ANAT: A Tool for Constructing and Analyzing Functional Protein Networks

Nir Yosef,[1,2] Einat Zalckvar,[3] Assaf D. Rubinstein,[3] Max Homilius,[4] Nir Atias,[5] Liram Vardi,[5] Igor Berman,[5] Hadas Zur,[5] Adi Kimchi,[3] Eytan Ruppin,[5,6] Roded Sharan[5]*

[1]Broad Institute of MIT and Harvard, 7 Cambridge Center, Cambridge, MA 02142, USA. [2]Center for Neurologic Diseases, Brigham & Women's Hospital, Harvard Medical School, Boston, MA 02115, USA. [3]Department of Molecular Genetics, Weizmann Institute of Science, Rehovot 76100, Israel. [4]Institute of Computer Science, Freie Universität Berlin, Takustr. 9, D-14195 Berlin, Germany. [5]Blavatnik School of Computer Science, Tel-Aviv University, Tel-Aviv 69978, Israel. [6]School of Medicine, Tel-Aviv University, Tel-Aviv 69978, Israel.

*Corresponding author. E-mail, roded@post.tau.ac.il

Genome-scale screening studies are gradually accumulating a wealth of data on the putative involvement of hundreds of genes in various cellular responses or functions. A fundamental challenge is to chart the molecular pathways that underlie these systems. ANAT is an interactive software tool, implemented as a Cytoscape plug-in, for elucidating functional networks of proteins. It encompasses a number of network inference algorithms and provides access to networks of physical associations in several organisms. In contrast to existing software tools, ANAT can be used to infer subnetworks that connect hundreds of proteins to each other or to a given set of "anchor" proteins, a fundamental step in reconstructing cellular subnetworks. The interactive component of ANAT provides an array of tools for evaluating and exploring the resulting subnetwork models and for iteratively refining them. We demonstrate the utility of ANAT by studying the crosstalk between the autophagic and apoptotic cell death modules in humans, using a network of physical interactions. Relative to published software tools, ANAT is more accurate and provides more features for comprehensive network analysis. The latest version of the software is available at http://www.cs.tau.ac.il/~bnet/ANAT_SI.

Introduction

The Need for Network Construction Tools

Most attempts to analyze and model biological networks involve dealing with uncertainty and information gaps (*1*). The challenge of minimizing those gaps and charting out the architecture that underlies a biological system of interest is a fundamental endeavor in systems biology (*2*). Methods that try to tackle this task by inferring specific subnetworks that are associated with some disease or phenotype within a global protein-protein interaction (PPI) network are becoming increasingly available (*3–8*).

In some situations where network reconstruction is useful, strong evidence connects the phenotype in question to a specific end target, or "anchor" point, to (or from) which signaling or regulatory pathways should lead. This anchor point can be an individual gene or protein or can be a set of functionally related genes or proteins. An example of this "anchored" phenotype is the change in gene expression that occurs in response to perturbation of a certain gene (*2*, *9*). In this case, one is interested in pathways that connect the differentially expressed genes to the perturbed gene, which acts as the single anchor of the inferred network. Another example of an anchored system is the process of apoptosis in mammalian cells, where the execution of the downstream cell death events is carried out by a small anchor set of caspase substrates (*8*).

The accumulation of large-scale screening data has expedited the development of tools for PPI network inference. One class of tools (*10–21*) is limited to localized views of the network, considering the close vicinity (typically node distance ≤3) of an input set of proteins. A second class of tools (*19*, *22*, *23*) performs computations of shortest paths between pairs of proteins. A third class of tools (*24*, *25*) uses global propagation heuristics to find network nodes that are functionally related to a given query set. Other related applications include functional enrichment analysis using a set of precompiled reference pathways (*26*) and detection of connected regions in a PPI network encoded by genes that exhibit similar expression patterns (*27–29*). All or most of the aforementioned capabilities are also available in commercial tools, such as Ingenuity Pathway Analysis (http://www.ingenuity.com), GeneGo (http://www.genego.com), and Pathway Studio (*30*).

Overview of the Features of ANAT

ANAT (Advanced Network Analysis Tool) is a publicly available tool for inference of functional networks of proteins. It is implemented through client-server architecture: The client side is a Cytoscape plug-in for visualization and query design; the server side contains a database of molecular interactions and annotations and a network inference engine. ANAT provides access to interaction data from several sources, advanced algorithms for network reconstruction, and a number of tools for exploring and evaluating the network models. ANAT is not restricted to localized or heuristic modes of operation. Instead, it conducts a network-wide search for the most plausible network models, using graph-theoretic algorithms with mathematically provable performance guarantees (with respect to the optimum solution) (*8*). Thus, relative to previously mentioned tools, ANAT analyzes all of the available interaction data and then identifies networks with a high confidence level.

ANAT supports four types of network-based analyses: (i) inferring an anchored network that connects a given set of proteins to a designated set of focal points; (ii) inferring high-confidence "general" networks when a focal point is not expected; (iii) finding the highest-confidence paths between pairs of proteins, and (iv) viewing the local neighborhood of a given set of proteins. In addition, ANAT provides an interface for adding expert knowledge (such as adding missing interactions or excluding irrelevant interactions) and iteratively refining the inferred models. A typical workflow in ANAT (Fig. 1) consists of three steps: (i) defining a background network; (ii) submitting a query for extracting a network model from the background network; and (iii) analyzing and refining the obtained network model.

Databases Available in ANAT

ANAT includes the PPI networks of 10 organisms: *Homo sapiens*, *Mus musculus*, *Rattus norvegicus*, *Drosophila melanogaster*, *Caenorhabditis elegans*, *Arabidopsis thaliana*, *Saccharomyces cerevisiae*, *Plasmodium falciparum*, *Escherichia coli*, and *Helicobacter pylori*. For each organism, we obtained a PPI network gathered from the literature (*31–39*) and from public databases (*40–42*). ANAT also contains protein-DNA interaction (PDI) information for yeast and humans (*43*, *44*). Network sizes, data sources, and gene identifier types are provided in table S1. All PPIs are assigned a confidence score according to the experimental evidence that supports them.
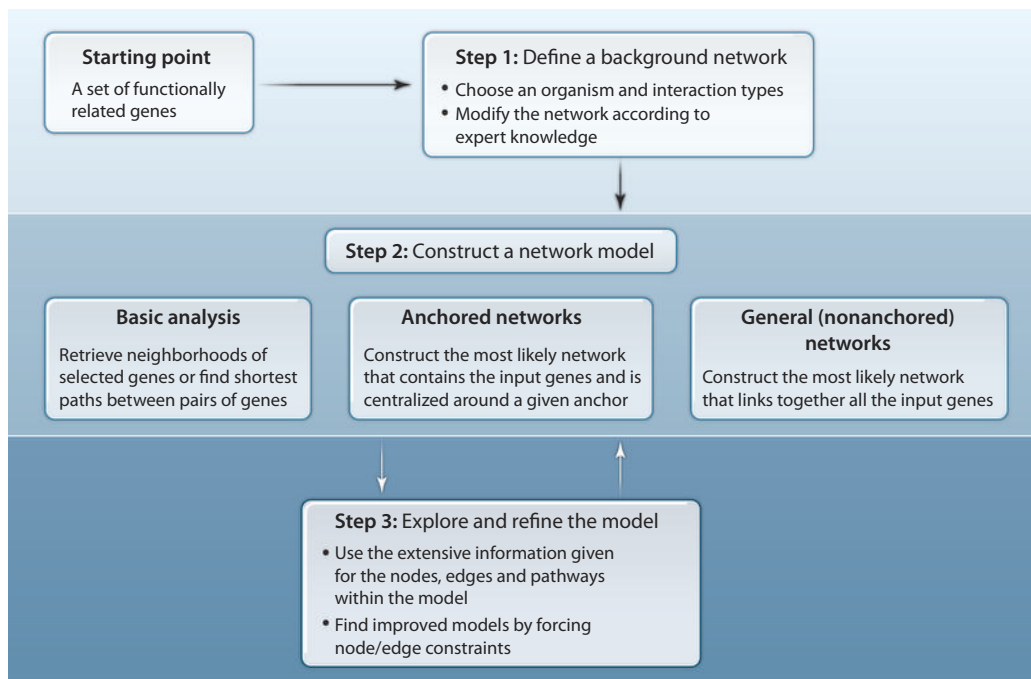
Fig. 1. Workflow in ANAT. Given a set of genes, the construction of a network of physical associations that links the proteins encoded by these genes consists of three steps: (Step 1) adding prior knowledge by defining a background network, (step 2) choosing the appropriate inference method and adjusting its parameters to construct the network model, and (step 3) exploring the obtained model and refining it.

Network Construction Processes in ANAT

Defining a background network

Large-scale data sets of physical interactions (as in ANAT's database) are often incomplete and suffer from inherent inaccuracies. For instance, the vast majority of the interactions lack directionality information, which if available (for example, kinase-substrate relations) can aid the reconstruction. Moreover, relevant interactions can be missing from the database and, if known, should be added before the analysis. Conversely, the database may include interactions that are irrelevant in the context of the studied system and should be discarded. Instead of excluding nodes or interactions, one may also like to prioritize their inclusion in the reconstructed model by setting their associated confidence level. ANAT provides a flexible mechanism for conducting such revisions before starting the actual network construction. This step is not mandatory; the user can choose to analyze the networks in ANAT's database without introducing any modifications.

Generating anchored networks

For creating anchored networks, the input to ANAT consists of two sets of proteins: "target" proteins (a set of proteins identified in a pertinent large-scale screen) and "anchor" proteins (a set of proteins that mediate the studied phenotype and around which the network should be constructed) (8). The output is a subnetwork of the background network that leads from each target to at least one of the anchors or vice versa (the direction of flow can be determined by the user).

The reconstruction scheme ANAT uses for anchored networks aims to optimize local and global features of the inferred network (8). The local criterion favors reliable pathways between the anchor and the targets. The global criterion favors a parsimonious network connecting the anchor to all the targets. Generally, local models are more appropriate when many true hits might be missing from the target set (that is, the case of partial coverage); global models are more appropriate when the knowledge on the target set is more complete. For network reconstruction by ANAT, the *alpha* parameter, ranging between 0 and 0.5, controls the relative importance of the local (represented by low parameter values) and global (represented by high parameter values) criteria in the reconstruction. This parameter can be adjusted in ANAT and is set to a default value of 0.25, which balances the importance of global and local models in building the network.

Generating general (nonanchored) networks

The second mode of ANAT applies to reconstruction scenarios in which the investigated system is not assumed to be centralized around a well-defined anchor. Instead, ANAT models the architecture of physical associations between the target proteins by searching for a high-confidence subnetwork that links them to each other. In some instances of this problem, all of the proteins will not form a single connected network module. This can be the case with "system-wide" or "distributed" phenotypes, such as modifications in the cell's transcriptional landscape in a disease state where we would expect to see several network modules that represent different cellular processes (*45*). Notably, there exists a natural tradeoff between the number of connected modules in a network and its overall confidence score. This tradeoff can be adjusted in ANAT through the *granularity* parameter. Low granularity values will yield many isolated modules, each containing a small portion of genes from the set, connected through highly confident links. High granularity values will produce fewer modules (possibly only one) of overall lower confidence, each containing a large portion of the genes in the set.

