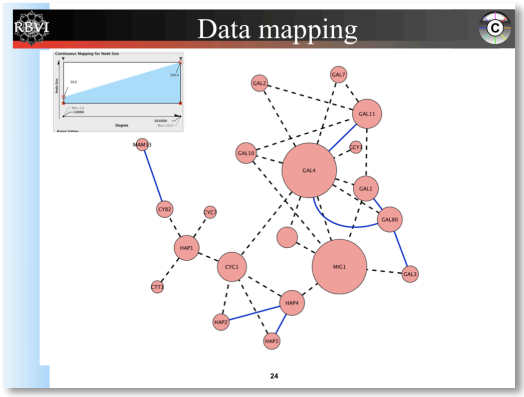


VizMapper provides a user interface for controlling the mapping of data attributes to visual attributes. There is a long list of available visual attributes that can be mapped to, including node properties such as fill color, border color, shape, width, height, opacity and label, and edge properties such as type, color, thickness, as well as arrow type, size and color.

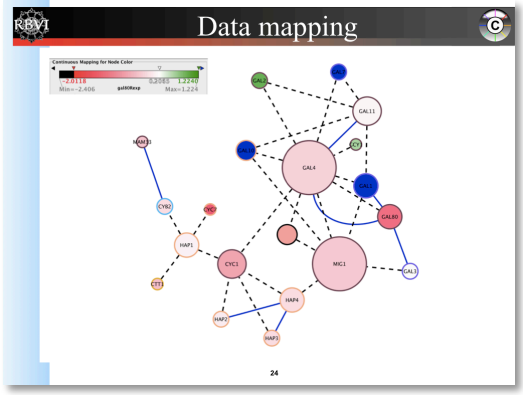
Notes

Data attributes can be mapped in three main ways: **Passthrough** – directly passing the data value to the visual attribute, e.g., labels. **Continuous** – mapping a continuous range of numerical values to a range of visual attributes, e.g., expression values to a color gradient. **Discrete** – mapping discrete data values (string or numeric) to specified visual attributes, e.g., five different categories to five different colors.

The next two examples focus on two types of Continuous data mapping, since these are the most useful and most challenging. First, we map degree-of-connectivity (the data attribute) to node size (the visual attribute), so that more connected nodes appear proportionally larger, thus highlighting potential "hubs." In the VizMapper interface, you would begin by double-clicking on 'Node Size' and selecting the data attribute containing degree information. Then select "Continuous Mapping" type and click the graphic to edit the mapping parameters. The min and max of the data attribute is given as the x-axis and the visual attribute is the y-axis.



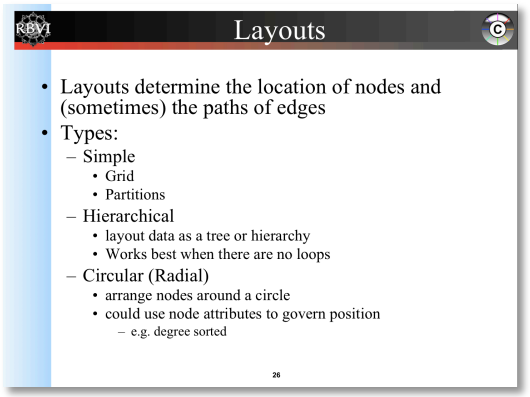
The second example maps continuous expression values to node color. Once again in the VizMapper interface, you would double-click the visual attribute, pull-down the data attribute and then choose "Continuous Mapping." When you click on the graphic, you will notice a different parameterization. Once again, the data attribute is given as the x-axis, but now instead of a y-axis, you will find thresholds that control the ends and mid-point of a color gradient as well as step-function thresholds to set discrete colors for values exceeding the gradient range. This is a handle tool for focusing the continuous mapping of color to a critical range of data.

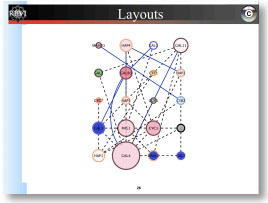


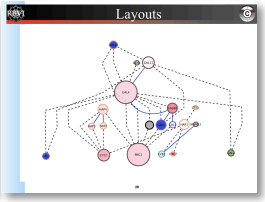
Notes

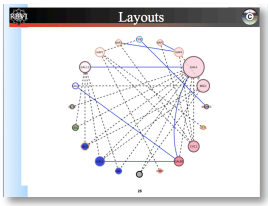
Layouts

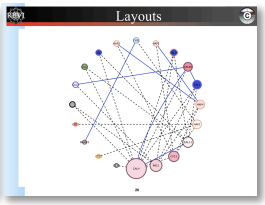
The majority of network information does not come with fixed coordinates. With the exception of manually curated pathway diagrams, networks typically rely on automated layout algorithms to position nodes and edges. Cytoscape comes with a wide variety of built-in layout algorithms that can be applied to any pathway or network. In addition, a number of plugin extensions have been developed to support additional layouts.











Here, we will describe the main layout types natively supported by Cytoscape. You can find these in the menu *Layout > Cytoscape Layouts*.

Grid Layout – a simple layout of nodes in arbitrary order arranged in a grid pattern. This layout does *not* take in account edge crossings, weights or degree of connectivity.

Group Attributes Layout – performs a grid layout but orders nodes according to a user-selected attribute, e.g., ascending order based on a numerical attribute.

Hierarchical – based on connectivity, this layout defines ordered layers of nodes in a tree structure, e.g., phylogenetic trees.

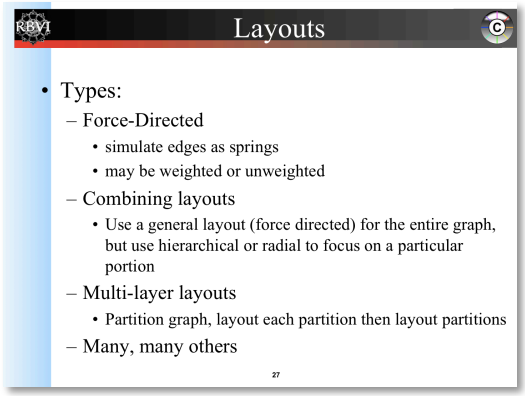
Circular Layout – arranges nodes around the circumference of a circle. The order of the nodes is arbitrary in the basic version. There two other versions: **Attribute Circle Layout**, which orders nodes based on a user-selected attribute, and **Degree Sorted Circle Layout**, which orders nodes based on their number of connections. *Pro-tip: The Degree Sorted Circle Layout calculates the degree for each node and creates a new attribute that can be used for other purposes as well, e.g., data mapping.*

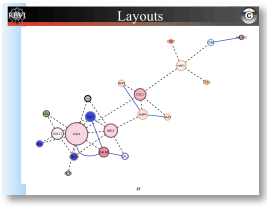
Notes

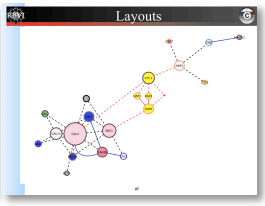
Force-Directed Layout – simulates edges as springs, resulting in clusters of highly connected nodes with minimally connected nodes spaced and in the periphery. You can also choose to influence the layout based on an edge attribute, if available. A related layout is **Spring Embedded**, which also simulates edges as springs. Both of these layouts also have explicit **Edge-Weighted-** versions that provide more control.

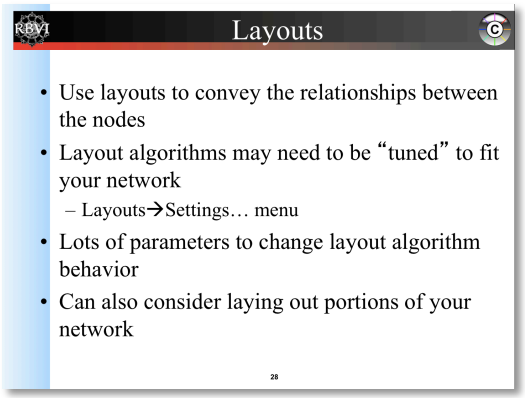
Furthermore, you can apply layouts to selected subsets of nodes. If you make a node selection prior to browsing the Layout menu, you will see an additional submenu option to apply the layout to "All nodes" or "Selected Nodes Only." By using this feature, you can effectively combine different layouts for a single network. For example, after applying a Force-Directed Layout, you could then select a connected subset and apply a Hierarchical Layout just to that set.

To achieve just the right visual layout for your network, you may need to "tune" a layout algorithm. You can do this by going to *Layout > Settings...* and then select the layout algorithm you want to tune. The settings expose the parameters of each algorithm so that you can explore different layout behaviours.









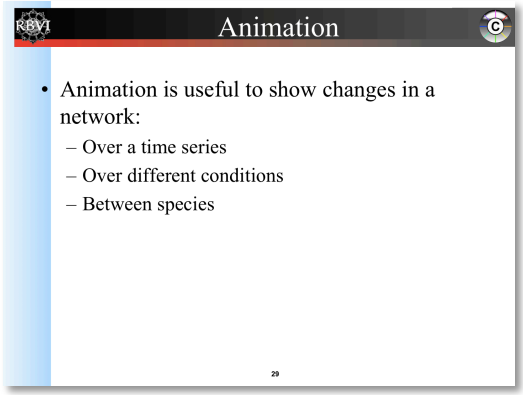
Notes

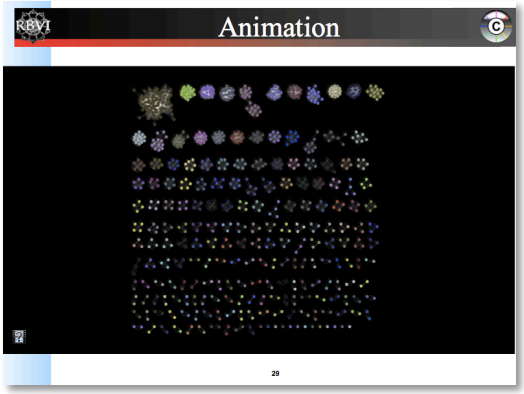
Animation

There are cases where a static image doesn't tell the whole story. Perhaps you have collected data in a time series or have more than one condition you want to compare. Animation is a key technique for visualizing change.

There are a handful of plugins for Cytoscape that support automatic, step-wise animation through defined VizMapper styles (e.g., *clusterMaker* and VistaClara). But there is also a dedicated plugin called **CyAnimator** that supports flexible animation creation.

Using CyAnimator, you choose the frames of the animation as you work. Interpolation fills-in the transitions between frames for a wide range of visual features, including position, size, color, and opacity. A final set of images are generated and ready to be imported into a number of free or commercial movie file generators.

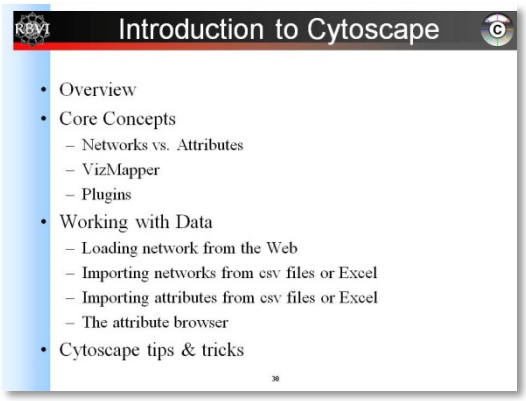


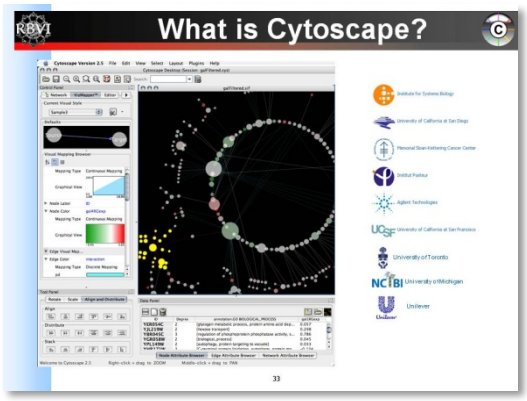


Notes

Introduction to Cytoscape

Cytoscape is an open source bioinformatics software platform for ***visualizing*** molecular interaction networks and biological pathways and ***integrating*** these networks with annotations, gene expression profiles and other state data. Although Cytoscape was originally designed for biological research, now it is a general platform for complex network analysis and visualization. Cytoscape *core* distribution provides a basic set of features for data integration and visualization. Additional features are available as ***plugins***. Plugins are available for network and molecular profiling analyses, new layouts, additional file format support, scripting, and connection with databases. Plugins may be developed by anyone using the Cytoscape open API based on Java™ technology and plugin community development is encouraged. Most of the plugins are freely available.



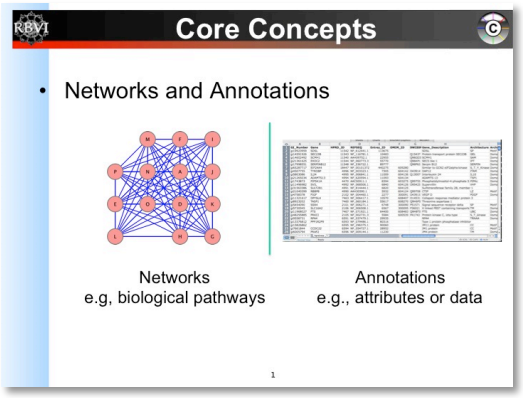


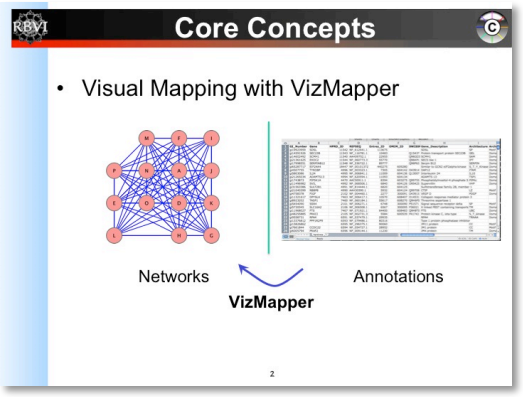
Cytoscape is a collaborative project between the Institute for Systems Biology (Leroy Hood lab), the University of California San Diego (Trey Ideker lab), Memorial Sloan-Kettering Cancer Center (Chris Sander lab), the Institut Pasteur (Benno Schwikowski lab), Agilent Technologies (Annette Adler team), the University of Toronto (Gary Bader Lab), and the David J Gladstone Institutes, University of California, San Francisco (Bruce Conklin and Tom Ferrin labs).

Notes

Core Concepts

Cytoscape creates networks, where nodes of the network represent objects (such as proteins) and connecting edges represent relationships between them (such as physical interactions). Each Edge connects two Nodes. Edges can be directed or undirected. In the case of a directed edge, there is a Source and a Target Node. Once this basic network is created, various attributes of the nodes and edges (such as protein expression levels or strength of interaction) can be added to the network and incorporated as visual cues like shape or color.



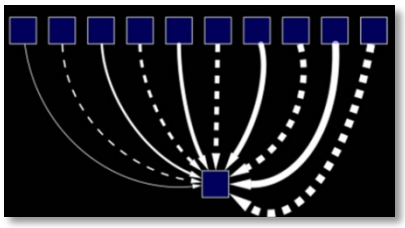


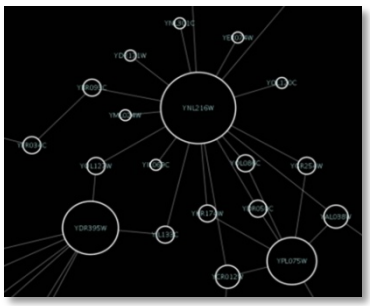
Notes

Visual Styles

One of Cytoscape's strengths in network visualization is the ability to allow users to encode any attribute of their data (name, type, degree, weight, expression data, etc.) as a visual property (such as color, size, transparency, or font type). A set of these encoded or mapped attributes is called a ***Visual Style*** and can be created or edited using the Cytoscape ***VizMapper***. With the VizMapper, the visual appearance of your network is easily customized. For example, you can:

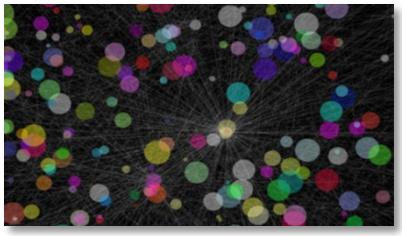
Use specific line types to indicate different types of interactions.

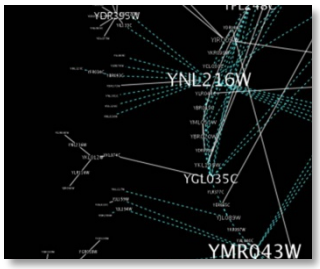




Set node sizes based on the degree of connectivity of the nodes.

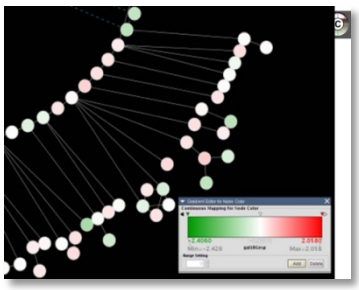
Browse extremely dense networks by controlling for the opacity of nodes.





Set node font sizes based on the degree of connectivity of the nodes.

Visualize Gene Expression data its biological context by superimposing colors onto the nodes based upon their Gene Expression data values.

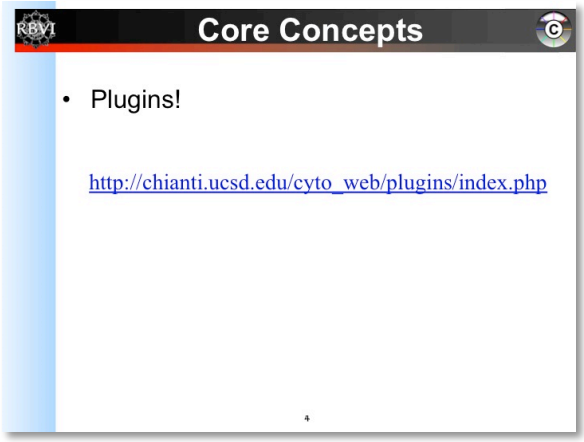


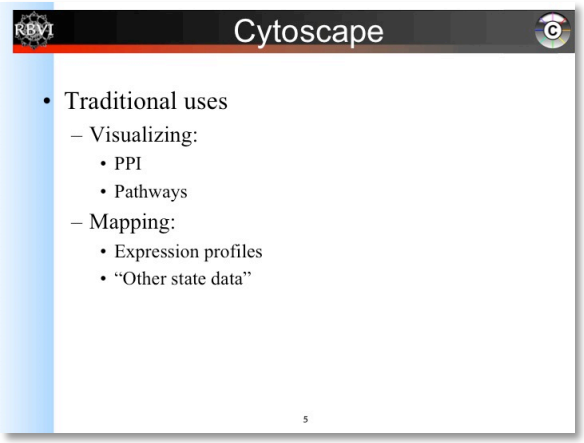
Plugins

Cytoscape allows users to extend its functionality by creating or downloading additional software modules known as "plugins". These plugins provide additional functionality in areas such as network data query and download services; network data integration and filtering; attribute-directed network layout; GO enrichment analysis[7]; as well as identification of network motifs, functional modules, proein complexes, or domain interactions.

Links to these plugins can be found at http://www.cytoscape.org/.

Altogether, Cytoscape and its plugins provide a powerful tool kit designed to help researchers answer specific biological questions using large amounts of cellular network and molecular profiling information.

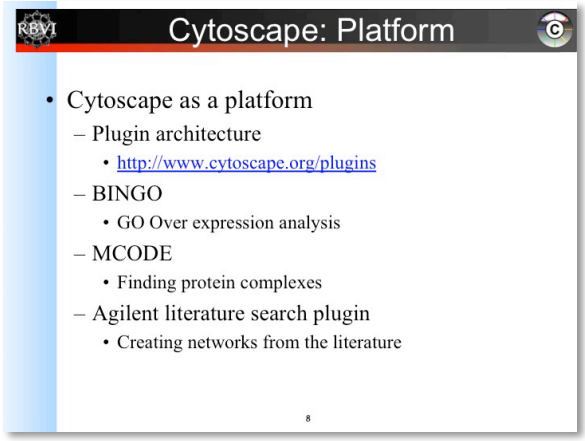




Notes

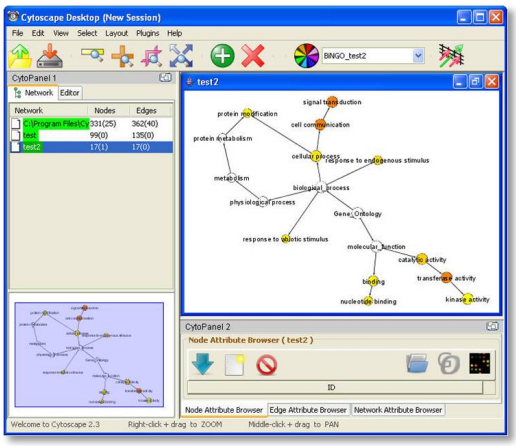
BiNGO

BiNGO is a Java-based tool to determine which Gene Ontology (GO) categories are statistically overrepresented in a set of genes or a subgraph of a biological network. BiNGO is implemented as a plugin for Cytoscape.



BiNGO maps the predominant functional themes of a given gene set on the GO hierarchy, and outputs this mapping as a Cytoscape graph. Gene sets can either be selected or computed from a Cytoscape network (as subgraphs) or compiled from sources other than Cytoscape (e.g. a list of genes that are significantly upregulated in a microarray experiment).

The main advantage of BiNGO over other GO tools is the fact that it can be used directly and interactively on molecular interaction graphs. Another plus is that BiNGO takes full advantage of Cytoscape's versatile visualization environment. This allows you to produce customized high-quality figures.

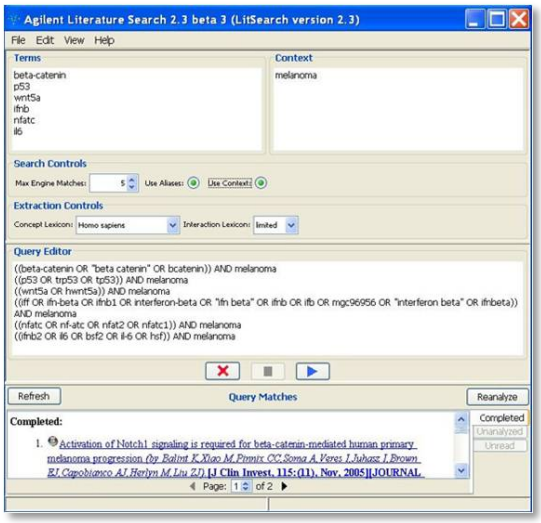


BiNGO features include:

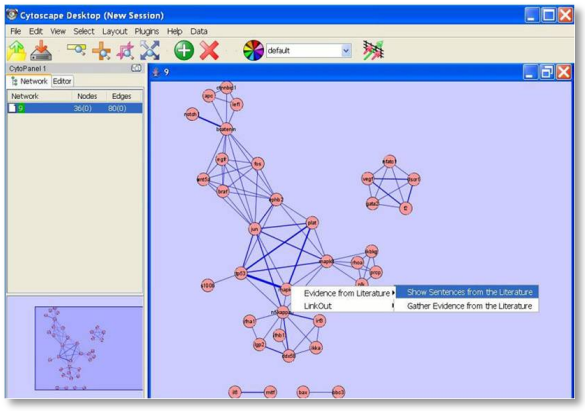
- Assessing overrepresentation or underrepresentation of GO categories
- Graph or gene list input
- Batch mode: analyze several clusters simultaneously using same settings
- Different GO and GOSlim ontologies
- Wide range of organisms
- Evidence code filtering
- Hypergeometric or binomial test for overrepresentation
- Multiple testing correction using Bonferroni (FWER) or Benjamini&Hochberg (FDR) correction
- Interactive visualization of results mapped on the GO hierarchy.
- Extensive results in tab-delimited text file format
- Ability to make and use custom annotations, ontologies and reference sets

Agilent Literature Search

Agilent Literature Search Software is a meta-search tool for automatically querying multiple text-based search engines (both public and proprietary) in order to aid biologists faced with the daunting task of manually searching and extracting associations among genes/proteins of interest.



Agilent Literature Search Software can be used in conjunction with Cytoscape, which provides a means of generating an overview network view of gene/protein associations.



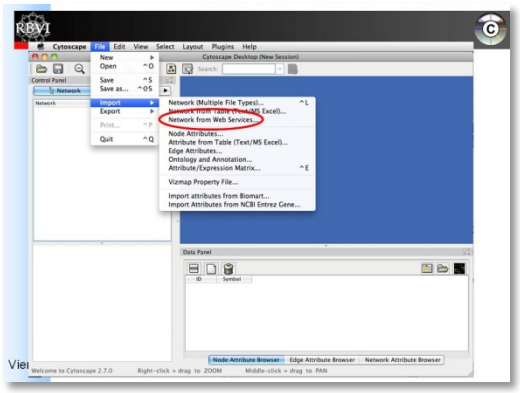
Agilent Literature Search software provides an easy-to-use interface to its powerful querying capabilities. When a query is entered, it is submitted to multiple user-selected search engines, and the retrieved results (documents) are fetched from their respective sources. Each document is then parsed into sentences and analyzed for protein-protein associations. Agilent Literature Search Software uses a set of "context" files (lexicons) for defining protein names (and aliases) and association terms (verbs) of interest. Associations extracted from these documents are collected into a Cytoscape network. The sentences and source hyperlinks for each association are further stored as attributes of the corresponding Cytoscape edges.

Agilent Literature Search Plugin Features:

- Meta-search engine combining Information Retrieval & Knowledge Extraction
- PubMed, OMIM, USPTO
- Load/Save/Reanalyze search results
- Paged Search results view
- User context-based aliasing
- File-based lexicon management
- Symbol identification, interaction extraction
- Cytoscape session load/save compatible
- Putative network generation from literature
- Literature-based evidence gathering for Cytoscape Edges
- Extend a Cytoscape network with associations extracted from the literature

Loading Networks

There are 4 different ways of creating networks in Cytoscape:

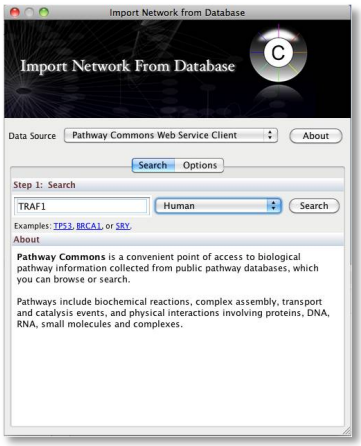


1. Importing networks from Web Service
2. Importing pre-existing, unformatted text or Excel files.
3. Importing pre-existing, formatted network files..
4. Creating an empty network and manually adding nodes and edges.

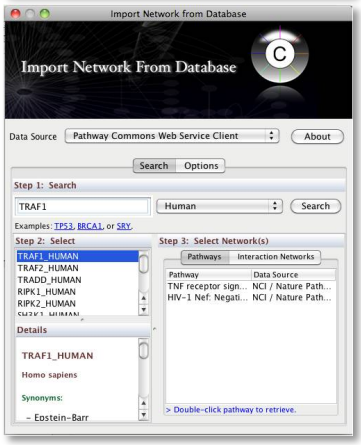
Loading Networks from a Web Service

In this section we will look at how to import networks using Web services.

First, select the **File→Import→Network** from Web Service menu item.



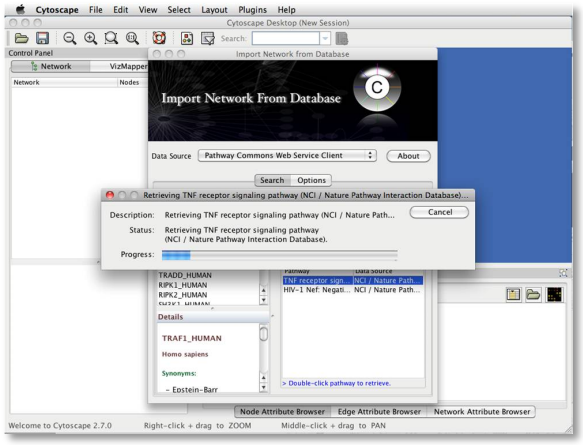
Step 1: Search. Select a Data Source and an organism. Type in a search term or set of search terms separated by commas. In this example we use the Pathway Commons Web Service Client as our Data Source, Human for Species, and enter TRAF1 as our search term.



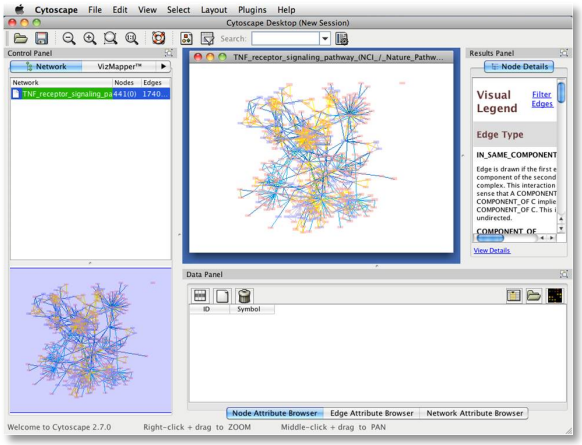
Step 2: Select. Select the protein or small molecule of interest. Full details regarding each molecule are shown in the bottom left panel.

Step 3: Select Network: Double-click on TNF receptor signaling pathway.

As the network is loading, you will see a progress dialog that indicates the percentage of the network that has been loaded.

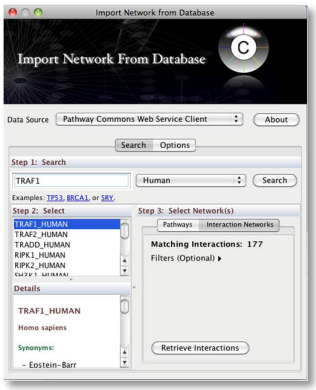


When the network has successfully loaded, you will see it displayed in the top center panel (Network View). There will also be a 'birds eye' overview in bottom-left panel that shows the entire network and a Visual Legend in the Results Panel at right, which gives details about the visual mappings for the network view.

