

FNV: Flash Network Viewer



MOUNT SINAI
SCHOOL OF
MEDICINE



USER MANUAL

Written by: Ruth Dannenfelser

December 22, 2010

Modified: May 5, 2011

Installation

FNV is available for download at: <http://www.maayanlab.net/FNV>

The download contains the FNV.swf file, a source code package for embedding, documentation, and some sample network XML files. Make sure that you extract all the files from the zipped archive before attempting to run FNV.

Running FNV

FNV requires that you have flash player installed on your machine. You can get the latest version of flash player from Adobe: <http://get.adobe.com/flashplayer/>

There are four ways to run FNV. You can run the flash movie locally, you can embed the viewer in a web document, you can embed the viewer in a pdf file, or you can run the viewer on our website: <http://www.maayanlab.net/FNV/useViewer.html>

To run the viewer locally, you must place the XML file you want to visualize in the same directory as

the FNV.swf file. The XML file must be named network.xml or the viewer will not display the network. To incorporate the flash viewer into a web page you can embed it using html as seen below. There are two ways to load the network into the viewer. Follow the same procedure as when using FNV locally (naming the network, network.xml and placing it in the same directory as FNV.swf file) or by feeding a network name into the viewer using flash vars (see code below lines highlighted in red). Simply replace the networkGENERATED.xml filename in the code below with the filename of the network you wish to visualize, to set the loadFile variable within the Flash viewer.

```
<object height="548" width="796">
<param value="graphviewer_copy_for_web.swf" name="movie" />
<param value="true" name="allowFullScreen" />
<param value="always" name="allowScriptAccess" />
<param value="loadFile=networkGENERATED.xml" name="FlashVars"
id="flashvar" />
<param value="transparent" name="vmode" />
<embed height="548" width="796" flashvars="loadFile=networkGENERATED.xml"
allowfullscreen="true" allowscriptaccess="always" vmode="transparent"
type="application/x-shockwave-flash" src="graphviewer_copy_for_web.swf" />
</object>
```

Note: The viewer has full screen functionality that won't be enabled unless the allowfullscreen attribute is set to true.

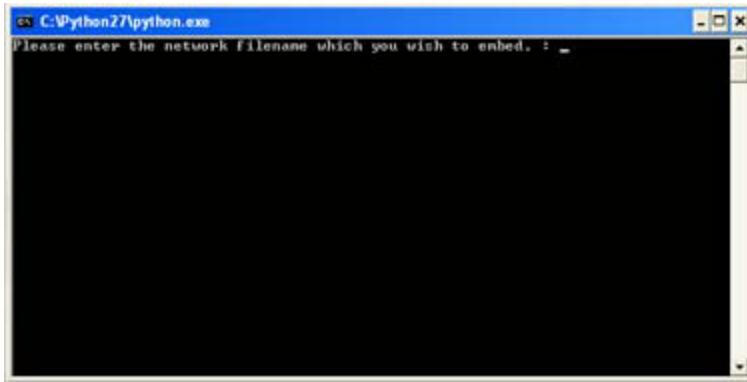
Embedding the viewer in a pdf file is a more advanced operation and requires that you have several programs installed on your machine.

Requirements:

1. Adobe Acrobat 9 Professional or higher. Without this software you will not be able to insert a Flash movie into a pdf.
2. An ActionScript 3.0 compiler. We recommend using Adobe Flash CS4 Professional but a free Flex 4 compiler is also available from Adobe.
3. A functional xml network that does not have any TRIANGLE nodes. A network with triangle nodes will not display when the viewer is embedded in a pdf.
4. Optional: Python 2.7 to utilize the automated embedding script.

Embedding Using the Python Script

If you have a functional network with no triangle nodes and all of the above software and want to continue, locate the "FNV for embedding" folder which was included in the downloaded archive. Those familiar with the source code for FNV will notice that one of the files GraphMain.as is missing from the folder. You will also notice that an extra file GraphMainIn.as has taken its place. GraphMainIn.as is the file in which the python script, embed.py, will use to create GraphMain.as. It is VERY important that you do not edit GraphMainIn.as if you intend to use the python script for embedding. (If you wish to manually insert a network into GraphMainIn.as then read the next section below.) Double click the python script to execute it. Once the python script is running you will be prompted to enter a file name.



Enter the filename where the network xml file is that you want to visualize and press enter. Once the script executes a new GraphMain.as file with an embedded network should be generated and is ready to be compiled.

Embedding Without Using the Python Script

If you wish to bypass the script you can manually edit GraphMainIn.as. First copy the contents of GraphMainIn.as into a new file and name it GraphMain.as. Secondly find the line with the comment, "//hardcoded network" The next line should contain the text, "var myXML:XML = new XML();". Insert the network between the parentheses in the format of a string, do not forget to escape the quotation marks inside the string.