Generating networks based on shortest paths or "local search"

The third mode of ANAT is the computation of the most likely path between a given pair (or pairs) of proteins. The fourth mode of ANAT provides an interface for gradual navigation through the selected background network. The starting point is a user-selected set of proteins and a desired neighborhood diameter d. ANAT returns all the proteins that are at most d links away from at least one protein in the set, and presents all the interactions between these proteins.

Customizable network parameters

ANAT provides several additional adjustable parameters as a part of its anchored networks, general networks, and shortest-path modes. In all three cases, the respective optimization process may yield several solutions with similar confidence scores. To avoid an arbitrary choice among equally good solutions, ANAT records multiple solutions that have an overall score within a user-defined margin from the best solution found. The output network is the union of a set of such optional solutions. The *margin* parameter can be adjusted by the user and is set by default to 0%. Another adjustable parameter is the *edge penalty*, which controls the balance between the size (number of edges) of the inferred subnetwork and its overall confidence. Higher values of this parameter will yield solutions with fewer edges, but possibly a lower overall confidence. A third adjustment can be made by setting a node-degree penalty. This feature tilts the algorithm toward the exclusion of hub nodes whose interactions might not be specific for the studied phenotype. The penalty is set through two parameters: The *curvature* parameter controls the proportionality between node degree (number of interaction partners) and the penalty score assigned to it; the *dominance* parameter adjusts the importance of node weights (or penalty scores) relative to that of edge weights. Note that this feature is not available in the general networks mode.

Network exploration and model refinement

The output networks produced by ANAT are accompanied by additional information on the nodes, edges, and submodels. Information on the nodes includes (i) statistical significance, evaluating the chance of the node to participate in a model constructed on random input; (ii) redundancy, evaluating the importance of the node in maintaining connectivity in the model (defined as the percentage of solutions, if more than one solution up to the given margin is found, that contain the respective node); (iii) Web links to various online data resources (provided as part of the default Cytoscape environment); and (iv) centrality, the number of target-anchor pairs (for anchored networks) or query pairs (for shortest paths) whose connecting paths visit this protein. In the general networks and local search modes, this is the number of interactions of the protein in the output model.

Information on the edges includes (i) confidence, an estimate of the reliability of an edge based on the supporting experimental data, and (ii) a list of references (provided as PubMed ID) to the supporting experimental data.

ANAT also analyzes specific subsets of the output model (submodels). Given an anchored network model, a submodel corresponds to a shortest path that connects a certain target protein to some anchor protein (note that these submodels might share nodes or edges). For a nonanchored network, a submodel is a connected component of the output network. The information provided for each submodel includes (i) a likelihood score, defined as the joint probability of the participating nodes and edges, and (ii) a functional coherency score based on "Biological Process" annotations from the Gene Ontology database (GO) (*46*) and curated signaling pathways from the KEGG (*47*) and MSigDB (*26*) databases.

Most of the edge-based and node-based information can be viewed from within the Cytoscape environment. Detailed reports providing the centrality and the list of mediated target proteins for each node, as well as the submodel analysis, are provided in external html files.

Complementing the network exploration stage, ANAT provides the means to modify and refine the resulting model. For example, if one of the nodes in the model is deemed irrelevant upon experimental validation or manual inspection, then the user can generate a new model that does not include this node and possibly replaces it with other nodes. Similarly, one can also exclude specific edges or force directionality on edges in the model.

A Case Study and Benchmarking ANAT

To demonstrate the utility of ANAT, we inferred pathways that link the autophagic cell death module to the apoptotic module in humans. The resulting anchored network suggests testable hypotheses for the wiring of the previously observed crosstalk between these two cellular processes (*48–50*). To illustrate the effectiveness of ANAT, we compared models generated with ANAT to those from six published software tools (*11*, *16*, *17*, *19*, *20*, *25*) using a telomere length maintenance data set in yeast as a benchmark. This comparison showed that ANAT generates more accurate models than the other tools and highlights the key features that are unique to ANAT.

Equipment

Personal computer equipped with OS X (version 4.0 and above), Windows, or Linux operating system

Internet connection

>   *Note: ANAT uses the standard HTTP protocol to communicate with the server. This does not require any additional special equipment.*

Java Runtime environment (version 6 and above; http://www.java.com/en/download/index.jsp)

>   *Note: The ANAT User Manual provides details for running ANAT with lower versions of Java. This document will be updated with new releases of the ANAT plug-in and made available from the ANAT Web site, http://www.cs.tau.ac.il/~bnet/ANAT_SI. The version available at the time of publication is provided as part of the Supplementary Materials.*

Cytoscape [version 2.7 and above (*15*); http://www.cytoscape.org]

The ANAT plug-in file

>   *Note: ANAT will be updated from time to time. The most current version of the software is available from the ANAT Web site http://www.cs.tau.ac.il/~bnet/ANAT_SI. The version available at the time of publication is provided as part of the Supplementary Materials.*

Sample files

>   *Note: These files are only needed to reproduce the examples used for illustration in the detailed instructions. They are not required for running ANAT.*

Instructions

ANAT Installation and Starting and Saving Work Sessions

1. Download the latest version of the plug-in from the ANAT Web site (http://www.cs.tau.ac.il/~bnet/ANAT_SI).

2. Place the ANAT plug-in file in the "plugins" folder within the Cytoscape directory.

3. To start ANAT, open Cytoscape and the main menu of ANAT will be displayed in a separate tab on Cytoscape's control panel.

4. To save a working session (including all generated models and user-supplied inputs), use the "Save" option in Cytoscape.

>   *Note: All the types of input provided to ANAT (such as the modifications made to the background network) can be exported and saved as XML files. These files can then be imported and used in future sessions. The XML files can also be generated directly by the user, thus avoiding the need for manual data entry. The structure of the XML files is described in the ANAT User Manual.*

Defining a Background Network

The next sections describe how to create a modified background network, using images from the case study network as examples. Figure 2 shows the main screen from which ANAT's commands are initiated; Fig. 3A shows the screen from which a background network can be defined. Possible modifications to a network in ANAT's database include adding or removing edges or nodes, setting node or edge confidence, and forcing directionality on edges. These modifications to the background network are optional. Background networks that are based entirely on user-defined data sets may also be analyzed with ANAT (see "Querying an External Database").

The detailed instructions for using ANAT include explicit commands, computer filenames, or menu names. They are set off by quotation marks. Do not type the quotation marks when typing the commands.
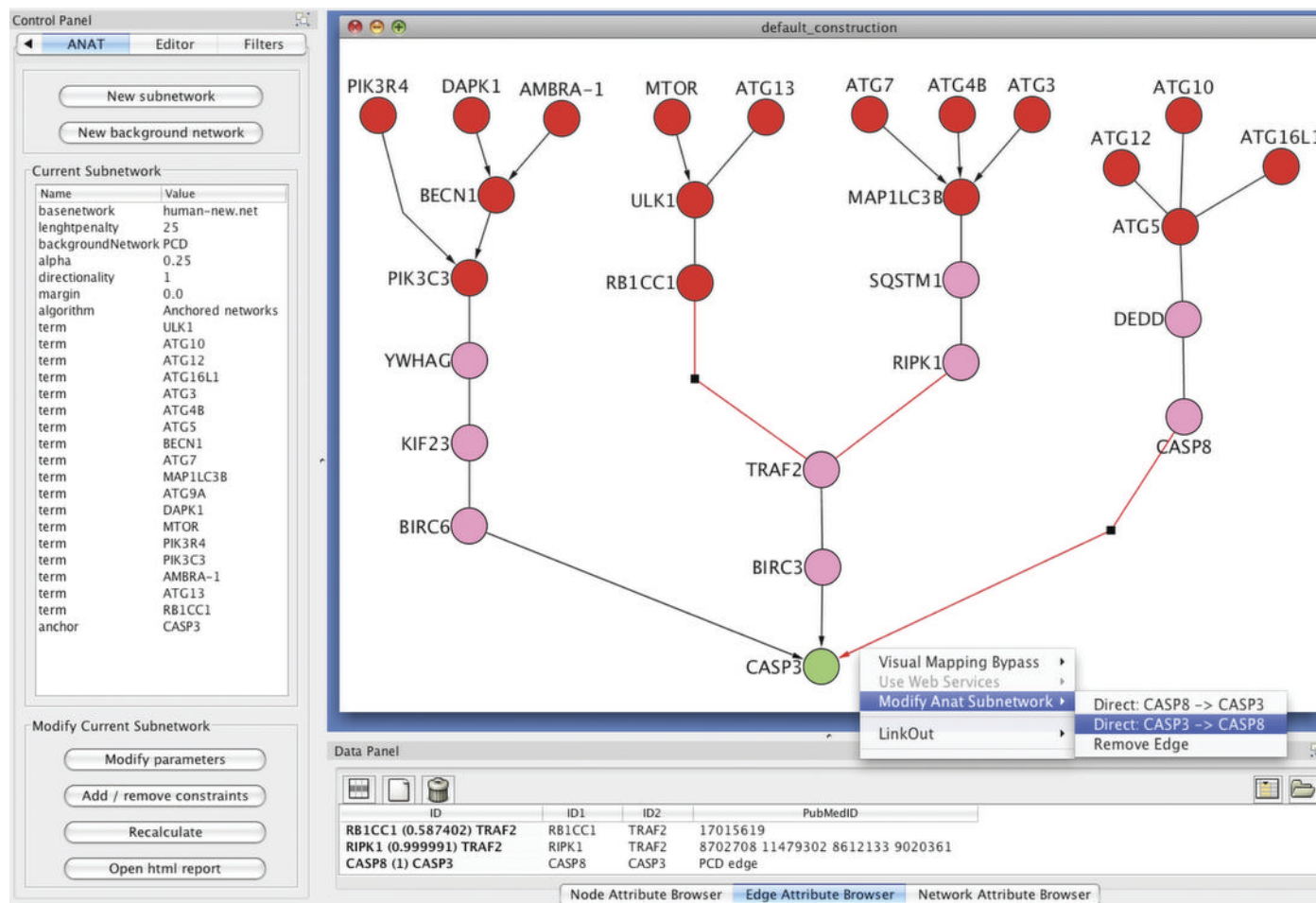
Fig. 2. The autophagy-apoptosis crosstalk network. The network model is depicted within the main screen of Cytoscape with ANAT panels and information on the left and at the bottom. Members of the autophagic core machinery, provided as a target set for the algorithm, are red. The apoptosis executioner and anchor node, caspase-3, is green. The nine nodes added by ANAT are pink. Additional information on every edge and node in the model is displayed in the attribute browsers of Cytoscape (bottom). Through ANAT's main panel (left), the user can edit the parameters of the algorithm, add or remove topological constraints (removing nodes or edges and forcing directionality on edges), and access reports generated for the current network model providing a submodel-level analysis or node-centrality analysis. Additional constraints can also be added by right-clicking a node or an edge and using ANAT's context menu that appears on the screen.

Creating a modified background network

1. From the control panel (Fig. 2, left), click either "New subnetwork" (to start a complete analysis) or "New background network" (only to create a background network, which can be saved for later use). This will load a new screen from which the available networks can be selected (Fig. 3A, left).