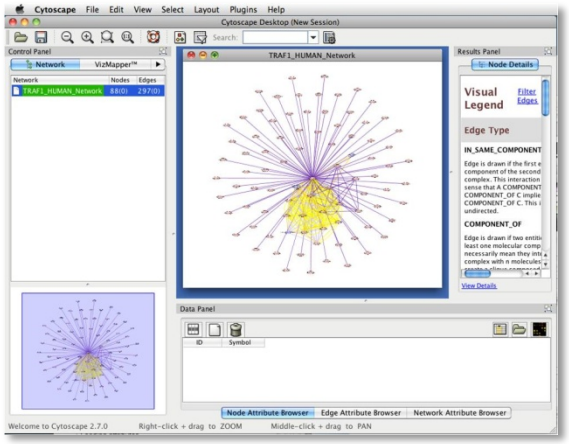


Notes

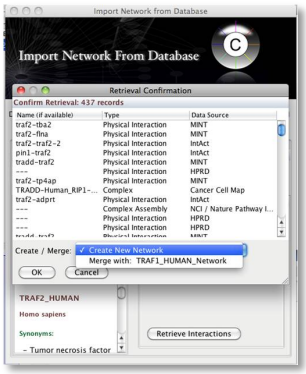
Now let's extend our network by merging in the known protein-protein interactions for TRAF1. Follow the same procedure as above, but this time select the **Interaction Networks** tab under Step 3: Select Network, then push the button labeled **Retrieve Interactions** and select **Create New Network** in the dialog box that appears.



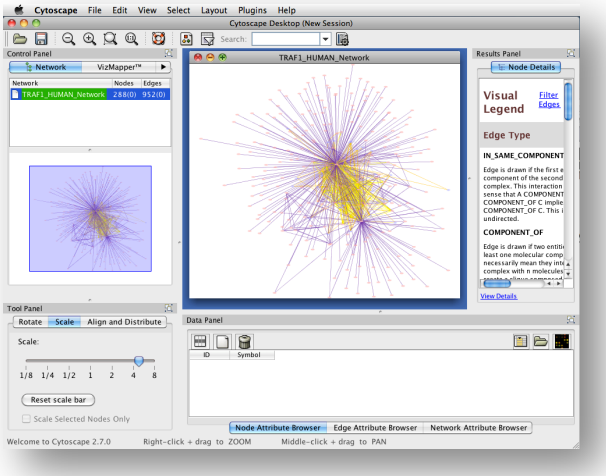
This will bring up the protein-protein interaction network for TRAF-1. Of course, this results in pretty much a star network (all nodes connect to TRAF1), so it might be interesting to expand our network by adding another of the TNF Receptor Associated Factors, TRAF2.



We follow the same procedure as above, selecting TRAF2. Now, though, when the dialog showing all of the interactions being imported comes up, we select the **Merge with TRAF1_HUMAN_Network** (the name of the network we created before) option.



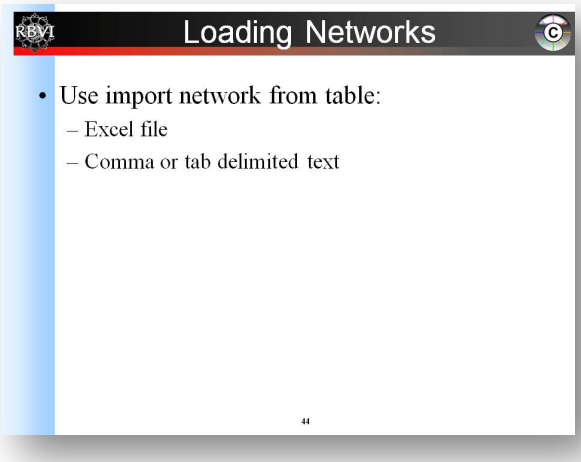
The combined TRAF1/TRAF2 protein-protein interaction network will be displayed.



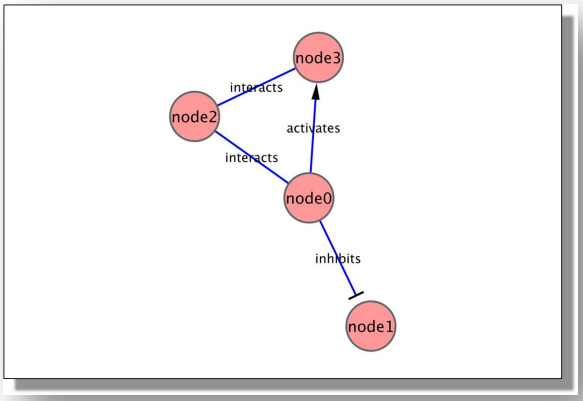
Load a Network from a Table

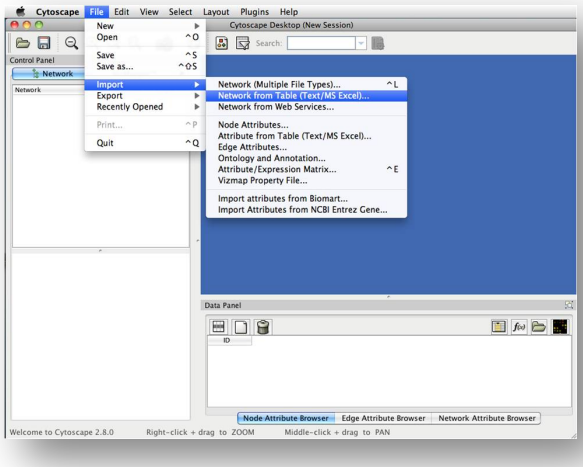
In this section we will explore how to create Cytoscape network by importing a pre-existing text or Excel file. The figure at right shows one such example network, consisting of four nodes and four edges.

Let's begin creating the network by selecting the **File→Import→Network from Table (Text/MS Excel)…** menu item.

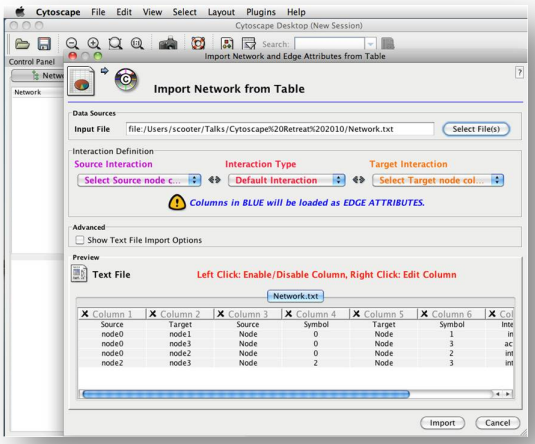


Notes

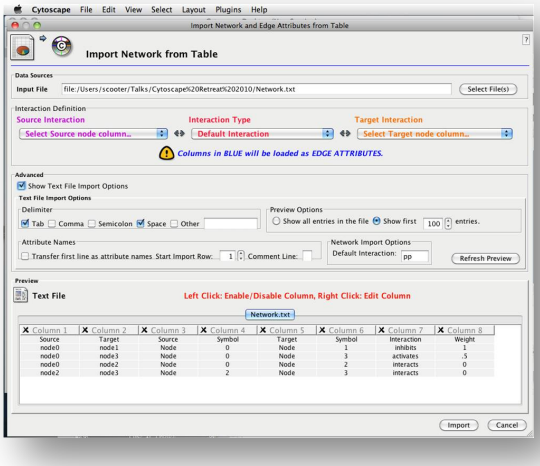




An interactive graphical user interface allows you to specify parsing options for specified files. The screen provides a preview that shows how the file will be parsed given the current configuration. As the configuration changes, the preview updates automatically. In addition to specifying how the file will be parsed, you also choose the columns that represent the Source nodes, the Target nodes, and an optional edge interaction type.

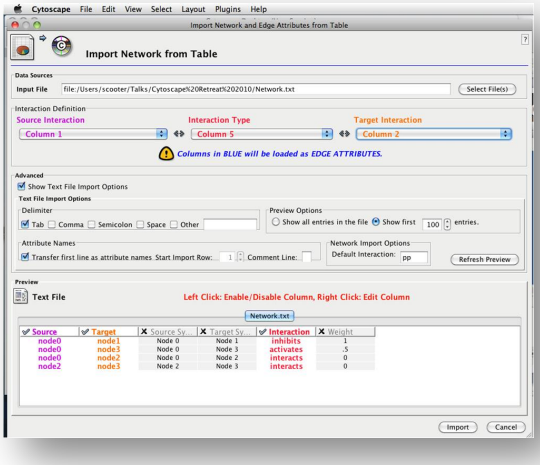


Under the **Advanced** section, check the checkbox labeled **Show Text File Import Options.**



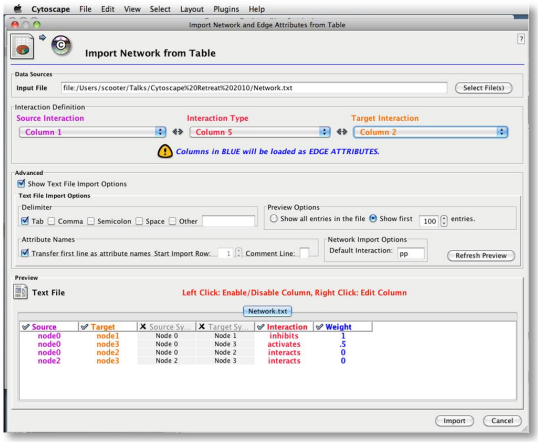
You will see a set of checkboxes appear. These allow you to choose the:

- **Delimiter**. The delimiter character that separates columns (fields) in the import file. This can be a tab, comma, semicolon, space, or any arbitrary delimiter character that you define.
- **Preview options**. This is a control for how many preview lines you see in the bottom Preview pane of the dialog. You can set this to preview all entries in the file or a subset of the entries (typically the first 100 entries).
- **Attribute Names**. You can choose whether to use the first line of the file to supply attribute names, one name per delimited column in the file.
- **Start import row**. You can set the import line number so that you can skip over any initial header or **comment** lines in the file.
- **Comment Line**. You can indicate a character, e.g. '#', to distinguish comment lines in the import files, so that they are not treated as network data.
- **Default Interaction**: You can set the name of the Default Interaction type, which is used to name an edge. The example in our figure uses 'pp' (for protein-protein interaction) as its default interaction.



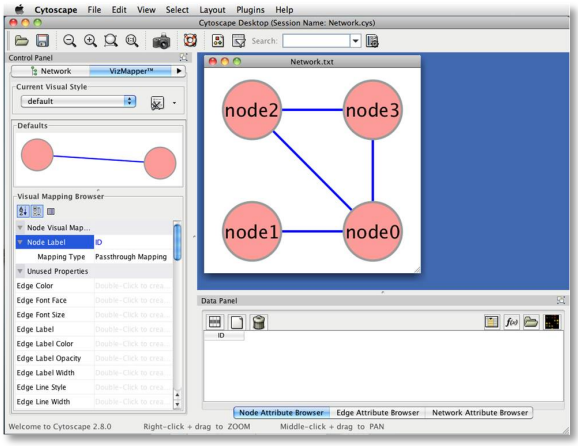
Now use the **Source Interaction**, **Interaction Type**, and **Target Interaction** combo boxes in the **Interaction Definition** to chose the columns for edge source, edge interaction type, and edge target, respectively. The figure above shows **Column 1** is being used for **Source Interaction, Column 5** for **Interaction Type**, and **Column 2** for **Target Interaction.**

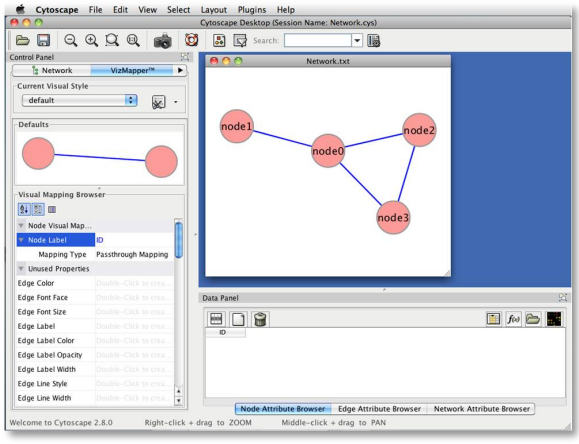
You can also use additional columns of data as edge attribute values by checking their column header. In this case the column labeled **Weight** has been checked and marked blue to indicate that its values will be loaded as edge attributes.



When you are satisfied with the settings, press the import button and the network will be imported. You will see a figure that looks like the figure on right.

The values of edge attributes can be used as arguments to graph layout and other computational operations. In the bottom example on right, the attribute for **Weight** is used in the calculation of coordinate positions using Cytoscape's **Force-Directed Layout.**





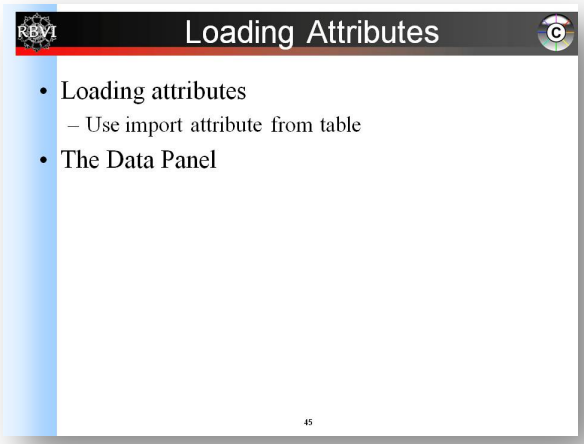
Notes

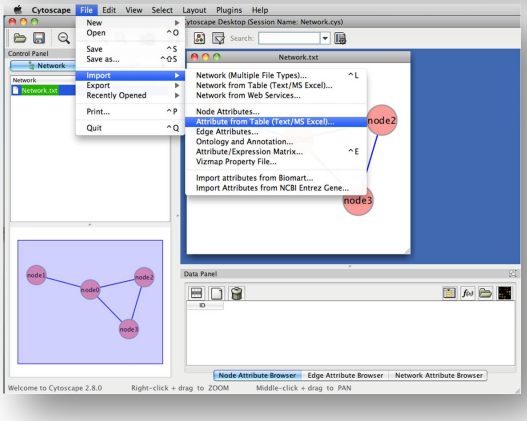
Load Attributes

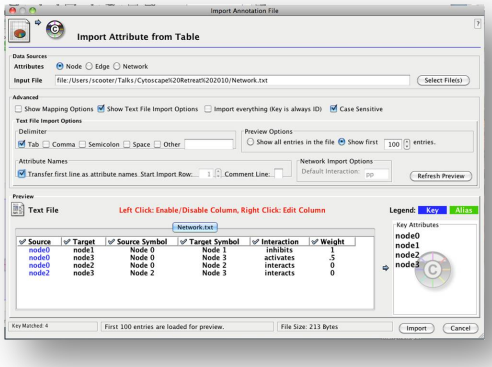
In this section we will explore how to create Cytoscape attributes and values by importing a pre-existing text or Excel file.

Let's begin creating the network by selecting the **File -> Import -> Attribute from Table (Text/MS Excel)…** menu item.

Select under **Advanced** section the **Show Mapping Options** and **Show Text File Import Options** checkboxes, which set up dialogs for controlling Text Import options (as in example with networks) and **Show Mapping Options,** which give you a number of options for associating network nodes with elements in the dataset. This enables us to encode the data as visual properties, such as color, shape, and overlay the network nodes with the values of those properties.

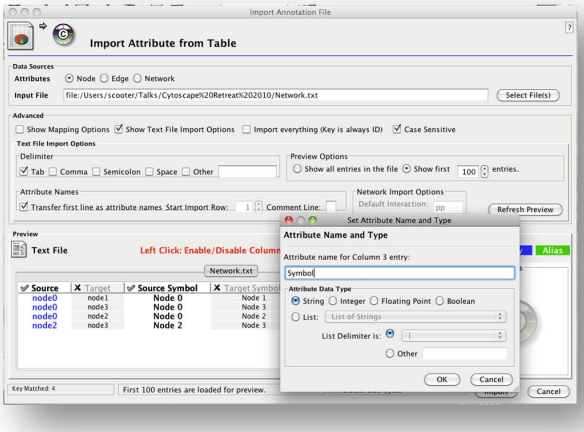






Notes

Now we are ready to map the nodes of the network to the data you have.



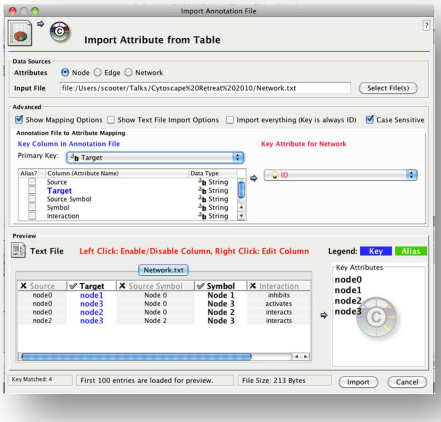
If you right-click on a column header, a dialog box will be displayed. You can fill in a name for the attribute. You can also set the type of the elements in the data column, to one of: the primitive data types that Cytoscape supports are: String , Integer, Floating Point, and Boolean. You can also set the datatype of the column to be a list of primitive elements of one datatype.

Now you need to map unique identifiers between the entries in the data and the nodes in the network. The key point of this is to identify which nodes in the network are equivalent to which entries in the table. This enables mapping of data values into visual properties like Color and Shape. This kind of mapping is typically done by comparing the unique Identifier attribute value for each node waith the unique Identifier value for each data value. As a default, Cytoscape looks for an attribute value of 'ID' in the network and a user-supplied *Primary Key* in the dataset. The user can change these values via combo boxes in the **Mapping** section:

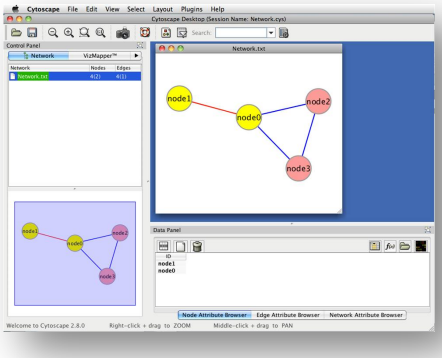
- **Primary Key:** combo box that allows you to choose the column that is to be used as key for mapping values in the dataset. You can also set an arbitrary number of columns as **aliaes** via checkbox, in which case those supplied alias will be used in addition to the Primary Key in the attempt to map identifiers.
- **Key Attribute for Network:** combo box that allow you to set the node attribute that is to be used as used as key to map to.

If there is a match between the value of a Primary Key in the dataset and and the value the Key Attribute For Network field in the network, then all attribute-value pairs associated with the element in the dataset are assigned as well to the matching node in the network.

You can control some of the options for ID Mapping by using the controls in the **Advanced** section. Select under **Advanced** section the **Show Mapping Options. Show Mapping Options,** which give you a number of options for associating network nodes with elements in the dataset. This enables us to encode the data as visual properties, such as color, shape, and overlay the network nodes with the values of those properties.

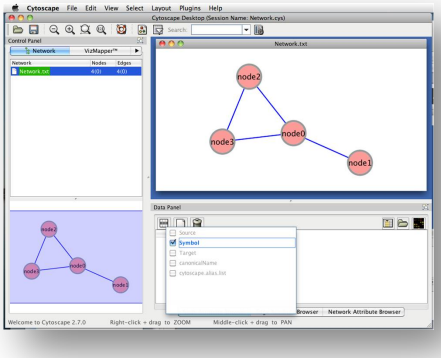


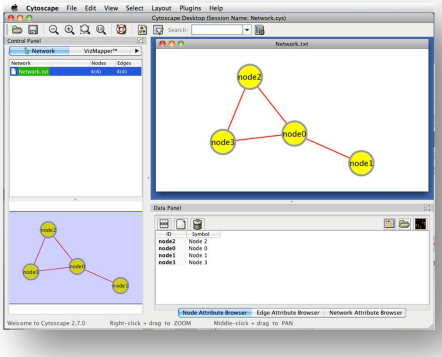
There is a checkbox entitled “Import everything (Key is always ID)”. If this is checked, Cytoscape will create an attribute value pair for every entry in the dataset, regardless of whether it maps to an equivalent attribute name. Now check **Show Mapping Options** checkbox under **Advanced** section. You can also check the **Case Sensitive** checkbox if the string comparison is to be done in a case insensitive way (e.g. if stat1 gene is the same as STAT2 gene).



Once attributes are imported, we can inspect them in the Data Panel. Right clicking on the leftmost icon in the toolbar under DataPanel title bar will bring up a menu of checkboxes. If you check a checkbox for any attribute in that attribute and its values will be displayed as a column in of values on the DataPanel.

If you a select a node in the network, its attribute values will be displayed in the control panel.

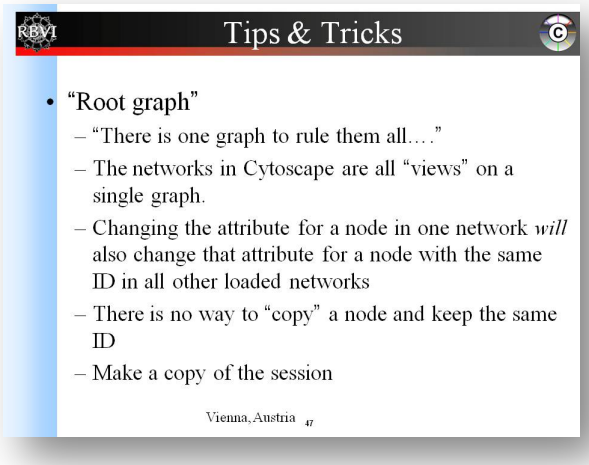




Notes

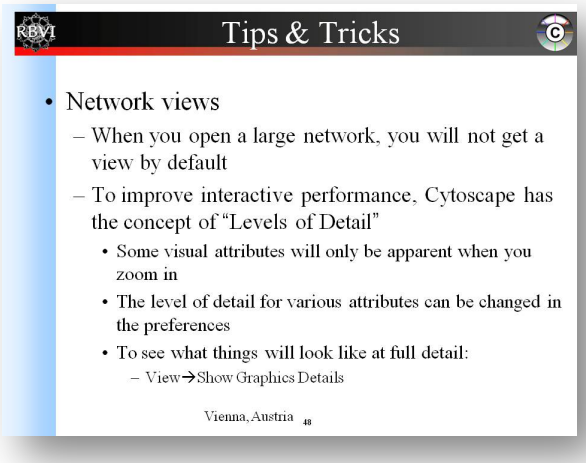
Tips and Tricks

Cytoscape is a large, complex, and dynamic software system. A little knowledge of the internals organization and operational model of the software will enable more efficient use of the software. Here are some useful Tips & Tricks to help you get the most out of your Cytoscape usage.



The ""Root Graph"

There is one central *root graph* that contains all nodes and edges. Thus all networks are 'views' on that single graph, and nodes and edges are unique across all networks. Modifying a node in one network will modify that node in all other networks that it appears in. There is no way to have two or more copies of a node with the same ID. The only workaround would be to make a copy of a Cytoscape session.



Network Views

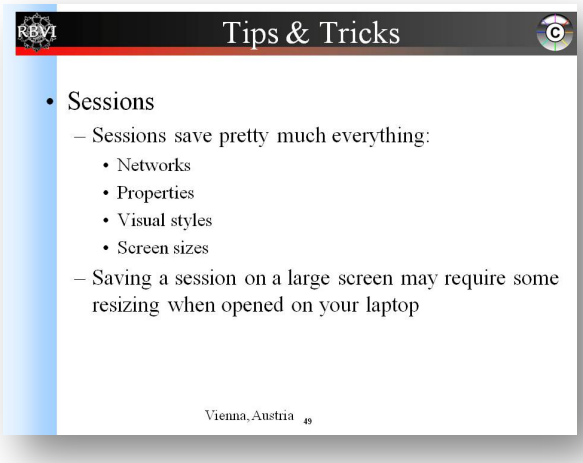
For efficiency in dealing with large networks, a view is not automatically generated when the size of the network is over a user-definable threshold. You can manually generate a Network View by right-clicking on its entry in the Network Navigator Panel (upper left o Cytoscape desktop)., then selecting 'Create View'. You can also use that right-menu item to 'Destroy View', 'Destroy Network", and edit the Network's title.

To improve interactive performance, Cytoscape has the concept of *Levels of Detail*? This is basically a mechanism for *semantic zooming*, where different levels of detail com into play at different levels of detail (think of the Google Maps interface where a City is represented by a yellow patch at high level then shows more of the structure of streets and avenues as you zoom in.

Some Cytoscape attributes will only be apparent when you zoom in. The level of detail for various attributes can be changed in the preferences. To see what things look like in full detail, select the **View→Show Graphics Details** menu item..

Sessions

Sessions save pretty much everything: Networks, Properties. Visual styles, Screen sizes, and many other types of information. When working on a complex study of workflow, it is often prudent to save one's intermediate results as a session, so that the current state of an activity is persisted and can be resumed without having to repeat earlier low-level operations. Not all state is the same, however. For example, saving a session on a large screen may require some resizing when re-opened.

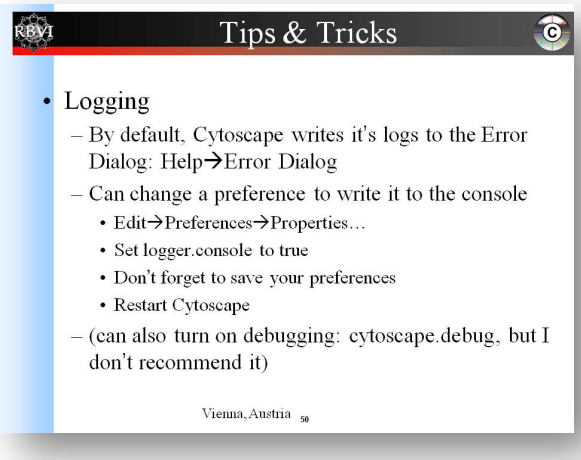


Logging

Logging can help you get to the bottom of operations that have gone awry. . By default, Cytoscape writes its logs to the Error Dialog: via the **Help → Error Dialog** menu item.

You can change a preference to write the log to the console via:

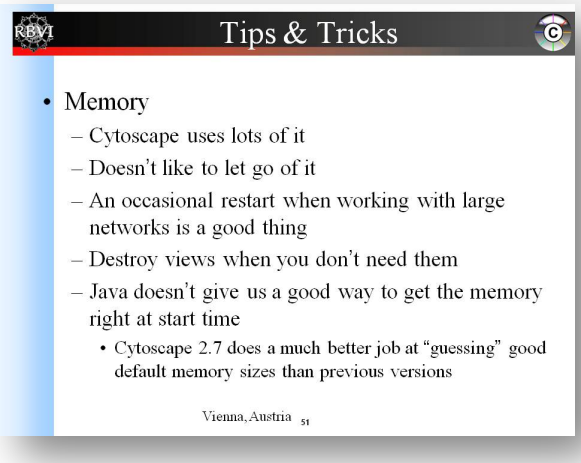
Edit→ Preferences→ Properties… menu item. To do this, set the Set the *logger.console* property to *true*. Don't forget to save your preferences. Then you can restart Cytoscape.



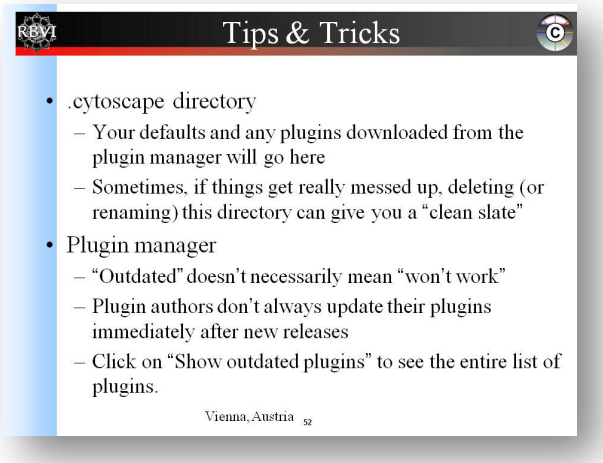
Memory

Cytoscape uses a lot of memory and, as a Java system, doesn't like to let go of it. When working with large networks, an occasional save session and restart will help clear out memory. Another efficiency measure is to destroy large network views when not needed.

One particular challenge is setting virtual memory sizes correctly upon startup. Java does not provide very good ways to do this, although Cytoscape from version 2.7 has become better at "guessing" good default memory sizes than previous versions.



Final points on Tips and Tricks



- .cytoscape directory
  - This directory is typically located under your home directory, for example on a Windows system it will be under

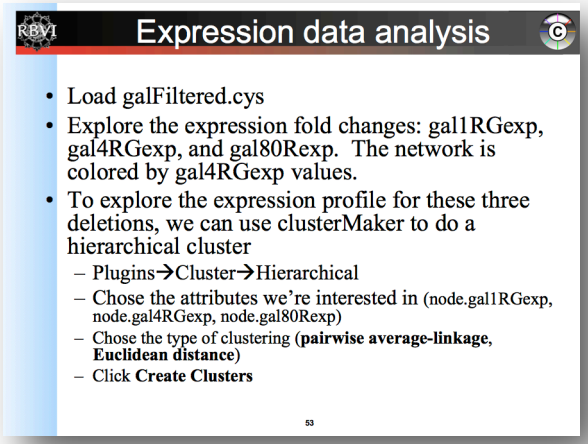
    C:\Documents and Settings\<username>
  - Your defaults and any plugins downloaded from the plugin manager will go in this directory. Also, plugins may use this directory to store configuration
  - Sometimes,, if things get really messed up, deleting (or renaming) this directory can give you a "clean slate"
- Plugin manager
  - When a plugin is labeled as "Outdated", it doesn't necessarily mean "won't work"
  - Plugin authors don't always update their plugins immediately after new releases
  - Click on "Show outdated plugins" to see the entire list of plugins.

