BMC
Research Notes

**TECHNICAL NOTE**                                                        **Open Access**

# CytoITMprobe: a network information flow plugin for Cytoscape

Aleksandar Stojmirović, Alexander Bliskovsky and Yi-Kuo Yu[*]

## Abstract

**Background:** Cytoscape is a well-developed flexible platform for visualization, integration and analysis of network data. Apart from the sophisticated graph layout and visualization routines, it hosts numerous user-developed plugins that significantly extend its core functionality. Earlier, we developed a network information flow framework and implemented it as a web application, called *ITM Probe*. Given a context consisting of one or more user-selected nodes, *ITM Probe* retrieves other network nodes most related to that context. It requires neither user restriction to subnetwork of interest nor additional and possibly noisy information. However, plugins for Cytoscape with these features do not yet exist. To provide the Cytoscape users the possibility of integrating *ITM Probe* into their workflows, we developed *CytoITMprobe*, a new Cytoscape plugin.

**Findings:** *CytoITMprobe* maintains all the desirable features of *ITM Probe* and adds additional flexibility not achievable through its web service version. It provides access to *ITM Probe* either through a web server or locally. The input, consisting of a Cytoscape network, together with the desired origins and/or destinations of information and a dissipation coefficient, is specified through a query form. The results are shown as a subnetwork of significant nodes and several summary tables. Users can control the composition and appearance of the subnetwork and interchange their *ITM Probe* results with other software tools through tab-delimited files.

**Conclusions:** The main strength of *CytoITMprobe* is its flexibility. It allows the user to specify as input any Cytoscape network, rather than being restricted to the pre-compiled protein-protein interaction networks available through the *ITM Probe* web service. Users may supply their own edge weights and directionalities. Consequently, as opposed to *ITM Probe* web service, *CytoITMprobe* can be applied to many other domains of network-based research beyond protein-networks. It also enables seamless integration of *ITM Probe* results with other Cytoscape plugins having complementary functionality for data analysis.

**Keywords:** Information flow, Interaction networks, Network analysis, Cytoscape plugin

## Findings

### Background

Cytoscape [1-3] is a popular and flexible platform for visualization, integration and analysis of network data. Apart from the sophisticated graph layout and visualization routines, its main strength is in providing an API that allows developers other than its core authors to produce extension plugins. Over the last decade, a large number of plugins have been released, supporting the features such as import and export of data, network

analysis, scripting and functional enrichment analysis. In this paper, we describe *CytoITMprobe*, a plugin that brings to Cytoscape new functionality founded on information flow.

Numerous approaches for analyzing biological networks based on information flow [4-10] have emerged in recent years. The main assumption of all such methods is information transitivity: information can flow through or can be exchanged via paths of biological interactions. Our contribution to this area [11,12] is a context-specific framework based on discrete-time random walks (or equivalently, diffusion) over weighted directed graphs. In contrast to most other approaches, our framework explicitly accommodates directed networks as well as the

*Correspondence: yyu@ncbi.nlm.nih.gov
National Center for Biotechnology Information, National Library of Medicine, NIH, Bethesda, MD 20894, USA

BioMed Central

information loss and leakage that generally occurs in all networks. Apart from the network itself and a user-specified context, it requires no prior restriction to the sub-network of interest nor additional and possibly noisy information. We implemented our framework as an application called *ITM Probe* [13] and made it available as a web service [14], where users can query protein-protein interaction (PPI) networks from several model organisms and visualize the results.

In addition to implementing network flow algorithms, the *ITM Probe* web service possesses a number of useful features. Using the Graphviz [15] suite for layout and visualization of graphs, it displays in the user's web browser the images of subnetworks consisting of nodes identified as significant by the information flow models and offers a choice of multiple coloring schemes. The entire query results can be retrieved in the CSV format or forwarded to a functional enrichment analysis tool to facilitate their interpretation. However, lacking a mechanism to decouple the algorithmic part from the interaction graph, the *ITM Probe* web service restricts users to querying only the few compiled PPI networks available on the website. Using a canned suite for graph layout, *ITM Probe* limits the users' ability to manipulate network images. For example, the only way to change the layout of significant subnetworks is to choose a different seed and re-compute the layout. Most importantly, not having an adequate interface to a well-designed platform such as Cytoscape, it is difficult to use the results of the *ITM Probe* service within the workflows involving additional data and algorithms from other sources. We thus developed *CytoITMprobe* to meet these challenges by (1) providing an explicit decoupling between the algorithmic part and the interaction graph, (2) utilizing the core graph manipulation functionality of Cytoscape for a broader visualization choices, and (3) adding an appropriate input/output interface for seamless integration with other resources available in Cytoscape.
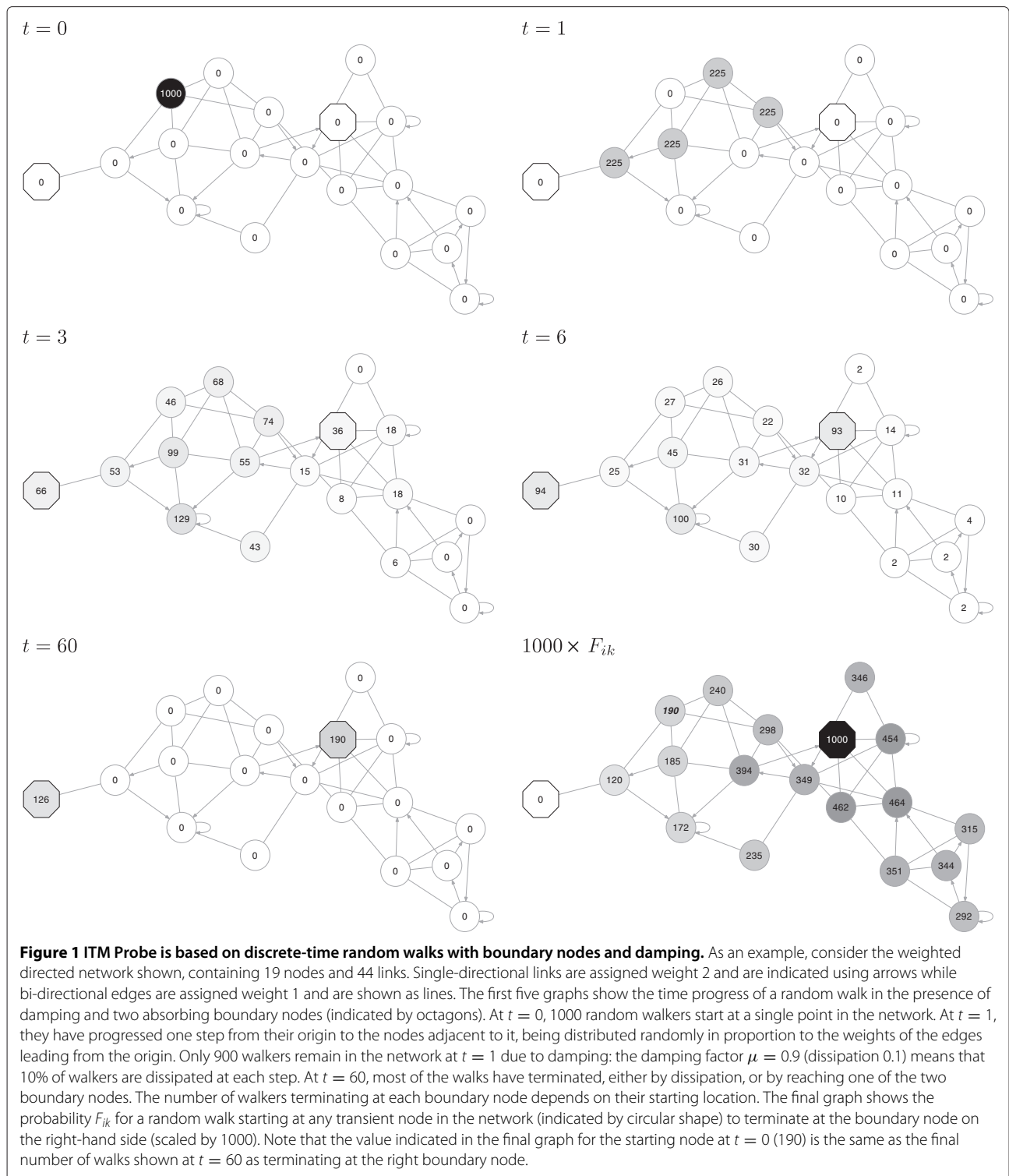
### Information Flow Framework