2. Select the desired organism and interaction type (PPI, PDI) from the "Base network (Species)" menu (Fig. 3A, bottom left).

3. Click the "Advanced" button at the bottom of the screen to enable the "Modify base network" menu (Fig. 3A, bottom left).

    Note: The "Advanced" button acts as a toggle and once it is selected, the button becomes "Simple" as seen in Fig. 3A.

4. Click the "New" button at the top of the "Modify base network (Optional)" panel (Fig. 3A, top right) and enter the desired title of the modified background network in the pop-up text window that opens.

    Note: The name of the modified background in Fig. 3A is "PCD."

5. Enter modifications using the "Modify node" and "Modify edge" panels.

    Note: Detailed instructions for each follow.

6. Click the "Export" button to save the modified background network as an XML file. This file can be saved anywhere and used in future ANAT sessions.

   Note: This step is optional. To upload a previously defined background network, click the "Import" button and select the corresponding XML file. An example file (ANAT_PCD_bckground.xml) is provided in the Autophagy Input files in the Supplementary Materials.

Modifying nodes in a background network

This section describes how to add or remove nodes in a background network, as well as how to assign node confidence values. These steps are optional; they allow the background network to be customized with additional knowledge or allow the user to specify how the algorithm computes the inclusion of certain nodes in the network. By setting a node confidence level, one can encourage the algorithm to use certain nodes (associated with confidence values close to 1) and exclude others (associated with confidence values near 0). By default, node confidence levels are not assigned. Once a confidence value is added to any node, this option then becomes enabled for all nodes. Nodes that were not assigned confidence values individually will be assigned a default value, which can be set by the user. Node confidence levels are ignored in the "General networks" algorithm option.

1. To add a new node, click "Add new node" in the "Modify node" panel, enter the node's name and, if desired, set its confidence value (Fig. 3A, right).

2. To remove an existing node, enter the node name under the "Name" column and select "REMOVE" from the "Action" column.

   Note: ANAT uses an auto-complete function for entering node names. Entering the first few letters will present a dropdown list of nodes that have the same prefix. A user cannot enter the name of a node that is not already included in the network (it either is in ANAT's database or was manually added using step 1). This functionality is active in all of ANAT's menus. The "Remove line" button will delete the currently selected entry. This functionality is available in all of ANAT's menus.

3. To set the confidence of an existing node, enter the node name under the "Name" column, select "SET" from the "Action" column, and enter its confidence value in the "Confidence" column.

4. Check the "Use node confidence" box to enable the use of node confidence values. Enter the default confidence value (for any node that was not assigned a specific confidence value) using the text box to the right.
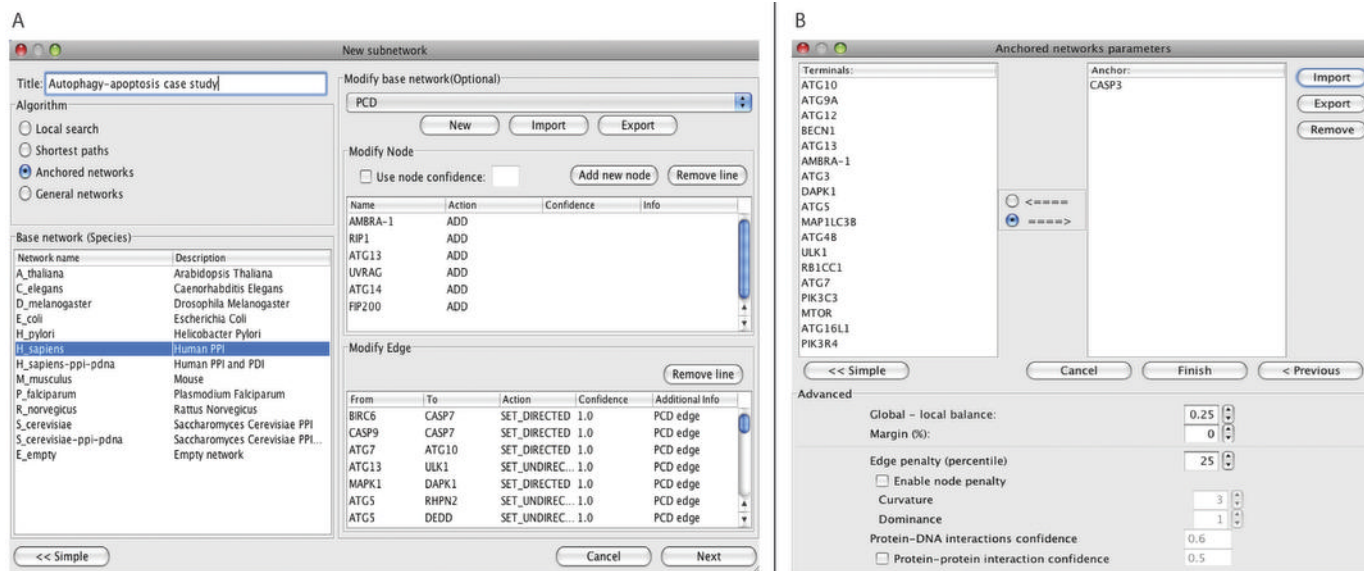

Fig. 3. ANAT menus. (**A**) Background network menu: After selecting a base network [organism and interaction type (PPI, PDI)], this menu provides an interface for various modifications of the baseline network, which is done before construction of the network model. To define the programmed cell death (PCD) background network, we selected the human PPI network from the "Base network (Species)" list. We added several PCD proteins that did not have any interactions recorded in the public databases used by ANAT in the "Modify node" panel and manually assembled and added interactions between PCD proteins and forced directionality of flow (denoting one protein as source and the other as target) on a subset of the PCD edges that had no directionality information in the public databases using the "Modify edge" panel. (**B**) Anchored network menu: The primary inputs for this menu are the target set (left) and an anchor set (right); the output is a network that connects each target to at least one of the anchors. To construct the autophagy-apoptosis crosstalk network, we entered a list of autophagic genes as targets and caspase-3, the main apoptosis executioner, as a sole anchor. This is where the network parameters are adjusted by choosing the "Advanced" button to display the screen shown.

Note: This step is optional. The box will be automatically checked if step 3 was carried out. Unchecking the box will cause ANAT to ignore node confidence values.

5. To enter additional information on nodes (in free text format), edit the "Info" column.

   Note: This step is optional.

Modifying edges in a background network

This section describes how to add a new relation or edge between two nodes, how to remove edges between nodes, and how to set the directionality of relations. These steps are optional. As with nodes, edges can have confidence values from 0 to 1. The confidence levels assigned with PPI edges that already exist in ANAT are based on the amount of supporting information. The PDI edges are assigned a fixed confidence value of 0.6. Because of insufficient data for some of the organisms, all PPI edges for those organisms are assigned a value of 1. Assigning confidence levels close to 1 will encourage the algorithm to include the respective edge, whereas confidence levels close to 0 will encourage exclusion.

1. To add a new edge, enter the names of the interacting nodes in the first two columns of the "Modify edge" panel (Fig. 3A, right).

2. Under "Action," choose "SET DIRECTED" for a directed edge (from the node in the first column to the node in the second column) or "SET UNDIRECTED" for undirected edge.

3. Under "Confidence," enter a confidence value. The default value is 0.2.

4. Enter additional information at the "Additional info" field.

5. To modify an existing edge, enter the names of the interacting nodes in the first two columns; then, if desired, choose "SET DIRECTED" under "Action" to force directionality on the edge (from the node on the first column to the node in the second) and, if desired, enter a confidence value to override the default confidence. Optional additional information may also be added.

6. To remove an existing edge, enter the names of the interacting nodes in the first two columns and choose "REMOVE EDGE" under "Action."

Generating a Network Model

The following instructions are for constructing an anchored network. The other three types of network construction tasks (general networks, shortest paths, and local search) have a similar workflow.

The menus for network construction include all the parameters that apply to the particular task. In the example of anchored networks (Fig. 3B), the menu includes the local-global parameter (*alpha*), the node and edge penalties, and the margin parameter. Additionally, the menu allows setting a constant confidence score to all of the PDI or PPI edges (Fig. 3B, bottom text boxes). These values will override the default confidence scores in ANAT.

1. From the control panel in the ANAT menu (Fig. 2, left), click "New subnetwork"; this will open the New subnetwork screen (Fig. 3A).

2. Enter the title of the analysis in the "Title" text box.

   Note: In the example, this is "Autophagy-apoptosis case study."

3. Select the desired construction algorithm from the "Algorithm" panel (in this case, "Anchored networks").

4. Select the desired organism and interaction type (PPI, PDI) from the "Base network (Species)" menu (Fig. 3A, bottom left)

5. Click the "Advanced" button at the bottom of the screen to enable the "Modify base network" menu (Fig. 3A, bottom left). Choose the modified network from the dropdown list (Fig. 3A, top right) or follow the steps above for importing or creating a modified background network.

   Note: Skip this last step if no modifications to the base network are needed.

6. Click "Next" and enter the respective list of anchor and target genes by either typing in the gene names in the "Terminals" and "Anchor" panels or by uploading an XML file using the "Import" button (Fig. 3B).

   Note: The input can be imported or exported with the XML "Export" or "Import" buttons. An example file (input.xml) is provided in the Autophagy Input files in the Supplementary Materials.

7. Click the "Advanced" button to set the relevant parameters

   Note: The "Advanced" button acts as a toggle and once it is selected, the button becomes "Simple" as seen in Fig. 3B. The alpha parameter is denoted as "Global-local balance" in the advanced screen.

8. Click "Finish" to send the query to the server, which will load a new screen that displays the generated network.

Exploring the Network

1. Explore the information about the nodes and edges by clicking the "Node Attribute Browser" or "Edge Attribute Browser" button, respectively (Fig. 2, bottom).

   Note: To view the information about a specific node or edge, left-click the node or edge of interest in the graphical display. Figure 2 shows the information for the edge connecting caspase-8 (CASP8) to caspase-3 (CASP3).

2. View the HTML reports, which provide information on node centrality and on the different submodels within the network, by clicking "Open html report" in the "Modify current subnetwork" panel (Fig. 2, bottom left).

   Note: The HTML reports are routinely deleted from the server. To keep these results, save them locally on your computer

3. View all of the immediate interactors of a given node by right-clicking a node and selecting "Expand node."

Refining the Network Model

Modifications can be made to nodes and edges or to general parameters of the model, such as *margin*, *edge penalty*, and *node-degree penalty*. General parameters are modified from the "Modify parameters" button that is part of the "Modify current subnetwork" panel of the main control panel. Modifications to nodes and edges can be made through contextual menus that are integrated within the graphical display or through a separate menu accessed from the "Modify current subnetwork" panel, both of which are described in detail.

Modifying a network using contextual menus

1. Right click the desired node or edge and choose "Modify ANAT subnetwork."

2. Choose the desired action ("Remove node," "Remove edge," or force directionality on edge).

3. Once the insertion of new constraints is complete, click the "Recalculate" button in the "Modify current subnetwork" panel (Fig. 2, bottom left) to generate the new model.