Notes

Demo/Sample use cases

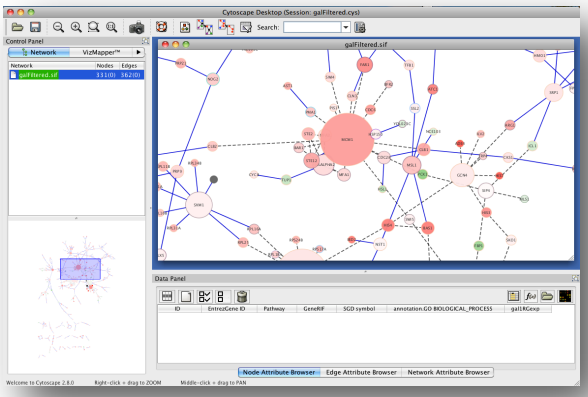
Use case 1: Expression data analysis

This use case highlights the visual display of expression data, integrated clustering features, and basic Gene Ontology overrepresentation analysis. *Note: we are starting with an expression dataset that has already been normalized, statistically analyzed, formatted, imported and associated with an interaction network.*



The dataset

Differential gene expression of GAL deletion mutants grown in the presence and absence of galactose[122]. Fold values were mapped onto a protein-protein interaction network focusing on galactose utilization in yeast, *Saccharomyces cerevisiae*. Additional annotations (e.g., gene and protein identifiers, GO terms, pathway associations, etc) have also been added as node attributes. This dataset is included with the download of Cytoscape in the sampleData folder and is called **galFiltered.cys**.

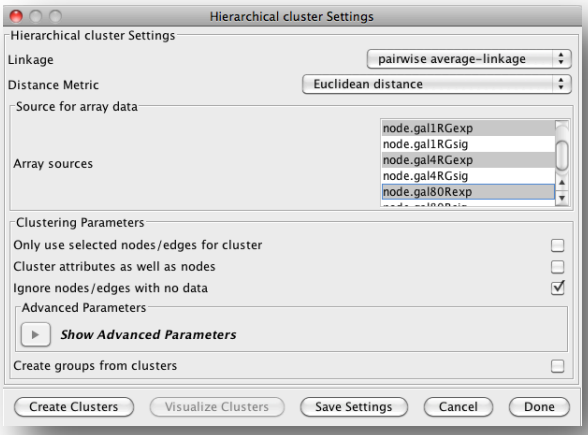


Locate and open the galFiltered.cys session file. Check out the VizMapper settings for node color. By default, the nodes are colored by the fold value **gal4RGexp.** Explore the visualization of other fold values in the dataset: **gal1RGexp** and **gal80Rexp**.

Notes

Cluster analysis

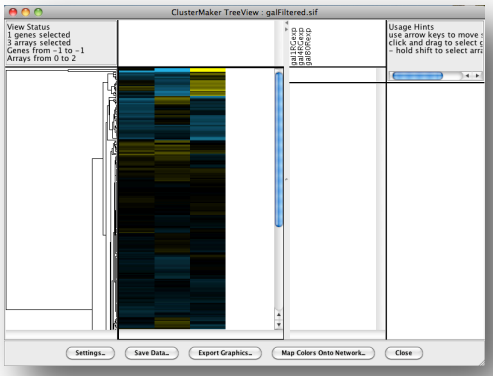
To explore the expression profiles for the three deletions, we can perform clustering within Cytoscape using the **clusterMaker** plugin.



In the Plugins menu, select Cluster > Hierarchical.

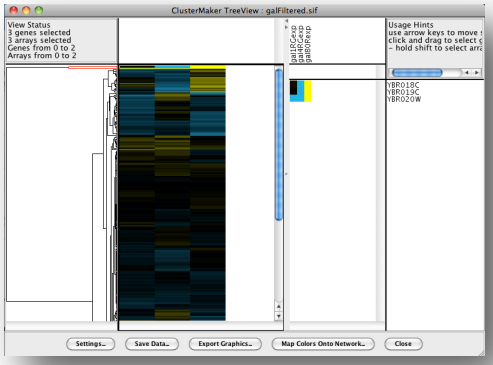
- Choose the type of clustering:
  - pairwise average-linkage
- Choose the attributes of array data:
  - node.gal1RGexp
  - node.gal4RGexp
  - node.gal80Rexp

- Click: Create Clusters
- When done, click: Visualize Clusters



This will bring up the TreeView of your cluster results. Each row is a gene and the three columns correspond to the three data attributes. A dendrogram to the left expresses the relationship between clusters, and the region to the right shows a close-up and labeled view of selected rows.

If the colors are too dark, or if you prefer other colors altogether, you can open Settings… and adjust a number of preferences.



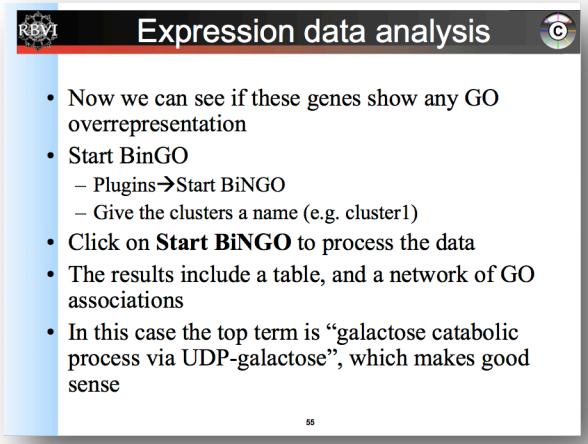
Now, select the top most branch of the dendrogram, as shown on the right. *Notice that selections in TreeView correspond to selections in the network!*

Notes

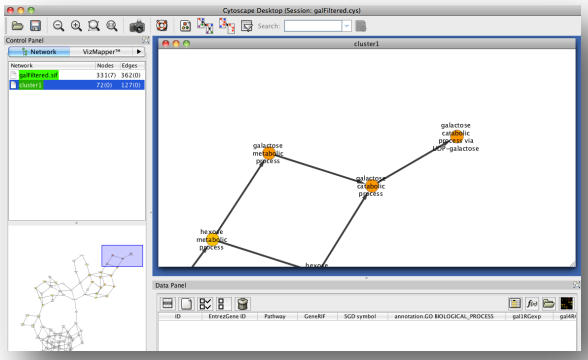
GO term overrepresentation analysis

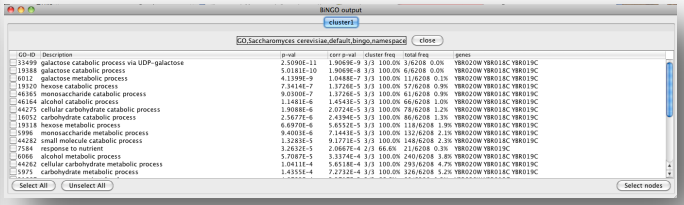
Now we can see if any of the selected genes from that first cluster show any GO term overrepresentation. In other words, are there particular GO terms that are enriched (or overrepresented) in this subset of genes? We can do this using the **BiNGO** plugin.

- Plugins > Start BiNGO
- Give the cluster a name
- Click: Start BiNGO
- *Note: there are many parameters you can play with. The defaults are usually sufficient for a first pass as major trends.*



The results are displayed as a table and as a network of GO associations. The GO terms are connected based on their inherent hierarchical relationship and they are colored based on the significance of their overrepresentation in your cluster.

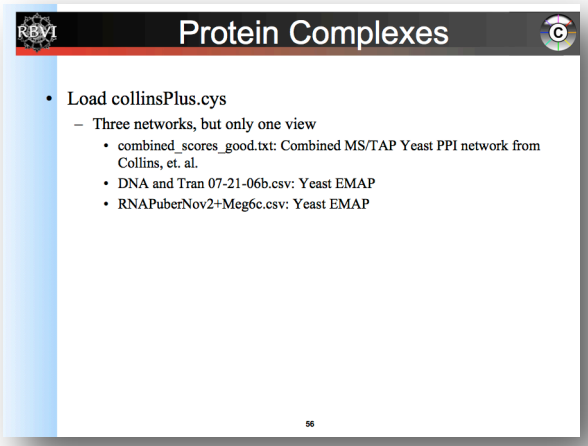




Notes

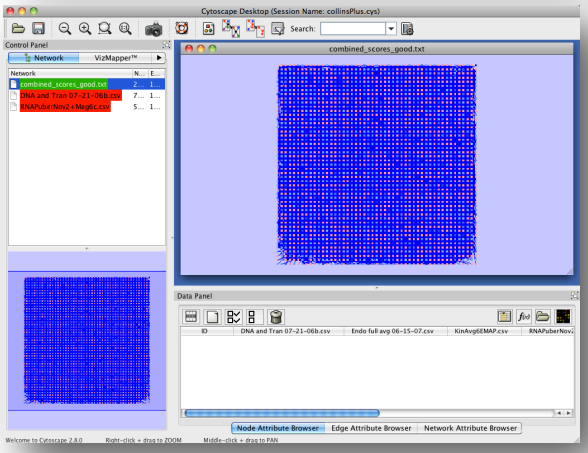
Use case 2: Protein complexes in protein-protein interaction networks

This use case highlights the combined use of MCL clustering of protein-protein interaction (PPI) networks and hierarchical clustering of epistatic mini-array profile (EMAP) data to explore potential biological protein complexes.



The dataset

We will be working with a Cytoscape session file containing three networks: one is a yeast PPI[27] and the other two are yeast EMAP datasets [7, 123]. *Note: we will not bother viewing the EMAP datasets as networks, but rather treat them as sets of nodes and attributes. You can perform clustering on sets of nodes without creating a network view!* The key to making this analysis work is having the same node identifiers in both the PPI and EMAPs.



The dataset is provided with this tutorial and is called **collinsPlus.cys**:

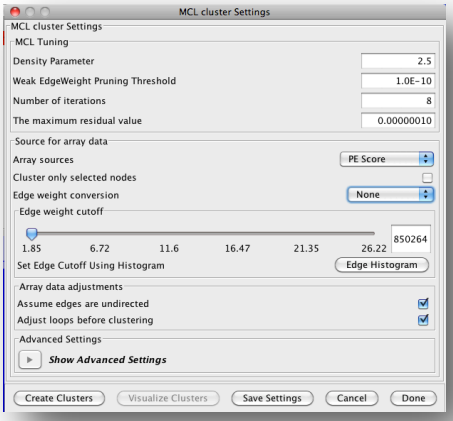
- combined_scores_good.txt (PPI)
- DNA and Tran 07-21-06b.csv (EMAP)
- RNAPuberNov2+Meg6c.csv (EMAP)

Notes

MCL clusters in the PPI network

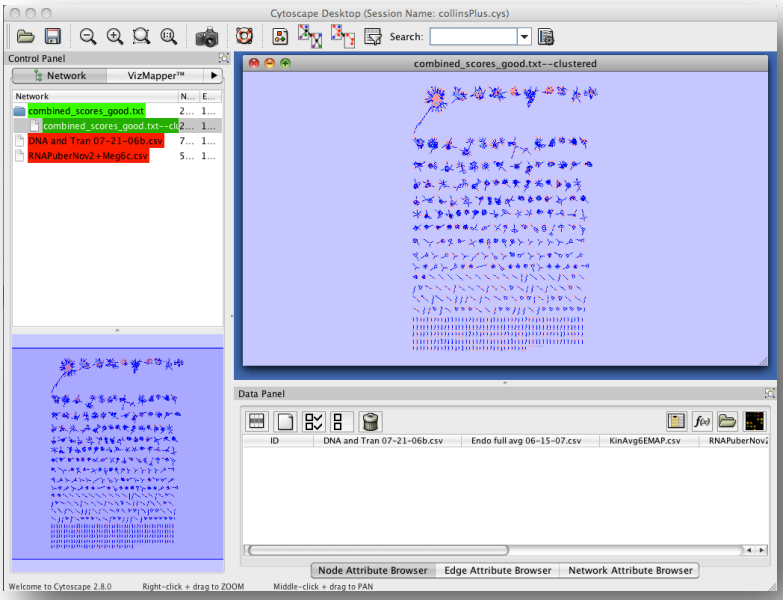
First, we will identify the MCL clusters in the protein-protein interaction network. Under the Plugins menu, choose Cluster and then MCL cluster:

- Density Parameter: 1.8
- Weak Edge Weight Pruning Threshold: 1.0E-10
- Maximum residual value: 0.00000010
- Array source: PE Score
- Click: Create Clusters
- When done, click: Visualize Clusters



There are your MCL clusters. Beautiful, aren't they! These are our first approximation of potential protein complexes based solely on tightly interacting protein clusters.

Next, we consider the clusters generated from EMAP data as an orthogonal form of evidence based on genetic interactions. Combining both cluster results provides a more complete picture.

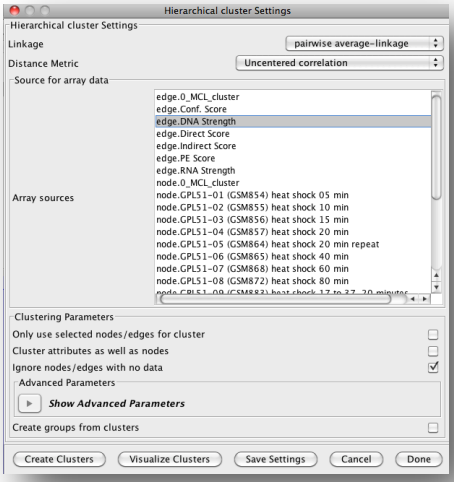


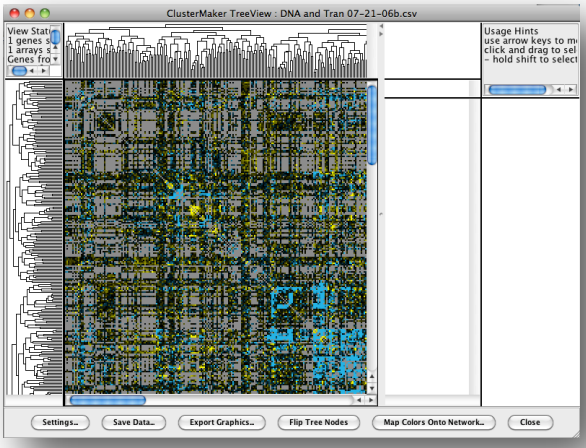
Notes

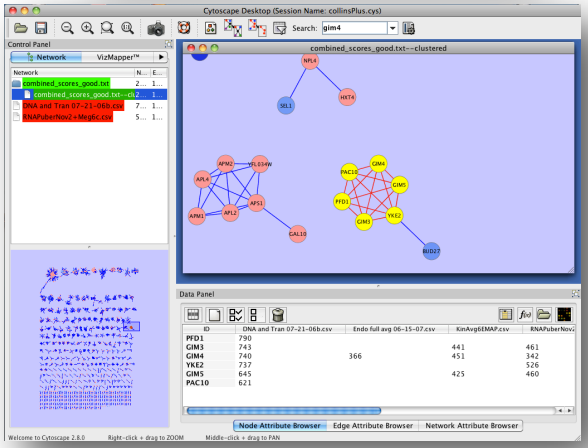
Hierarchical clustering of EMAP data

Select the "DNA and Tran…" dataset in the Network panel on the left. *Note: the red highlight simply indicates that no network view has been created. No problem.* Once again, go to Plugins > Cluster > Hierarchical Cluster:

- Linkage: pairwise average-linkage
- Distance Metric: Uncentered correlation
- Array sources: edge.DNA Strenth
- Create Clusters, then Visualize Clusters







The EMAP clusters identify potential complexes based on genetic (functional) interactions. Now, we can explore the correspondence of evidence from these two methods. For example, search for GIM5 and select the entire cluster. Notice how the corresponding interactions are dynamically highlighted in the TreeView. Notice how both EMAP and PPI data do not provide strong support for the inclusion of BUD27 in this potential complex.

Notes

Hands-on tutorial: Introduction to Cytoscape

This tutorial will cover:

1. Navigating Cytoscape
2. Visualizing Data on Networks
3. Network and Pathway Resources
4. Plugin Manager
5. Plugin Demos

By the end of the tutorial, you should be able to use Cytoscape to import networks and attributes and visualize those attributes on the network.

Notes

Hands-on tutorial: Working with data

This tutorial will introduce you to:

1. Searching Internet interaction databases with query terms.
2. Mapping Identifiers of different types to networks.
3. Finding your query terms in the downloaded network.

The second half of the tutorial will introduce you to some advanced basics in Cytoscape:

1. Apply filters to filter out low-confidence edges.
2. Perform basic edits using the Cytoscape graph editor.

Notes

Hands-on tutorial: Analysis of microarray data

This tutorial will introduce you to:

1. Combining data from two different sources: experimental data in the form of microarrray expression data and network data in the form of interaction data.
2. Visualizing networks using expression data.
3. Filtering networks based on expression data.

NOTE: The expression data used in this example has been pre-processed to work with the interaction network used.

Notes

Bibliography

1. Ideker T, Ozier O, Schwikowski B, Siegel AF: **Discovering regulatory and signalling circuits in molecular interaction networks**. *Bioinformatics* 2002, **18 Suppl 1**:S233-240.
2. Breitling R, Amtmann A, Herzyk P: **Graph-based iterative Group Analysis enhances microarray interpretation**. *BMC Bioinformatics* 2004, **5**:100.
3. Bandyopadhyay S, Kelley R, Ideker T: **Discovering regulated networks during HIV-1 latency and reactivation**. *Pac Symp Biocomput* 2006:354-366.
4. Qiu YQ, Zhang S, Zhang XS, Chen L: **Detecting disease associated modules and prioritizing active genes based on high throughput data**. *BMC Bioinformatics* 2010, **11**:26.
5. Bader GD, Hogue CW: **An automated method for finding molecular complexes in large protein interaction networks**. *BMC Bioinformatics* 2003, **4**:2.
6. Bandyopadhyay S, Kelley R, Krogan NJ, Ideker T: **Functional maps of protein complexes from quantitative genetic interaction data**. *PLoS computational biology* 2008, **4**(4):e1000065.
7. Collins SR, Miller KM, Maas NL, Roguev A, Fillingham J, Chu CS, Schuldiner M, Gebbia M, Recht J, Shales M *et al*: **Functional dissection of protein complexes involved in yeast chromosome biology using a genetic interaction map**. *Nature* 2007, **446**(7137):806-810.
8. Krogan NJ, Cagney G, Yu H, Zhong G, Guo X, Ignatchenko A, Li J, Pu S, Datta N, Tikuisis AP *et al*: **Global landscape of protein complexes in the yeast Saccharomyces cerevisiae**. *Nature* 2006, **440**(7084):637-643.
9. Vlasblom J, Wu S, Pu S, Superina M, Liu G, Orsi C, Wodak SJ: **GenePro: a Cytoscape plug-in for advanced visualization and analysis of interaction networks**. *Bioinformatics* 2006, **22**(17):2178-2179.
10. Florez AF, Park D, Bhak J, Kim BC, Kuchinsky A, Morris JH, Espinosa J, Muskus C: **Protein network prediction and topological analysis in Leishmania major as a tool for drug target selection**. *BMC Bioinformatics* 2010, **11**:484.
11. Bansal M, Della Gatta G, di Bernardo D: **Inference of gene regulatory networks and compound mode of action from time course gene expression profiles**. *Bioinformatics* 2006, **22**(7):815-822.
12. Emig D, Salomonis N, Baumbach J, Lengauer T, Conklin BR, Albrecht M: **AltAnalyze and DomainGraph: analyzing and visualizing exon expression data**. *Nucleic acids research* 2010, **38**(Web Server issue):W755-762.
13. Liu S, Zhang C, Zhou Y: **Domain graph of Arabidopsis proteome by comparative analysis**. *J Proteome Res* 2005, **4**(2):435-444.
14. Kann MG, Jothi R, Cherukuri PF, Przytycka TM: **Predicting protein domain interactions from coevolution of conserved regions**. *Proteins* 2007, **67**(4):811-820.
15. Vailaya A, Bluvas P, Kincaid R, Kuchinsky A, Creech M, Adler A: **An architecture for biological information extraction and representation**. *Bioinformatics* 2005, **21**(4):430-438.
16. Kohler S, Bauer S, Horn D, Robinson PN: **Walking the interactome for prioritization of candidate disease genes**. *Am J Hum Genet* 2008, **82**(4):949-958.
17. Kann MG: **Protein interactions and disease: computational approaches to uncover the etiology of diseases**. *Brief Bioinform* 2007, **8**(5):333-346.

18. Mohammad Shafkat Amin AB, Russel L. Finley, Jr., Hasan Jamin: **A stochastic approach to candidate disease gene subnetwork extraction**. In: *2010 ACM Symposium on Applied Computing*. 2010.
19. Dobrin R, Zhu J, Molony C, Argman C, Parrish ML, Carlson S, Allan MF, Pomp D, Schadt EE: **Multi-tissue coexpression networks reveal unexpected subnetworks associated with disease**. *Genome biology* 2009, **10**(5):R55.
20. King JY, Ferrara R, Tabibiazar R, Spin JM, Chen MM, Kuchinsky A, Vailaya A, Kincaid R, Tsalenko A, Deng DX *et al*: **Pathway analysis of coronary atherosclerosis**. *Physiol Genomics* 2005, **23**(1):103-118.
21. Chuang H, Ressenti, L., Ideker, T., Kipps, T.: **Interactome-based modeling and diagnosis of Chrinic Lymphocytic Leukemia**. In: *50th Annual Meeting of the American Society of Hematology*. 2008.
22. Chuang HY, Lee E, Liu YT, Lee D, Ideker T: **Network-based classification of breast cancer metastasis**. *Mol Syst Biol* 2007, **3**:140.
23. Mileyko Y, Joh RI, Weitz JS: **Small-scale copy number variation and large-scale changes in gene expression**. *Proceedings of the National Academy of Sciences of the United States of America* 2008, **105**(43):16659-16664.
24. Chen L, Xuan J, Wang Y, Hoffman EP, Riggins RB, Clarke R: **Identification of condition-specific regulatory modules through multi-level motif and mRNA expression analysis**. *Int J Comput Biol Drug Des* 2009, **2**(1):1-20.
25. McLendon R, et. al.: **Comprehensive genomic characterization defines human glioblastoma genes and core pathways**. *Nature* 2008, **455**(7216):1061-1068.
26. Moraru, II, Schaff JC, Slepchenko BM, Blinov ML, Morgan F, Lakshminarayana A, Gao F, Li Y, Loew LM: **Virtual Cell modelling and simulation software environment**. *IET Syst Biol* 2008, **2**(5):352-362.
27. Collins SR, Kemmeren P, Zhao XC, Greenblatt JF, Spencer F, Holstege FC, Weissman JS, Krogan NJ: **Toward a comprehensive atlas of the physical interactome of Saccharomyces cerevisiae**. *Mol Cell Proteomics* 2007, **6**(3):439-450.
28. Keiser MJ, Roth BL, Armbruster BN, Ernsberger P, Irwin JJ, Shoichet BK: **Relating protein pharmacology by ligand chemistry**. *Nat Biotechnol* 2007, **25**(2):197-206.
29. Atkinson HJ, Morris JH, Ferrin TE, Babbitt PC: **Using sequence similarity networks for visualization of relationships across diverse protein superfamilies**. *PLoS One* 2009, **4**(2):e4345.
30. Yildirim MA, Goh KI, Cusick ME, Barabasi AL, Vidal M: **Drug-target network**. *Nat Biotechnol* 2007, **25**(10):1119-1126.
31. Enright AJ, Van Dongen S, Ouzounis CA: **An efficient algorithm for large-scale detection of protein families**. *Nucleic Acids Res* 2002, **30**(7):1575-1584.
32. Scheeff ED, Bourne PE: **Structural evolution of the protein kinase-like superfamily**. *PLoS computational biology* 2005, **1**(5):e49.
33. Jaccard P: **Distribution de la flore alpine dans le bassin des Dranses et dans quelques régions voisines.** *Bulletin del la Société Vaudoise des Sciences Naturelles* 1901, **37**:241-272.
34. Tanimoto TT. In: *IBM Internal Report.* 1957.
35. Tversky A: **Features of similarity**. *Psychological Review* 1977, **84**(4):327-352.
36. **Daylight Theory: Fingerprints** [http://www.daylight.com/dayhtml/doc/theory/theory.finger.html]

37. Altschul SF, Gish W, Miller W, Myers EW, Lipman DJ: **Basic local alignment search tool**. *Journal of molecular biology* 1990, **215**(3):403-410.
38. Altschul SF, Madden TL, Schaffer AA, Zhang J, Zhang Z, Miller W, Lipman DJ: **Gapped BLAST and PSI-BLAST: a new generation of protein database search programs**. *Nucleic acids research* 1997, **25**(17):3389-3402.
39. Smith TF, Waterman MS: **Identification of common molecular subsequences**. *Journal of molecular biology* 1981, **147**(1):195-197.
40. Shindyalov IN, Bourne PE: **Protein structure alignment by incremental combinatorial extension (CE) of the optimal path**. *Protein Eng* 1998, **11**(9):739-747.
41. Holm L, Sander C: **Mapping the protein universe**. *Science* 1996, **273**(5275):595-603.
42. Taylor WR, Flores TP, Orengo CA: **Multiple protein structure alignment**. *Protein Sci* 1994, **3**(10):1858-1870.
43. Ilyin VA, Abyzov A, Leslin CM: **Structural alignment of proteins by a novel TOPOFIT method, as a superimposition of common volumes at a topomax point**. *Protein Sci* 2004, **13**(7):1865-1874.
44. Kolbeck B, May P, Schmidt-Goenner T, Steinke T, Knapp EW: **Connectivity independent protein-structure alignment: a hierarchical approach**. *BMC Bioinformatics* 2006, **7**:510.
45. Guerler A, Knapp EW: **Novel protein folds and their nonsequential structural analogs**. *Protein Sci* 2008, **17**(8):1374-1382.
46. Bausch P, Bumgardner, J.: **Make a Flickr-Style Tag Cloud**. In: *Flickr Hacks.* O'Reilly Press.; 2006.
47. **ISO/IEC JTC1/SC34/WG3** [http://www.isotopicmaps.org/]
48. Pegg SC, Brown SD, Ojha S, Seffernick J, Meng EC, Morris JH, Chang PJ, Huang CC, Ferrin TE, Babbitt PC: **Leveraging enzyme structure-function relationships for functional inference and experimental design: the structure-function linkage database**. *Biochemistry* 2006, **45**(8):2545-2555.
49. **Graph theory** [http://en.wikpedia.org/wiki/Graph_theory]
50. Bondy JA, Murty, U.S.R.: **Graph Theory with Applications**. In.
51. Barabasi AL, Oltvai ZN: **Network biology: understanding the cell's functional organization**. *Nat Rev Genet* 2004, **5**(2):101-113.
52. Jeong H, Mason SP, Barabasi AL, Oltvai ZN: **Lethality and centrality in protein networks**. *Nature* 2001, **411**(6833):41-42.
53. Jeong H, Tombor B, Albert R, Oltvai ZN, Barabasi AL: **The large-scale organization of metabolic networks**. *Nature* 2000, **407**(6804):651-654.
54. Fell DA, Wagner A: **The small world of metabolism**. *Nat Biotechnol* 2000, **18**(11):1121-1122.
55. Ma HW, Zeng AP: **The connectivity structure, giant strong component and centrality of metabolic networks**. *Bioinformatics* 2003, **19**(11):1423-1430.
56. Maslov S, Sneppen K: **Specificity and stability in topology of protein networks**. *Science* 2002, **296**(5569):910-913.
57. Rzhetsky A, Gomez SM: **Birth of scale-free molecular networks and the number of distinct DNA and protein domains per genome**. *Bioinformatics* 2001, **17**(10):988-996.
58. Wuchty S: **Scale-free behavior in protein domain networks**. *Mol Biol Evol* 2001, **18**(9):1694-1702.

59. Tong AH, Lesage G, Bader GD, Ding H, Xu H, Xin X, Young J, Berriz GF, Brost RL, Chang M *et al*: **Global mapping of the yeast genetic interaction network**. *Science* 2004, **303**(5659):808-813.
60. van Noort V, Snel B, Huynen MA: **The yeast coexpression network has a small-world, scale-free architecture and can be explained by a simple model**. *EMBO Rep* 2004, **5**(3):280-284.
61. Featherstone DE, Broadie K: **Wrestling with pleiotropy: genomic and topological analysis of the yeast gene expression network**. *Bioessays* 2002, **24**(3):267-274.
62. Agrawal H: **Extreme self-organization in networks constructed from gene expression data**. *Phys Rev Lett* 2002, **89**(26):268702.
63. Khanin R, Wit, E.: **How Scale-Free Are Biological Networks**. *Journal of Computational Biology* 2006, **13**(3):810-818.
64. Albert R, Jeong H, Barabasi AL: **Error and attack tolerance of complex networks**. *Nature* 2000, **406**(6794):378-382.
65. Stumpf MP, Wiuf C, May RM: **Subnets of scale-free networks are not scale-free: sampling properties of networks**. *Proceedings of the National Academy of Sciences of the United States of America* 2005, **102**(12):4221-4224.
66. Stumpf MP, Ingram, P.J.: **Probablity models for degree distribution of protein interaction networks.** *Europhys Lett* 2005, **71**(1):152-158.
67. Stumpf MP, INgram, P.J., Nouvel, I., Wiuf, C.: **Statistical model selection methods applied to biological network data.** *Trans Comp Syst Biol* 2005, **3**:65-77.
68. Barabasi AL, Albert R: **Emergence of scaling in random networks**. *Science* 1999, **286**(5439):509-512.
69. Barabasi AL: **Scale-free networks: a decade and beyond**. *Science* 2009, **325**(5939):412-413.
70. Gilbert EN: **Random Graphs**. *Ann Math Stat* 1959, **30**(4):1141-1144.
71. Erdős P, Rényi, A.: **On Random Graphs, I.** *Publicationes Mathematicae* 1959, **6**:290-297.
72. Erdős P, Rényi, A.: **The Evolution of Random Graphs**. *Magyar Tud Akad Mat Kutató Int Közl* 1960, **5**:17-61.
73. **Erdős–Rényi model** [http://en.wikipedia.org/wiki/Erd%C5%91s%E2%80%93R%C3%A9nyi_model]
74. Watts DJ, Strogatz SH: **Collective dynamics of 'small-world' networks**. *Nature* 1998, **393**(6684):440-442.
75. Ward JH: **Hierarchical Grouping to Optimize an Objective Function**. *Journal of the Americal Statistical Association* 1963, **58**(301):236-244.
76. Eisen MB, Spellman PT, Brown PO, Botstein D: **Cluster analysis and display of genome-wide expression patterns**. *Proc Natl Acad Sci U S A* 1998, **95**(25):14863-14868.
77. MacQueen JB: **Some Methods for classification and Analysis of Multivariate Observations**. In: *5th Berkeley Symposium on Mathematical Statistics and Probability: 1967*. University of California Press: 281-297.
78. Steinhaus H: **Sur la division des corps matériels et parties**. *Bull Acad Polon Sci* 1956, **4**(12):801-804.
79. Newman ME, Girvan M: **Finding and evaluating community structure in networks**. *Phys Rev E Stat Nonlin Soft Matter Phys* 2004, **69**(2 Pt 2):026113.
80. van Dongen S: **A cluster algorithm for graphs**. In. Amsterdam: National Research Institue in the Netherlands; 2000.