*ITM Probe* extracts *context-specific* information from networks. We elaborated on the information flow framework underlying *ITM Probe* in our previous publications [11,12] and here we provide a non-technical explanation. Given a context consisting of one or more user-selected network nodes, the aim is to retrieve a set of other network nodes most related to that context. We model networks as weighted directed graphs, where nodes are linked by directional edges and each edge is assigned a positive weight. One can consider a random walker that wanders among network nodes in discrete steps. The rule of the walk is that the walker starts at a certain node and in each step moves randomly to some adjacent node with probability proportional to the weight of the edge linking these

nodes (Figure 1). If the graph is connected, that is, if there is a directed path linking any two nodes, such a walk never terminates and the walker will eventually visit every node in the graph.

Our main idea is to set termination or *boundary* nodes for the walkers while using random walks to explore the neighborhoods of the context nodes. Provided there is a directed path linking any node to a boundary node, every random walk here will eventually terminate. Furthermore, the nodes visited by a walker before termination will vary depending on the origin of the walk. Since a random walk is a stochastic process, and each walk is different, we are interested in the cumulative behavior of infinitely many walkers following the same rules. On average, we expect that the nodes more relevant to the context will be more visited than those that are less relevant. Thus, the main quantity of interest is the average number of visits to a node given the selected origins and destinations of the walk.
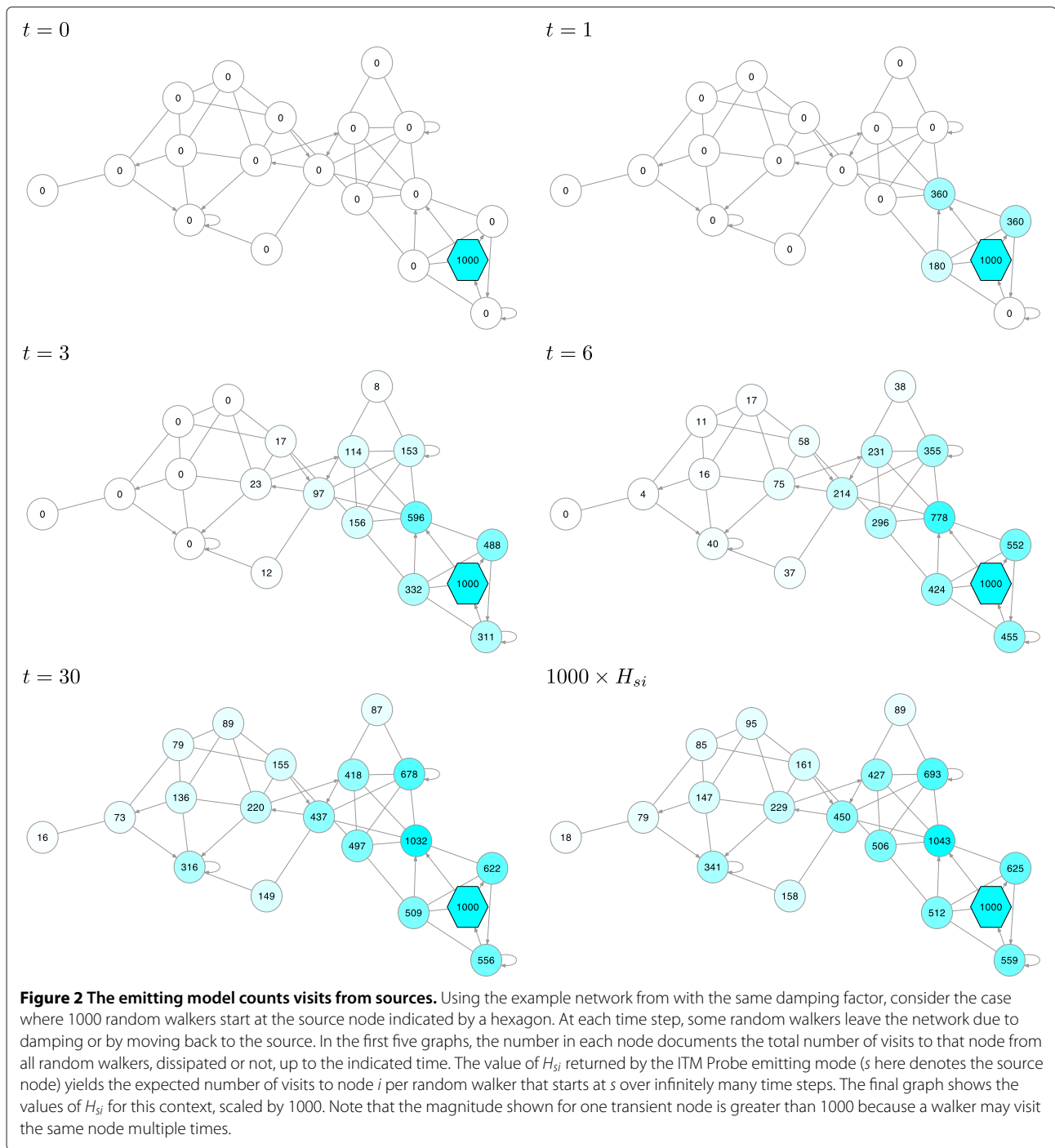
A problem with the above approach is that random walkers may spend too much time in the graph if the origins and destinations of the walk are far apart. This could mean that the entire graph is visited so that the most significant nodes are just those with the largest degree. To ensure that the significant nodes are relatively close to the context nodes, our framework contains an additional ingredient, *damping*: at each step of a walk, we assign a certain probability for the walker to dissipate, that is, to leave the network. We still evaluate the average number of visits to each node, but now only count the visits prior to the walker leaving the network. Evidently, the nodes that are close to the walker's origin will be significantly visited. In addition to forcing locality, damping is also natural in physical or biological contexts. If we treat random walkers as information propagating through the network, it is natural to assume that some information is lost during transmission. For protein-protein interaction networks, where nodes are proteins and links are physical bindings between proteins, damping could be associated with protein degradation by proteases, which would diminish the strength of information propagation.