Modifying a network using the constraint menu

1. Click the "Add/remove constraints" button (Fig. 2) to open a window that shows the status of all the currently used constraints (any removed nodes, removed edges, or directed edges).

   Note: The status of a constraint is "active" if the network was recalculated after its addition, and "pending" otherwise.

2. Use the Nodes and Edges panels to add or remove constraints.

   Note: This menu also allows the user to save the modifications and upload previously saved modifications by clicking the "Import" or "Export" buttons.

3. Once the insertion of new constraints is complete, click "Close" and then click the "Recalculate" button in the "Modify current subnetwork" panel (Fig. 2, bottom left) to generate the new model.

Reproducing the Case Study Network

As a case study, we applied ANAT to generate a network that connected apoptotic and autophagic proteins in humans. The following instructions describe how to reproduce this network. The files needed to execute these steps are provided in the Autophagy Input files as part of the Supplementary Materials.

1. Click the "New subnetwork" button.

2. From the list of networks, select "H. sapiens"; from the list of algorithms, choose "Anchored networks."

3. Enter the analysis title ("Autophagy-apoptosis case study") in the title field.

4. Click the "Advanced" button.

5. In the "background network" menu, import the file "ANAT_PCD_bckground.xml" and click "Next."

6. In the Anchored Networks screen, import the file "input.xml."

7. Click "Finish."

8. To view the analysis of the pathways in the resulting autophagy-apoptosis crosstalk network, click the "Open html report" button.

Reproducing the Case Study Alternative Models

The following steps assume that the autophagy-apoptosis crosstalk network is the currently active network in ANAT.

1. Click "Modify parameters."

2. Click "Advanced."

3. To generate the first alternative model, check the "Enable node penalty" checkbox.

4. To generate the second alternative model, enter "1" in the "Margin" text box.

5. Click "Finish."

Querying an External Database

ANAT can be employed as an algorithmic engine for querying external databases. This can be done by defining a new background network that only contains the nodes and edges in the external database:

1. Click "New subnetwork."

2. From the list of networks, select "E_empty."

3. Click "Advanced" and then use the background network menu to enter the nodes and edges from the external database.

   Note: These data can be entered manually or by importing a user-generated XML file. Details for the format of the XML can be found in the ANAT User Manual in the Supplementary Materials.

4. Continue with the analysis as described above.

Related Techniques

We compared ANAT to several published resources for network inference: BisoGenet (*19*), PPISpider (*17*), STRING (*20*), FunCoup (*16*), BioPixie (*11*), and GeneMANIA (*25*) (Table 1). BisoGenet is a Cytoscape plug-in that performs a neighbor selection procedure, which is similar to the local search task in ANAT. The next four tools have a Web-based interface and perform heuristic network expansion, starting from an initial query set of proteins. For instance, PPISpider conducts a greedy selection of connected components made of proteins of up to two edges away from the original set. The last tool (GeneMANIA) has both Cytoscape and Web interfaces and, given an input node set, it ranks the remaining nodes using a global propagation procedure and returns a network composed of the query set and the top ranking genes. An additional tool, ToppGene (*24*), which uses a global propagation operation, could not be applied here because it contains only human PPI data.

As a benchmark data set, we used the telomere length maintenance (TLM) system in yeast. Briefly, telomeres are specialized DNA-protein structures that protect the ends of eukaryotic chromosomes and get shorter with replicative age (*51*). Telomeric DNA is synthesized by the enzyme telomerase, which, in mammals, is expressed at the early stages of development but not in most somatic cells of the adult (*52*). Three large-scale screens were conducted to evaluate the effect of gene deletion on telomere length in yeast (*53–55*); 360 genes were identified, of which 207 had a short telomere phenotype and 153 had an elongating effect on telomere length. We added to this "TLM set" a group of 23 additional proteins that were collected from the literature (*8*). We constructed the TLM network with the anchored network option in ANAT using different margins (0%, 1%, and 2%) and default values for the remaining parameters. As the anchor of the TLM system, we used a set of 10 telomere-binding proteins, defined in (*8*), including subunits of the telomerase and telomerase-interacting proteins. The target nodes were the 383 proteins in the TLM set. The benchmark software tools were applied with different bounds on network size (Table 2). All the other parameters were left at their default value.

We previously used a subset of the TLM set (*53*, *54*) to evaluate the different algorithms included in ANAT (*8*). Here, we used performance measures similar to those in (*8*) to compare the anchored-networks component of ANAT with the benchmark tools (Table 2; see the Supplementary Text for a full description of the performance measures). An important aspect of network construction performance is the ability to infer true positives that were not present in the input screen. We tested this function of the tools in a cross-validation setting, hiding one screen at a time and using the other two to infer the TLM network. ANAT achieved the best performance in successfully retrieving hidden TLM nodes, particularly with the margin parameter set to 1% (Table 2).

A second measure of performance is the functional coherence of the different submodels (paths from terminals to anchor) within the network model. We tested two properties: (i) enrichment in GO Biological Process annotation (*8*, *46*) (FE in Table 2), and (ii) enrichment in specific telomeric phenotype (short or long) (MC in Table 2). We also tested how well one can use the inferred network to predict telomeric phenotypes (PP in Table 2). Overall, relative to most benchmark tools, ANAT's network models exhibited superior performance in these tests. The one exception was BioPixie with stringent confidence cutoff (limited to interactions ranking in the top 25% confidence level), which provided better performance; however, the BioPixie model spanned only ~30% of the TLM genes, whereas ANAT's model

spanned ~88%. Using the default confidence cutoff of the software, the resulting BioPixie model spans the majority (~95%) of the TLM genes but is then outperformed by ANAT.

ANAT has several unique features (Table 1). Unlike the other tools, ANAT offers a range of network inference algorithms, the selection of which depends on the specific data set. Furthermore, among the tested tools, GeneMANIA and ANAT are the only ones that do not limit the search distance and allow the inclusion of user-defined background networks. However, ANAT is the only tool that provides the means to integrate expert knowledge into the inference procedure by setting node confidence levels and to conduct an iterative refinement by modifying the background network and adding node and edge constraints.

The background networks used by GeneMANIA, STRING, FunCoup, and BioPixie include functional links between proteins, inferred from heterogeneous data (such as coexpression). In contrast, the database of ANAT includes only physical interactions. Nevertheless, if a user is interested in inferring a network of functional interactions, ANAT can be used as an algorithmic engine to query those external databases (see Instructions).

Table 1. Comparison of the features of ANAT and other network construction tools. The features compared for each software tool include (i) network inference algorithms; (ii) the presence of absence of the ability of the user to perform functional enrichment analysis (FE), node significance estimation, active model refinement, and inclusion of expert knowledge; (iii) whether the software includes links to articles (in PubMed) that reported the interactions in the model; and (iv) number of organisms supported (#O).

Tool	Algorithms	FE	Node significance	Model refinement	PubMed links	#O
ANAT	List all neighbors up to a given distance; shortest paths; provable approximation of high-probability anchored/ general networks; no bound on distance	Yes	Yes	Yes	Yes	10
STRING	List all neighbors up to distance 5	Yes	No	No	Yes	630
FunCoup	List all neighbors up to distance 3; several heuristics available for filtering nodes/ edges and generating smaller subnetworks	Yes	Yes	No	Yes	9
GeneMANIA	Propagation-based node ranking; display a selected number of top candidates; no bound on distance	Yes	Yes	No	Yes	6
BioPixie	Greedy node selection, up to distance 2	Yes	Yes	No	Yes	1
BisoGenet	List all neighbors up to distance 3; shortest paths within a computed model	No	No	No	Yes	8
PPISpider	Greedy node selection, up to distance 3	Yes	Yes	No	No	9
	Greedy node selection, up to distance 3	Yes	Yes	No	No	9

Table 2. Comparison of the performance of ANAT with other network construction tools. The different tools were used to construct the TLM network with different bounds on network size. All the other parameters were left at their default value. For ANAT, we used three margin values (m = 0%, 1%, and 2%); for STRING, we used two distance limits by setting the "network depth" parameter [D = 1: (default) include queried genes, D = 2: include neighbors of distance 1]; for FunCoup and BioPixie, we used two quality thresholds ("all," output network with default parameters; "top 75%," retain only interactions with confidence score above the 75th percentile); for GeneMANIA, we examined the top 50 and the top 200 ranked nodes; for BisoGenet, we used two distance limits (L = 1: include neighbors of distance 1, L = 2: include neighbors of distance 2); and, for PPISpider, we used the three distance limits provided by the software (D = 1, 2, and 3). The left part of the table presents the performance on the TLM benchmark. Displayed measures include number of nodes and edges in the inferred models (N/E); percentage of input TLM genes included in the model (T); the ability to recover unannotated TLM proteins (cross-validation score, CV); functional enrichment (fraction of functionally coherent submodels, FE); monochromaticity (mean coherence in telomere length phenotype, MC); phenotype prediction accuracy (precision/recall, PP). See the Supplementary Materials for details on the determination of the performance metrics.

Tool	N/E	T	CV	FE	MC	PP
ANAT (default)	559 / 594	88.1	7.5	81.2	73.3	72.2 / 11
ANAT (m = 1%)	620 / 840	88.1	8.2	81.6	75.1	77.1 / 10
ANAT (m = 2%)	632 / 913	88.4	7.8	80.2	73	73.9 / 9
STRING* (D = 1)	337 / 2048	88.1	—	82.3	60.7	60.7 / 31
STRING* (D = 2)	369 / 2369	89.7	4.8	66.6	59.5	59.9 / 25
FunCoup* (top 75%)	390 / 988	45.4	3.4	68.8	60.9	60.9 / 12
FunCoup* (all)	459 / 1520	52.4	3.3	64.9	51.3	51.3 / 11
GeneMANIA* (top 50)	430 / 30,555	99.7	6.1	47.1	76.7	58.9 / 24
GeneMANIA* (top 200)	580 / 63,115	99.7	7.4	44	67.3	61.5 / 16
BioPixie* (top 75%)	941 / 2687	34.3	7.7	100	76.5	76.5 / 4
BioPixie* (all)	5546 / 50,145	95.8	5.1	79.7	62.1	62.1 / 5
BisoGenet (L = 1)	62 / 116	3.9	2	—	—	—
BisoGenet (L = 2)	2723 / 29,003	41.4	3.7	35	72.2	85.7 / 5
PPISpider (D = 1)	214 / 422	56.1	NA†	53.7	73.4	66.2 / 26
PPISpider (D = 2)	468 / 794	87.6	3	57.1	71.8	60.6 / 39
PPISpider (D = 3)	652 / 2777	88.7	5.6	34.1	68	58.8 / 25

*The values shown in the phenotype and functional enrichment fields (FE, MC, and PP) were calculated only with PPI edges.
†The D = 1 model of PPISpider only includes input proteins and is therefore not applicable for this test.