81. van Dongen S: **Graph Clustering by Flow SImulation**. University of Utrecht; 2000.
82. Meila M, Shi, J.: **A random walks view of spectral segmentation**. In: *International Workshop on AI and Statistics*. 2001.
83. Fiedler M: **Algebraic connectivity of graphs**. *Czech Math J* 1973, **23**:298-305.
84. Fiedler M: **A property of eigenvectors of non-negative symmetric matrices and its application to graph theory.** *Czech Math J* 1975, **25**:619-633.
85. Donath WE, Hoffman, A.J.: **Lower bounds for paritioning of graph.s**. *IBM J Res Develop* 1973, **17**:420-425.
86. Paccanaro A, Casbon JA, Saqi MA: **Spectral clustering of protein sequences**. *Nucleic acids research* 2006, **34**(5):1571-1580.
87. Frey BJ, Dueck D: **Clustering by passing messages between data points**. *Science* 2007, **315**(5814):972-976.
88. Givoni IE, Frey BJ: **A binary variable model for affinity propagation**. *Neural Comput* 2009, **21**(6):1589-1600.
89. Nepusz T, Sasidharan R, Paccanaro A: **SCPS: a fast implementation of a spectral method for detecting protein families on a genome-wide scale**. *BMC Bioinformatics* 2010, **11**:120.
90. Wittkop T, Baumbach J, Lobo FP, Rahmann S: **Large scale clustering of protein sequences with FORCE -A layout based heuristic for weighted cluster editing**. *BMC Bioinformatics* 2007, **8**:396.
91. Wittkop T, Emig D, Lange S, Rahmann S, Albrecht M, Morris JH, Bocker S, Stoye J, Baumbach J: **Partitioning biological data with transitivity clustering**. *Nat Methods* 2010, **7**(6):419-420.
92. Newman AM, Cooper JB: **AutoSOME: a clustering method for identifying gene expression modules without prior knowledge of cluster number**. *BMC Bioinformatics* 2010, **11**:117.
93. Apeltsin L, Morris JH, Babbitt PC, Ferrin TE: **Improving the quality of protein similarity network clustering algorithms using the network edge weight distribution**. *Bioinformatics* 2011, **27**(3):326-333.
94. Shi J, Malik, J.: **Normalized Cuts and IMage Segmentation**. In: *International Conference on Computer Vision and Pattern Recognition: June 1997; San Juan, Puerto Rico*. 1997.
95. Shi J, Malik, J.: **Normalized Cuts and Image Segmentation**. In.: UC Berkeley; 1997.
96. Vlasblom J, Wodak SJ: **Markov clustering versus affinity propagation for the partitioning of protein interaction graphs**. *BMC Bioinformatics* 2009, **10**:99.
97. **Network Motif** [http://en.wikipedia.org/wiki/Network_motif]
98. Alon U: **Network motifs: theory and experimental approaches**. *Nat Rev Genet* 2007, **8**(6):450-461.
99. Milo R, Shen-Orr S, Itzkovitz S, Kashtan N, Chklovskii D, Alon U: **Network motifs: simple building blocks of complex networks**. *Science* 2002, **298**(5594):824-827.
100. Shen-Orr SS, Milo R, Mangan S, Alon U: **Network motifs in the transcriptional regulation network of Escherichia coli**. *Nature genetics* 2002, **31**(1):64-68.
101. Milo R, Itzkovitz S, Kashtan N, Levitt R, Shen-Orr S, Ayzenshtat I, Sheffer M, Alon U: **Superfamilies of evolved and designed networks**. *Science* 2004, **303**(5663):1538-1542.
102. Mangan S, Alon U: **Structure and function of the feed-forward loop network motif**. *Proceedings of the National Academy of Sciences of the United States of America* 2003, **100**(21):11980-11985.

103. Mangan S, Zaslaver A, Alon U: **The coherent feedforward loop serves as a sign-sensitive delay element in transcription networks**. *Journal of molecular biology* 2003, **334**(2):197-204.
104. Kalir S, Mangan S, Alon U: **A coherent feed-forward loop with a SUM input function prolongs flagella expression in Escherichia coli**. *Mol Syst Biol* 2005, **1**:2005 0006.
105. Mangan S, Itzkovitz S, Zaslaver A, Alon U: **The incoherent feed-forward loop accelerates the response-time of the gal system of Escherichia coli**. *Journal of molecular biology* 2006, **356**(5):1073-1081.
106. Entus R, Aufderheide B, Sauro HM: **Design and implementation of three incoherent feed-forward motif based biological concentration sensors**. *Syst Synth Biol* 2007, **1**(3):119-128.
107. Glossop NR, Lyons LC, Hardin PE: **Interlocked feedback loops within the Drosophila circadian oscillator**. *Science* 1999, **286**(5440):766-768.
108. Hau LD, Kwon YK: **The effects of feedback loops on disease comorbidity in human signaling networks**. *Bioinformatics* 2011.
109. Ferrazzi F, Engel FB, Wu E, Moseman AP, Kohane IS, Bellazzi R, Ramoni MF: **Inferring cell cycle feedback regulation from gene expression data**. *J Biomed Inform* 2011.
110. Sevim V, Gong X, Socolar JE: **Reliability of transcriptional cycles and the yeast cell-cycle oscillator**. *PLoS computational biology* 2010, **6**(7):e1000842.
111. Eichenberger P, Fujita M, Jensen ST, Conlon EM, Rudner DZ, Wang ST, Ferguson C, Haga K, Sato T, Liu JS *et al*: **The program of gene transcription for a single differentiating cell type during sporulation in Bacillus subtilis**. *PLoS biology* 2004, **2**(10):e328.
112. Lee TI, Rinaldi NJ, Robert F, Odom DT, Bar-Joseph Z, Gerber GK, Hannett NM, Harbison CT, Thompson CM, Simon I *et al*: **Transcriptional regulatory networks in Saccharomyces cerevisiae**. *Science* 2002, **298**(5594):799-804.
113. Boyer LA, Lee TI, Cole MF, Johnstone SE, Levine SS, Zucker JP, Guenther MG, Kumar RM, Murray HL, Jenner RG *et al*: **Core transcriptional regulatory circuitry in human embryonic stem cells**. *Cell* 2005, **122**(6):947-956.
114. Ashburner M, Ball CA, Blake JA, Botstein D, Butler H, Cherry JM, Davis AP, Dolinski K, Dwight SS, Eppig JT *et al*: **Gene ontology: tool for the unification of biology. The Gene Ontology Consortium**. *Nature genetics* 2000, **25**(1):25-29.
115. Draghici S, Khatri P, Martins RP, Ostermeier GC, Krawetz SA: **Global functional profiling of gene expression**. *Genomics* 2003, **81**(2):98-104.
116. Cho RJ, Huang M, Campbell MJ, Dong H, Steinmetz L, Sapinoso L, Hampton G, Elledge SJ, Davis RW, Lockhart DJ: **Transcriptional regulation and function during the human cell cycle**. *Nature genetics* 2001, **27**(1):48-54.
117. Khatri P, Draghici S, Ostermeier GC, Krawetz SA: **Profiling gene expression using onto-express**. *Genomics* 2002, **79**(2):266-270.
118. Zhang S, Cao J, Kong YM, Scheuermann RH: **GO-Bayes: Gene Ontology-based overrepresentation analysis using a Bayesian approach**. *Bioinformatics* 2010, **26**(7):905-911.
119. Dunn OJ: **Multiple Comparisons Among Means**. *Journal of the Americal Statistical Association* 1961, **56**:52-64.
120. Benjamini Y, Hochberg, Y.: **Controlling the false discovery rate: a practical and powerful approach to multiple testing**. *Journal of the Royal Statistical Society* 1995, **57**(1):289-300.

121. Maere S, Heymans K, Kuiper M: **BiNGO: a Cytoscape plugin to assess overrepresentation of gene ontology categories in biological networks**. *Bioinformatics* 2005, **21**(16):3448-3449.
122. Ideker T, Thorsson V, Ranish JA, Christmas R, Buhler J, Eng JK, Bumgarner R, Goodlett DR, Aebersold R, Hood L: **Integrated genomic and proteomic analyses of a systematically perturbed metabolic network**. *Science* 2001, **292**(5518):929-934.
123. Schuldiner M, Collins SR, Thompson NJ, Denic V, Bhamidipati A, Punna T, Ihmels J, Andrews B, Boone C, Greenblatt JF *et al*: **Exploration of the function and organization of the yeast early secretory pathway through an epistatic miniarray profile**. *Cell* 2005, **123**(3):507-519.

Tutorial:Introduction to Cytoscape

Cytoscape is an open source software platform for *integrating*, *visualizing*, and *analyzing* measurement data in the context of networks. This tutorial will cover:

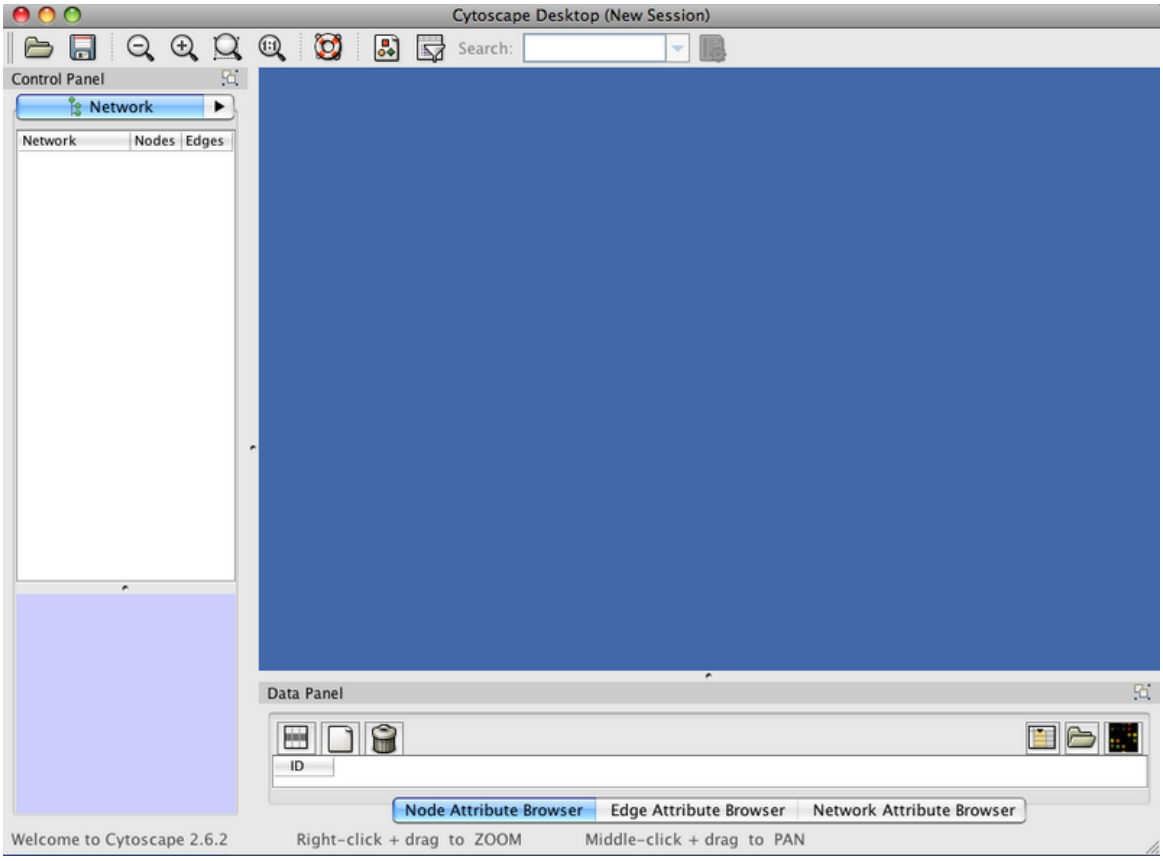
1. Navigating Cytoscape
2. Visualizing Data on Networks
3. Network and Pathway Resources
4. Plugin Manager
5. Plugin Demos

Navigating Cytoscape

This section will introduce the Cytoscape user interface. First of all we will look at the basic UI of Cytoscape. Then we will show all menu features of Cytoscape and the extended functionality provided by plugins.

Cytoscape Layout and User Interface

Launch Cytoscape. You should see a window that looks like this:



- At the top of the Cytoscape Desktop window is the toolbar, which contains the command buttons. The name of each command button is shown when the mouse pointer hovers over it.
- In the upper right is the Main Network View window, where network data will be displayed. This region is initially blank.
- At left is the Control Panel (Network Management) Panel. This lists the available networks by name and provides information on the number of nodes and edges.
- Immediately below the Control Panel is the Network Overview Pane

- At lower right is the Data Panel which can be used to display node, edge, and network attribute data

The Network Management and Data browser panels are dockable tabbed panels known as CytoPanels. You can undock any of these panels by clicking on the Float Window control in the upper-right corner of the CytoPanel. The Data Panel starts off with three tabs: **Node Attribute Browser**, **Edge Attribute Browser**, and **Network Attribute Browser**; the Network Management panel starts off with four tabs: **Network**, **VizMapper**, **Editor**, and **Filters**. Loaded plugins might add tabs to either of these CytoPanels.

Cytoscape Menus

We will briefly run through all the menus available in Cytoscape.

File

The File menu contains basic file functionality:

- **File → Open** for opening a Cytoscape session file
- **File → New** for creating a new network
- **File → Save** for saving a session file
- **File → Import** for importing data such as networks and attributes
- **File → Export** for exporting data and images.
- **File → Print** allows printing
- **File → Quit** closes all windows of Cytoscape and exits the program

Edit

The Edit menu contains:

- Undo and Redo functions which undo and redo edits made in the Attribute Browser, the Network Editor and the Layout.
- Options for creating and destroying views (graphical representations of a network) and networks
- Options for deleting selected nodes and edges from the current network.
- All deleted nodes and edges can be restored to the network via **Edit → Undo**.
- **Edit → Preferences → Properties** to edit preferences for properties and plugins

View

The View menu allows you to display or hide:

- The network management panel (Control Panel)
- The attribute browser (Data Panel)
- Results Panel
- VizMapper

Select

The Select menu contains:

- Options for selecting nodes and edges
- The **Select → Use Filters** option allows filters to be created for automatic selection of portions of a network whose node or edge attributes meet a filtering criterion (see below for the filters section).

Layout

The Layout menu has an array of features for visually organizing the network:

- Rotate, Scale, Align and Distribute are tools for manipulating the network visualization.
- The bottom section of the menu lists a variety of layout algorithms which automatically lay a network out.

Plugins

The Plugins menu contains options for managing your plugins (install/update/delete) and may have options added by plugins that have already been installed, such as the Agilent Literature Search or Merge Networks.

- Depending on which plugins are loaded, the plugins that you see may be different than what appear here.

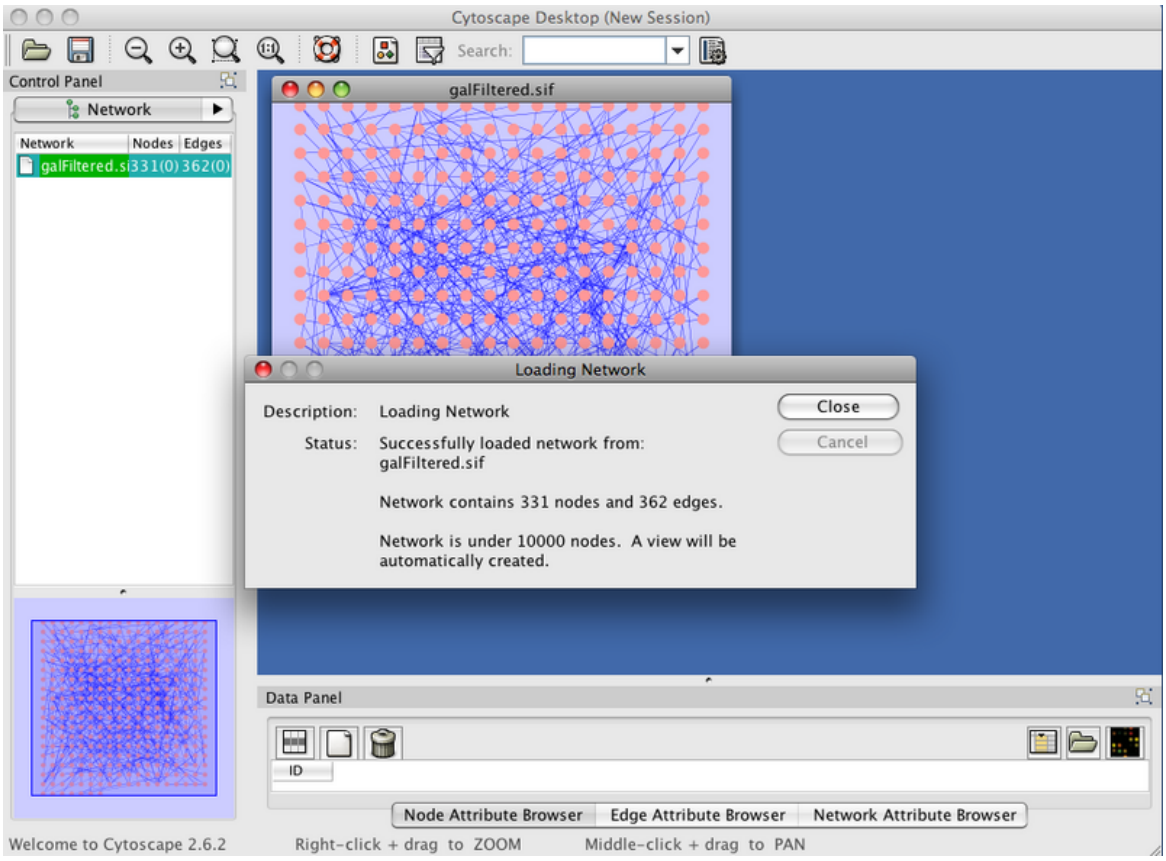
Help

- The Help menu allows you to launch the help viewer and browse the table of contents for this manual.
- The "About…" option displays information about the running version of Cytoscape.

Loading a Simple Network

- Go to **File-> Import -> Network (multiple file types)**
- You should see the Import Network File Dialog
- For Data Source Type select **Local** and then click **Select**
- Open the **sampleData** folder and select **galFiltered.sif** and then click on **Open** and then **Import**

You should see the following:



The SIF file format is about as simple as it gets. It consists of 3 columns: source, interaction type, and target. "Source" and "target" are gene/protein identifiers that are used to define nodes, while "interaction type" serves to label the edge connecting each pair of nodes.

Manipulating Your Network

Now that you have a network loaded, you can interact with it in a number of ways:

- Start by clicking on the node at the upper left corner of the network. The node with turn

yellow. If you hold your mouse down over the node and drag it around the node will move on the screen.

- Now add another node to the selection by holding down the Shift key and clicking on a node. Note that both nodes are now selected (yellow). Again, move the nodes around. Node that both nodes will move.
- To select a group of nodes, hold the mouse down in the upper left-hand corner and drag your mouse over a region of the network. Again, a group of nodes will be selected and can be moved around on the screen.
- To zoom in on the selected nodes, click on the  icon.
- To move the window around the network, you can either use the middle mouse button, or drag the small window outlined in blue around in the Network Overview Pane.
- Finally, zoom your network out by clicking on the  icon.

While useful, hand selecting nodes in dense networks can be error-prone and difficult. However, you can specifically search for a node by name or attribute:

- In the **Search:** box at the top of the screen, type in *ynr050c*. This will select that node and zoom the display to focus on it.

The **Search:** box will also allow you to select nodes by other attributes, but first, we need to import more attributes...

Visualizing Data on Networks

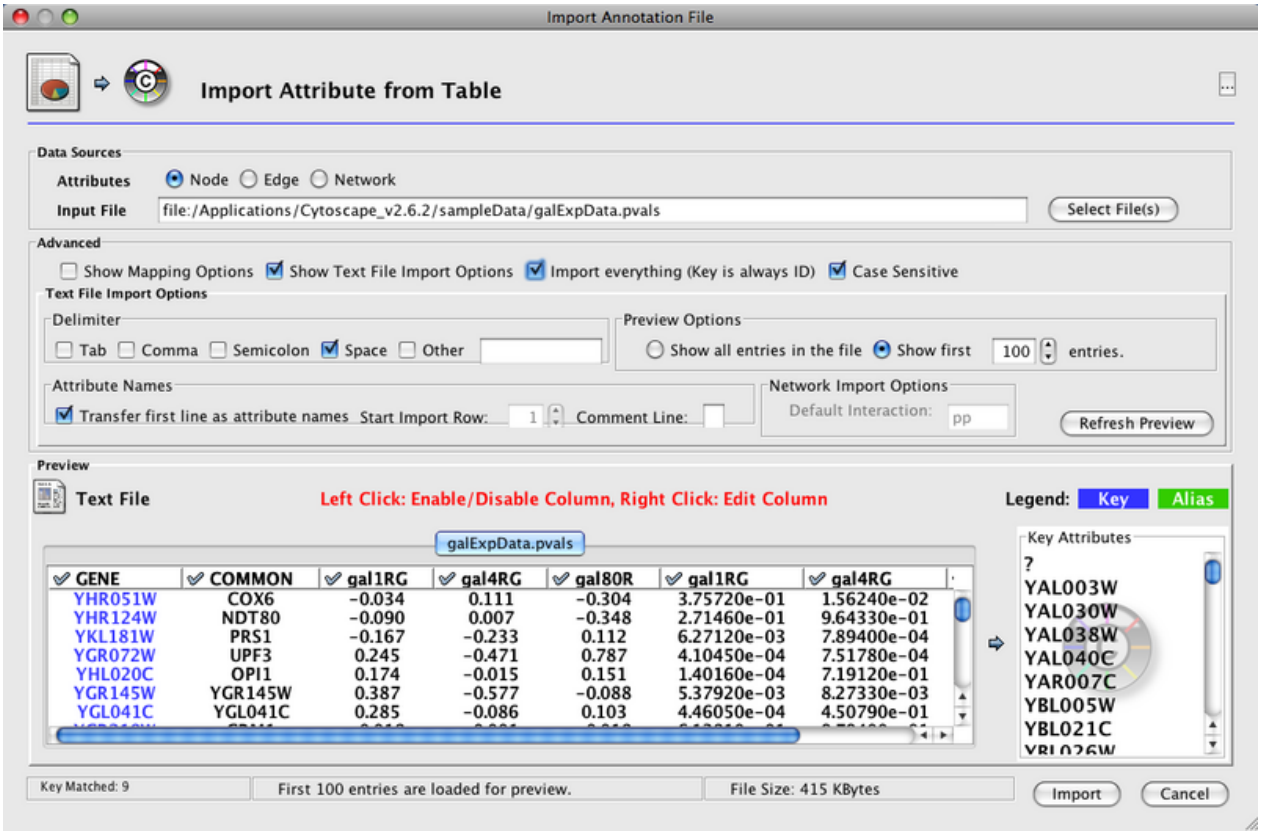
Cytoscape provides a number of features to load arbitrary data and visualize that data by mapping the attribute values to visual styles.

Importing Your Data

Cytoscape can read file structures that are delimited Text or Excel files.

- Go to **File-> Import->Attribute from Table (Text/MS Excel)**
- You should see the **Import Attribute from Table** Dialog Box
- Make sure the **Node** radio attribute button is selected in **Data Sources Attributes**
- In **Data Sources**, click on **Select File**
- Select **galExpData.pvals** from the **sampleData** folder and then click on **Open**
- In the **Advanced** section, select **Show Text File Import Options**
- By default, **Tab** is selected in the **Delimiter** section. Instead, select **Space**.
- In the **Attribute Names** section, select **Transfer first line attribute names**
- Also in the **Advanced** section, select **Import Everything**

*The 'Import Everything' option tells Cytoscape to load **all** of your data, not just the records that match currently loaded networks.*



If you were to click 'Import' now you would see a pop-up message complaining about duplicate attribute names. Notice that columns 6, 7 and 8 have the same names as 3, 4 and 5. The next steps show you how to fix column names without having to start over from Excel.

- Right click on the column header of the first duplicate name (column 6, gal1RG) to open the **Set Attribute Name and Type** dialog box
- Add the suffix 'pval' to the name (e.g, 'gal1RGpval') to distinguish the column as containing p-values
- Repeat these last two steps on columns 7 and 8 (gal4RG and gal80R)
- Now click **Import**

You've successfully loaded data! Now explore the **Data Panel** to confirm the mapping of the data to the network.

- Locate the **Select Attributes** button  in the **Data Panel**
- Select the data attributes: gal1RG, gal4RG, and gal80R
- Return to the **Data Panel** by click away from the attribute list
- Select some nodes in the network (ctrl-A selects all nodes) and see the associated data in the **Data Panel**

You now have access to your networked data and can begin playing with visualization.

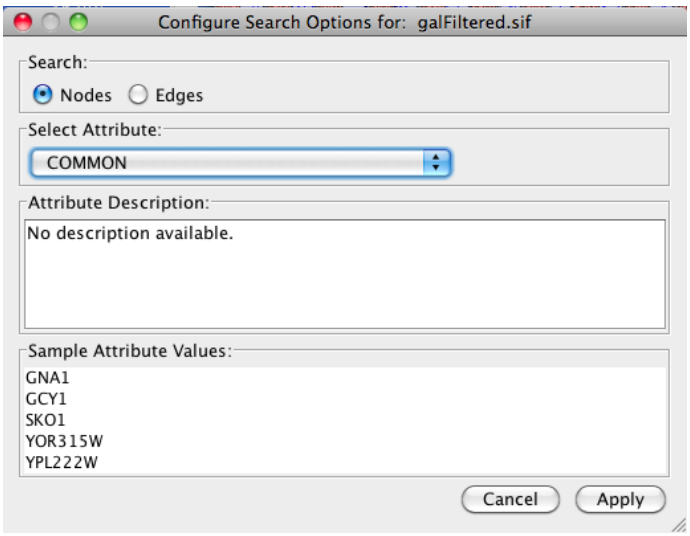
- Go to **Layout -> Cytoscape Layouts -> Force-Directed Layout**
- Go to the **VizMapper** tab in the **Control Panel** and select **Sample 1**

These are default visual styles. In the next section you will customize a visual style to highlight your data values.

Viewing Attributes

In the previous section, you were able to select a couple of attributes to display in the Data Panel. In this section, we will explore a little more about attributes and the Data Panel.

- First click on the **Configure search options** icon: .
- Change the **Select Attribute:** to **COMMON**. You should see this screen:



- Click on **Apply**
- Now type *mcm1* in the **Search:** box. This will select the node:*YMR043W*, and display the attributes for that node in the **Data Panel**.
- We're going add a new attribute for *MCM1*. Click on the icon and select **String Attribute**.
- Type in **pdb** for the name of the attribute -- this will define a new string attribute for nodes, and add it to the **Data Panel**.
- Now click into the empty cell for newly-created **pdb** attribute for **YMR043W** and type *1mnm*, which is the PDB ID for the yeast protein *mcm1*. You need to hit Return or Tab to enter that data.
- Move the **pdb** attribute to be the second column by dragging the column header to be behind the **ID** column
- Finally, select a number of nodes and note that the attributes for all of the nodes are shown in the **Data Panal**
- By clicking on the column header, you can sort the columns. Clicking again changes the order of the sort.

Visualizing Data with VizMapper

- Go to **File-> Open** (click 'Yes' to losing current session)
- You should see the Open a Session File Dialog
- Open the **sampleData** folder and select **galFiltered.cys** and then click on **Open**

Notice how the galFiltered Style maps multiple data and annotation values to:

- Edge Color
- Edge Line Style
- Node Border
- Node Color
- Node Label
- Node Size
- Node Tooltip

Modifying a Visual Style

Customising the way you visualize and manipulate networks is a key function of Cytoscape. This is achieved through the use of the VizMapper tool.