Compiling the ActionScript files

To compile using Flash CS4, open the FNV.fla file and select test movie. You should now see an FNV.swf file in the directory with your ActionScript files. Double click the FNV.swf file to make sure the network is displaying. If no network is visible make sure that the network xml file embedded has the correct syntax and try again. Once the embedded network is displaying, it is ready to be imported into the pdf.

Inserting the Flash movie in a pdf

Create a new Adobe Acrobat 9 file or open an existing one and navigate to tools->multimedia->flash tool. Draw an area on the pdf where you want to place the FNV. You will then be prompted to browse for the .swf file, find FNV.swf and select it. Select the arrow tool and press the play button to make sure the movie works. If the FNV controls display but no network, make sure that no triangle nodes are in the embedded network, and try again.

XML File Syntax

XML files read by the viewer have the following basic structure:

```
<network>
  <node>
    <desc/>
  </node>
  <edge>
    <desc/>
  </edge>
</network>
```

Network Element

The network element attributes' control the aesthetics of the network and the underlying algorithm used to render the network.

XML Example:

```
<network name="my network" backgroundcolor="0xFFFFFFFF"  
nodeselectioncolor="0xF76541" edgeselectioncolor="0x6698FF" fade="true"  
description="true" doubleclick="true" layoutmode="force">
```

Color Codes: The viewer requires that all colors be defined using hexadecimal values. Unlike color codes used in HTML these values must start with a "0x" instead of a "#". The three pairs of two digit hex values represent red, green, and blue. More information about hexadecimal color codes including a color picker can be found at: http://www.w3schools.com/HTML/html_colors.asp.

Attributes: `name`– Used to assign a name to the network.

`backgroundcolor`– A hex color code which specifies the background for the network.

`nodeselectioncolor`– A hex color code which specifies the highlight color of the node when the node is hovered over.

`edgeselectioncolor`– A hex color code which specifies the highlight color of the edge when hovered over.

`fade`– A Boolean (true or false) value that turns subnetwork fading on mouseover on or off.

`description`– A Boolean (true or false) value that adds or removes the description box.

`doubleclick`– A Boolean (true or false) value used to enable full screen mode.

`layoutmode`– Allows for the selection of a graph drawing algorithm. All of the graph drawing algorithms render planar straight-line drawings. Note: with the exception of the predefined graph algorithm all graphs with more than 70 nodes will be drawing using the default algorithm potentially producing undesired results.

Options:

`predefined`– Allows the user to specify their own position, in pixels, for each node. Additional attributes `xPos` and `yPos` need to be added to the node representing the x position on the page and the y pixel position on the page. This mode is the preferred method for visualizing networks over 100 nodes.

force - Positions the nodes using attraction and repulsion forces so that all the edges are roughly equal in length while attempting to minimize edge overlap. When dragging and repositioning a node the force of repulsion may reposition some of the surrounding nodes.

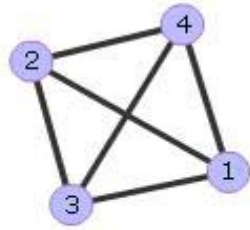


Figure 1. An undirected graph drawn using the force algorithm.

hierarchy - Arranges a directed graph into layers flowing from a source node to one or more sink nodes.

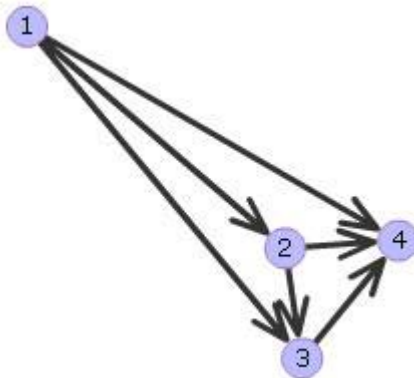


Figure 2. A directed graph drawn using the hierarchy algorithm.

radial - Arranges the network circularly around a root node. The additional attribute layer can be used to set the positions of the nodes in the circle. Layer 0 represents the center of the circle while larger layer values are placed at increasingly larger radii from the center.

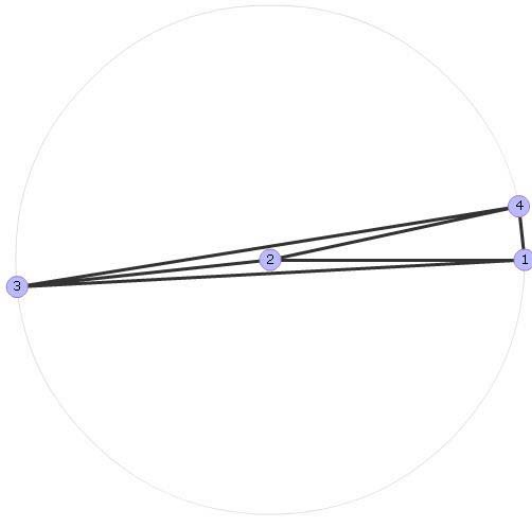


Figure 3. An undirected graph with two layers, 0 and 1.

spectral-

Renders the network using eigenvectors of the adjacency matrix associated with the graph. Attempts to reduce edge overlap.