Notes and Remarks

Node, Edge, and Submodel Confidence Levels

ANAT assigns every protein-protein interaction (edge) with a confidence score $0 \leq p \leq 1$ based on the available experimental evidence for it, using a published logistic regression model (*56*). Because experimental data are insufficient, ANAT does not compute logistic models for rat, mouse, *H. pylori*, or plasmodium and instead assigns a fixed confidence level of "1" to all interactions. The inferred confidence scores (for the other six species) can be changed from ANAT's menus (Fig. 3, A and B). The confidence of the protein-DNA interactions is set by default to 0.6 and can be adjusted from ANAT menus (Fig. 3B). To control for the size (number of edges) of the inferred subnetworks, we follow the approach of Shachar *et al.* (*57*) and multiply each edge confidence by an edge penalty factor, $0 < e_p < 1$. The edge penalty is assigned by default to the probability of an edge at the 25th percentile and can be adjusted by the user.

The proteins (nodes) can be assigned confidence levels as well (by default, ANAT does not assign node confidence scores). This can be done either directly (providing a confidence level for every protein in the network using the background network menu) or through the node-degree penalty option. Using the latter option, the confidence of a node is determined as inversely proportional to its degree, penalizing hub proteins. Specifically, we set the confidence level of node v with degree $\deg(v)$ to

$$p(v) = \exp\left[-a \cdot \deg(v)^{1/b}\right]$$

where a is the dominance parameter and b is the curvature parameter.

The confidence scores assigned to an edge e and a node v are denoted by $p(e)$ and $p(v)$, respectively. Assuming independence of edges and nodes, the overall confidence of a submodel H (as presented in the HTML reports) is

$$p(H) = \prod_{e \in E(H)} p(e) \prod_{v \in V(H)} p(v)$$

where $V(H)$ and $E(H)$ are the nodes and edges in H, respectively.

To facilitate the optimization algorithms in ANAT, we treat the protein interaction network as a directed weighted graph $G = (V, E, w)$, where the weight w of an edge or a node is defined as the negative logarithm of its confidence.

Anchored Networks and the Global-Local Parameter

Given an anchor node r and a set of target proteins X, ANAT attempts to construct a connected subnetwork H that connects the anchor to the target set. In the global variant of the problem, the solution is a subnetwork H with minimum overall weight:

$$F_G(H) = \sum_{e \in E(H)} w(e) + \sum_{v \in V(H)} w(v)$$

In the local variant of the problem, the solution is a collection of all lowest-weight paths from the anchor to each protein in the set X:

$$F_L(H) = \sum_{x \in X}\left[\sum_{e \in P_H(x,r)} w(e) + \sum_{v \in P_H(x,r)} w(v)\right]$$

where $P_H(x,r)$ is a shortest path from a node x to the anchor r in H. When the anchor set contains more than one member, ANAT adds an additional "pseudonode" and connects it by directed edges of weight zero to all the members of the anchor set. This pseudonode serves as the single anchor point r.

Finding a single network that simultaneously optimizes the local and global objectives (up to a constant factor times their optimal value) is feasible in undirected graphs (*58*). However, it can be shown that in directed graphs the two objectives cannot be simultaneously approximated to within a factor of less than $O(k^{1/2})$, where k is the number of proteins in the set. Moreover, optimizing only the global objective can only guarantee an approximation bound of $O(k)$ with respect to the local one, and vice versa (*8*). The intermediate algorithm presented in (*8*) and implemented in ANAT provides approximation guarantees of $O(k^{1-\alpha})$ and $O(k^{1/2+\alpha})$ for the global and local objectives, respectively. The parameter $0 \leq \alpha \leq 0.5$, which controls the tradeoff between the two objectives, can be adjusted by the user. The running time of the intermediate algorithm is $O\left[|X|^2 \cdot |E| + |X| \cdot |E| \cdot \log(|V|)\right]$, where E and V are the sets of edges and nodes, respectively, in the background network.

General Networks and the Granularity Parameter

Given a set of proteins, ANAT seeks a minimum-weight subnetwork H that connects the proteins in this set. The general network construction in ANAT is done in an undirected setting (ignoring edge directions) and uses only edge weights (ignoring node weights). Finding the best network under these settings translates to a Steiner tree problem [see (*8*) for details of the algorithm].

To allow greater flexibility, ANAT does not force H to be connected, but rather allows it to have several connected components. The number of components $c(H)$ is controlled by a granularity factor λ, which is combined into the overall subnetwork score as follows:

$$F(H) = \lambda \cdot c(H) + \sum_{e \in E(H)} w(e)$$

. To this end, ANAT adds one auxiliary node to the network, connecting it to all the other nodes by edges of weight λ, and applies the basic Steiner tree approximation algorithm to the resulting network. The value of λ is defined as

$$\lambda = -\frac{g}{100} \cdot \log(e_p)$$

, where e_p is the edge-penalty factor and g is the granularity value that was provided by the user. The running time of the algorithm is $O\left[|X| \cdot |E| \cdot \log(|V|) + |X|^2 \right]$.

Computing Multiple Solutions

To avoid an arbitrary choice among equally good solutions, ANAT's implementation records multiple solutions. This is done by randomly selecting the order by which the nodes in the background network are processed, and repeating the randomization 50,000 times. The random ordering affects the way ties are handled during the run of the algorithms, thus producing different solutions. The returned solution is the union of all solutions obtained. The redundancy of a node is computed as the percentage of solutions in which it is included.

By setting the *margin* parameter, the user can include in the solution networks whose overall weight (sum of log confidence values of the nodes and edges) is up to x% from the weight of the guaranteed solution. We previously described how to compute suboptimal solutions (up to x%) for the shortest-path problem (*50*). In ANAT, this procedure was extended to facilitate margin computation in the general and anchored network inference problems by using suboptimal shortest paths and by modifying the respective inference algorithms. Note that ANAT does not guarantee to return all x% margin solutions, which may be computationally intractable, but rather samples the solution space.

Statistical Evaluation of Nodes

The output networks produced by ANAT typically contain "new" nodes that are required for connectivity and were not part of the original input provided by the user. In all the algorithms used in ANAT, the key factor for the inclusion of a new node by chance (that is, with a random input set) is the number of shortest paths that go through it (taking into account all node pairs in the background network). The statistical significance of the new nodes in ANAT's models is based on this attribute.

Shortest paths between members of the input set are the starting point for two of the algorithms: general networks and shortest-path networks. The new nodes in the resulting model are the proteins that constitute these shortest paths. To evaluate the probability for inclusion of a node v in a model by chance, ANAT computes the following binomial score:

$$\sum_{i=k+1}^{m} \binom{m}{i} p^i (1-p)^{m-i}$$

where m (the number of "trials") is the number of pairs connected by one or more shortest path(s) in the model, and k (the number of "successes") is the number of pairs in the model whose shortest path goes through v. The resulting score is the probability of obtaining at least k successes out of m trials where the probability for success is $p = K/M$, where K is the overall number of pairs in the background network whose shortest linking path goes through v, and M is the overall number of node pairs in the background network.

The algorithm for anchored networks uses a more complex scheme; it is also based on shortest paths, but every target protein u is linked to the anchor node r by a pair of shortest paths, going through a middle "pivot" node w [see (*8*) for details; it is possible that $u = w$ or $w = r$]. ANAT uses a binomial score of a given node v adjusted as follows: m (the number of trials) is the number of target proteins; k (the number of successes) is the number of target proteins whose path to the anchor goes through v; and p (the background probability for success) is $2(K/M) + [1/(N-2)]$, taking an upper bound on the probability for v to appear in either of the two pathways (from u to w or from w to r) or being selected as a pivot ($v = w$). The variable N is the overall number of nodes in the background network. The variables M and K are defined as before.

Because the computation of the overall number of pairs whose shortest path goes through a given node is time-consuming, ANAT uses pre-computed probabilities, obtained from the original background networks, with the default parameters of ANAT. ANAT also maintains access to pre-computed probabilities, considering various levels of the *margin* parameter (from 2% to 40%). This is required because higher *margin* values result in more paths (which can be suboptimal) that go through each node (that is, K increases). Note that the node significance values are based on the original background networks. In future releases, ANAT will support node significance computation with modified background networks as well.

Analysis of Submodels

The main resource for assigning functional annotations and enrichment scores to the submodels is the Gene Ontology (GO) database (46); additional data sources include KEGG (47) (for yeast and human) and MSigDB (26) (for human). The enrichment of a submodel with each functional category is computed by a hypergeometric *P* value as in (8). These *P* values are compared to 100 *P* values computed for random sets of proteins of the same size, yielding empirical *P* values. In all cases, ANAT reports the annotation with the highest significance, using a cutoff of 5% for the empirical *P* values.

The submodels are also assigned a confidence score as defined in "Node, Edge, and Submodel Confidence Levels."

## Case Study: Constructing the Autophagy-Apoptosis Crosstalk Network

Programmed cell death (PCD) is a genetically controlled cell death that plays a major role in different biological processes. Defective regulation of PCD is widely recognized as the basis for a spectrum of diseases, including degenerative disorders and cancer (59). Two forms of PCD are caspase-dependent apoptotic cell death (apoptosis) and caspase-independent autophagic cell death. These pathways are activated by different signals and lead to distinct death morphologies (60). Much effort has been made to elucidate the functional interplay that exists between these two cell death modalities, and several autophagic proteins have been implicated in apoptosis (61); however, the molecular pathways underlying these observations are still poorly characterized.

Previously, we used a shortest-path computation to infer a single crosstalk pathway connecting caspase-3, a central effector caspase in the apoptotic cascade that is not connected by any known PPI to the autophagic network, and Atg5, an autophagic protein (50). We used the anchored network algorithm of ANAT to extend this work and explore a larger portion of the pathways that connect autophagy with apoptosis in humans. As targets, we used a set of 17 proteins that participate in autophagy, and as an anchor point we used caspase-3. The 17 target proteins were chosen because they participate in autophagy as central regulators or effectors (or both), without making any prior assumptions about their role in apoptosis. We were interested in examining the pathways identified with ANAT that connect the target set with the anchor point, as well as dissecting the way in which some of the targets may arrange into clusters that share one or more common pathways.

Following the ANAT workflow (Fig. 1), we chose the human PPI network as the analysis starting point (Fig. 3A, bottom left) and then defined a PCD background network by adding 167 interactions that were either undirected or absent (Fig. 3A, right). Where possible, we assigned the directions of these interactions by the order of the respective elements in the PCD signaling cascades (for example, directing an edge from an enzyme to its substrate). After obtaining a preliminary model linking the autophagic targets to caspase-3, we continued with an iterative process of model analysis and refinement by manually investigating the literature on each interaction and adding edge or node constraints where needed; for example, an interaction between beclin-1 and the nuclear transcription repressor GFI1B (growth factor–independent protein 1B) was manually removed because these proteins reside in different subcellular locations and the interaction is therefore unlikely to occur in vivo. In another case, the undirected interaction between caspase-3 and its substrate paxillin was directed from the former to the latter, subsequently eliminating pathways that do not follow this directionality (see table S2 for the complete list of refinements made).

The final network obtained following the iterative refinement process contained nine proteins in addition to the target and anchor sets (Fig. 2). Information on the additional nodes can be viewed from within the Cytoscape environment by clicking the "Node attribute browser" tab (Fig. 2). The submodel-level analysis showed that the majority of the target-anchor paths are enriched with relevant functions such as autophagic vesicle formation and caspase activation. These data can be viewed in HTML reports. The autophagy-apoptosis Cytoscape session (Autophagy-apoptosis.cys) and HTML reports (Autophagy Network Reports) are provided as Supplementary Materials.