- To launch the VizMapper, either select the VizMapper tab on the Control Panel or click on the VizMapper icon  at the top of the tool bar
- Find **Node Color** in the **Visual Mapping Browser** and expand it by clicking on the triangle icon for expand/collapse
- Click on the value 'gal4RGexp' to select an alternative data value to map to node color: select 'gal80Rexp'. Notice the changes in the network display.
- Click on the gradient color mapping to open the **Gradient Editor**
- Control the values and colors of the mapping by means of triangular handles and endpoint markers
- Double-click on any handle or marker to change it's color: change green to blue and change red to yellow. Notice the immediate changes to the network
- Click-and-drag any handle to slide its value between the min and max
- To save your changes, close the editor
- Find **Edge Color** in the **Visual Mapping Browser** and expand it by clicking on the triangle icon
- Notice the discrete mapping of colors to values. Click on any color mapping to change the color.
- Close the editor to save your changes

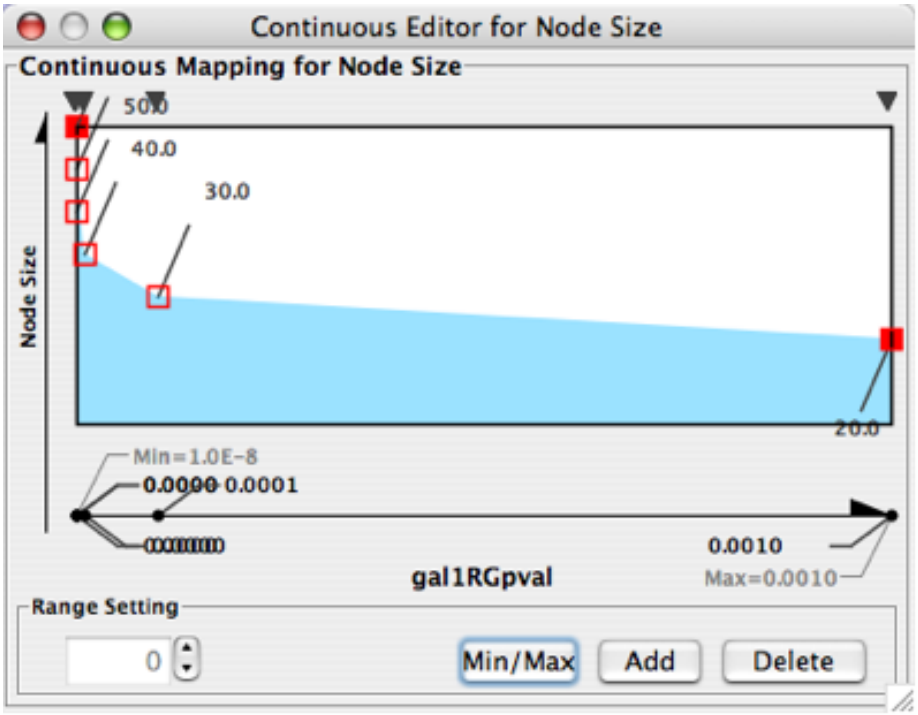
Creating a Visual Style (Advanced)

- In the **Current Visual Style** section, start a new visual style by clicking the **Options** button  and selecting **"Create new Visual Style"**
- Enter a name for your custom visual style when prompted, click on **OK**
- Find **Node Color** in the list of **Unused Properties** and double-click to activate
- For **Node Color** select the value 'gal1RGexp' to map expression fold values. For **Mapping Type** select 'Continuous'. For **Graphical View** click on the gradient to open the dialog:
  - Click on **Min/Max** button and set to -1 and 1, respectively
  - Double-click on gradient handles to set colors, e.g.
    - below -1.0 = green
    - -0.8 = green
    - 0.8 = red
    - above 1.0 = red
  - Click **Add** to add another gradient handle (added to max by default). Drag to center at 0.0. Leave as white color.
  - Close gradient dialog
- Next, activate **Node Size**. Select 'gal1RGpval' to map p-values. For **Mapping Type** select 'Continuous'. Click on gradient to open dialog:
  - Note: We want smaller p-values (more significant) to show as larger nodes
  - Set **Min/Max** to 1.0E-8 and 1.0E-3, respectively.
  - Double-click on solid red box for p-values *below the minimum* (far left) to set the *maximum* node size. Set to 70.0
  - Double-click on solid red box for p-values *below the maximum* (far right) to set the *minimum* node size. Set to 20.0
  - Select handles along the top border to drag the x-position (p-value) of the gradient points. When selected, you can also set the value by typing in the **Range Setting** field. Drag (or double-click) the open red boxes to set the

y-position (node size). Set the following x and y values. Add new gradient points when needed:

- Minimum: x=1.0E-8, y=70.0
- Maximum: x=1.0E-3, y=20.0
- x=10E-4, y=30.0
- x=10E-5, y=40.0
- x=10E-6, y=50.0
- x=10E-7, y=60.0

This creates a pseudo-exponential gradient mapping:



- Close the gradient dialog and **explore** the visualization you've created!
- Zoom by selecting an area and clicking 
- Use the bird's-eye-view panel (bottom-left) to pan around network
- Return to VizMapper and switch mappings to another column of data:
  - For **Node Color** select 'gal4RG'
  - For **Node Size** select 'gal4RGpval'

Notice the change in the view of the network! You can reuse the mappings across multiple datasets. Now is a good time to save your Cytoscape session, to save the visual style you've created.

Laying Out Your Network

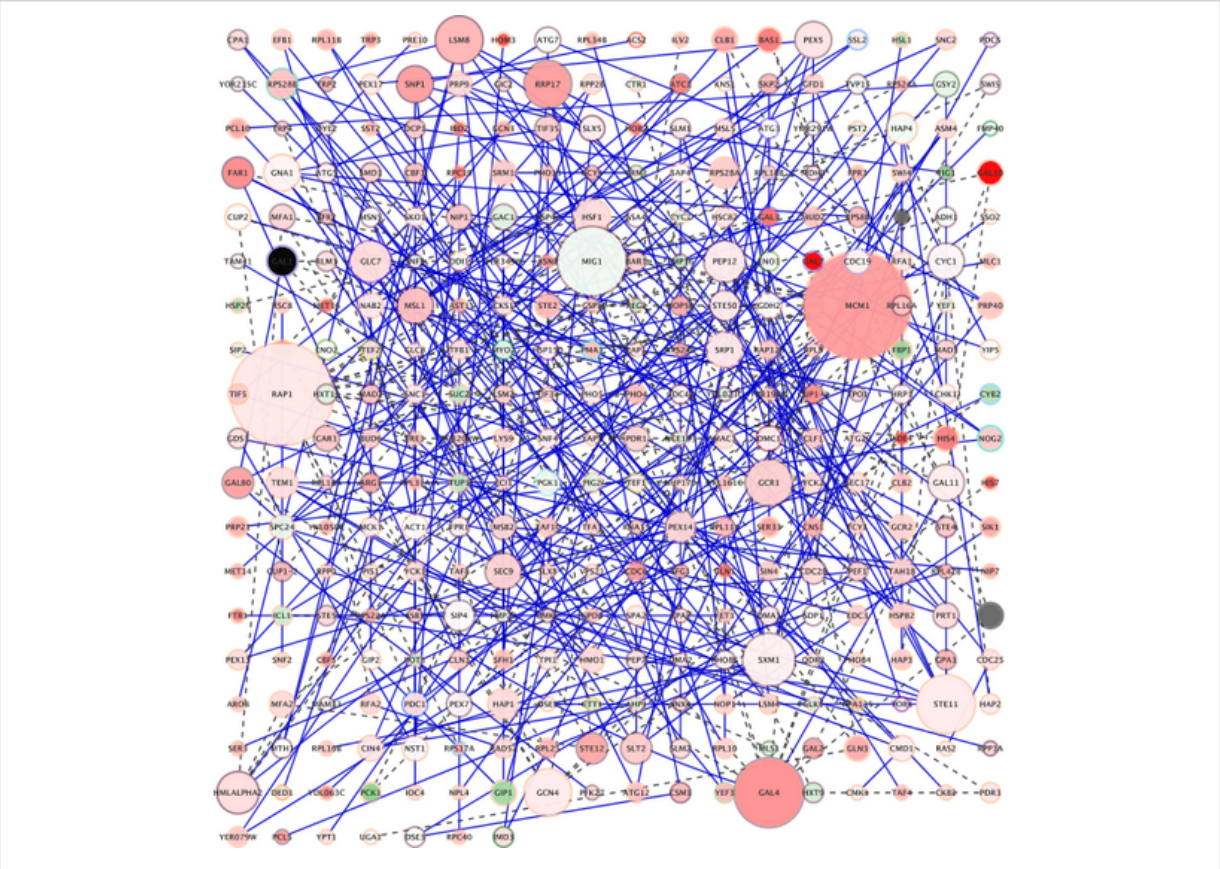
A network layout is a process that positions the nodes and edges for the network. There are a large variety of layouts in Cytoscape and plugins might add new layouts. All of the layouts will appear under the **Layouts** menu. In this section, we'll explore some of the layouts in the **Cytoscape Layouts** category, which are the core layouts supported by the Cytoscape team. All of the **Cytoscape Layouts** support the ability to only layout a portion of the network, and most expose parameters that can be used to tune the layout algorithm.

Simple Layouts

Grid Layout

The simplest layout that Cytoscape provides is the **Grid Layout**, which simply places all of the nodes in a grid arrangement.

- Using the network you loaded before, select **Layout->Cytoscape Layouts->Grid**. You should see the image below:

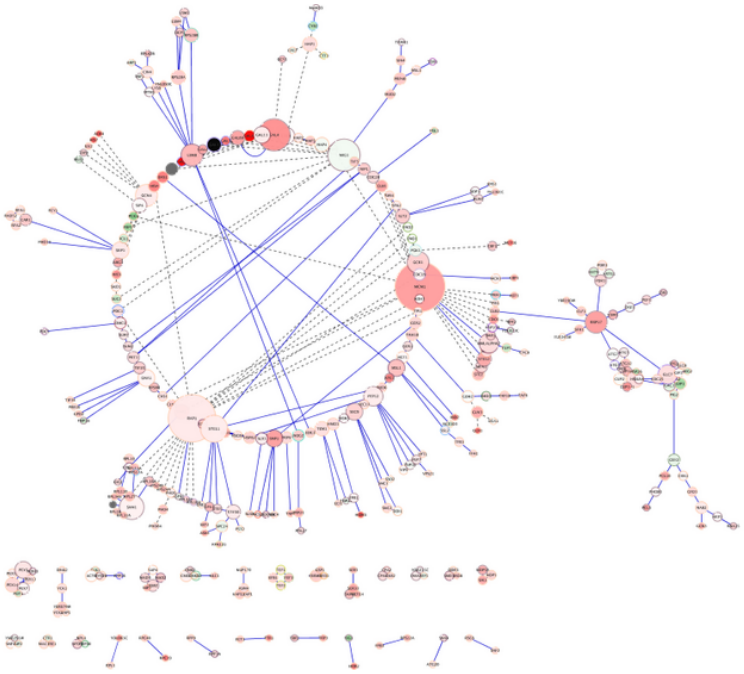


- Now, select **Edit->Undo Grid Layout Layout** and notice how your network reverts to the previous condition.
  - Many (but not all) of Cytoscape's actions may be undone.
- Select a group of nodes in your network and again select **Layout->Cytoscape Layouts->Grid**.
  - Notice that this now has a sub-menu with two options: **All Nodes** and **Selected Nodes Only**.
- Choose **Selected Nodes Only**.
  - Note that only the nodes you selected are changed.

Circular Layout

Grid is very fast, but often not very helpful. Alternatives:

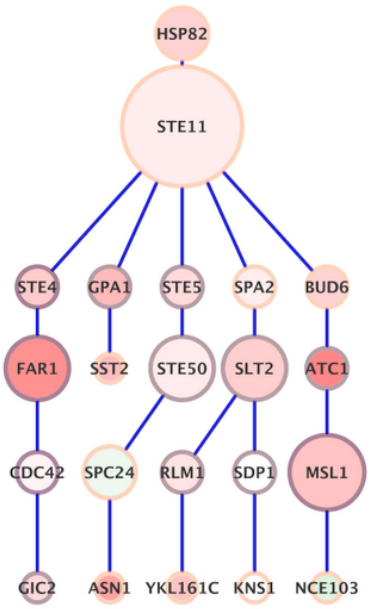
- **Circular Layout** places all of the nodes in a circular arrangement.
  - Very quick
  - Usually not very informative.
  - *Partitions* the network into disconnected parts and independently lays out those parts.



Circular Layout

Hierarchical Layout

- **Hierarchical Layout** forces the nodes into a tree structure.
  - Works best when the network is naturally tree-structured
  - Also works reasonably well when the network is mostly hierarchical.



Hierarchical Layout

Data-Driven Simple Layouts

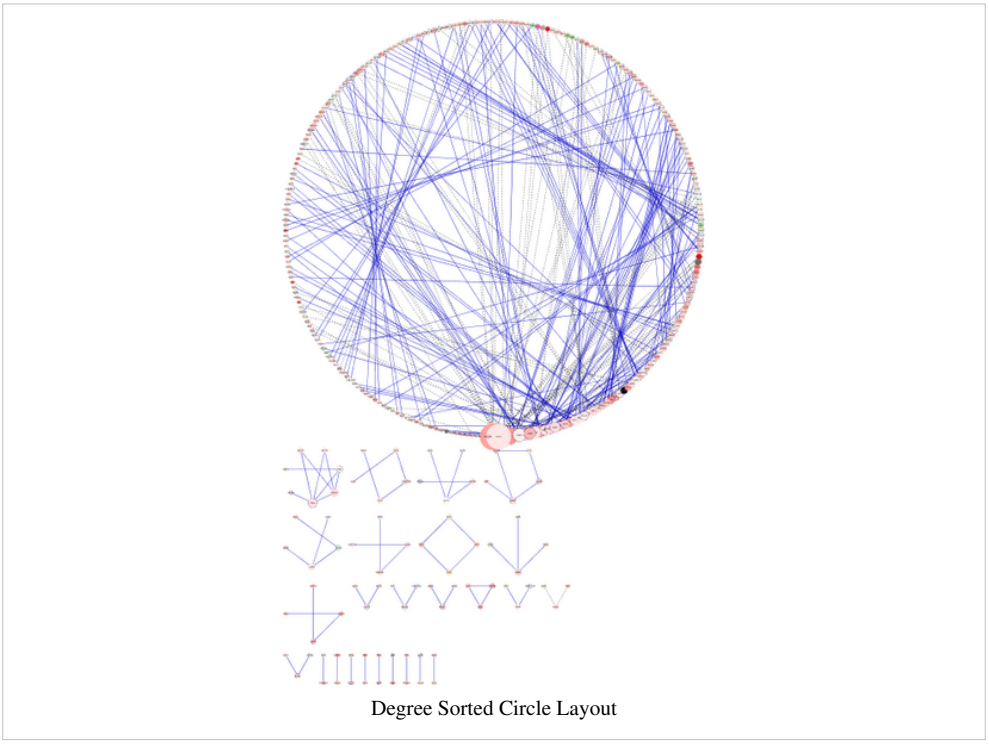
Often what is desired is to organize the nodes in space to reflect some data property of the nodes themselves. We'll look at three of the simple ones:

- **Degree Sorted Circle**: Orders the node around a circle based on node degree (number of edges)
- **Attribute Circule Layout**: Orders the node around a circle based on the value of some attribute
- **Group Attributes Layout**: Groups the nodes based on the value of some attribute

Degree Sorted Circle Layout

For example, assume you are interested in hubs (nodes with high degree):

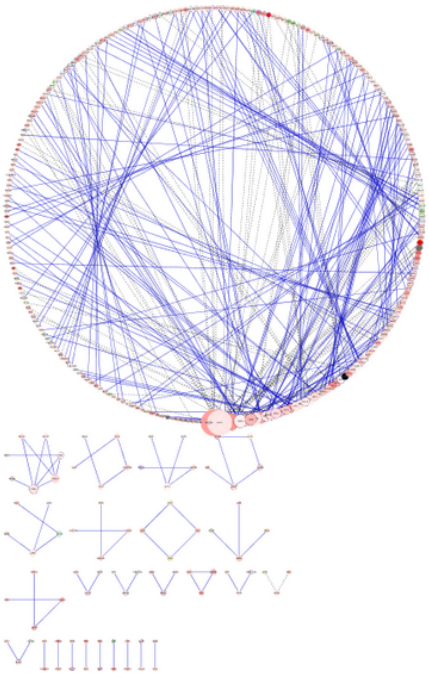
- Select **Layout->Degree Sorted Circle Layout**.
  - Note the highest degree nodes are in the same region of the circle and the degree decreases as you proceed counter-clockwise around the circle.
  - Also note that this layout supports partitioning of the network into disconnected components.



Degree Sorted Circle Layout

Attribute Circle Layout

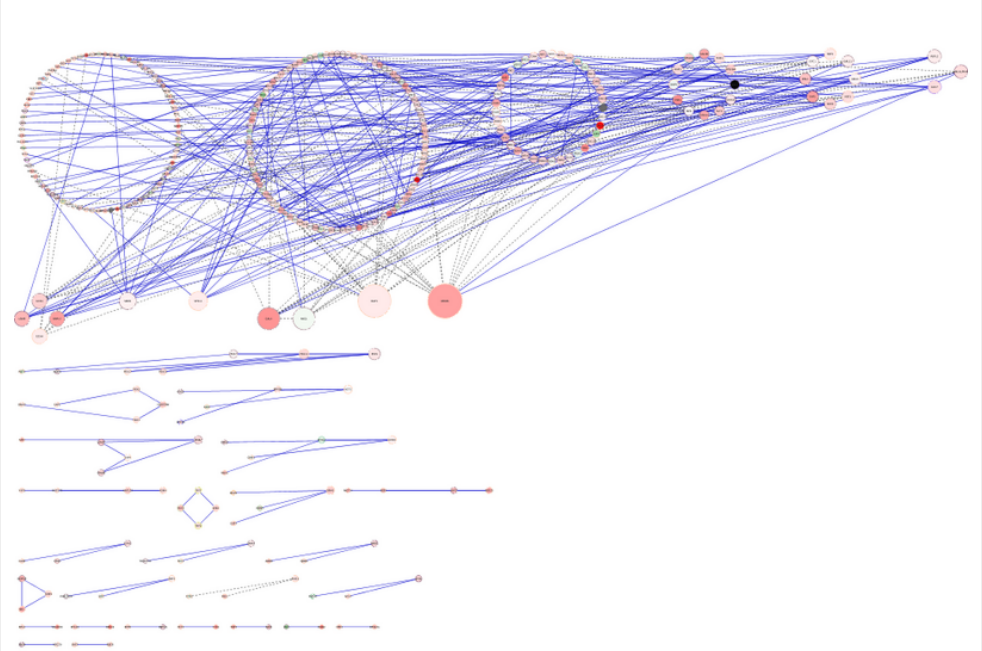
- Now try **Layout->Attribute Circle Layout->Degree**.
  - This should give you a very similar layout to the Degree sorted circle layout.
  - You should also try using other attributes to layout.



Attribute Circle Layout

Group Attributes Layout

- Finally, select **Layout->Cytoscape Layouts->Group Attributes Layout->Degree**.
  - Note that this layout organizes all of the nodes with the same degree into a circle and then positions the circles into a grid.
  - Partitions the graph before layouts.



Group Attributes Layout

These layouts are useful to represent data attributes associated with the nodes of a network, however, they don't provide much information about the network structure itself. For that, using more complicated layouts are required.

Force-Directed Layouts

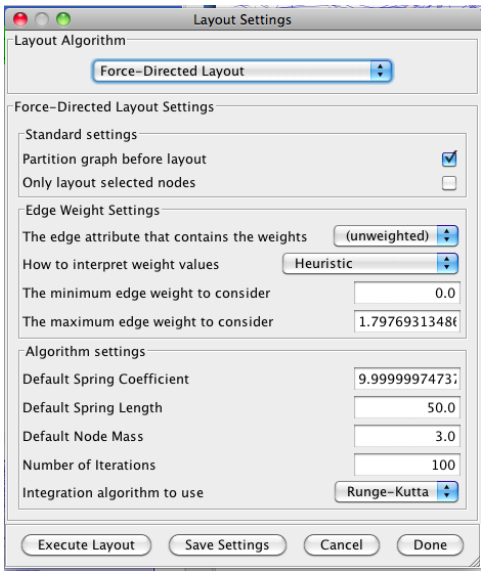
The force-directed methods for laying out graphs all use some kind of physical simulation that models the nodes as physical objects and the edges as springs connecting those objects together.

- Cytoscape provides 4 force-directed layouts:
  - Edge-weighted Force directed (BioLayout)
  - Edge-weighted Spring Embedded
  - Spring Embedded
  - Force-Directed

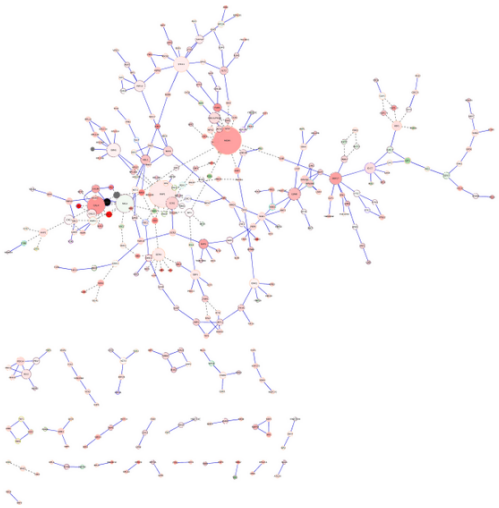
  In this exercise, we'll use the **Force-Directed Layout**, which is a port of the layout by the same name in the *prefuse* package (see http://prefuse.org http://perfuse.org).

Force-Directed Layout

- We'll do all of our work from the Layout settings dialog, so bring up the dialog by **Layout->Settings...**
- In the **Layout Settings** dialog:
  - Select **Force-Directed Layout** under **Select algorithm to view settings**.
  - The result should look like the image on the right
- Lets start by creating a layout using the default parameters.
  - Select **Execute Layout**.
  - Note how this exposes more of the structure of the network than any of the previous layouts.



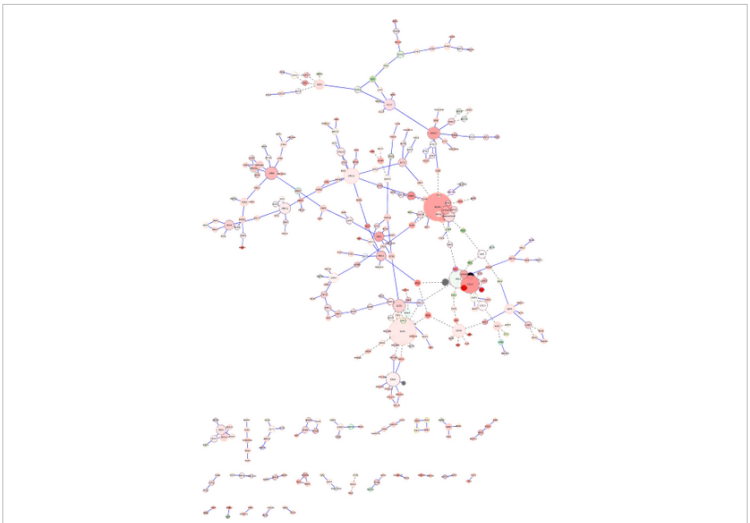
Force Directed Settings Dialog



Force Directed Layout

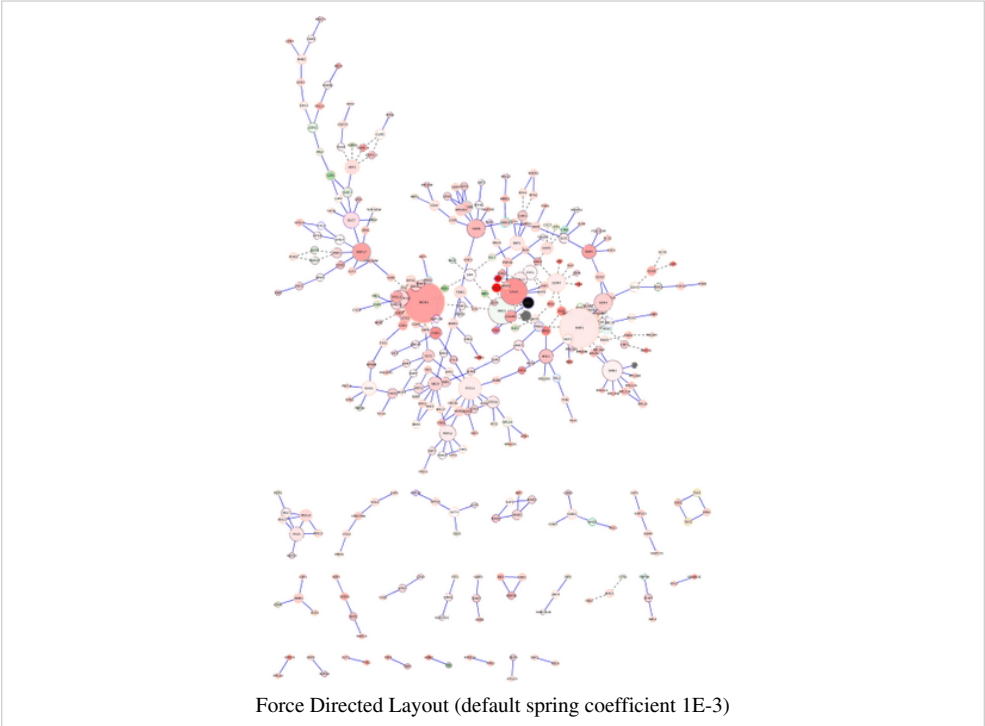
Now, let's see what the parameters do.

- Start by changing the **Default Spring Length** to **20**, and select **Execute Layout**.
  - The default spring length is the length of a spring (edge) with no forces exerted upon it. Essentially, it's the length of an edge connecting two nodes with no other connections.
- Note how the layout has changed.
  - The resulting network tends to be more closely packed in the denser regions.



Force Directed Layout (default spring length 20)

- Finally, lets change **Default Spring Length** back to **50** and the **Default Spring Coefficient** to **1E-3**.
  - The spring coefficient is a measure of the strength of the spring.
- Now click **Execute Layout**.
  - Note that the network remains pretty compact, even though we've reset the default spring length back to 50.



Force Directed Layout (default spring coefficient 1E-3)

Force-Directed Layout is also a good layout to use for circumstances when you want to have the length of your edges reflect some numeric weight on each edge.

- Select the attribute containing your edge weights from the menu **The edge attribute that contains the weights**.
- This often requires some tuning of the spring coefficient and the spring length to get an aesthetically pleasing layout.

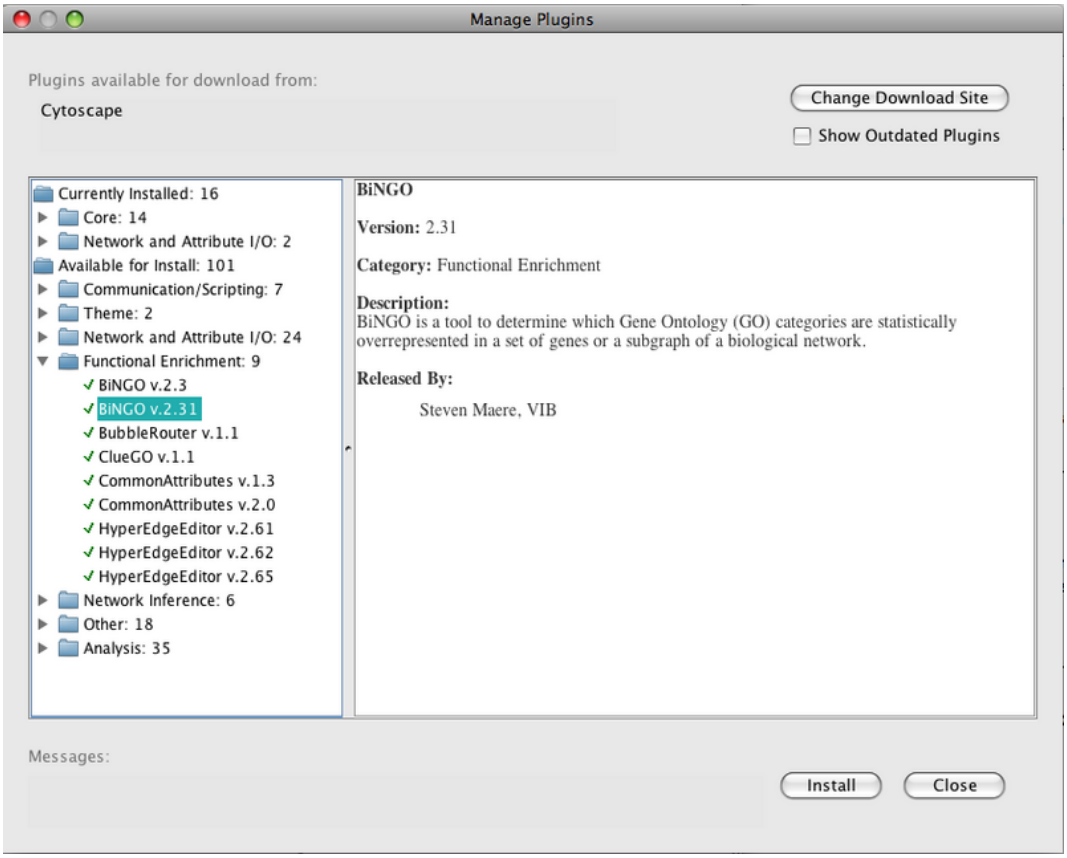
Plugin Manager

The Plugin Manager allows users to quickly and conveniently add extra features to Cytoscape directly from within Cytoscape, eliminating the need for manual searches through different websites to install and update plugins.

If you do not have Internet access enabled, you will not see the list of available plugins or be able to automatically update existing ones; however, you will still be able to view and delete previously installed plugins.

Install new plugins

- Go to the **Plugin Manager at Plugins -> Manage Plugins**. On the left side of the window that pops up, you will see plugin folders labeled **Currently Installed and Available for Install**. Double-clicking on these will show sub-folders, and then the plugins themselves. To find out more about a specific plugin, click on its name to display some basic information on the right-hand side of the window.