*ITM Probe* framework contains three models: *emitting*, *absorbing* and *channel*. In the absorbing model (Figure 1), the context nodes are interpreted as destinations or *sinks* of random walks, while every non-boundary or *transient* node is considered as a potential origin. For each transient node $i$ and each sink $k$, the model computes $F_{ik}$, the average number of visits to the terminating node $k$ by random walks originating at the node $i$. Since a walk can either terminate at one sink or the other, $F_{ik}$ can also be interpreted as the probability that a random walk from $i$ reaches $k$. In the absence of damping, the sum of $F_{ik}$ over all sinks will be exactly 1 for any transient node $i$. However, in the presence of damping, the sum of $F_{ik}$ over all

**Figure 1 ITM Probe is based on discrete-time random walks with boundary nodes and damping.** As an example, consider the weighted directed network shown, containing 19 nodes and 44 links. Single-directional links are assigned weight 2 and are indicated using arrows while bi-directional edges are assigned weight 1 and are shown as lines. The first five graphs show the time progress of a random walk in the presence of damping and two absorbing boundary nodes (indicated by octagons). At $t = 0$, 1000 random walkers start at a single point in the network. At $t = 1$, they have progressed one step from their origin to the nodes adjacent to it, being distributed randomly in proportion to the weights of the edges leading from the origin. Only 900 walkers remain in the network at $t = 1$ due to damping: the damping factor $\mu = 0.9$ (dissipation 0.1) means that 10% of walkers are dissipated at each step. At $t = 60$, most of the walks have terminated, either by dissipation, or by reaching one of the two boundary nodes. The number of walkers terminating at each boundary node depends on their starting location. The final graph shows the probability $F_{ik}$ for a random walk starting at any transient node in the network (indicated by circular shape) to terminate at the boundary node on the right-hand side (scaled by 1000). Note that the value indicated in the final graph for the starting node at $t = 0$ (190) is the same as the final number of walks shown at $t = 60$ as terminating at the right boundary node.

sinks may be much less than 1 (Figure 1). The emitting model (Figure 2), offers a dual point of view. Here, the context nodes are interpreted as origins or *sources* of random walks. The walks terminate by dissipating or by returning to the sources – the sources form an emitting boundary.

Since the origins of the walks are fixed, the quantity of interest is the visits to the transient nodes. Specifically, for each source $s$ and each transient node $i$, the emitting model returns $H_{si}$, the average number of visits to $i$ by walkers originating at $s$.

**Figure 2 The emitting model counts visits from sources.** Using the example network from with the same damping factor, consider the case where 1000 random walkers start at the source node indicated by a hexagon. At each time step, some random walkers leave the network due to damping or by moving back to the source. In the first five graphs, the number in each node documents the total number of visits to that node from all random walkers, dissipated or not, up to the indicated time. The value of $H_{si}$ returned by the ITM Probe emitting mode (*s* here denotes the source node) yields the expected number of visits to node *i* per random walker that starts at *s* over infinitely many time steps. The final graph shows the values of $H_{si}$ for this context, scaled by 1000. Note that the magnitude shown for one transient node is greater than 1000 because a walker may visit the same node multiple times.

The values of $F_{ik}$ and $H_{si}$ can be efficiently computed by solving (sparse) systems of linear equations. Let $W_{ij}$ denote the weight of the directed link $i \rightarrow j$ and let $0 < \mu < 1$ denote the damping factor. For all pairs of nodes $i, j$, construct the random walk evolution operator **P**, where $P_{ij} = \frac{\mu W_{ij}}{\sum_{j'} W_{ij}}$. The operator **P** includes damping and hence $\sum_j P_{ij} < 1$. Let $\mathbf{P}_{TT}$ denote the sub-operator of **P** with domain and range restricted only to transient

nodes and let $\mathbf{G} = (\mathbb{I} - \mathbf{P}_{TT})^{-1}$, where $\mathbb{I}$ stands for the identity matrix. Then, it can be shown [11], that

$$F_{ik} = \sum_j G_{ij} P_{jk}, \qquad \text{and}$$

$$H_{si} = \sum_j P_{sj} G_{ji}.$$

More details, including the cases where $\mu = 0$, $\mu = 1$ or non-uniform damping are covered in [11,12].

The channel model combines the emitting and the absorbing model, with both sources and sinks on the boundary. It illuminates the most likely paths from sources to sinks. For each source node $s$, transient node $i$ and sink node $k$, it computes $\Phi_{i,k}^s = H_{si}F_{ik}$, the average number of visits to $i$ by a random walker that originates at $s$ and terminates at $k$. *ITM Probe* does not report $\Phi_{i,k}^s$ directly, but instead shows a simpler, *normalized* quantity $\hat{\Phi}_{i,K}^s$ (Figure 3), which is defined for each source $s$ and transient node $i$ by

$$\hat{\Phi}_{i,K}^s = \frac{\sum_k H_{si}F_{ik}}{\sum_{k'} F_{sk'}}. \tag{1}$$

Here, the numerator $\sum_k H_{si}F_{ik} = \sum_k \Phi_{i,k}^s$ gives the average number of visits, in the presence of damping, to $i$ by a random walker starting at $s$ and terminating at any sink. The denominator gives the total probability of a walker starting at $s$ to terminate at any sink. Hence, with the denominator off-setting the effect of damping, the value of $\hat{\Phi}_{i,K}^s$ counts the average number of visits to $i$ by walkers that start at $s$ and terminate at any of the sinks as if no dissipation is present. Generally, damping in the emitting or the absorbing model determines how far the flow can reach away from its origins. In contrast, the damping parameter for the normalized channel model plays a different role (Figure 4): it effectively determines the 'width' of the channel from sources to sinks. When damping is very strong, only the nodes on the shortest path from a source to its nearest sink will be visited.

Given the close relationship between random walks and diffusion, it is also possible to interpret *ITM Probe* models through information diffusion (or information flow). Within that paradigm, a fixed amount information is constantly replenished at the source nodes while leaving the network at all boundary nodes and through dissipation. At equilibrium, when the rate of flow entering equals the rate of leaving, the amount of information occupying each transient node is equivalent to the average number of visits to that node (using the aforementioned non-replenishing random walk interpretation [11]). We call the set of nodes most influenced by the flow an *Information Transduction Module* or ITM.