Fig. 4. The autophagy-apoptosis crosstalk network with the default background. The resulting network model when ANAT is applied directly on the human PPI network instead of the PCD background network (which contains additional cell death–specific information such as missing nodes and edge directions).
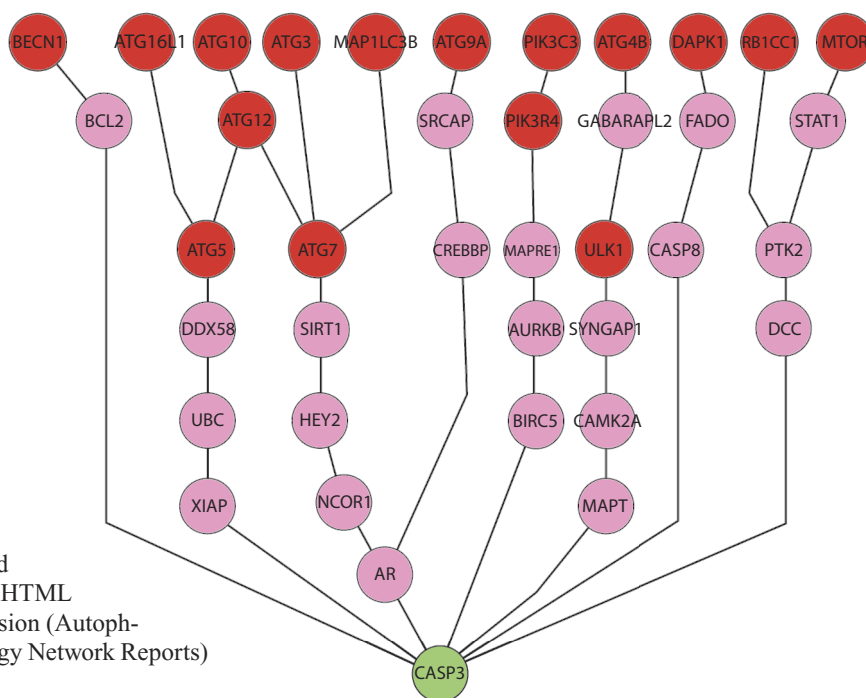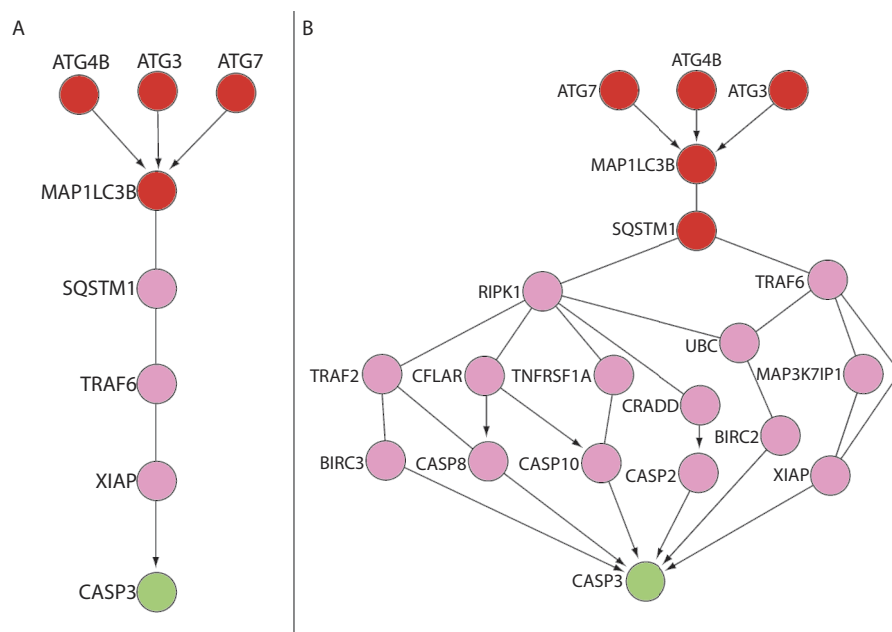
Fig. 5. Alternative pathways linking SQSTM1 to caspase-3 constructed by adjusting the network parameters. (**A**) Adding a node degree penalty results in a pathway connecting SQSTM1 to caspase-3 (CASP3) through TRAF6 and the IAP protein XIAP. (**B**) Using a margin of 1% instead of the default 0% revealed an expanded pathway in which SQSTM1 was connected to TRAF6 in addition to the connection to TRAF2 through RIPK1. Three additional pathways connecting RIPK1 to CASP3 (in addition to the original TRAF2-BIRC3 connection) converged into the upstream regulatory caspases—CASP8 (CASP8), CASP10, and CASP2—that process or activate CASP3.

Comparing the resulting model to the one obtained without augmenting the background network (Fig. 4) highlights the importance of incorporating expert knowledge in the process of network inference. Specifically, the public databases used for the background network of human PPI only partially described what is currently known about the relatively new and constantly developing study of autophagy. Consequently, the resulting model (Fig. 4) missed some relevant nodes (such as ATG13 and AMBRA-1) and interactions (such as the ones from DAPK1 to BECN1 and MTOR to ULK1) that are included in the model produced from the augmented network (Fig. 2).

The genes associated with autophagy converged through three major pathways to caspase-3 (Fig. 2). Some of the intermediate proteins had not been previously associated with cell death signaling, such as YWHAG (a phosphoserine- and phosphothreonine-binding protein) and KIF23 (a kinesin-like protein), and others such as TRAF2 [a tumor necrosis factor (TNF) receptor–associated factor] and RIPK1 (a kinase) were identified in some settings of cell death but had not been associated with autophagy. Two of the three pathways end with a member of the inhibitor of apoptosis (IAP) family (BIRC3, also known as c-IAP2, or BIRC6, also known as Bruce), which func-

tions as an inhibitor of caspase-3 (*62*). This suggests that signals that control autophagy may also regulate apoptosis through modulation of IAP proteins. Of particular interest is the pathway connecting the autophagic protein MAP1LC3B (LC3B) to caspase-3 through SQSTM1 (also known as P62) (Fig. 2). SQSTM1 participates in autophagic degradation of ubiquitinated or aggregated proteins through direct binding to MAP1LC3B; thus, unlike the 17 selected autophagic proteins that participate in the formation of autophagosomes (the main organelles that execute the process), the role of SQSTM1 is to deliver the ubiquitinated or aggregated proteins to the autophagosomes (*63*). According to the proposed network, SQSTM1 emerges as a potentially important mediator targeting BIRC3 through RIPK1 and TRAF2 (Fig. 2). To further extend the basis role of SQSTM1 as a link between autophagy and apoptosis, we explored alternatives to this pathway by either enabling the node penalty option (using the default penalty settings; Fig. 5A) or increasing the margin to 1% (Fig. 5B). Both options resulted in networks in which SQSTM1 was connected to caspase-3 through TRAF6, which (like TRAF2) is a TNF receptor–associated factor that mediates a variety of signal transduction pathways. Together, the networks produced with ANAT (Figs. 2 and 5) indicate that SQSTM1 is a mediator of crosstalk between autophagy and apoptosis and acts through pathways involving members of the IAP family, which is supported by functional evidence linking SQSTM1 to both autophagy and apoptosis (*49*).

In summary, by following the ANAT workflow and using expert knowledge to refine the results, we constructed a protein subnetwork connecting autophagy with apoptosis. The pathways suggested by the model serve as practical scientific leads.

Supplementary Materials

Supplementary Text: Performance evaluation of the TLM network.

Table S1: Network sizes, data sources, and gene identifier types.

Table S2: Constraints used for the autophagy-apoptosis crosstalk network.

ANAT Plugin File (AnatPlugin.jar).

ANAT User Manual (ANAT_manual.pdf).

Gene Identifiers: A compressed archive of text files containing all the gene identifiers used in ANAT's database (Gene_identifiers.zip).

Autophagy Input Files: A compressed archive of the files for creating the autophagy-apoptosis crosstalk network (Autophagy_input.zip).

PROTOCOL

Autophagy Network Reports: A compressed archive of the HTML reports for the autophagy-apoptosis crosstalk network (Autophagy-NetworkReports.zip).

Autophagy-Apoptosis File: A Cytoscape session file for the autophagy-apoptosis crosstalk networks (Autophagy-apoptosis.cys).