- **Currently Installed** folder contains a number of default plugins that are fully integrated in every copy of Cytoscape, as well as any additional installed plugins
- **Available for Install** folder displays plugins that may be installed.
- To install **jActiveModules**: Go to the **Available for Install** folder, select the **Analysis** folder and then select **jActiveModules v2.22**. Click on the Install button at the bottom of the window
- Click on close to exit the Plugin Manager
- Go to the **Plugins** menu and **jActiveModules** should appear in the list of plugins

To manually install a plugin

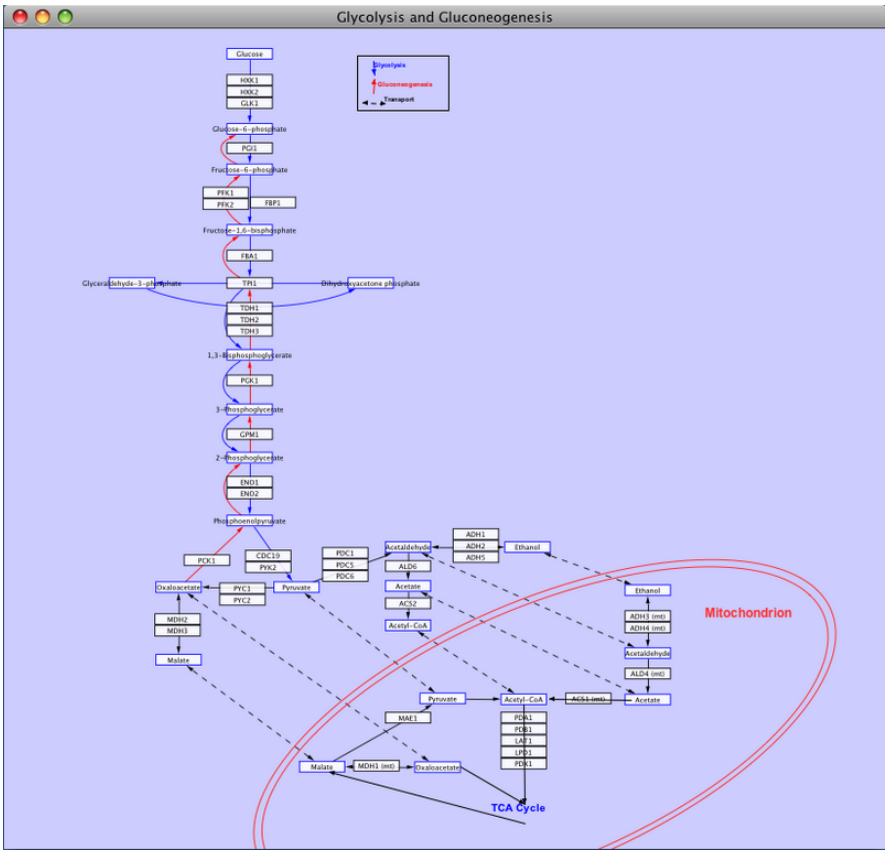
- Go to the Cytoscape plugins page (http://cytoscape.org/plugins),
- Scroll down to find the plugin
- Click on the appropriate link to download the file, and then save it in the Cytoscape/plugins folder on your hard drive.
- Cytoscape will require a restart in order to load the manually installed plugin.

Network and Pathway Resources

But what if you don't have a network? In this section we will cover some of the ways you can retrieve, construct and infer networks and pathways from public sources using Cytoscape and selected plugins.

WikiPathways

- Begin by downloading the GPML-Plugin from the **Plugin Manager**, which can be found under the **Network and Attribute I/O** folder.
- Select **File-> Import -> Network from web services**
- From the drop-down menu, select the **WikiPathways Web Service Client**
- Enter a search term ('TCA') and select a species ('Saccharomyces cerevisiae')
- A list of relevant pathways is returned
- Double-click on 'Glycolysis and Gluconeogenesis' to import the pathway

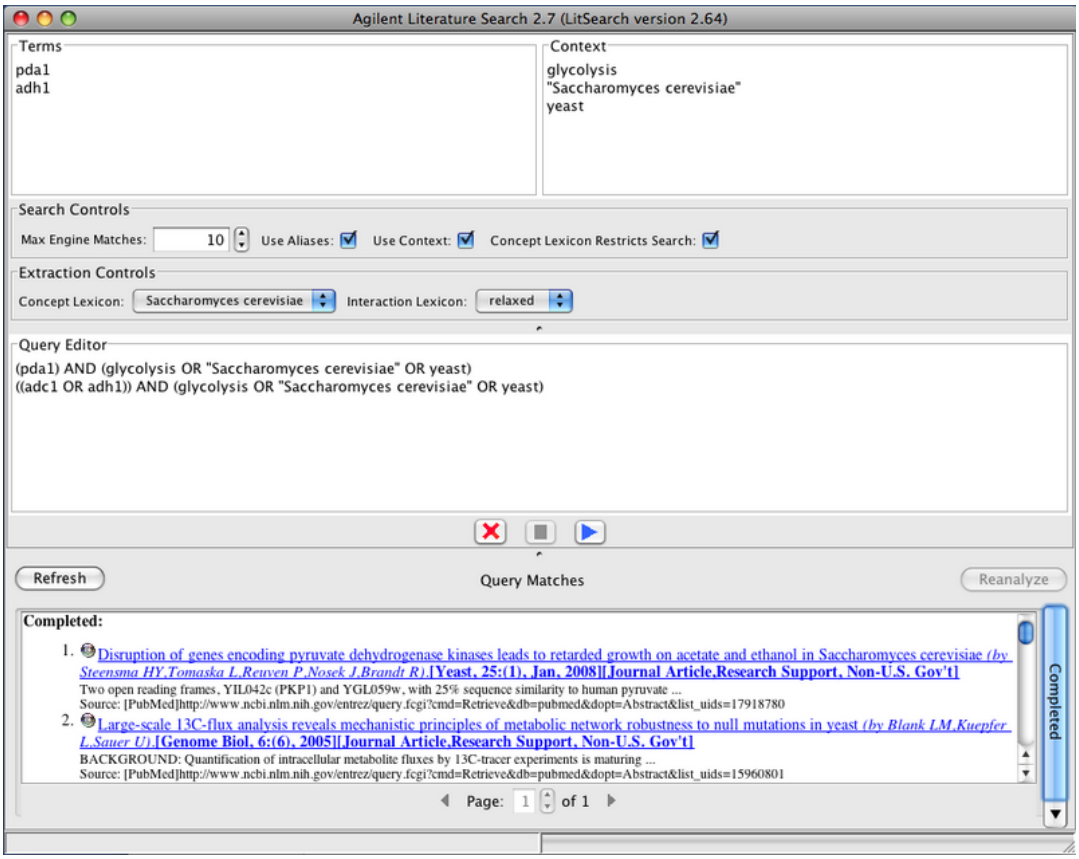


WikiPathways uses the GPML data format, which includes curated coordinates, graphical annotations and labels, in addition to the node and edge network. You can hide the annotation layer using ***View>Toggle GPML Annotations*** *and then treat the network like any other network in Cytoscape.*

Agilent Literature Search

Agilent Literature Search uses text mining technology to generate an "association" network from information extracted from the scientific literature. This can be useful in understanding how genes and proteins may interact in the context of a disease or other biological process.

- To open Agilent Literature Search, go to **Plugins->Agilent Literature Search**
- In **Terms**, enter "pda1" and "adh1"
- In **Context**, enter "glycolysis"
- Select **Use Aliases**. Notice the synonym for "adh1"
- Select **Concept Lexicon**: *Saccharomyces cerevisiae*
- Select **Interaction Lexicon**: "relaxed"
- Click the blue **Play button** to execute the search. Your results will appear in the Query Matches window.

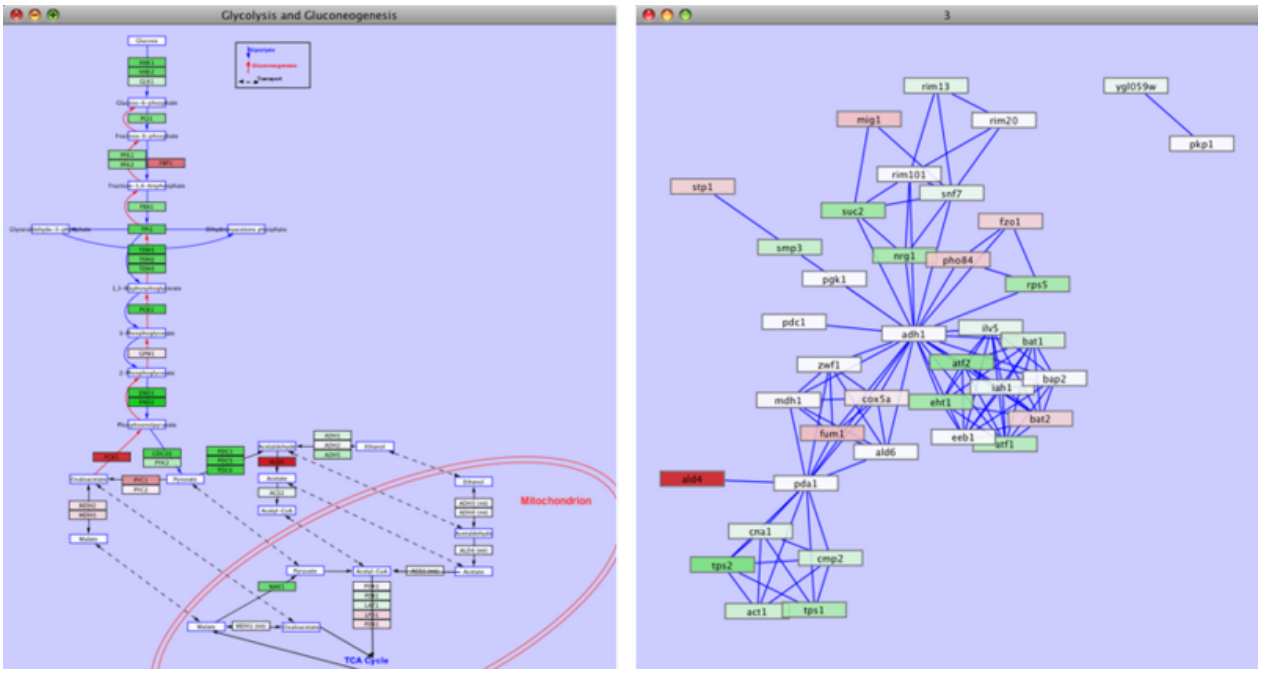


Non-default Data Mapping

When mapping data onto imported networks and pathways, you have to be aware of the key identifiers used in both your data and the imported network. The Attribute Importer provides advanced mapping options for such cases.

- Return to **File-> Import->Attribute from Table (Text/MS Excel)**
- Select **galExpData.pvals** from the **sampleData** folder and then click on **Open**.
- In the **Advanced** section, click **Case Sensitive** to deselect
- In the **Advanced** section, select **Show Text File Import Options**
- In the **Delimiter** section, select **Space**
- In the **Attribute Names** section, select **Transfer first line attribute names**
- Now select **Show Mapping Options**
- Change “GENE” to “COMMON”, and change “ID” to “canonicalName”

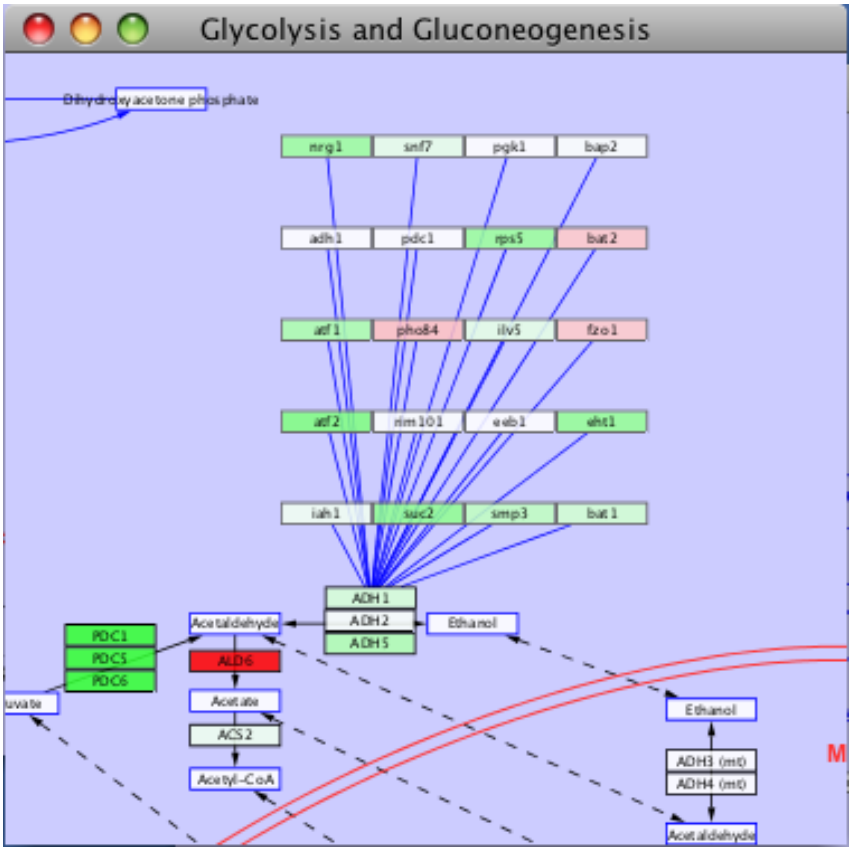
- Click on the headers of the second set of redundant columns in the **Preview** to simply exclude them from import, or rename them by right-clicking
- Click **Import**
- Now go to the **VizMapper** tab in the **Control Panel** and locate the **Node Color** property. Double-click to activate.
- Choose “gal1RG” and then choose “Continous”
- Click on the color gradient to open the **Gradient Editor**
- Set colors for end points and handles to create a color gradient
- Notice the visualization of data on all imported pathways and networks!



Extending Networks

This plugin also has an interactive feature that lets you expand a network using literature search results.

- Go to the imported pathway “Glycolysis and Gluconeogenesis” and use the **Search** box to find “ADH1”
- Right-click on the node and select **Evidence from Literature > Extend Network from Literature**
- Select and drag new nodes near ADH1, go to **Layout > CyLayouts > Grid > Selected Only**, reposition grid in available space near ADH1



- Notice the automatic data mapping
- Right-click on edges to new nodes to **Show Sentences from Literature**
- You can use this procedure to extend known pathways based on literature findings and your visualized data

More Plugin Demos

If continuing from the above tutorial, you may want to restart Cytoscape at this time to clean the slate of unused networks, nodes and attributes. The following plugins all use the same demo file:

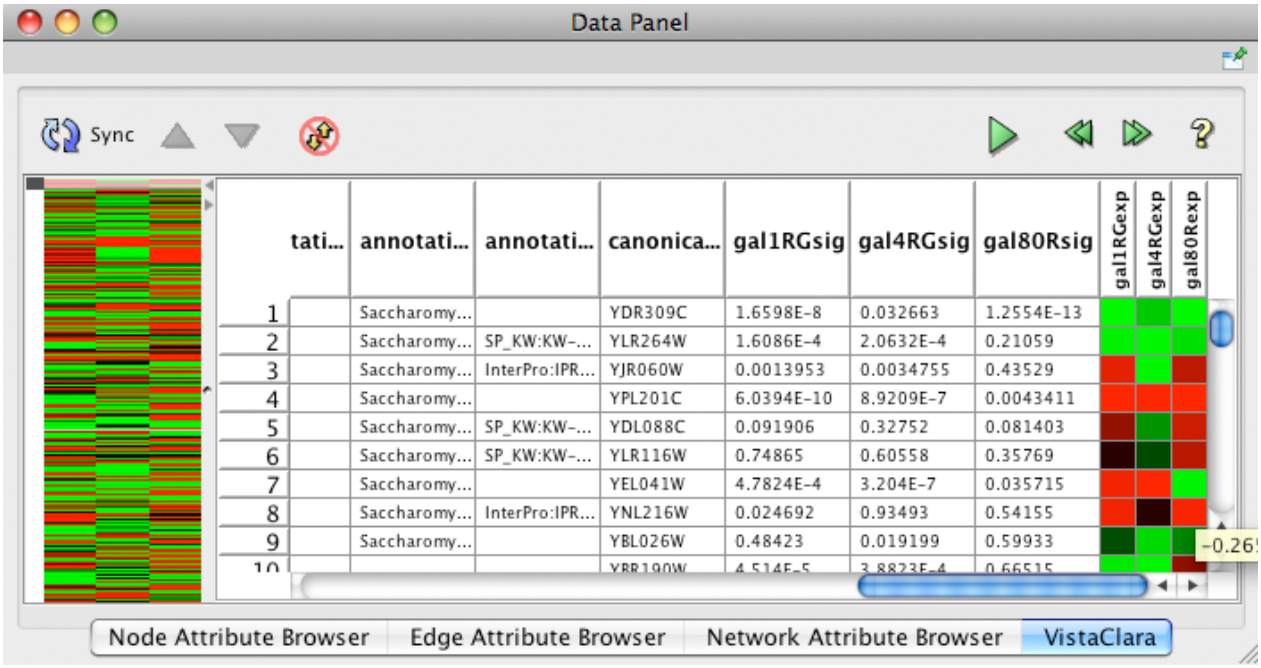
- Go to **File-> Open** (click 'Yes' to losing current session)
- You should see the Open a Session File Dialog
- Open the **sampleData** folder and select **galFiltered.cys** and then click on **Open**

Plugin 1: Vista Clara

Usage: explore data in an interactive, visual spreadsheet

The VistaClara plugin is available from the **Plugin Manager** (refer to Section 1.3)

- Select the **VistaClara** tab below the **Data Panel**
- Click **sync**
- It may be helpful to expand or undock the **Data Panel**



The first column on the left displays the automatically generated heatmap for all of your data. The last columns show the heatmap values for the selected range of data. Note: some columns of data may not be recognized (e.g., gal80R), or may not be properly formatted.

- To fix the formatting of any column of data, simply right-click on the column header, choose format and the correct value. Try 'log2 ratio' for all the fold values in the galFiltered dataset.
- Right-click on any column and select sort to sort by values
- Select or drag anywhere in the heatmap to scroll through the data values
- Select a column of heatmap data to visualize the data on the network
- Click the **play** button to cycle through all heatmap columns
- Notice that a 'VistaClara' visual style has been created in the **VizMapper**. You can select your previous visual styles to return to your custom view at any time

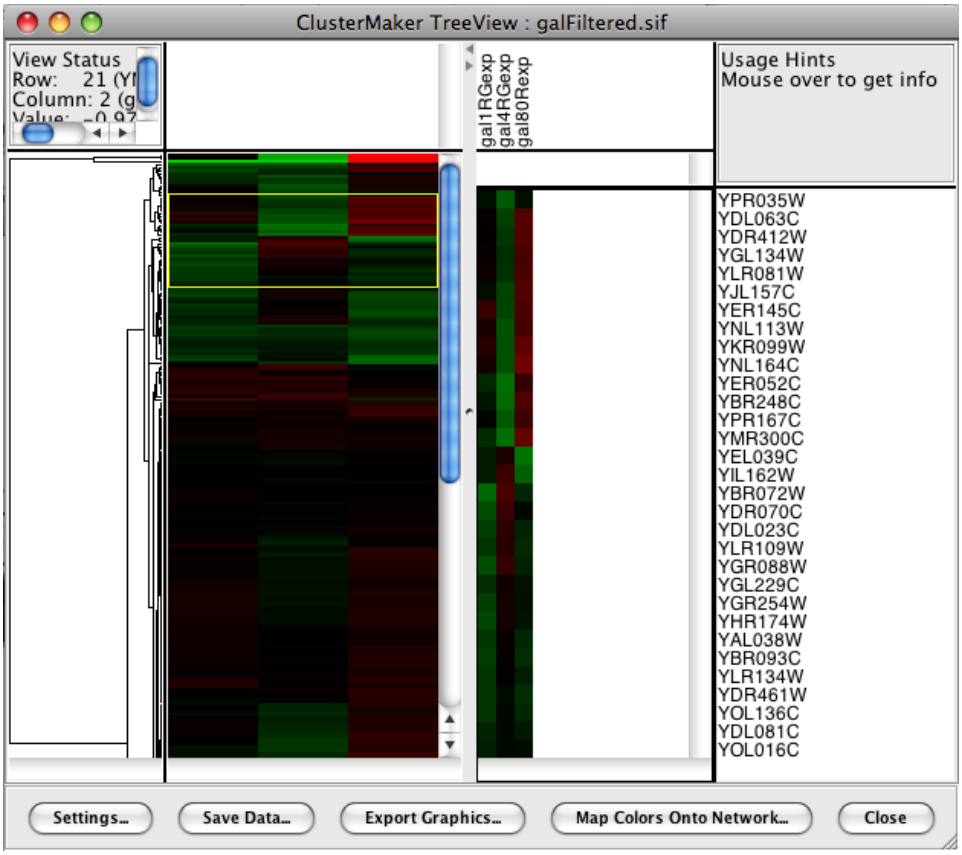
Plugin 2: Cluster Maker

The clusterMaker plugin provides a general framework for clustering and visualizing clusters of Cytoscape node and edge attributes. One of the primary use cases for clusterMaker is to support the analysis of expression data using either hierarchical or k-means clustering. Other uses include the analysis of epistatic mapping (E-MAP) data as well as clustering protein similarity networks to assign nodes to putative protein families.

Usage: perform and visualize cluster analyzes

The ClusterMaker plugin is available from the **Plugin Manager** (refer to Section 1.3)

- Select **Cluster->Hierarchical cluster** from the **Plugins** menu
- Use ctrl (or cmd on Macs) to select the three columns of fold value data (gal1RGexp, gal4RGexp, and gal80Rexp)
- Click **Only use selected nodes/edges for cluster** to deselect
- Click **Create Clusters**
- Click **Visualize Clusters**



This is a standard cluster view with the complete heatmap on the left, the selection heatmap in the middle and the gene names on the right.

- Click **Settings…** to adjust the **contrast, color gradient**, or **data range**
- Click and drag on the heatmap to select rows; Shift-click on heatmap to select individual columns
- Select a column and then click **Map Colors Onto Network…** to create a visual style from the cluster heatmaps, visualizing the data on your network
- If you select a row in the heatmap and then click **Map Colors Onto Network…**, a dialog will pop up asking which columns you want to use; select all three and click **Animate Vizmap**

*Now zoom out and watch your network come alive with data! You can use this view to identify "hot spots" in your network. Hit **Stop Animation** to end the cycle. And use VizMapper to switch back to any prior visual style.*

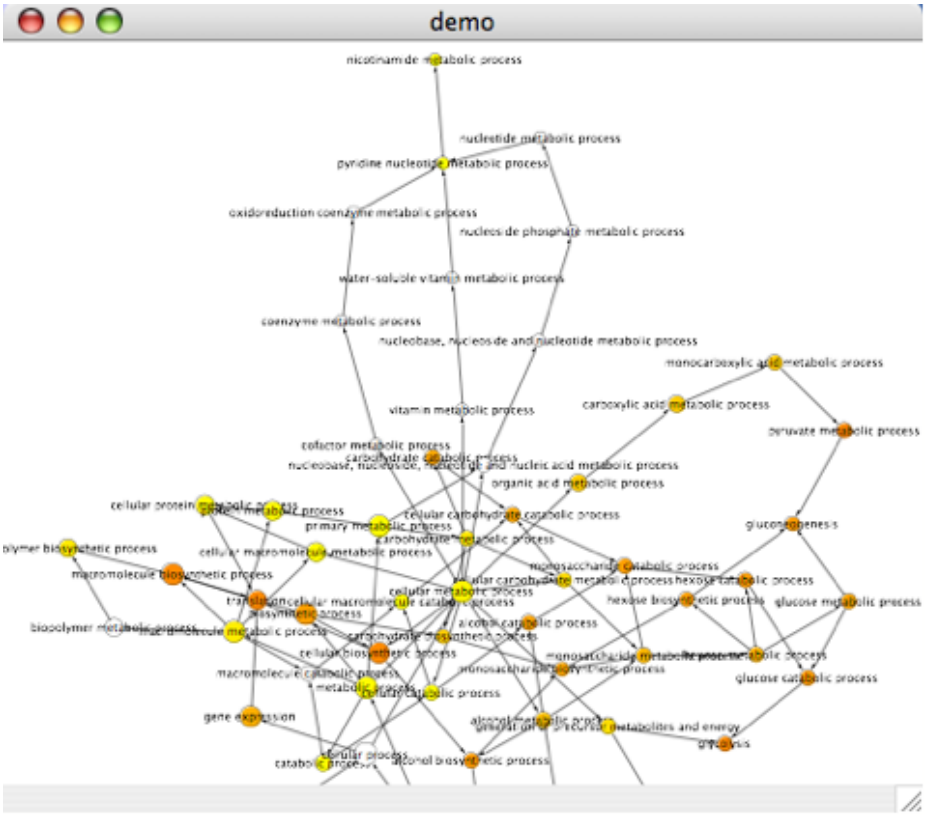
Plugin 3: BiNGO

Usage: perform and visualize overrepresentation analyzes

The BiNGO plugin is available from the **Plugin Manager** (refer to Section 1.3)

- Select **BiNGO** from the **Plugins** menu
- Select a subset of nodes from the network, e.g.
  - Select a cluster from the **ClusterMaker** analysis
  - Select based on sorted or filtered data values in the **Data Panel**
- Provide a name for the cluster
- Review other settings (optional for this tutorial)
- Click 'Start BiNGO'
- You will be greeted with a table of ranked GO terms and statistics, and a network visualization of the GO terms colored by significance, and a legend window explaining the color gradient

The long GO term names can make the network difficult to read. You can use ***Force-directed*** *layout and* ***Layout>Scale*** *to increase the spacing between nodes.* ***Hierarchical*** *layout is also nice, though the names overlap even worse.*



- Play with the cluster selection and BiNGO parameters to explore the analytical possibilities with your data.

Article Sources and Contributors

Tutorial:Introduction to Cytoscape *Source*: http://opentutorials.rbvi.ucsf.edu/opentutorials/index.php?oldid=763 *Contributors*: AlexanderPico, AnnaKuchinsky, ScooterMorris

Image Sources, Licenses and Contributors

Image:screen.png *Source*: http://opentutorials.rbvi.ucsf.edu/opentutorials/index.php?title=File:Screen.png *License*: unknown *Contributors*: -
Image:loadingnetwork.png *Source*: http://opentutorials.rbvi.ucsf.edu/opentutorials/index.php?title=File:Loadingnetwork.png *License*: unknown *Contributors*: -
File:Zoom.png *Source*: http://opentutorials.rbvi.ucsf.edu/opentutorials/index.php?title=File:Zoom.png *License*: unknown *Contributors*: -
File:ZoomOut.png *Source*: http://opentutorials.rbvi.ucsf.edu/opentutorials/index.php?title=File:ZoomOut.png *License*: unknown *Contributors*: -
Image:importing.png *Source*: http://opentutorials.rbvi.ucsf.edu/opentutorials/index.php?title=File:Importing.png *License*: unknown *Contributors*: -
Image:attributebutton.png *Source*: http://opentutorials.rbvi.ucsf.edu/opentutorials/index.php?title=File:Attributebutton.png *License*: unknown *Contributors*: -
File:QuickFind.png *Source*: http://opentutorials.rbvi.ucsf.edu/opentutorials/index.php?title=File:QuickFind.png *License*: unknown *Contributors*: -
Image:QuickFindConfiguration.png *Source*: http://opentutorials.rbvi.ucsf.edu/opentutorials/index.php?title=File:QuickFindConfiguration.png *License*: unknown *Contributors*: -
File:NewAttribute.png *Source*: http://opentutorials.rbvi.ucsf.edu/opentutorials/index.php?title=File:NewAttribute.png *License*: unknown *Contributors*: -
Image:vizmapper.png *Source*: http://opentutorials.rbvi.ucsf.edu/opentutorials/index.php?title=File:Vizmapper.png *License*: unknown *Contributors*: -
Image:visualstyle.png *Source*: http://opentutorials.rbvi.ucsf.edu/opentutorials/index.php?title=File:Visualstyle.png *License*: unknown *Contributors*: -
Image:gradient.png *Source*: http://opentutorials.rbvi.ucsf.edu/opentutorials/index.php?title=File:Gradient.png *License*: unknown *Contributors*: -
Image:zoombutton.png *Source*: http://opentutorials.rbvi.ucsf.edu/opentutorials/index.php?title=File:Zoombutton.png *License*: unknown *Contributors*: -
Image:GridLayout.png *Source*: http://opentutorials.rbvi.ucsf.edu/opentutorials/index.php?title=File:GridLayout.png *License*: unknown *Contributors*: -
Image:CircularLayout.png *Source*: http://opentutorials.rbvi.ucsf.edu/opentutorials/index.php?title=File:CircularLayout.png *License*: unknown *Contributors*: -
Image:HierarchicalLayout.png *Source*: http://opentutorials.rbvi.ucsf.edu/opentutorials/index.php?title=File:HierarchicalLayout.png *License*: unknown *Contributors*: -
Image:DegreeSortedLayout.png *Source*: http://opentutorials.rbvi.ucsf.edu/opentutorials/index.php?title=File:DegreeSortedLayout.png *License*: unknown *Contributors*: -
Image:GroupAttributesLayout.png *Source*: http://opentutorials.rbvi.ucsf.edu/opentutorials/index.php?title=File:GroupAttributesLayout.png *License*: unknown *Contributors*: -
Image:ForceDirectedSettings.png *Source*: http://opentutorials.rbvi.ucsf.edu/opentutorials/index.php?title=File:ForceDirectedSettings.png *License*: unknown *Contributors*: -
Image:ForceDirectedDefaultLayout.png *Source*: http://opentutorials.rbvi.ucsf.edu/opentutorials/index.php?title=File:ForceDirectedDefaultLayout.png *License*: unknown *Contributors*: -
Image:ForceDirected50Layout.png *Source*: http://opentutorials.rbvi.ucsf.edu/opentutorials/index.php?title=File:ForceDirected50Layout.png *License*: unknown *Contributors*: -
Image:ForceDirected1e-3Layout.png *Source*: http://opentutorials.rbvi.ucsf.edu/opentutorials/index.php?title=File:ForceDirected1e-3Layout.png *License*: unknown *Contributors*: -
Image:plugins.png *Source*: http://opentutorials.rbvi.ucsf.edu/opentutorials/index.php?title=File:Plugins.png *License*: unknown *Contributors*: -
Image:wikipathways.png *Source*: http://opentutorials.rbvi.ucsf.edu/opentutorials/index.php?title=File:Wikipathways.png *License*: unknown *Contributors*: -
Image:litsearch.png *Source*: http://opentutorials.rbvi.ucsf.edu/opentutorials/index.php?title=File:Litsearch.png *License*: unknown *Contributors*: -
Image:pathway_litSearch_viz.png *Source*: http://opentutorials.rbvi.ucsf.edu/opentutorials/index.php?title=File:Pathway_litSearch_viz.png *License*: unknown *Contributors*: -
Image:extendingnetworks.png *Source*: http://opentutorials.rbvi.ucsf.edu/opentutorials/index.php?title=File:Extendingnetworks.png *License*: unknown *Contributors*: -
Image:vistaclara2.png *Source*: http://opentutorials.rbvi.ucsf.edu/opentutorials/index.php?title=File:Vistaclara2.png *License*: unknown *Contributors*: -
Image:clustermaker.png *Source*: http://opentutorials.rbvi.ucsf.edu/opentutorials/index.php?title=File:Clustermaker.png *License*: unknown *Contributors*: -
Image:bingo.png *Source*: http://opentutorials.rbvi.ucsf.edu/opentutorials/index.php?title=File:Bingo.png *License*: unknown *Contributors*: -

License

Tutorial:Network Loading And ID Mapping

Slideshow Network Loading and ID Mapping (20 min) [1]
Handout Network_Loading_And_ID_Mapping.pdf (7 pages) [2]
Tutorial Curators Mike Smoot

Cytoscape is an open source software platform for *integrating*, *visualizing*, and *analyzing* measurement data in the context of networks. This tutorial will introduce you to:

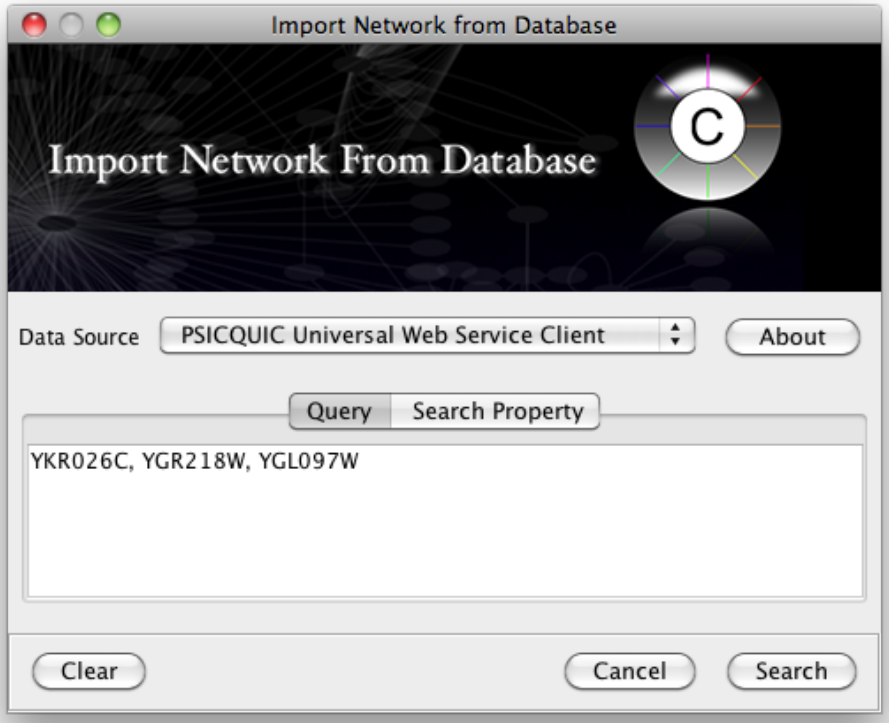
- Searching internet interaction databases with query terms.
- Mapping Identifiers of different types to networks.
- Finding your query terms in the downloaded network.