### Software architecture

*CytoITMprobe* architecture consists of two parts: the user interface front end and computational back end. The user interface, written in Java [16] using Cytoscape API, is accessed as a Cytoscape plugin. It consists of the query form, the results viewer and the ITM subnetwork (Figure 5). The back end is the standalone *ITM Probe* program, written in Python, which can be installed locally or accessed through a web service. In either configuration, *CytoITMprobe* takes the user input through the graphical user interface, validates it, and passes a query to the back end. Upon receiving from the back end the entire query results, *CytoITMprobe* stores them as the node and network attributes of the original network. Consequently, the query output can be edited or manipulated within Cytoscape, as well as saved for later use.

Standalone *ITM Probe* is a part of the *qmbpmn-tools* Python package, which also contains the code supporting the *ITM Probe* and *SaddleSum* web services, as well as the scripts for constructing the underlying datasets. The *ITM Probe* part depends on Numpy and Scipy [17] packages for numerical computations. The performance of *ITM Probe* critically depends on the routines for computing direct solutions of large, sparse, nonsymmetric systems of linear equations. Scipy supports two sparse direct solver libraries (both written in C): SuperLU [18] as default and UMFPACK [19] as an optional add on through SciKits collection [20]. In our experience, UMFPACK runs faster than SuperLU and Scipy always uses it if available. However, for optimal performance, UMFPACK requires well-tuned Basic Linear Algebra Subroutines (BLAS) libraries and may not be easy to install. To support users who are inclined not to install UMFPACK or Scipy, *CytoITMprobe* supports remote queries by default.

### Input

*CytoITMprobe* requires as input a weighted directed graph and the *ITM Probe* model parameters that include a selection of boundary nodes and a dissipation probability.

### *Step one: defining a query graph*

In *CytoITMprobe* graph connectivity is specified by selecting a Cytoscape network. In addition, each link must be assigned a weight and a direction through the query form. Edge weights are set using the *Weight attribute* dropdown box, which lists all available floating-point edge attributes of the selected network and the default option (*NONE*). If the default option is selected, *CytoITMprobe* assumes a weight 2 for any self-pointing edge and 1 for all other edges. If an attribute is selected, the weight of an edge is set to the value of the selected attribute for that edge. Null attribute values are treated as zero weights.

Since Cytoscape edges are always internally treated as directed, the user must also indicate the directedness of each edge type through the query form. Whenever a new Cytoscape network is selected, *CytoITMprobe* updates the query form and places all of the network's edge types into the *undirected* category. The user can use arrow buttons to move some edge types to the *directed* or *ignored* category. Undirected edges are treated as bidirectional, with the same weight in both directions. Directed edges have a specified weight assigned
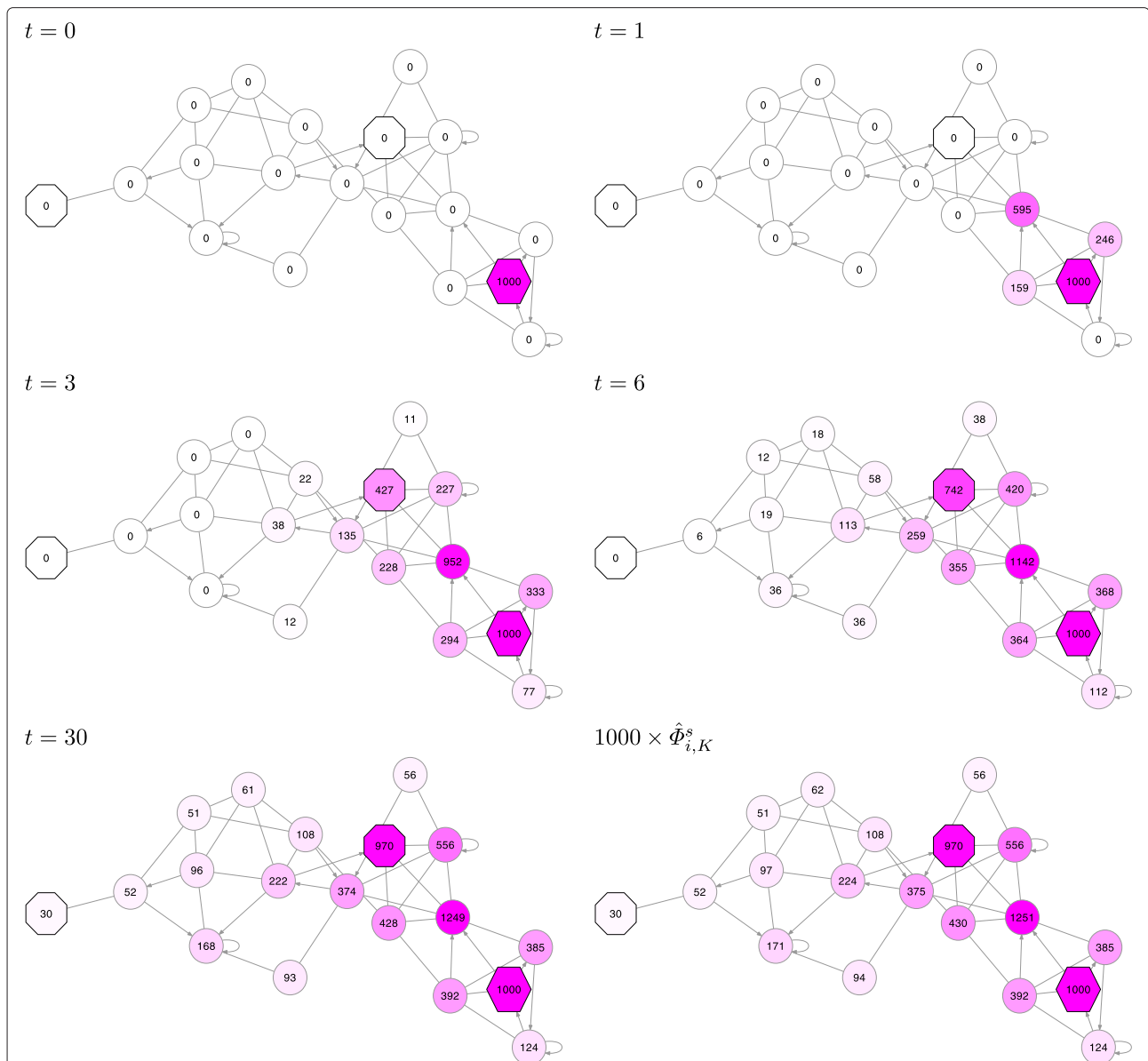
**Figure 3 The channel model highlights the directed flow from origins to destinations.** Consider once again the example network from Figures 1 and 2, now with a single source (hexagon) and two sinks (octagons). In common with the case from Figure 2, the walkers start at the source, but in this case can terminate only by reaching the sinks. The damping factor is implicit: it determines how far the walkers are allowed to deviate from the shortest path towards one of the sinks. In the first five graphs, the number in each transient node documents the total number of visits to that node from all random walkers up to the indicated time. However, the value in each sink node represents the likelihood to reach that sink from the source at the indicated time. The value of $\hat{\Phi}_{i,K}^{s}$ returned by the ITM Probe normalized channel mode yields the expected number of visits to node $i$ per random walker that starts at $s$ over infinitely many time steps. Note that the sink nodes split the flow from the source depending on their location. In this example, over infinitely many time steps, the node closer to the source captures 970 walkers, while the farther sink gets only the remaining 30.

only in the forward direction, with the backward direction receiving the zero weight. Ignored edges have zero weight in both directions. Since Cytoscape allows multiple edges of different types between the same nodes, *CytoITMprobe* collapses multiple edges in each direction into a single edge by appropriately summing their weights (Figure 6).