References and Notes

1. S. J. Wodak, S. Pu, J. Vlasblom, B. Séraphin, Challenges and rewards of interaction proteomics. *Mol. Cell. Proteomics* **8**, 3–18 (2009).
2. E. Yeger-Lotem, L. Riva, L. J. Su, A. D. Gitler, A. G. Cashikar, O. D. King, P. K. Auluck, M. L. Geddie, J. S. Valastyan, D. R. Karger, S. Lindquist, E. Fraenkel, Bridging high-throughput genetic and transcriptional data reveals cellular responses to α-synuclein toxicity. *Nat. Genet.* **41**, 316–323 (2009).
3. M. A. Pujana, J. D. Han, L. M. Starita, K. N. Stevens, M. Tewari, J. S. Ahn, G. Rennert, V. Moreno, T. Kirchhoff, B. Gold, V. Assmann, W. M. Elshamy, J. F. Rual, D. Levine, L. S. Rozek, R. S. Gelman, K. C. Gunsalus, R. A. Greenberg, B. Sobhian, N. Bertin, K. Venkatesan, N. Ayivi-Guedehoussou, X. Solé, P. Hernández, C. Lázaro, K. L. Nathanson, B. L. Weber, M. E. Cusick, D. E. Hill, K. Offit, D. M. Livingston, S. B. Gruber, J. D. Parvin, M. Vidal, Network modeling links breast cancer susceptibility and centrosome dysfunction. *Nat. Genet.* **39**, 1338–1349 (2007).
4. H. Goehler, M. Lalowski, U. Stelzl, S. Waelter, M. Stroedicke, U. Worm, A. Droege, K. S. Lindenberg, M. Knoblich, C. Haenig, M. Herbst, J. Suopanki, E. Scherzinger, C. Abraham, B. Bauer, R. Hasenbank, A. Fritzsche, A. H. Ludewig, K. Büssow, S. H. Coleman, C. A. Gutekunst, B. G. Landwehrmeyer, H. Lehrach, E. E. Wanker, A protein interaction network links GIT1, an enhancer of huntingtin aggregation, to Huntington's disease. *Mol. Cell* **15**, 853–865 (2004).
5. S. E. Calvano, W. Xiao, D. R. Richards, R. M. Felciano, H. V. Baker, R. J. Cho, R. O. Chen, B. H. Brownstein, J. P. Cobb, S. K. Tschoeke, C. Miller-Graziano, L. L. Moldawer, M. N. Mindrinos, R. W. Davis, R. G. Tompkins, S. F. Lowry; Inflamm and Host Response to Injury Large Scale Collab. Res. Program, A network-based analysis of systemic inflammation in humans. *Nature* **437**, 1032–1037 (2005).
6. M. R. Said, T. J. Begley, A. V. Oppenheim, D. A. Lauffenburger, L. D. Samson, Global network analysis of phenotypic effects: Protein networks and toxicity modulation in *Saccharomyces cerevisiae*. *Proc. Natl. Acad. Sci. U.S.A.* **101**, 18006–18011 (2004).
7. M. S. Scott, T. Perkins, S. Bunnell, F. Pepin, D. Y. Thomas, M. Hallett, Identifying regulatory subnetworks for a set of genes. *Mol. Cell. Proteomics* **4**, 683–692 (2005).
8. N. Yosef, L. Ungar, E. Zalckvar, A. Kimchi, M. Kupiec, E. Ruppin, R. Sharan, Toward accurate reconstruction of functional protein networks. *Mol. Syst. Biol.* **5**, 248 (2009).
9. S. Rashi-Elkeles, R. Elkon, N. Weizman, C. Linhart, N. Amariglio, G. Sternberg, G. Rechavi, A. Barzilai, R. Shamir, Y. Shiloh, Parallel induction of ATM-dependent pro- and antiapoptotic signals in response to ionizing radiation in murine lymphoid tissue. *Oncogene* **25**, 1584–1592 (2006).
10. F. Iragne, M. Nikolski, B. Mathieu, D. Auber, D. Sherman, ProViz: Protein interaction visualization and exploration. *Bioinformatics* **21**, 272–274 (2005).
11. C. L. Myers, D. Robson, A. Wible, M. A. Hibbs, C. Chiriac, C. L. Theesfeld, K. Dolinski, O. G. Troyanskaya, Discovery of biological networks from diverse functional genomic data. *Genome Biol.* **6**, R114 (2005).
12. E. G. Cerami, G. D. Bader, B. E. Gross, C. Sander, cPath: Open source software for collecting, storing, and querying biological pathways. *BMC Bioinformatics* **7**, 497 (2006).
13. R. Aragues, D. Jaeggi, B. Oliva, PIANA: Protein interactions and network analysis. *Bioinformatics* **22**, 1015–1017 (2006).
14. S. I. Berger, J. M. Posner, A. Ma'ayan, Genes2Networks: Connecting lists of gene symbols using mammalian protein interactions databases. *BMC Bioinformatics* **8**, 372 (2007).
15. M. S. Cline, M. Smoot, E. Cerami, A. Kuchinsky, N. Landys, C. Workman, R. Christmas, I. Avila-Campilo, M. Creech, B. Gross, K. Hanspers, R. Isserlin, R. Kelley, S. Killcoyne, S. Lotia, S. Maere, J. Morris, K. Ono, V. Pavlovic, A. R. Pico, A. Vailaya, P. L. Wang, A. Adler, B. R. Conklin, L. Hood, M. Kuiper, C. Sander, I. Schmulevich, B. Schwikowski, G. J. Warner, T. Ideker, G. D. Bader, Integration of biological networks and gene expression data using Cytoscape. *Nat. Protoc.* **2**, 2366–2382 (2007).
16. A. Alexeyenko, E. L. Sonnhammer, Global networks of functional coupling in eukaryotes from comprehensive data integration. *Genome Res.* **19**, 1107–1116 (2009).
17. A. V. Antonov, S. Dietmann, I. Rodchenkov, H. W. Mewes, PPI spider: A tool for the interpretation of proteomics data in the context of protein-protein interaction networks. *Proteomics* **9**, 2740–2749 (2009).
18. J. H. Konieczka, K. Drew, A. Pine, K. Belasco, S. Davey, T. A. Yatskievych, R. Bonneau, P. B. Antin, BioNetBuilder2.0: Bringing systems biology to chicken and other model organisms. *BMC Genomics* **10** (suppl. 2), S6 (2009).
19. A. Martin, M. E. Ochagavia, L. C. Rabasa, J. Miranda, J. Fernandez-de-Cossio, R. Bringas, BisoGenet: A new tool for gene network building, visualization and analysis. *BMC Bioinformatics* **11**, 91 (2010).
20. D. Szklarczyk, A. Franceschini, M. Kuhn, M. Simonovic, A. Roth, P. Minguez, T. Doerks, M. Stark, J. Muller, P. Bork, L. J. Jensen, C. von Mering, The STRING database in 2011: Functional interaction networks of proteins, globally integrated and scored. *Nucleic Acids Res.* **39** (Database issue), D561–D568 (2011).
21. M. Laakso, S. Hautaniemi, Integrative platform to translate gene sets to networks. *Bioinformatics* **26**, 1802–1803 (2010).
22. R. Elkon, R. Vesterman, N. Amit, I. Ulitsky, I. Zohar, M. Weisz, G. Mass, N. Orlev, G. Sternberg, R. Blekhman, J. Assa, Y. Shiloh, R. Shamir, SPIKE—A database, visualization and analysis tool of cellular signaling pathways. *BMC Bioinformatics* **9**, 110 (2008).
23. E. Demir, O. Babur, U. Dogrusoz, A. Gursoy, G. Nisanci, R. Cetin-Atalay, M. Ozturk, PATIKA: An integrated visual environment for collaborative construction and analysis of cellular pathways. *Bioinformatics* **18**, 996–1003 (2002).
24. J. Chen, E. E. Bardes, B. J. Aronow, A. G. Jegga, ToppGene Suite for gene list enrichment analysis and candidate gene prioritization. *Nucleic Acids Res.* **37** (Web Server issue), W305–W311 (2009).
25. J. Montojo, K. Zuberi, H. Rodriguez, F. Kazi, G. Wright, S. L. Donaldson, Q. Morris, G. D. Bader, GeneMANIA Cytoscape plugin: Fast gene function predictions on the desktop. *Bioinformatics* **26**, 2927–2928 (2010).
26. A. Subramanian, P. Tamayo, V. K. Mootha, S. Mukherjee, B. L. Ebert, M. A. Gillette, A. Paulovich, S. L. Pomeroy, T. R. Golub, E. S. Lander, J. P. Mesirov, Gene set enrichment analysis: A knowledge-based approach for interpreting genome-wide expression profiles. *Proc. Natl. Acad. Sci. U.S.A.* **102**, 15545–15550 (2005).
27. T. Ideker, O. Ozier, B. Schwikowski, A. F. Siegel, Discovering regulatory and signalling circuits in molecular interaction networks. *Bioinformatics* **18** (suppl. 1), S233–S240 (2002).
28. I. Ulitsky, R. Shamir, Identification of functional modules using network topology and high-throughput data. *BMC Syst. Biol.* **1**, 8 (2007).
29. D. Beisser, G. W. Klau, T. Dandekar, T. Müller, M. T. Dittrich, BioNet: An R-Package for the functional analysis of biological networks. *Bioinformatics* **26**, 1129–1130 (2010).
30. A. Nikitin, S. Egorov, N. Daraselia, I. Mazo, Pathway studio—the analysis and navigation of molecular networks. *Bioinformatics* **19**, 2155–2157 (2003).
31. J. C. Rain, L. Selig, H. De Reuse, V. Battaglia, C. Reverdy, S. Simon, G. Lenzen, F. Petel, J. Wojcik, V. Schächter, Y. Chemama, A. Labigne, P. Legrain, The protein-protein interaction map of *Helicobacter pylori*. *Nature* **409**, 211–215 (2001).
32. S. Li, C. M. Armstrong, N. Bertin, H. Ge, S. Milstein, M. Boxem, P. O. Vidalain, J. D. Han, A. Chesneau, T. Hao, D. S. Goldberg, N. Li, M. Martinez, J. F. Rual, P. Lamesch, L. Xu, M. Tewari, S. L. Wong, L. V. Zhang, G. F. Berriz, L. Jacotot, P. Vaglio, J. Reboul, T. Hirozane-Kishikawa, Q. Li, H. W. Gabel, A. Elewa, B. Baumgartner, D. J. Rose, H. Yu, S. Bosak, R. Sequerra, A. Fraser, S. E. Mango, W. M. Saxton, S. Strome, S. Van Den Heuvel, F. Piano, J. Vandenhaute, C. Sardet, M. Gerstein, L. Doucette-Stamm, K. C. Gunsalus, J. W. Harper, M. E. Cusick, F. P. Roth, D. E. Hill, M. Vidal, A map of the interactome network of the metazoan *C. elegans*. *Science* **303**, 540–543 (2004).
33. D. J. LaCount, M. Vignali, R. Chettier, A. Phansalkar, R. Bell, J. R. Hesselberth, L. W. Schoenfeld, I. Ota, S. Sahasrabudhe, C. Kurschner, S. Fields, R. E. Hughes, A protein interaction network of the malaria parasite *Plasmodium falciparum*. *Nature* **438**, 103–107 (2005).
34. L. Giot, J. S. Bader, C. Brouwer, A. Chaudhuri, B. Kuang, Y. Li, Y. L. Hao, C. E. Ooi, B. Godwin, E. Vitols, G. Vijayadamodar, P. Pochart, H. Machineni, M. Welsh, Y. Kong, B. Zerhusen, R. Malcolm, Z. Varrone, A. Collis, M. Minto, S. Burgess, L. McDaniel, E. Stimpson, F. Spriggs, J. Williams, K. Neurath, N. Ioime, M. Agee, E. Voss, K. Furtak, R.

Renzulli, N. Aanensen, S. Carrolla, E. Bickelhaupt, Y. Lazovatsky, A. DaSilva, J. Zhong, C. A. Stanyon, R. L. Finley, Jr, K. P. White, M. Braverman, T. Jarvie, S. Gold, M. Leach, J. Knight, R. A. Shimkets, M. P. McKenna, J. Chant, J. M. Rothberg, A protein interaction map of *Drosophila melanogaster*. *Science* **302**, 1727–1736 (2003).

35. J. F. Rual, K. Venkatesan, T. Hao, T. Hirozane-Kishikawa, A. Dricot, N. Li, G. F. Berriz, F. D. Gibbons, M. Dreze, N. Ayivi-Guedehoussou, N. Klitgord, C. Simon, M. Boxem, S. Milstein, J. Rosenberg, D. S. Goldberg, L. V. Zhang, S. L. Wong, G. Franklin, S. Li, J. S. Albala, J. Lim, C. Fraughton, E. Llamosas, S. Cevik, C. Bex, P. Lamesch, R. S. Sikorski, J. Vandenhaute, H. Y. Zoghbi, A. Smolyar, S. Bosak, R. Sequerra, L. Doucette-Stamm, M. E. Cusick, D. E. Hill, F. P. Roth, M. Vidal, Towards a proteome-scale map of the human protein-protein interaction network. *Nature* **437**, 1173–1178 (2005).

36. U. Stelzl, U. Worm, M. Lalowski, C. Haenig, F. H. Brembeck, H. Goehler, M. Stroedicke, M. Zenkner, A. Schoenherr, S. Koeppen, J. Timm, S. Mintzlaff, C. Abraham, N. Bock, S. Kietzmann, A. Goedde, E. Toksöz, A. Droege, S. Krobitsch, B. Korn, W. Birchmeier, H. Lehrach, E. E. Wanker, A human protein-protein interaction network: A resource for annotating the proteome. *Cell* **122**, 957–968 (2005).