Setup: Load necessary plugins

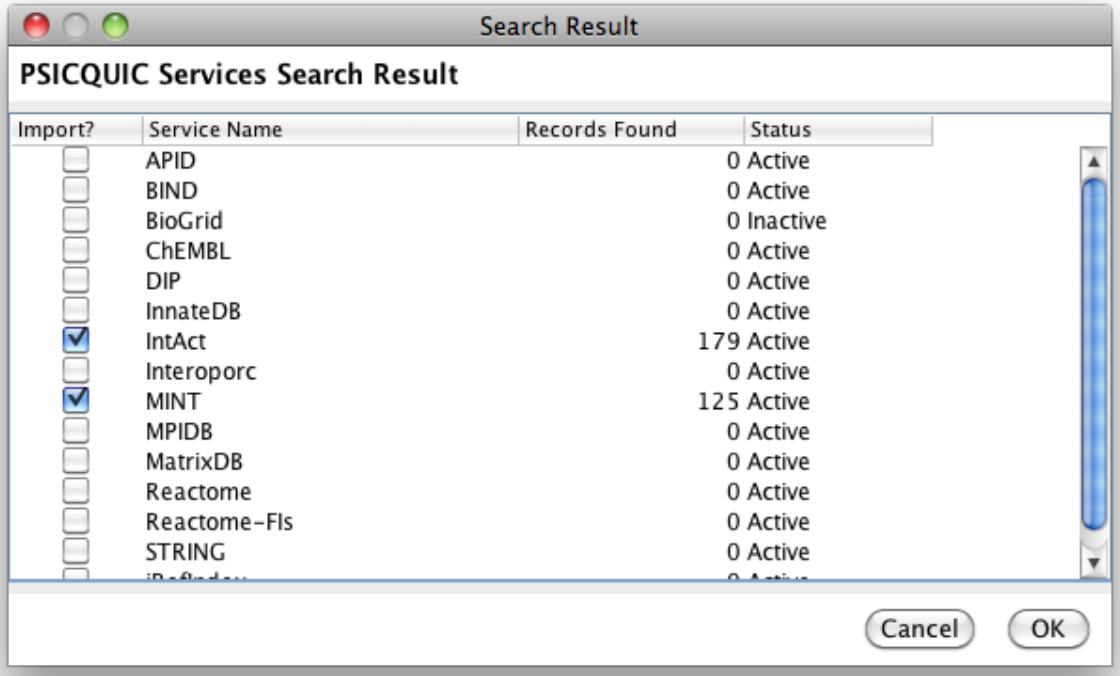
- Start Cytoscape.
- Under the **Plugins** menu, select **Manage Plugins**.
- In the search dialog type "PSICQUIC".
- Find the "PSICQUICUniversalClient" plugin and if it is not already installed, click the **Install** button.
- Wait for the plugin to install.
- Repeat for the plugin "EnhancedSearch".
- Repeat for the plugin "CyThesaurus".
- Close the Plugin Manager.

Search for a Network

- Under the **File** menu, select **Import → Network from Web Services...**.
- In the dialog that pops up choose *PSICQUIC Universal Web Service Client* from the **Data Source** combo box. (Other data sources may work for this tutorial as well, so feel free to try one of those instead.)
- In the **Query** field, copy the following gene IDs: *YKR026C, YGR218W, YGL097W* and then click the **Search** button.
  - (This example is for yeast, but any species should work. Likewise, any ID type should work. **However**, different PSICQUIC service providers may deal with some IDs or species better than others, so your mileage may vary!)

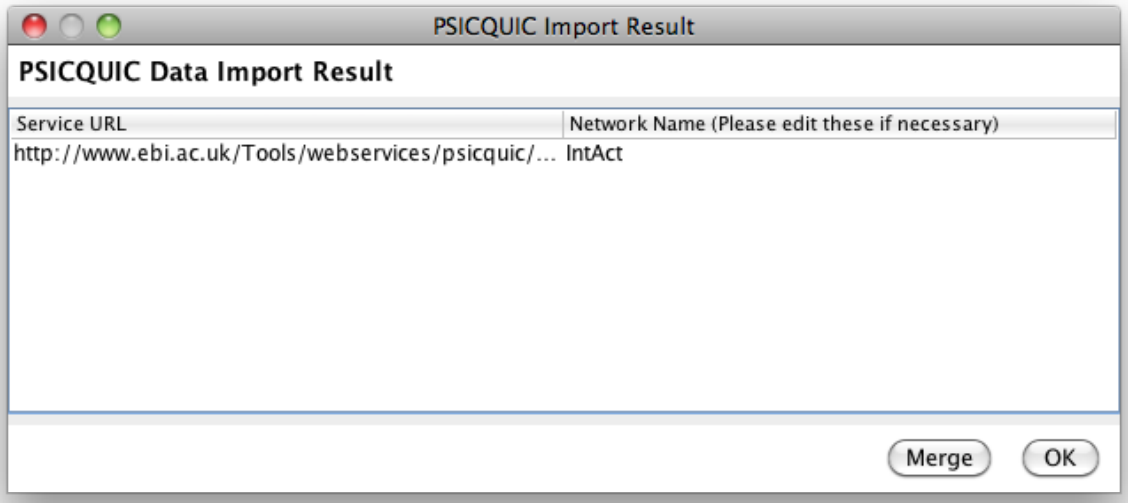


- After a few seconds you should see a new dialog pop-up after the initial search has happened. Click **Yes** to indicate that you'd like to create a new network.
- The **Search Result** dialog will show you a list of service providers which check boxes next to those that have query results to provide. Uncheck all except the *IntAct* service.



- Click OK.

- Click OK in the **Import Result** box as well.



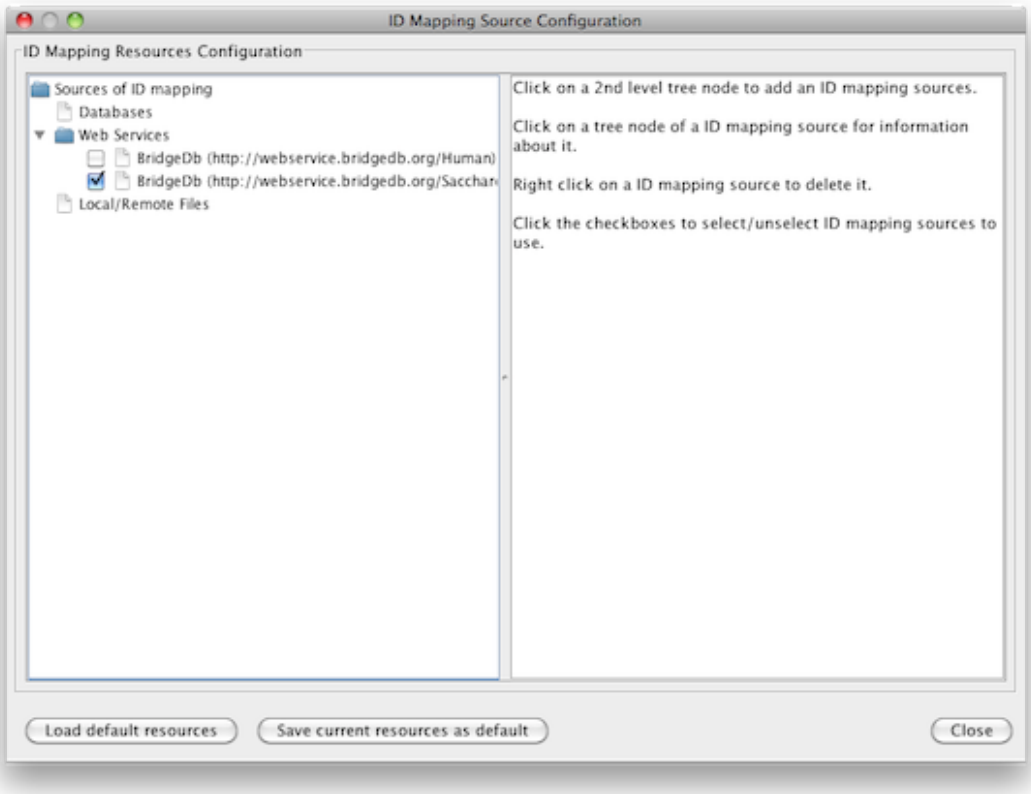
- (The other option, **Merge** will pop up Cytoscape's Network Merge utility and will attempt to merge the specified networks.)

You should now see two networks created: an overview network and the network itself. For this tutorial, you can safely ignore the overview network. One problem with the network that was loaded is that the node identifiers are most likely not the search terms you entered. Instead what you see are identifiers provided by the PSICQUIC service provider based on how they interpreted your search terms. In addition to node identifiers, the PSICQUIC plugin also provided several attributes for the network, among which are alternative identifiers for the nodes. Our next step will be to use these alternative identifiers to map back to our original identifiers.

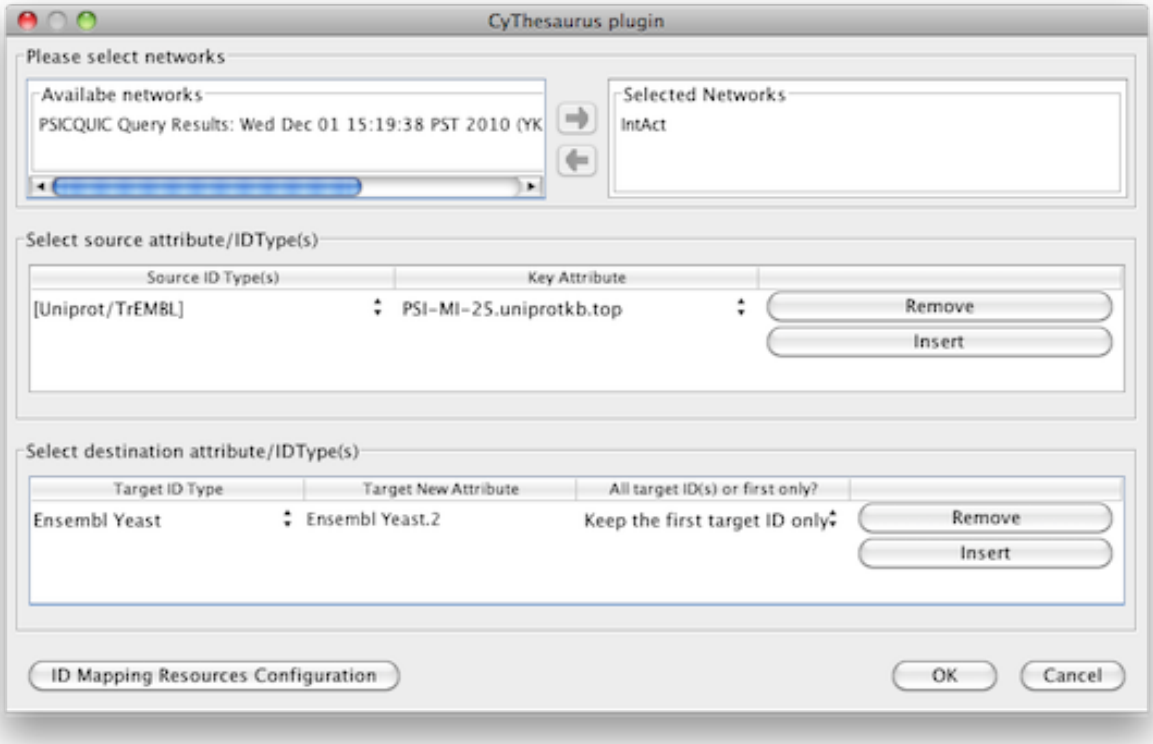
Map Identifiers

This section will demonstrate how to use the *CyThesaurus* plugin to create a new attribute with an identifier of a specific type based on an existing identifier of a different type.

- Under the **Plugins** menu, select **CyThesaurus**
- Before creating a mapping, we need to configure the data source for CyThesaurus. Click the **ID Mapping Resources Configuration** button at the bottom of the CyThesaurus dialog.
- Select the **Web Services** entry in the "Sources of ID Mapping" tree on the left side of the dialog.
- In the dialog that pops up, select the "BridgeDB web service" option, and in the next combo box that appears, choose the base URL appropriate for the species you're working with. Since our search terms were yeast genes, select http://webservice.bridgedb.org/Saccharomyces *cerevisiae*. Click OK.
  - (If the species you're working with doesn't appear in the list, try other web service options.)
- Now ensure that the proper web service is checked in the "Sources of ID Mapping" tree on the left side of the dialog.



- Click the **Save current resources as default** at the bottom of the dialog and then click the **Close** button.
- You should now be back to the main CyThesaurus dialog.
- In the top section of the dialog select the network you wish to work with. This should be named after the database that generated the network (e.g. "MINT" or "IntAct").
- In the "Select source attribute IDType(s)" section, click on the "Key Attribute" column. You should see a choice box listing the available attributes for the network.
- Select "PSI-MI-25.uniprotkb.top". This action will trigger CyThesaurus to guess at the type of the attribute and in the column to the left of "PSI-MI-25.uniprotkb.top" you should see "[Uniprot/TrEMBL]" get selected. You may select additional or alternative ID types by clicking on this field and checking the appropriate boxes. For this tutorial, "[Uniprot/TrEMBL]" is sufficient.
- In the "Select destination attribute/IDType(s)" section, click the "Target ID Type" field. Choose "Ensembl yeast" from the list.
- This will create a new attribute, however the type of attribute will be a List of strings, rather than the single string we'd like. To change this click on the "All target ID(s) or first only?" column in the "Select destination attribute/IDType(s)" section. Choose "Keep the first target ID only". This will create a simple string attribute with first synonym that CyThesaurus finds.

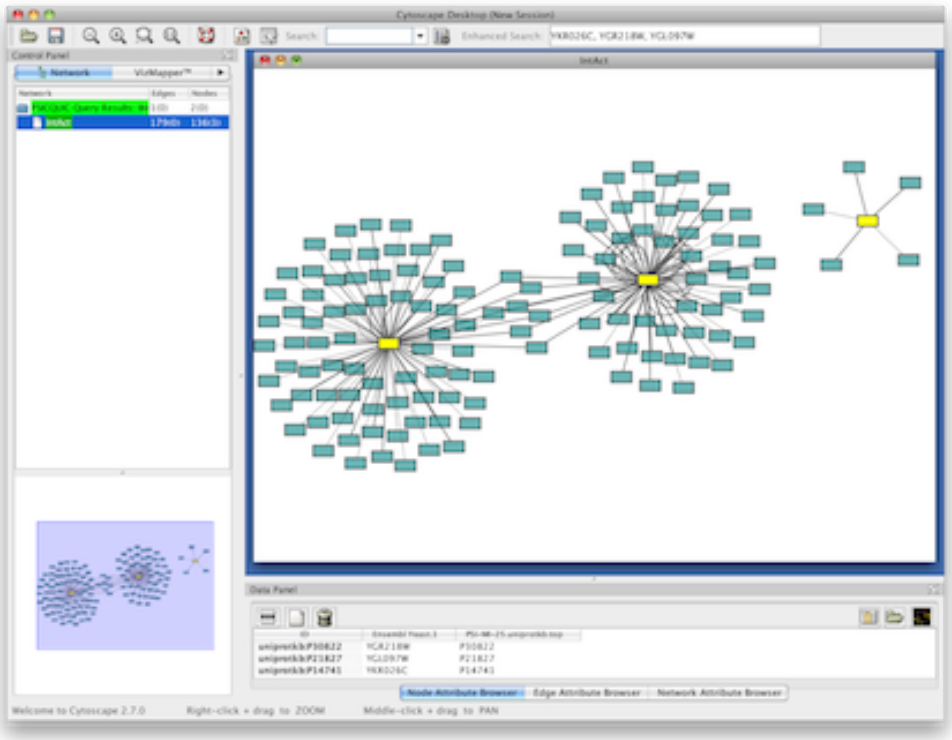


- Click OK.
- You should now see a new attribute available in the attribute browser called something like "Ensembl Yeast".

Find the nodes you care about!

The final step in this tutorial is to find your query terms in the network that you just loaded. It is possible to simply scroll through the attribute browser to find your terms, however it is much more efficient to use the EnhancedSearch plugin.

- Simply type one or more of your search terms into the "Enhanced Search:" text field in the tool bar at the top of the main Cytoscape window.
- Enter your original terms *YKR026C, YGR218W, YGL097W* and you should see 3 nodes get selected in your network!



Possible Next Steps

Now that you have a network with the nodes you care about identified, there are many ways to proceed.

- Import attributes using the "Ensembl Yeast" attribute as a key.
- Analyze the topology of the network using the "Network Analyzer" plugin.
- Grow the network by searching other databases for the same query terms and merging the results into your existing network.
- Load attributes (e.g. expression measurements) and then use the VizMapper to create an informative visualization of the network.

References

[1] http://opentutorials.rbvi.ucsf.edu/index.php?title=Tutorial:Network_Loading_And_Attribute_Mapping&ce_slide=true&ce_style=cytoscape

[2] http://opentutorials.rbvi.ucsf.edu/index.php/File:Network_Loading_And_ID_Mapping.pdf

Article Sources and Contributors

Tutorial:Network Loading And ID Mapping *Source*: http://opentutorials.rbvi.ucsf.edu/index.php?oldid=1204 *Contributors*: KristinaHanspers, MikeSmoot

Image Sources, Licenses and Contributors

File:psicquic_query_dialog.png *Source*: http://opentutorials.rbvi.ucsf.edu/index.php?title=File:Psicquic_query_dialog.png *License*: unknown *Contributors*: MikeSmoot

File:psicquic_query_result.png *Source*: http://opentutorials.rbvi.ucsf.edu/index.php?title=File:Psicquic_query_result.png *License*: unknown *Contributors*: MikeSmoot

File:psicquic_import_result.png *Source*: http://opentutorials.rbvi.ucsf.edu/index.php?title=File:Psicquic_import_result.png *License*: unknown *Contributors*: MikeSmoot

File:cythesaurus_mapping_resource_dialog.png *Source*: http://opentutorials.rbvi.ucsf.edu/index.php?title=File:Cythesaurus_mapping_resource_dialog.png *License*: unknown *Contributors*: MikeSmoot

File:cythesaurus_mapping_configured.png *Source*: http://opentutorials.rbvi.ucsf.edu/index.php?title=File:Cythesaurus_mapping_configured.png *License*: unknown *Contributors*: MikeSmoot

File:network_loaded_ids_mapped_result.png *Source*: http://opentutorials.rbvi.ucsf.edu/index.php?title=File:Network_loaded_ids_mapped_result.png *License*: unknown *Contributors*: MikeSmoot

License

Tutorial:Filtering and Editing in Cytoscape

Slideshow Filtering and Editing in Cytoscape (15 min) [1]
Handout Filtering_and_Editing_in_Cytoscape.pdf [2]
Tutorial Sources Cytoscape Wiki [3]
Tutorial Curators Kristina Hanspers

This tutorial will introduce you to some advanced basics in Cytoscape:

- Apply filters to filter out low-confidence edges.
- Perform basic edits using the Cytoscape graph editor.

Prerequisites

This tutorial features the following plugin and datasets:

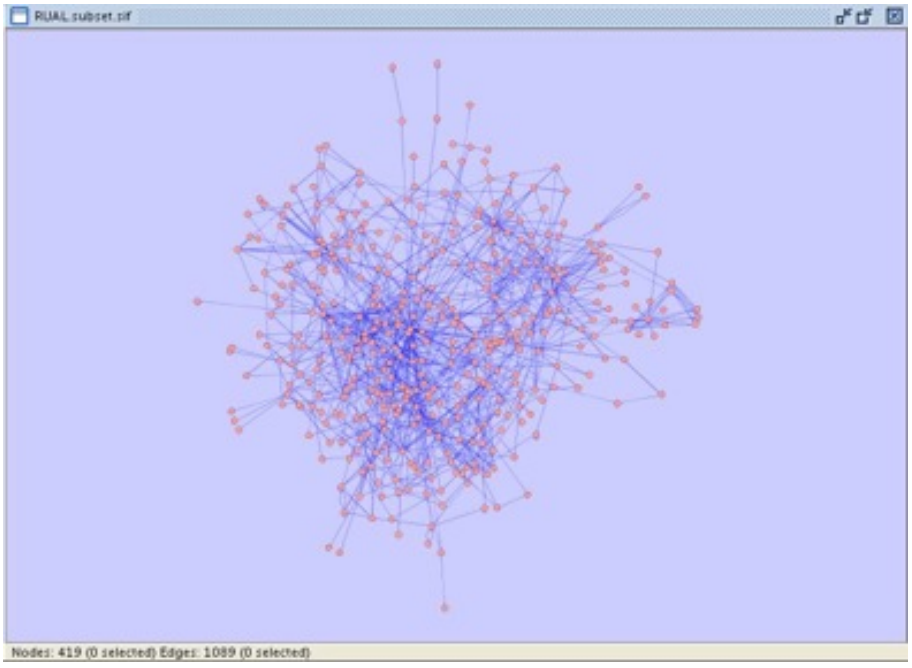
- The Network Filter [4] plugin by Rowan Christmas at the Institute for Systems Biology [5].
- RUAL.subset.sif [6] , a portion of a human interaction data set published 2005 Oct 20 in Rual et al., Nature 437(7062):1173-8 and available from Cytoscape [7].
- Its attribute set RUAL.na [8] , available at the same site.

Before starting, please download these data sets to your computer by right-clicking on the links to the files and saving them.

Loading a network and attributes

- Launch Cytoscape. For details on downloading and installing Cytoscape, see Getting Started [9].
- Load the RUAL.subset.sif network file by going to the **File** menu on the Cytoscape desktop, then select **Import → Network (Multiple file types)...**, and then specifying the location you have downloaded the file to.
- Load the node attribute file RUAL.na by going to **File → Import → Node Attributes...**.
- Generate a yFiles organic layout for your network.

Your Cytoscape window should now appear as shown:



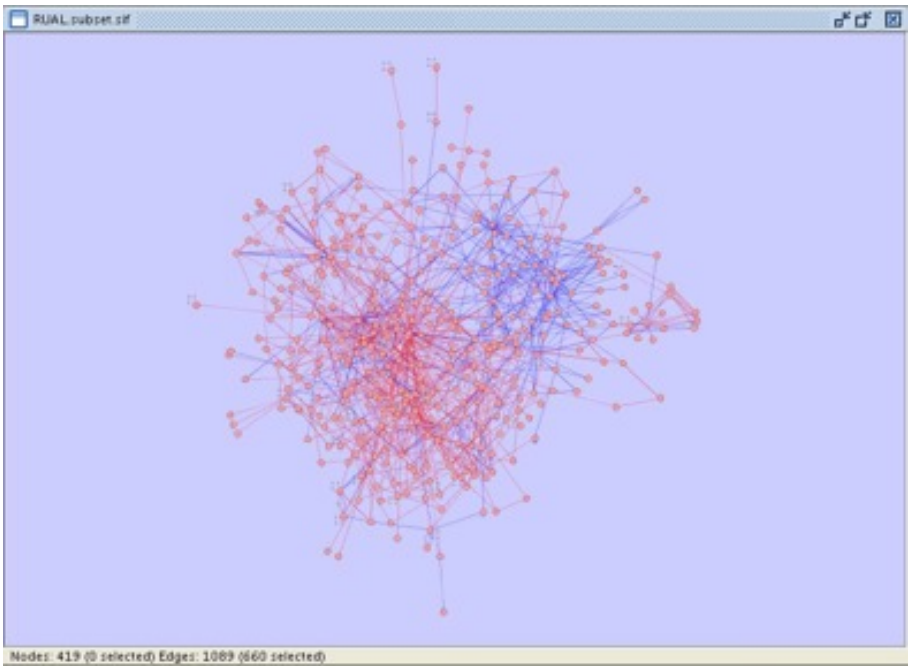
You can see the node attributes as follows:

- The node attribute file you have just loaded defines the official node name, called Official. To see how this attribute is specified, open the file RUAL.na with your favorite text editor.
- In the Cytoscape desktop, under the **Node Attribute Browser**, click on the **Select Attributes** button.
- Notice the attribute named **Official HUGO Symbol**. Select it by clicking on it with the left mouse button.
- Exit the menu with the right mouse button.
- You should now see two columns in the Node Attribute Browser, one labeled **ID** and one labeled **Official HUGO Symbol**. Select some nodes on the Cytoscape canvas. You should see their IDs (Entrez gene IDs, in this case), and their official gene names.

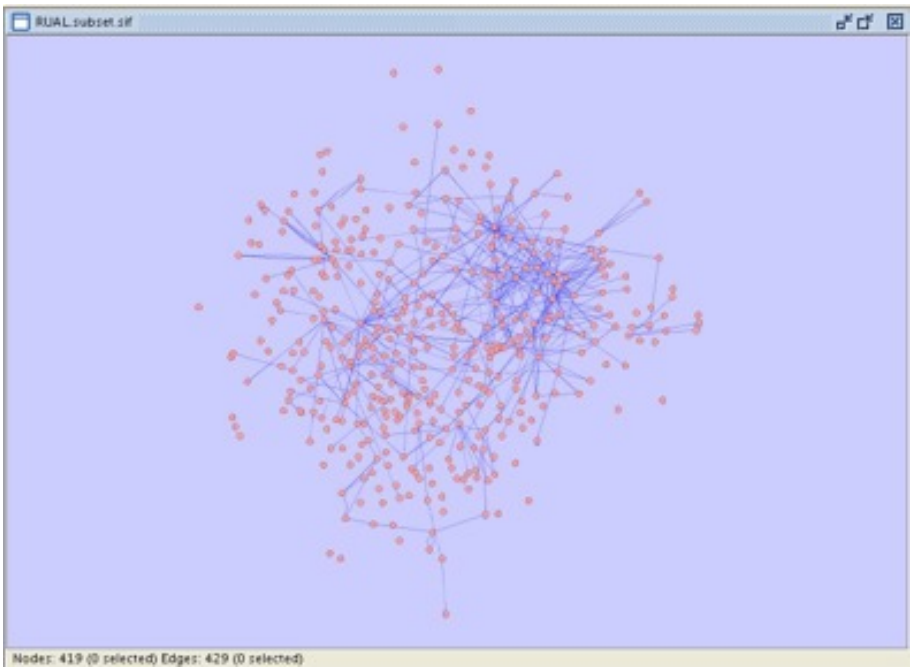
Using Filters

Your network contains several types of edges: Y2H yeast two-hybrid interactions, coAP GST pull-down interactions, and three types of literature-based interactions, listed in order of increasing confidence: non-core, core, hypercore. Verify this by examining the edge types under the Edge Attribute Browser. Remember that to select edges, you must first go to the **Select** menu on the Cytoscape Desktop, and then to **Mouse Drag Selects...**. We will now use Cytoscape's filters to remove the lower-confidence non-core edges.