**Step two: selecting a model and boundary nodes**

In addition to a weighted directed graph, *ITM Probe* requires an information flow model (emitting, absorbing or normalized channel), a selection of sources and/or sinks, and dissipation probability. The choice of the model determines the types of boundary nodes that need to be specified, as well as the ways in which the damping factor
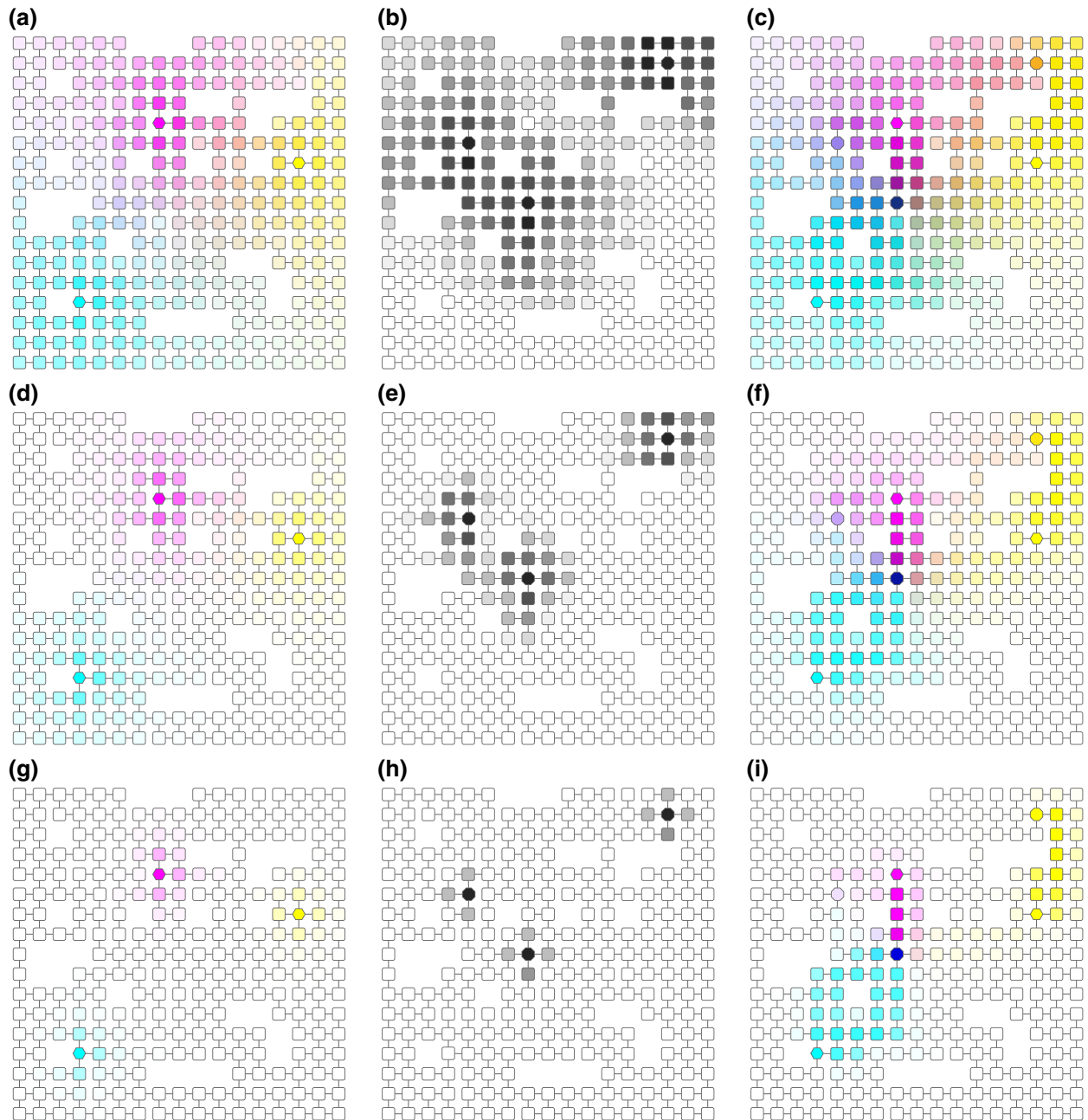
**Figure 4 An example of the results of running different *ITM Probe* models.** Here we see the results of running the emitting (a,d,g), absorbing (b,e,h) and channel (c,f,i) model of *ITM Probe* with the same sources and sinks but different dissipation coefficients. The underlying undirected graph is derived from a square lattice by removing random nodes and edges. Sources are shown as hexagons, sinks as octagons, and transient nodes as squares. The top row (a,b,c) shows the runs with damping factor $\mu = 0.95$ (dissipation 0.05), the middle (d,e,f) with $\mu = 0.75$ and the bottom with $\mu = 0.25$. For the emitting and channel model, each basic cyan, magenta or yellow color is associated with a source. The coloring of each node arises by mixing the basic color in proportion to the strength of information flow from their respective sources. For the absorbing model, the nodes are shaded according to the total probability of absorption at any sink on a logarithmic scale.
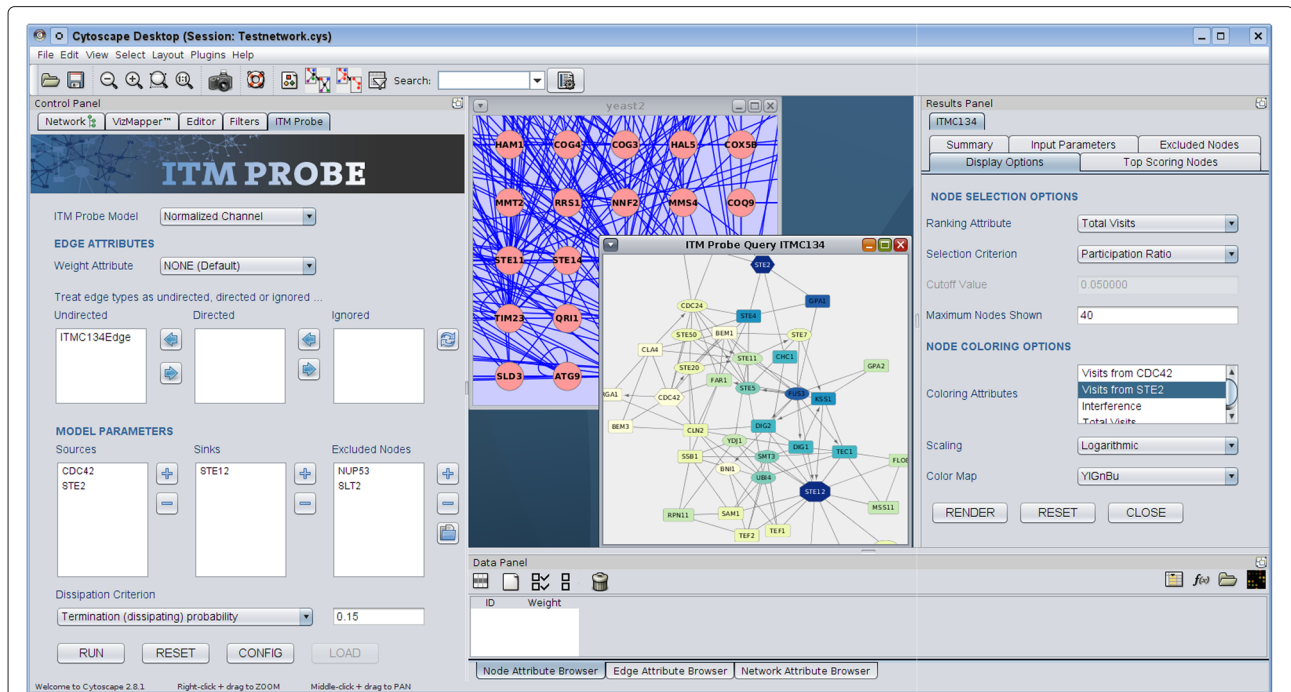
**Figure 5 CytoITMprobe interface.** At startup from the Plugins menu, *CytoITMprobe* embeds its query form into the Control Panel (left). After performing a query or loading previously obtained search results, it creates an ITM subnetwork showing significant nodes and a viewer embedded into Results Panel (right). The viewer allows closer examination of the results and manipulation of the contents and the look of the ITM subnetwork. The overall visual styling of *CytoITMprobe* components closely resembles that of the *ITM Probe* web version.