37. R. M. Ewing, P. Chu, F. Elisma, H. Li, P. Taylor, S. Climie, L. McBroom-Cerajewski, M. D. Robinson, L. O'Connor, M. Li, R. Taylor, M. Dharsee, Y. Ho, A. Heilbut, L. Moore, S. Zhang, O. Ornatsky, Y. V. Bukhman, M. Ethier, Y. Sheng, J. Vasilescu, M. Abu-Farha, J. P. Lambert, H. S. Duewel, I. I. Stewart, B. Kuehl, K. Hogue, K. Colwill, K. Gladwish, B. Muskat, R. Kinach, S. L. Adams, M. F. Moran, G. B. Morin, T. Topaloglou, D. Figeys, Large-scale mapping of human protein-protein interactions by mass spectrometry. *Mol. Syst. Biol.* **3**, 89 (2007).

38. M. Arifuzzaman, M. Maeda, A. Itoh, K. Nishikata, C. Takita, R. Saito, T. Ara, K. Nakahigashi, H. C. Huang, A. Hirai, K. Tsuzuki, S. Nakamura, M. Altaf-Ul-Amin, T. Oshima, T. Baba, N. Yamamoto, T. Kawamura, T. Ioka-Nakamichi, M. Kitagawa, M. Tomita, S. Kanaya, C. Wada, H. Mori, Large-scale identification of protein-protein interaction of *Escherichia coli* K-12. *Genome Res.* **16**, 686–691 (2006).

39. H. Yu, N. M. Luscombe, H. X. Lu, X. Zhu, Y. Xia, J. D. Han, N. Bertin, S. Chung, M. Vidal, M. Gerstein, Annotation transfer between genomes: Protein-protein interologs and protein-DNA regulogs. *Genome Res.* **14**, 1107–1118 (2004).

40. B. J. Breitkreutz, C. Stark, T. Reguly, L. Boucher, A. Breitkreutz, M. Livstone, R. Oughtred, D. H. Lackner, J. Bähler, V. Wood, K. Dolinski, M. Tyers, The BioGRID Interaction Database: 2008 update. *Nucleic Acids Res.* **36** (Database issue), D637–D640 (2008).

41. I. Xenarios, L. Salwínski, X. J. Duan, P. Higney, S. M. Kim, D. Eisenberg, DIP, the Database of Interacting Proteins: A research tool for studying cellular networks of protein interactions. *Nucleic Acids Res.* **30**, 303–305 (2002).

42. S. Peri, J. D. Navarro, R. Amanchy, T. Z. Kristiansen, C. K. Jonnalagadda, V. Surendranath, V. Niranjan, B. Muthusamy, T. K. Gandhi, M. Gronborg, N. Ibarrola, N. Deshpande, K. Shanker, H. N. Shivashankar, B. P. Rashmi, M. A. Ramya, Z. Zhao, K. N. Chandrika, N. Padma, H. C. Harsha, A. J. Yatish, M. P. Kavitha, M. Menezes, D. R. Choudhury, S. Suresh, N. Ghosh, R. Saravana, S. Chandran, S. Krishna, M. Joy, S. K. Anand, V. Madavan, A. Joseph, G. W. Wong, W. P. Schiemann, S. N. Constantinescu, L. Huang, R. Khosravi-Far, H. Steen, M. Tewari, S. Ghaffari, G. C. Blobe, C. V. Dang, J. G. Garcia, J. Pevsner, O. N. Jensen, P. Roepstorff, K. S. Deshpande, A. M. Chinnaiyan, A. Hamosh, A. Chakravarti, A. Pandey, Development of human protein reference database as an initial platform for approaching systems biology in humans. *Genome Res.* **13**, 2363–2371 (2003).

43. A. Lachmann, H. Xu, J. Krishnan, S. I. Berger, A. R. Mazloom, A. Ma'ayan, ChEA: Transcription factor regulation inferred from integrating genome-wide ChIP-X experiments. *Bioinformatics* **26**, 2438–2444 (2010).

44. T. I. Lee, N. J. Rinaldi, F. Robert, D. T. Odom, Z. Bar-Joseph, G. K. Gerber, N. M. Hannett, C. T. Harbison, C. M. Thompson, I. Simon, J. Zeitlinger, E. G. Jennings, H. L. Murray, D. B. Gordon, B. Ren, J. J. Wyrick, J. B. Tagne, T. L. Volkert, E. Fraenkel, D. K. Gifford, R. A. Young, Transcriptional regulatory networks in *Saccharomyces cerevisiae*. *Science* **298**, 799–804 (2002).

45. H. Y. Chuang, E. Lee, Y. T. Liu, D. Lee, T. Ideker, Network-based classification of breast cancer metastasis. *Mol. Syst. Biol.* **3**, 140 (2007).

46. M. A. Harris, J. Clark, A. Ireland, J. Lomax, M. Ashburner, R. Foulger, K. Eilbeck, S. Lewis, B. Marshall, C. Mungall, J. Richter, G. M. Rubin, J. A. Blake, C. Bult, M. Dolan, H. Drabkin, J. T. Eppig, D. P. Hill, L. Ni, M. Ringwald, R. Balakrishnan, J. M. Cherry, K. R. Christie, M. C. Costanzo, S. S. Dwight, S. Engel, D. G. Fisk, J. E. Hirschman, E. L. Hong, R. S. Nash, A. Sethuraman, C. L. Theesfeld, D. Botstein, K. Dolinski, B. Feierbach, T. Berardini, S. Mundodi, S. Y. Rhee, R. Apweiler, D. Barrell, E. Camon, E. Dimmer, V. Lee, R. Chisholm, P. Gaudet, W. Kibbe, R. Kishore, E. M. Schwarz, P. Sternberg, M. Gwinn, L. Hannick, J. Wortman, M. Berriman, V. Wood, N. de la Cruz, P. Tonellato, P. Jaiswal, T. Seigfried, R. White, Gene Ontology Consortium, The Gene Ontology (GO) database and informatics resource. *Nucleic Acids Res.* **32** (database issue), D258–D261 (2004).

47. M. Kanehisa, The KEGG database. *Novartis Found. Symp.* **247**, 91–101, discussion 101–103, 119–128, 244–252 (2002).

48. R. Mathew, C. M. Karp, B. Beaudoin, N. Vuong, G. Chen, H. Y. Chen, K. Bray, A. Reddy, G. Bhanot, C. Gelinas, R. S. Dipaola, V. Karantza-Wadsworth, E. White, Autophagy suppresses tumorigenesis through elimination of p62. *Cell* **137**, 1062–1075 (2009).

49. J. Moscat, M. T. Diaz-Meco, p62 at the crossroads of autophagy, apoptosis, and cancer. *Cell* **137**, 1001–1004 (2009).

50. E. Zalckvar, N. Yosef, S. Reef, Y. Ber, A. D. Rubinstein, I. Mor, R. Sharan, E. Ruppin, A. Kimchi, A systems level strategy for analyzing the cell death network: Implication in exploring the apoptosis/autophagy connection. *Cell Death Differ.* **17**, 1244–1253 (2010).

51. T. de Lange, Shelterin: The protein complex that shapes and safeguards human telomeres. *Genes Dev.* **19**, 2100–2110 (2005).

52. K. Collins, J. R. Mitchell, Telomerase in the human organism. *Oncogene* **21**, 564–579 (2002).

53. S. H. Askree, T. Yehuda, S. Smolikov, R. Gurevich, J. Hawk, C. Coker, A. Krauskopf, M. Kupiec, M. J. McEachern, A genome-wide screen for *Saccharomyces cerevisiae* deletion mutants that affect telomere length. *Proc. Natl. Acad. Sci. U.S.A.* **101**, 8658–8663 (2004).

54. T. Gatbonton, M. Imbesi, M. Nelson, J. M. Akey, D. M. Ruderfer, L. Kruglyak, J. A. Simon, A. Bedalov, Telomere length as a quantitative trait: Genome-wide survey and genetic mapping of telomere length-control genes in yeast. *PLoS Genet.* **2**, e35 (2006).

55. L. Ungar, N. Yosef, Y. Sela, R. Sharan, E. Ruppin, M. Kupiec, A genome-wide screen for essential yeast genes that affect telomere length maintenance. *Nucleic Acids Res.* **37**, 3840–3849 (2009).

56. N. Yosef, M. Kupiec, E. Ruppin, R. Sharan, A complex-centric view of protein network evolution. *Nucleic Acids Res.* **37**, e88 (2009).

57. R. Shachar, L. Ungar, M. Kupiec, E. Ruppin, R. Sharan, A systems-level approach to mapping the telomere length maintenance gene circuitry. *Mol. Syst. Biol.* **4**, 172 (2008).

58. S. Khuller, B. Raghavachari, N. Young, Balancing minimum spanning trees and shortest-path trees. *Algorithmica* **14**, 305–321 (1995).

59. D. A. Nelson, T. T. Tan, A. B. Rabson, D. Anderson, K. Degenhardt, E. White, Hypoxia and defective apoptosis drive genomic instability and tumorigenesis. *Genes Dev.* **18**, 2095–2107 (2004).

60. L. Galluzzi, M. C. Maiuri, I. Vitale, H. Zischka, M. Castedo, L. Zitvogel, G. Kroemer, Cell death modalities: Classification and pathophysiological implications. *Cell Death Differ.* **14**, 1237–1243 (2007).

61. A. Eisenberg-Lerner, S. Bialik, H. U. Simon, A. Kimchi, Life and death partners: Apoptosis, autophagy and the cross-talk between them. *Cell Death Differ.* **16**, 966–975 (2009).

62. D. L. Vaux, J. Silke, IAPs, RINGs and ubiquitylation. *Nat. Rev. Mol. Cell Biol.* **6**, 287–297 (2005).

63. M. Komatsu, S. Waguri, M. Koike, Y. S. Sou, T. Ueno, T. Hara, N. Mizushima, J. Iwata, J. Ezaki, S. Murata, J. Hamazaki, Y. Nishito, S. Iemura, T. Natsume, T. Yanagawa, J. Uwayama, E. Warabi, H. Yoshida, T. Ishii, A. Kobayashi, M. Yamamoto, Z. Yue, Y. Uchiyama, E. Kominami, K. Tanaka, Homeostatic levels of p62 control cytoplasmic inclusion body formation in autophagy-deficient mice. *Cell* **131**, 1149–1163 (2007).

Citation: N. Yosef, E. Zalckvar, A. D. Rubinstein, M. Homilius, N. Atias, L. Vardi, I. Berman, H. Zur, A. Kimchi, E. Ruppin, R. Sharan, ANAT: A tool for constructing and analyzing functional protein networks. *Sci. Signal.* **4**, pl1 (2011).

Article Tools	Visit the online version of this article to access the personalization and article tools: http://stke.sciencemag.org/content/4/196/pl1
Supplemental Materials	*"Supplementary Materials"* http://stke.sciencemag.org/content/suppl/2011/10/21/4.196.pl1.DC1
Related Content	The editors suggest related resources on *Science*'s sites: http://stke.sciencemag.org/content/sigtrans/5/220/eg5.full http://stke.sciencemag.org/content
References	This article cites 63 articles, 34 of which you can access for free at: http://stke.sciencemag.org/content/4/196/pl1#BIBL
Permissions	Obtain information about reproducing this article: http://www.sciencemag.org/about/permissions.dtl