- Go to **Select → Use Filters** from the Cytoscape desktop menu. This should open the Filters window in the Control Panel.
- In the **Filters** window, click on the **Options** drop-down and select **Create new filter**.
- In the **Filter Definition** field, select edge.interaction in the drop-down. Click the **Add** button.
- In the **Advanced** field, in the **interaction** field, select non_core in the drop-down. Click **Apply** to apply the filter.
- Close the **Use Filters** window and verify that 660 edges have been selected. Your Cytoscape window should appear as shown below.



- Under **Edit** in the Cytoscape Desktop, select **Delete Selected Nodes/Edges**.
- Your Cytoscape window should now appear as shown:



Compared to the network you started with, the network you have now has fewer edges, but all the edges are determined either through experimentation or by higher-confidence literature-based methods. For some types of analysis, this is a more appropriate set of edges. This will leave you with several nodes having no edges, which you may now want to filter out. Here is one method for doing so:

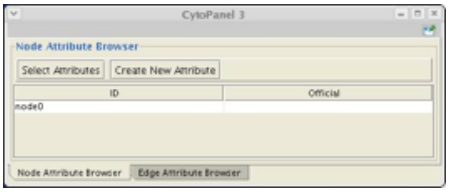
- Create another filter to select objects of type Edge with a value for text attribute interaction that matches the wild card pattern *. This can be done in the same way as the previous filter, but this time type * instead of non_core in the interaction field under Advanced. This filter should select every edge on the canvas. Click **Apply** and verify that all edges are selected.
- Under the **File** menu, select **New → Network → From selected nodes, selected edges**. This should create a "child" network of 257 nodes and 429 edges.
- Apply the spring-embedded layout. This should generate a network as shown.

Note that this will not filter out nodes that only have self-edges. But at this point, such nodes are easy to select with the mouse and delete.

Editing

At times, it can be very useful to modify a network slightly: to add or remove nodes or edges. For instance, if you have prior knowledge of some biological process, you might want to add some nodes for proteins that you know are involved in the process, but that don't appear in your data set. This section will describe how to do so.

- In the **Control Panel** at the left of the Cytoscape Desktop, click on the **Editor** tab.
- Add a node to the canvas by clicking the left mouse button on the node labeled **Add a Node**, and holding down the left mouse button, dragging it onto the Cytoscape canvas. You should see a new, blank node on your canvas.
- Select the node with your mouse. In the **Node Attribute Browser**, you should see a node with ID of node0, as shown below:



- Give your new node a name by going to the **Node Attribute Browser**, clicking on the entry for the name, and entering a name under the column labeled **Official HUGO Symbol**. Notice that you cannot enter a new internal ID for the node.
- Enter an edge between your new node and some other node, as follows:
  - In the **Cytoscape Editor**, click on **Directed Edge**.
  - Holding down the left mouse button, drag the mouse from **Directed Edge** to your new node on the Cytoscape canvas. When you release the left mouse button, you should see two things: when the mouse is on top of your node, it should have a thick black border; and as you move your mouse away from the node, a thick black line should follow the mouse. Click on another node, and a new edge should appear between the two nodes.
  - Create several new edges between your node and existing nodes.
  - Delete your new node (and any edges attached to it) by selecting the node with your mouse, and selecting **Delete Selected Nodes/Edges**. Note that if you delete a node, any edges running to it are also deleted.

Congratulations! You are now finished the advanced course in Cytoscape menu operation. That is worth at least a nice snack!

References

[1] http://opentutorials.rbvi.ucsf.edu/index.php?title=Tutorial:Filtering_and_Editing_in_Cytoscape&ce_slide=true&ce_style=cytoscape
[2] http://opentutorials.rbvi.ucsf.edu/index.php/File:Filtering_and_Editing_in_Cytoscape.pdf
[3] http://cytoscape.wodaklab.org/wiki/Presentations/02_Filter_Edit
[4] http://www.cytoscape.org/plugins/NetworkFilter/rowan.jar
[5] http://www.systemsbiology.org/
[6] http://opentutorials.rbvi.ucsf.edu/index.php/File:RUAL.subset.sif
[7] http://www.cytoscape.org/cgi-bin/moin.cgi/Data_Sets/
[8] http://opentutorials.rbvi.ucsf.edu/index.php/File:RUAL.na
[9] http://opentutorials.rbvi.ucsf.edu/index.php/Tutorial:Getting_Started_with_Cytoscape#Before_Getting_Started

Article Sources and Contributors

Tutorial:Filtering and Editing in Cytoscape *Source*: http://opentutorials.cgl.ucsf.edu/index.php?oldid=1067 *Contributors*: KristinaHanspers

Image Sources, Licenses and Contributors

Image:FiltersFig1.jpg *Source*: http://opentutorials.cgl.ucsf.edu/index.php?title=File:FiltersFig1.jpg *License*: unknown *Contributors*: KristinaHanspers
Image:select.png *Source*: http://opentutorials.cgl.ucsf.edu/index.php?title=File:Select.png *License*: unknown *Contributors*: KristinaHanspers
Image:FiltersFig6.jpg *Source*: http://opentutorials.cgl.ucsf.edu/index.php?title=File:FiltersFig6.jpg *License*: unknown *Contributors*: KristinaHanspers
Image:FiltersFig7.jpg *Source*: http://opentutorials.cgl.ucsf.edu/index.php?title=File:FiltersFig7.jpg *License*: unknown *Contributors*: KristinaHanspers
Image:FiltersFig24.png *Source*: http://opentutorials.cgl.ucsf.edu/index.php?title=File:FiltersFig24.png *License*: unknown *Contributors*: KristinaHanspers

License

Tutorial:Basic Expression Analysis in Cytoscape

Slideshow Basic Expression Analysis in Cytoscape (25 min) [1]
Handout Basic_Expression_Analysis_in_Cytoscape.pdf (9 pages) [2]
Tutorial Sources
Tutorial Curators Kristina Hanspers, Alex Pico, Mike Smoot

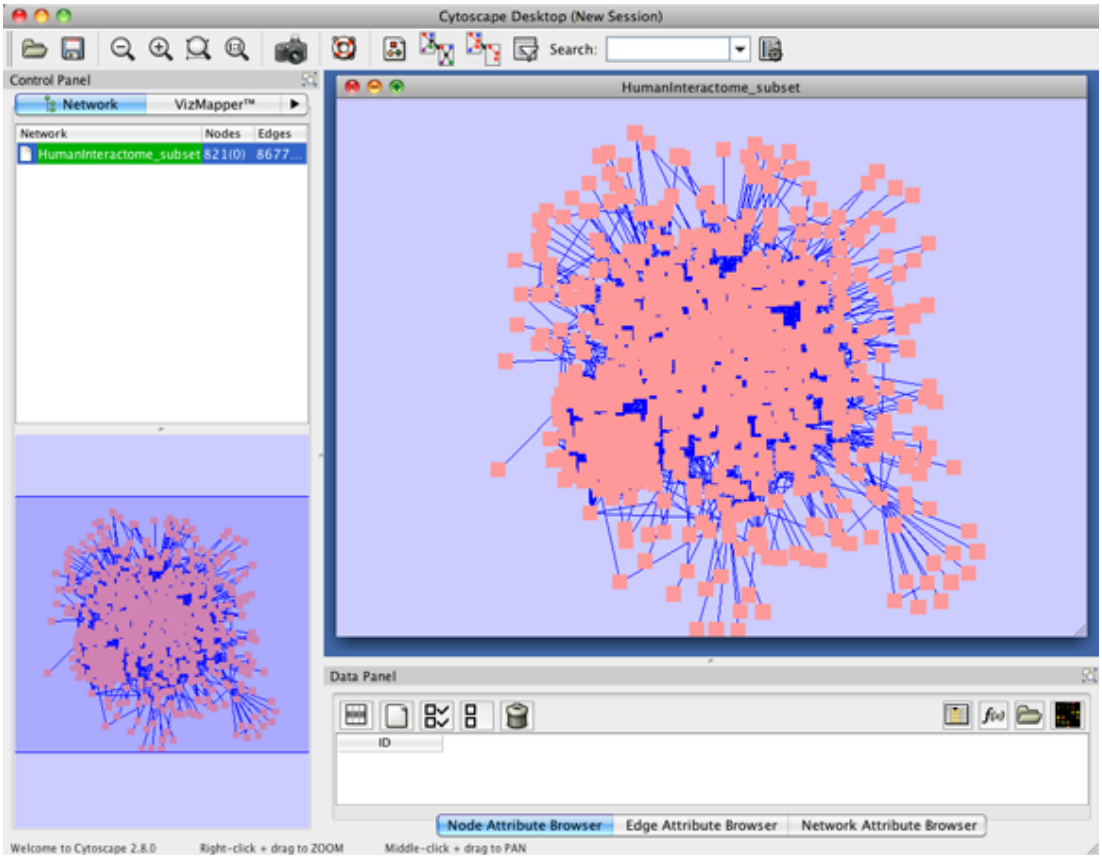
Cytoscape is an open source software platform for *integrating*, *visualizing*, and *analyzing* measurement data in the context of networks. This tutorial will introduce you to:

- Combining data from two different sources: experimental data in the form of microarrray expression data and network data in the form of interaction data.
- Visualizing networks using expression data.
- Filtering networks based on expression data.

NOTE: The expression data used in this example has been pre-processed to work with the interaction network used.

Loading Network

- Start Cytoscape and load the network HumanInteractome_subset.cys.
- Cytoscape should now look similar to this:

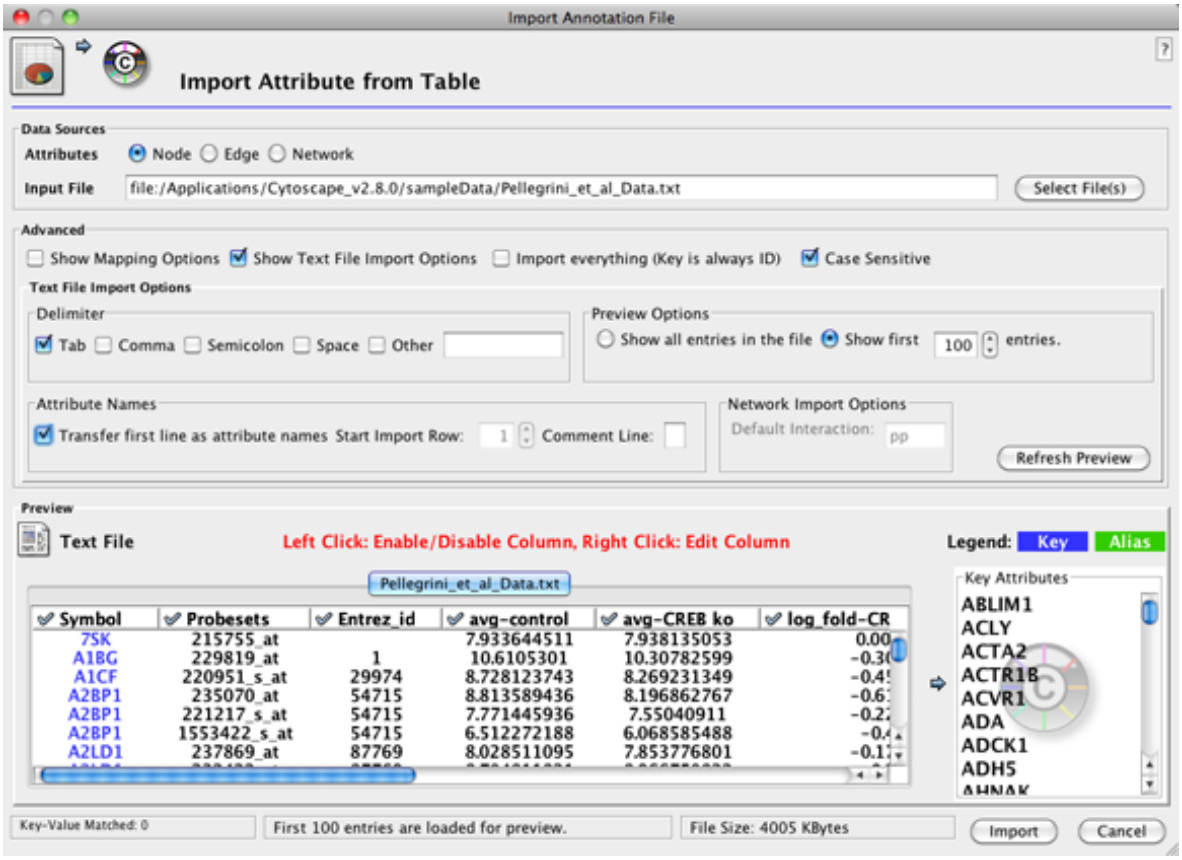


Loading expression data

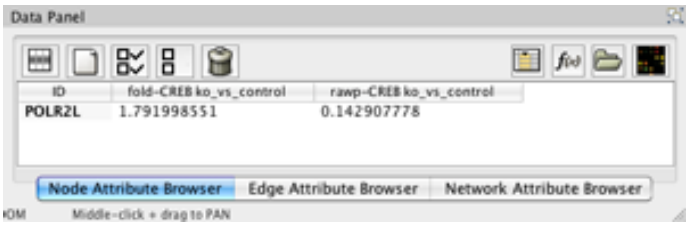
- Using your favorite text editor, open the file Pellegrini_et_al_Data.txt. The first few lines of the file are as follows:

```
Symbol      Probesets      avg control      avg CREB kd      log fold CREB kd vs con      fold CREB kd vs con      p value CREB kd vs con

SLC39A8      222935_x_at      9.428228086      6.494626196      -2.933601889              -7.640154905              5.38E-05

KLHL32      1553765_a_at      7.139322897      4.747085457      -2.392237439              -5.249708947              0.002462526

SH2D3C      215639_at      8.042897921      5.911418475      -2.131479446              -4.381665787           0.000311415
```

- You should note the following information about the file:
  - The first line consists of labels.
  - All columns are separated by a single tab character.
  - The first column contains node names, and must match the names of the nodes in your network exactly!
  - The second column contains Affymetrix probe set IDs. This column is optional, and the data is not currently used by Cytoscape, but this column may be useful for analysis in other microarray analysis tools.
  - The remaining columns contain experimental data; average expression for experimental and control groups, fold change and log fold for the comparison of experimental and control group, and raw and adjusted p value for the comparison.
- Under the **File** menu, select **Import → Attribute from Table (Text/MS Excel)**.
  - Click "Node" for the type of attribute to import.
  - Select the file Pellegrini_et_al_Data.txt.
  - Click the "Text File Import Options" check box.
  - Make sure the "Tab" check box in the "Delimiter" section is selected and that no other check box under "Delimiter" is selected. The preview should indicate that it is importing multiple columns of data.
  - Click the "Transfer first line as attribute names" check box in the "Attribute Names" section. The preview should now show be using the first row of the input file as column names and the import window should look like the image below.
  - Click the "Import" button to import the attribute data.



- Now we will use the **Node Attribute Browser** to browse through the expression data, as follows.
  - Select a node on the Cytoscape canvas by clicking on it.
  - In the **Node Attribute Browser**, click the **Select Attributes** button  , and select the attributes "fold CREB kd vs con", "log fold CREB kd vs con" and "p value CREB kd vs con" by left-clicking on them. Right-click to close the menu.
  - Under the **Node Attribute Browser**, you should see your node listed with their expression values, as shown.



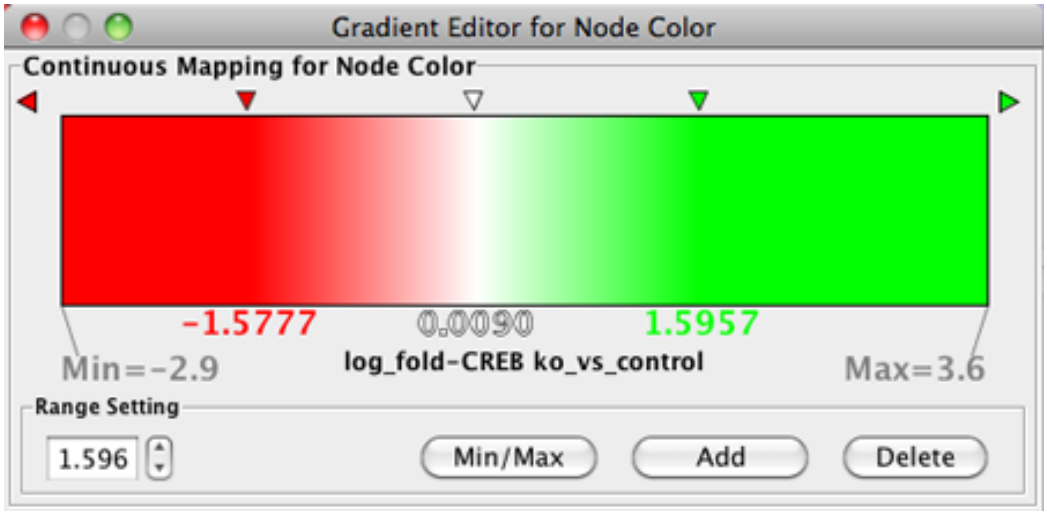
Visualizing Expression Data

Probably the most common use of expression data in Cytoscape is to set the visual attributes of the nodes in a network according to expression data. This creates a powerful visualization, portraying functional relation and experimental response at the same time. Here, we will walk through the steps for doing this.

Color the nodes

- Define the node color of this visual style.
  - Double-Click the **Node Color** row in the **Visual Mapping Browser** in the **Unused Visual Properties** Section.
  - This action will move **Node Color** to the top of the **Visual Mapping Browser**.
  - Click the "Please select a value!" cell in the **Node Color** section.
  - This will produce a drop-down menu of available attribute names. Select "log fold CREB kd vs con".

- Click the "Please select a mapping" cell in the **Node Color** section.
- This will produce a drop-down menu of available mapping types. Select "Continuous Mapping".
- This action will produce a basic black to white color gradient.
- Click on the **Min/Max** button and type in the minimum and maximum data values for "log fold CREB kd vs con" (-2.9 and 3.6). If you don't know the range of values in your data, open the data file in Excel and sort on the column if interest to get the min and max values.
- Click on the color gradient to change the colors. This will pop-up a gradient editing dialog.
- Double-Click on the left-most black triangle to change the low boundary color. Choose a bright red color.
- Repeat for the second black triangle. This will change the full gradient from red to white.
- Click on the left most white triangle and slide it towards the center of the scale so that its value is close to 0.
- Click the **Add** button to add a new white triangle to the scale.
- Double-Click this new triangle and select a bright green color.
- Double-Click the far right white triangle and select the same bright green color.
- This should produce a full Red-White-Green Color gradient like the image below.
- Close the gradient adjustment dialog and verify that the nodes in the network reflect the new coloring scheme.

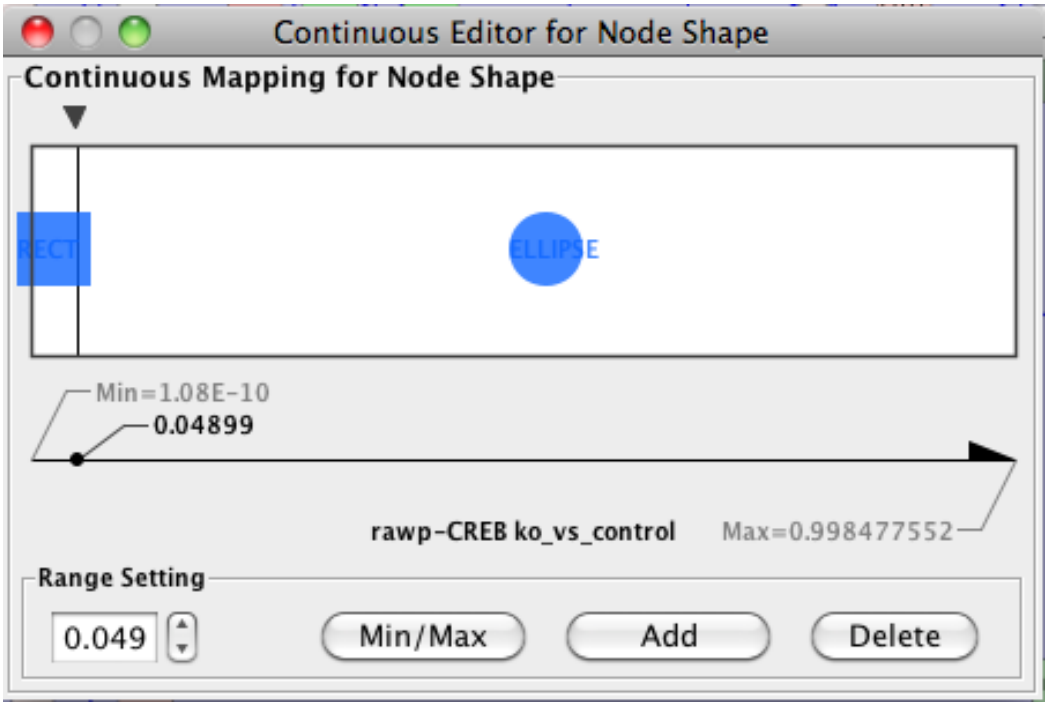


Set the default node color

- Note that the default node color of pink falls within this spectrum. A useful trick is to choose a color outside this spectrum to distinguish nodes with no defined expression value and those with slight repression.
  - Click the **Defaults** network icon in the **VizMapper** panel.
  - Click the **Node Color** entry and choose a dark gray color.
  - Zoom out on the network view to verify that a few nodes have been colored gray.

Set the Node Shape

- We imported both a fold change value and a p value for the comparison. We can use the p values to change the shape of the nodes so that measurements we have confidence in appear as squares while potentially bad measurements appear as circles.
  - Double-Click the **Node Shape** row in the **Visual Mapping Browser** in the **Unused Visual Properties** Section.
  - This action will move **Node Shape** to the top of the **Visual Mapping Browser**.
  - Click the "Please select a value!" cell in the **Node Shape** section.
  - This will produce a drop-down menu of available attribute names. Select "p value CREB kd vs con".
  - Click the "Please select a mapping" cell in the **Node Shape** section.
  - This will produce a drop-down menu of available mapping types. Select "Continuous Mapping".
  - This will create an empty icon in the "Graphical View" row of the **Node Shape** section. Click on this icon.
  - This action will pop-up a continuous shape selection dialog.
  - Click the **Add** button.
  - This action will split the range of values with a slider down the middle with a node shape icon to either side of the slider.
  - Double-Click on the left node icon (a circle).
  - This will pop-up a node shape selection dialog.
  - Choose the **Rectangle** shape and click the **Apply** button.
  - The continuous shape selection dialog should now show both a square and a circle node shape icon.
  - Click on the black triangle and move the slider to the left, to slightly lower that 0.05, our threshold for significance.
  - Close the continuous shape selection dialog and verify that some nodes have a square shape and some nodes have a circular shape.



Data analysis features

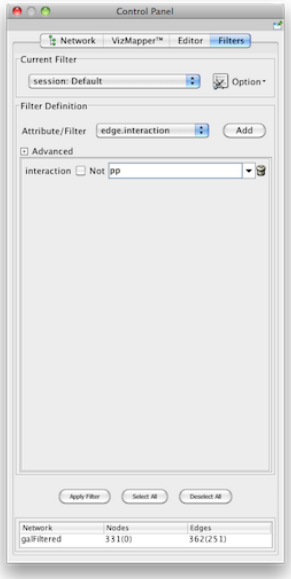
This section presents a few examples of features in Cytoscape that can be used to further analyze the network and associated data.

First, here is some background on your data. The data is from an experiment in a human myeloid leukemia cell line. The cAMP Response Element Binding Protein, CREB, was knocked down by shRNA and the expression profile of knockdown cells was compared to that of control cells from the same cell line. See Pellegrini et al [3]

Filter Interactions

Your network contains protein-protein interactions detected by multiple methods. Here, we filter out interactions based on the interaction detection method.

- Click the **Filters** tab in the **Control Panel**.
- Click the **Attribute/Filter** chooser in the **Filter Definition** and choose "edge.Interaction detection methods".
- Click the **Add** button in the **Filter Definition** section to add the selected attribute to the filter.
- This action will create a text search box entry in the filter.
- Type the letters "coip" into the text search box. This indicates that we're searching for all edge interaction attributes that match the string "coip".
- Click the **Apply Filter** button at the bottom of the Filters panel.

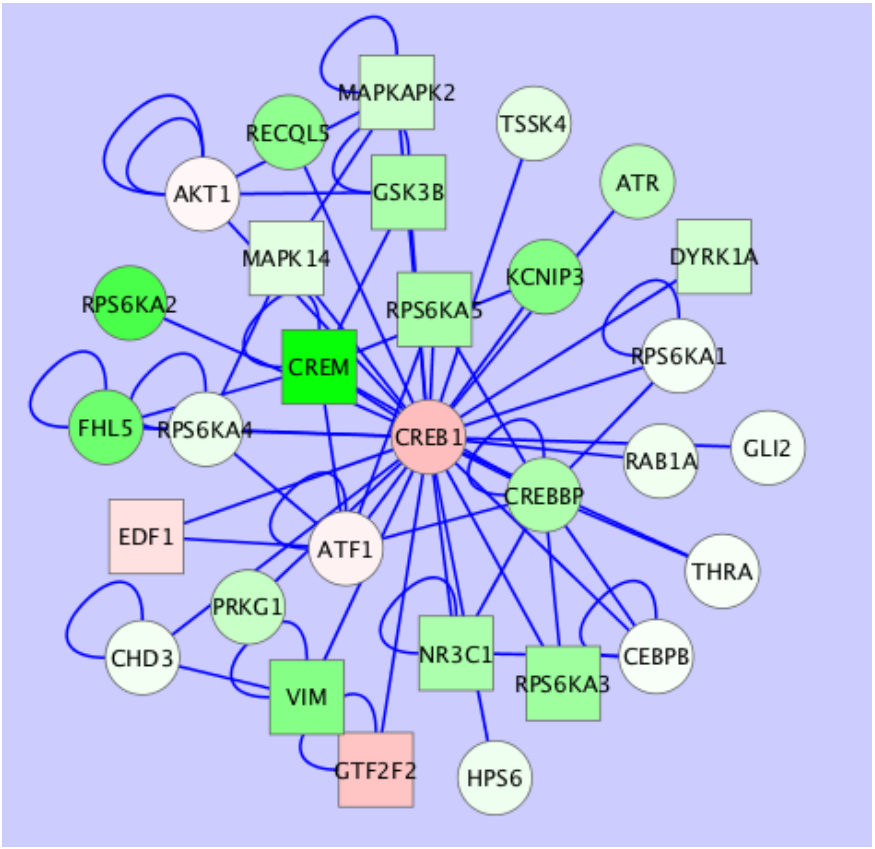


- You should now see many edges in the network selected (i.e. colored red).
- Select the menu **Edit → Delete Selected Nodes and Edge**.
- Select the menu **Layout → Cytoscape Layouts → Force-Directed Layout** to clean up the network visualization. You will see some unconnected nodes once the layout is finished.

Search for a node

We will now search for the CREB1 (CREB) node in the network.

- In the toolbar, to the right of the **Search** ***box, click the icon for*** **'Configure search options**.
- In the dialog that opens, select the radio button for **Nodes** and make sure **Unique Identifier** is selected in the drop-down. Click **Apply**.
- In the search field, type in "CREB". In the list of hits that is generated, you will see that there is one node named CREB1, which is an alias name for the CREB transcription factor. Select this node from the list and click Enter.
- The CREB node will be highlighted in the network.
- To make it easier to explore the interactions immediately surrounding CREB, we can create a network based on the first degree neighbors of CREB by clicking **Select → Nodes → First Neighbors of Selected Nodes**.
- A set of nodes should now be highlighted. Click **File → New → Network → From Selected Nodes, All Edges**. A new network will be produced.
- Clean up the network by applying a force-directed layout.
- The network should now look like this:



Exploring Nodes

- Right click on the node CREB1.
- Select the menu **LinkOut → Entrez → Gene**.
- This action will pop-up a browser window and search the Entrez Gene database for CREB.

References

[1] http://opentutorials.rbvi.ucsf.edu/index.php?title=Tutorial:Basic_Expression_Analysis_in_Cytoscape&ce_slide=true&ce_style=cytoscape

[2] http://opentutorials.rbvi.ucsf.edu/index.php/File:Basic_Expression_Analysis_in_Cytoscape.pdf

[3] http://www.ncbi.nlm.nih.gov/sites/ppmc/articles/PMC2647550/

Article Sources and Contributors

Tutorial:Basic Expression Analysis in Cytoscape *Source*: http://opentutorials.cgl.ucsf.edu/index.php?oldid=1232 *Contributors*: AlexanderPico, KristinaHanspers, MikeSmoot

Image Sources, Licenses and Contributors

File:Interactome_subnet.png *Source*: http://opentutorials.cgl.ucsf.edu/index.php?title=File:Interactome_subnet.png *License*: unknown *Contributors*: KristinaHanspers
File:LoadExpData.png *Source*: http://opentutorials.cgl.ucsf.edu/index.php?title=File:LoadExpData.png *License*: unknown *Contributors*: KristinaHanspers
File:select.png *Source*: http://opentutorials.cgl.ucsf.edu/index.php?title=File:Select.png *License*: unknown *Contributors*: KristinaHanspers
File:BrowseData.png *Source*: http://opentutorials.cgl.ucsf.edu/index.php?title=File:BrowseData.png *License*: unknown *Contributors*: KristinaHanspers
File:NodeColorGradient.png *Source*: http://opentutorials.cgl.ucsf.edu/index.php?title=File:NodeColorGradient.png *License*: unknown *Contributors*: KristinaHanspers
File:NodeShapeEditor.png *Source*: http://opentutorials.cgl.ucsf.edu/index.php?title=File:NodeShapeEditor.png *License*: unknown *Contributors*: KristinaHanspers
File:filter_panel.png *Source*: http://opentutorials.cgl.ucsf.edu/index.php?title=File:Filter_panel.png *License*: unknown *Contributors*: MikeSmoot
File:CREB-network.png *Source*: http://opentutorials.cgl.ucsf.edu/index.php?title=File:CREB-network.png *License*: unknown *Contributors*: KristinaHanspers

License