can be set (see 'Step three: specifying dissipation probability' below). The query form also allows users to specify *excluded nodes*. Any flow reaching excluded nodes is fully dissipated. This is a way to remove those nodes that do not participate in information propagation in the desired context or that introduce undesirable shortcuts.

***Step three: specifying dissipation probability***
The values of $H$, $F$, and $\hat{\Phi}$, all implicitly depend on the dissipation probability. In *ITM Probe* the user can set the dissipation probability directly or specify a related quantity that can, using Newton's method, determine the
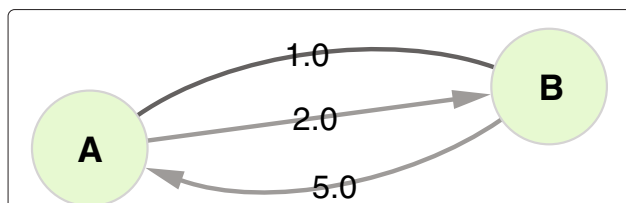


**Figure 6 Edge weights example.** Consider the following example: Suppose A and B are nodes in a Cytoscape network linked by three edges of two types with shown edge weights. Assume two type I edges (lighter gray), $A \rightarrow B$ and $B \rightarrow A$ are directed, while a single type II edge (darker gray) $A \rightarrow B$ is undirected. At query time, *CytoITMprobe* creates two directed edges, $A \rightarrow B$ and $B \rightarrow A$, with weights 3 and 6, respectively.

dissipation probability. The choice of the alternative quantity depends on the selected model. For the emitting model, this quantity is the average path length before termination, which we denote by $\bar{t}$. For example, the user can require a random walker to make on average three steps before terminating. The formula for $\bar{t}$ is

$$\bar{t} = 1 + \frac{1}{n_S} \sum_s \sum_j H_{sj}, \qquad (2)$$

where $n_S$ denotes the number of sources. For the normalized channel model, the path length before termination is given by

$$\bar{t} = 1 + \frac{1}{n_S} \sum_s \sum_j \hat{\Phi}^s_{j,K}. \qquad (3)$$

Since the normalized channel model counts only the random walkers actually terminating at sinks, $\bar{t}$ is in this case bounded below by the length of the shortest path from any source to any sink. Hence, *ITM Probe* accepts the desired value of $\bar{t}$ in terms of length deviation from the shortest path. There are two ways to set the average path-length deviation: in absolute units (steps) or as a proportion of the length of the shortest path. The absorbing model allows users to obtain the dissipation probability by

setting the average absorption probability, denoted $\bar{r}$. The formula for $\bar{r}$ is

$$\bar{r} = \frac{1}{n_T} \sum_i \sum_k F_{ik}, \qquad (4)$$

where $k$ ranges over all sinks, $i$ ranges over all transient nodes *that are connected to at least one sink*, and $n_T$ is the total number of such nodes. The value of $\bar{r}$ represents the likelihood of a random walk starting at a randomly selected point in the network to reach a sink. The dissipation probability obtained in this way is larger if the sinks are well-connected hubs near the center of the network, in contrast to the case when the chosen sinks are not as well connected.

### Step four: submitting a query

After specifying all necessary input, the user submits a query by pressing the *QUERY* button on the query form. The time required for a run depends on whether the query is local or remote, as well as on the size of the submitted graph and the number of selected sources and/or sinks.

### Output

For every completed query, *CytoITMprobe* displays its results in a viewer embedded in Cytoscape Results Panel and a new Cytoscape network consisting of significant nodes (ITM subnetwork). The results viewer has five tabs: *Top Scoring Nodes, Summary, Input Parameters, Excluded Nodes,* and *Display Options*. The first four tabs contain information about the query and the results, while the last one contains a form that allows users to manipulate the ITM subnetwork. The form controls two aspects of the subnetwork: composition (what nodes are selected and how many) and node coloring.

### Displaying significant nodes

Subnetwork nodes are selected through a *ranking attribute*, which assigns a numerical value from *ITM Probe* results to each node. The nodes are listed in descending order of the ranking attribute and top nodes are displayed as the ITM subnetwork. The number of top nodes is determined by specifying a *selection criterion*, which can be simply a number of nodes to show, a cutoff value or the 'participation ratio'. Specifying a cutoff value $x$ selects the nodes with their ranking attribute greater than $x$. Participation ratio estimates the number of 'significant' nodes by considering all values of the ranking attribute in a scale-independent manner [11]. The available choices for the ranking attribute depend on the *ITM Probe* model and the number of boundary points. For the emitting and normalized channel model, the user can select visits to a node from each source or the sum of visits

from all sources. It is also possible to use *interference* [11], which denotes the minimum number of node visits, taken over all sources. For the absorbing model, the available attributes are absorbing probabilities to each sink and the total probability of termination at a sink. The values of all attributes for the subnetwork nodes are displayed in the *Top Scoring Nodes* tab.

The colors of the subnetwork nodes are determined by selecting *coloring attributes*, a *scaling function* and a *color map*. The list of coloring attributes is the same as the list of ranking attributes but the user can select up to three coloring attributes. If a single attribute is selected, node colors are determined by the selected eight-category ColorBrewer [21] color map. Otherwise, they are resolved by color mixing: each coloring attribute is assigned a single basic color (cyan, magenta or yellow), and the final node color is obtained by mixing the three basic colors in proportion to the values of their associated attributes at that node. The scaling function serves to scale and discretize the coloring attributes to the ranges appropriate for color maps. Figure 4 shows examples of mixed color scheme with three boundary points (left and right columns) and of a coloring using a single attribute (center column).

### Manipulating node attributes

Since the *ITM Probe* query results are saved as Cytoscape attributes of the original network, they can be arbitrarily modified through Cytoscape. Any changes made are reflected in the results viewer and the corresponding ITM subnetwork after pressing the *RESET* button on the Display Options form. Using the *CytoITMprobe* attribute nomenclature, users can create additional attributes to be used for ranking or coloring. Consider the following usage example. A user has run an emitting model query with three sources, `S1`, `S2`, and `S3`, and obtained the results in a viewer labeled `ITME243`. At the end of the run, *CytoITMprobe* created the attributes `ITME243[S1]`, `ITME243[S2]` and `ITME243[S3]` for the nodes of the input network and saved the results as their values. The user creates a new floating-point node attribute with a label `ITME243[avgS1S2]` and fills it with an average of `ITME243[S1]` and `ITME243[S2]`. After resetting the Display Options form, an item 'Custom `[avgS1S2]`' is available for selection as a ranking or coloring attribute. This gives the user the flexibility to reinterpret `S1` and `S2` as if they were a single source of equal weight as `S3`. Another possibility is to combine the results of queries with different boundaries and display them together on the same subnetwork.

### Saving and restoring results

The query network together with its attributes containing *ITM Probe* results can be saved as a Cytoscape session and later retrieved. After reloading the session, the user can

regenerate the results viewer and the corresponding sub-network for a stored ITM by pressing the *LOAD* button on the *CytoITMprobe* query form and selecting the desired ITM from a list.

Alternatively, the *ITM Probe* results can be exported to tab-delimited text files through the Cytoscape *Export* menu. Each exported tab-delimited file contains all the information necessary to restore the results except the query network and can be easily manipulated both by humans and by external programs or scripts. The results from tab-delimited files can be imported into any selected Cytoscape network through the *Import* menu. Since the selected network may be different from the original query network, only the results for the nodes in the selected network whose IDs match the IDs from the imported file will be loaded. After importing the results, *CytoITMprobe* generates a new results viewer and a subnetwork, as if the results originated from a direct *ITM Probe* query.

### Discussion

The main function of *ITM Probe*, also applicable to domains other than PPI networks, is to retrieve information from large and complex networks by discovering the possible interface between network nodes that are hypothesized to be related. This paradigm works best with large networks, where such information cannot be easily accessed by other means. For examples of applications of the *ITM Probe* frameworks to protein-protein interaction networks, consult our earlier papers [11-13].

With a network as an *encyclopedia* of domain-specific knowledge, *ITM Probe* enables a direct access to its specific portions related to a specified context. The user can learn about the objects representing individual nodes by setting them as sources and/or sinks and retrieving information about the most significant objects in the resulting ITM. This approach not only extracts a relevant subnetwork but also produces context-specific weights for each node. With their interpretation as average numbers of node visits, or equivalently, as average numbers of paths passing through a node, the ITM weights signify the relative importance of network nodes in the context of the query and thus can be used to refine its interpretation as a whole. For example, a single node with a large weight in an ITM resulting from a normalized channel model query represents a choke point *in the particular context of the query*. The same node need not have a high global centrality.

Containing both sources and sinks, the normalized channel model offers the users the ability to formulate and evaluate network based hypotheses *in silico*. Since information flow that reaches one sink cannot subsequently terminate at any other, sink nodes can be associated with alternative hypotheses, such as different biological functions if the network is PPI. The information flow from each source will then, depending on the dissipation coefficient used, mainly trace the path towards the sink most likely to be reached first from that source (see Figure 4, right column). The *ITM Probe* framework considers all weighted paths from sources to sinks and hence produces more robust results than approaches involving only the shortest paths. The path weights are tunable using the dissipation probability.

Compared to the previously described web interface to *ITM Probe* [13], *CytoITMprobe* significantly benefits from being a part of the Cytoscape platform. Although the *Display Options* form is very similar to the web version, the sophisticated network visualization functionality provided by Cytoscape allows significantly more versatility in displaying ITMs. For example, Cytoscape GUI allows users to manually alter node placements, rotate network views, or arbitrarily change the look of a network. In addition, Cytoscape interface enables users to directly manipulate node attributes representing *ITM Probe* results and possibly create new node summary variables appropriate to their problem. The newly created variables can be immediately reflected in the graphical representation of an ITM, which is not possible in the web setting. Most importantly, the results of *ITM Probe* can be integrated into workflows involving other Cytoscape plugins that provide complementary functionality. For instance, output ITMs can be related to terms from controlled vocabularies such as Gene Ontology [22] using functional enrichment analysis plugins such as PinGO [23] or our own recently released *CytoSaddleSum* [24]. The graph-theoretic structure of ITM subnetworks can be analyzed using a variety of algorithms such as MCODE [25] or GraphletCounter [26,27].

The architecture of *CytoITMprobe* with a Cytoscape front end and an *ITM Probe* back end offers flexibility for a variety of usage scenarios. In contrast to the web version, it allows users to use *ITM Probe* with arbitrary networks and edge weights, rather than being limited to compiled PPIs from few model organisms. Most users will be content with accessing *ITM Probe* through the web server. However, the option to download and install the *qmbpmn-tools* package provides not only faster running times for queries but also the ability to use the command line interface for *ITM Probe* to perform batch queries and to locally reproduce its web service. The separation of the presentation layers (web or Cytoscape) from the 'business' layer (standalone *ITM Probe*) facilitates easy future updates to any components.

### Conclusion

*CytoITMprobe* is a plugin that brings the previously unavailable network flow algorithms of *ITM Probe* to the Cytoscape platform. It enables users to extract context-specific subnetworks from large networks by specifying

the origins and/or destinations of information flow. *CytoITMprobe* significantly extends the features of the previously released web version of *ITM Probe.*

The main novelty of *CytoITMprobe* is that it allows the user to specify as input any Cytoscape network, rather than being restricted to the PPI networks available through the *ITM Probe* web service. Using Cytoscape attributes to hold their desired values, users may easily supply their own edge weights and denote edge directionalities. Additionally, the ability to manipulate and add new node attributes through Cytoscape reduces the workload required for visualizing various combinations of ITM components. In the context of biological cellular networks, this additional flexibility may lead to constructions of new node attributes that can better reflect biological significance, hence facilitating more educated hypothesis forming.

By bringing *ITM Probe* to Cytoscape, *CytoITMprobe* enables seamless integration of *ITM Probe* results with other Cytoscape plugins having complementary functionality for data analysis. By decoupling the query network from the information flow algorithm, the newly developed *CytoITMprobe* can be applied to many other domains of network-based research beyond protein-networks.

## Availability and requirements
### *CytoITMprobe* plugin
**Project name:** CytoITMprobe
**Project home page:** http://www.ncbi.nlm.nih.gov/ CBBresearch/Yu/downloads/itmprobe.html
**Documentation:** http://www.ncbi.nlm.nih.gov/ CBBresearch/Yu/mn/itm_probe/doc/cytoitmprobe.html
**Video tutorial:** http://www.youtube.com/watch?v=4Cdf-mSKtWo
**Operating system(s):** Platform independent
**Programming language:** Java
**Other requirements:** Java SE 6 or higher and Cytoscape 2.7 or higher
**License:** All components written by the authors at the NCBI are released into Public Domain. Components included from elsewhere are available under their own open source licenses and attributed in the source code.

### Standalone *ITM Probe* (optional for *CytoITMprobe*)
**Project name:** qmbpmn-tools
**Project home page:** http://www.ncbi.nlm.nih.gov/ CBBresearch/Yu/downloads/itmprobe.html
**Documentation:** http://www.ncbi.nlm.nih.gov/ CBBresearch/Yu/mn/itm_probe/doc/
**Operating system(s):** Platform independent
**Programming language:** Python
**Other requirements:** Python 2.6 or 2.7, Numpy 1.3 or higher and Scipy 0.7 or higher. UMFPACK Scikit is recommended for good performance.

**License:** All components written by the authors at the NCBI are released into Public Domain. Components included from elsewhere are available under their own open source licenses and attributed in the source code.

**References**
1. Cline MS, Smoot M, Cerami E, Kuchinsky A, Landys N, Workman C, Christmas R, Avila-Campilo I, Creech M, Gross B, Hanspers K, Isserlin R, Kelley R, Killcoyne S, Lotia S, Maere S, Morris J, Ono K, Pavlovic V, Pico AR, Vailaya A, Wang PL, Adler A, Conklin BR, Hood L, Kuiper M, Sander C, Schmulevich I, Schwikowski B, Warner GJ, Ideker T, Bader GD: **Integration of biological networks and gene expression data using Cytoscape.** *Nat Protoc* 2007, **2**(10):2366–82.
2. Shannon P, Markiel A, Ozier O, Baliga NS, Wang JT, Ramage D, Amin N, Schwikowski B, Ideker T: **Cytoscape: a software environment for integrated models of biomolecular interaction networks.** *Genome Res* 2003, **13**(11):2498–504.
3. Smoot ME, Ono K, Ruscheinski J, Wang PL, Ideker T: **Cytoscape 2.8: new features for data integration and network visualization.** *Bioinformatics* 2011, **27**(3):431–2.
4. Nabieva E, Jim K, Agarwal A, Chazelle B, Singh M: **Whole-proteome prediction of protein function via graph-theoretic analysis of interaction maps.** *Bioinformatics* 2005, **21**(Suppl 1):302–310.
5. Tu Z, Wang L, Arbeitman M, Chen T, Sun F: **An integrative approach for causal gene identification and gene regulatory pathway inference.** *Bioinformatics* 2006, **22:**e489–496.
6. Suthram S, Beyer A, Karp R, Eldar Y, Ideker T: **eQED: an efficient method for interpreting eQTL associations using protein networks.** *Mol Syst Biol* 2008, **4:**162.
7. Zotenko E, Mestre J, O'Leary DP, Przytycka TM: **Why do hubs in the yeast protein interaction network tend to be essential: reexamining the connection between the network topology and essentiality.** *PLoS Comput Biol* 2008, **4**(8):e1000140.
8. Missiuro P, Liu K, Zou L, Ross B, Zhao G, Liu J, Ge H: **Information flow analysis of interactome networks.** *PLoS Comput Biol* 2009, **5**(4):e1000350.
9. Voevodski K, Teng S, Xia Y: **Spectral affinity in protein networks.** *BMC Syst Biol* 2009, **3:**112.
10. Kim YA, Przytycki JH, Wuchty S, Przytycka TM: **Modeling information flow in biological networks.** *Phys Biol* 2011, **8**(3):035012.
11. Stojmirović A, Yu YK: **Information flow in interaction networks.** *J Comput Biol* 2007, **14**(8):1115–43.
12. Stojmirović A, Yu YK: **Information flow in interaction networks II: channels, path lengths, and potentials.** *J Comput Biol* 2012, **19**(4):379–403.
13. Stojmirović A, Yu YK: **ITM Probe: analyzing information flow in protein networks.** *Bioinformatics* 2009, **25**(18):2447–9.
14. **ITM Probe Web Service** [http://www.ncbi.nlm.nih.gov/CBBresearch/Yu/mn/itm_probe]

15. Gansner ER, North SC: **An open graph visualization system and its applications to software engineering.** *Software — Practice and Experience* 2000, **30**(11):1203–1233.
16. **Java** [http://www.java.com]
17. Jones E, Oliphant T, Peterson P, et al.: **SciPy: Open source scientific tools for Python.** 2001. [http://www.scipy.org/]
18. Demmel JW, Eisenstat SC, Gilbert JR, Li XS, Liu JWH: **A supernodal approach to sparse partial pivoting.** *SIAM J. Matrix Analysis and Applications* 1999, **20**(3):720–755.
19. Davis TA: **Algorithm 832: UMFPACK V4.3—an unsymmetric-pattern multifrontal method.** *ACM Trans Math Softw* 2004, **30**(2):196–199.
20. **SciKits** [http://scikits.appspot.com/]
21. Harrower M, Brewer C: **ColorBrewer.org: An Online Tool for Selecting Colour Schemes for Maps.** *Cartogr J* 2003, **40**:27–37.
22. Ashburner M, Ball CA, Blake JA, Botstein D, Butler H, Cherry JM, Davis AP, Dolinski K, Dwight SS, Eppig JT, Harris MA, Hill DP, Issel-Tarver L, Kasarskis A, Lewis S, Matese JC, Richardson JE, Ringwald M, Rubin GM, Sherlock G: **Gene ontology: tool for the unification of biology. The Gene Ontology Consortium.** *Nat Genet* 2000, **25**:25–29.
23. Smoot M, Ono K, Ideker T, Maere S: **PiNGO: a Cytoscape plugin to find candidate genes in biological networks.** *Bioinformatics* 2011, **27**(7):1030–1.
24. Stojmirović A, Bliskovsky A, Yu YK: **CytoSaddleSum: a functional enrichment analysis plugin for Cytoscape based on sum-of-weights scores.** *Bioinformatics* 2012, **28**(6):893–4.
25. Bader GD, Hogue CWV: **An automated method for finding molecular complexes in large protein interaction networks.** *BMC Bioinformatics* 2003, **4**:2.
26. Whelan C, Sönmez K: **Computing graphlet signatures of network nodes and motifs in Cytoscape with GraphletCounter.** *Bioinformatics* 2012, **28**(2):290–1.
27. Milenković T, Przulj N: **Uncovering biological network function via graphlet degree signatures.** *Cancer Inform* 2008, **6**:257–73.