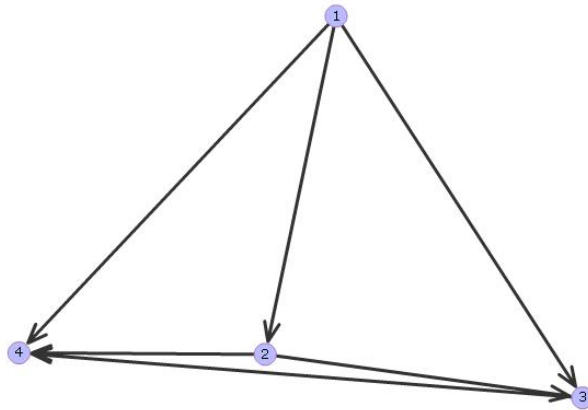


Figure 4. A directed graph drawn using the spectral algorithm.

spectral-hierarchy-

This option first arranges the nodes hierarchically then arranges each layer in the hierarchy using the eigenvectors of the adjacency matrix.

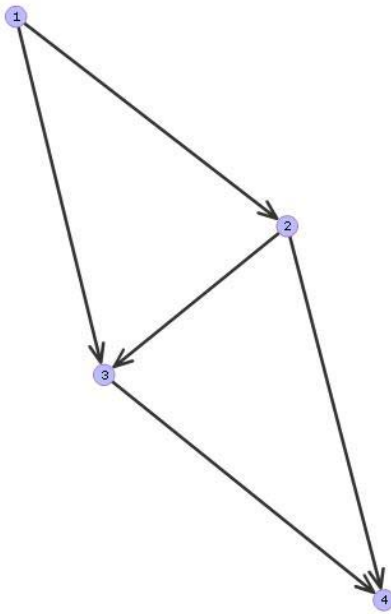


Figure 5. A directed graph drawn using the spectral-hierarchy algorithm.

default - When no valid option is entered the viewer simply places all nodes in a stack on top of each other.

Node Element

The node element is used to create nodes for the graph. The attributes allow users to specify the aesthetics of the nodes (which can be used to distinguish classes of nodes), provide additional information to the chosen layout algorithm, and add a hyperlink to be displayed within the description box.

Note: Since FNV was designed to visualize small to moderately sized networks it is not recommended that users create more than 70 nodes, unless the predefined graph drawing algorithm is used.

XML Examples:

```

<node id="1" name="SRI" bgcolor="0xbbbbff"
url="http://www.uniprot.org/uniprot/?query=SRI&sort=score" shape="roundbox"
textsize="12" textcolor="0x000000" level="0">
<node id="2" name="Dopamine" bgcolor="0xFF9999"
url="http://stke.sciencemag.org/cgi/cm/stkecm;CMN_21164" shape="circle"
textsize="20" textcolor="0x330000" nodesize="3" xPos="750" yPos="0">
  
```


id - Unique number used to identify the node.
(IMPORTANT: Node ids must be numeric, in order, start with 1, and increment by one with no skipping.)


name - Name of the node which is displayed on the node.

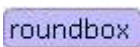
`bgcolor` – A hex color code which specifies the background color of the node, useful for delineating node types.

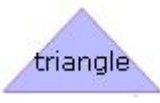
`url` – (optional) A link that will automatically open when the node description is clicked.

`shape` – Used to specify node shape can be used to distinguish particular types of nodes.
Options:

`box:` 

`circle:` 

`roundbox:` 

`triangle:` 

Alternatively an image can be used by using the `image` option.

`textsize` – Controls the font size of the name on the node.

`textcolor` – A hex color code value specifying the color of the text on the node.

Node Specific Attributes:

`nodesize` – When circle nodes are created this option can be used to set the diameter.

`imageurl` – When image node is selected the option specifies the location of the image to be used.

Layout Algorithm Specific Attributes:

`level` – Used only when the radial layout mode is selected. Levels represent distance from the radius, therefore level 0 is in the center of the circle.

`xPos` – Used only when the predefined option is selected. Specifies in pixels the x coordinate for the location of the node.

`yPos` – Used only when the predefined option is selected. Specifies in pixels the y coordinate for the location of the node.

A description can be placed in-between the <node>tags enclosed in <desc>tags. This description can contain plain text and images. It is displayed in the description box whenever the node is moused over. Note: The description is limited to 400 pixels wide. Images wider will be cutoff.

Edge Element

The edge element is used to connect two nodes. The attributes are used to specify the type and look of the edges.

XML Examples:

```
<edge id="1" name="Binding" type="bidirect" thickness="3" url=""
source="1"target="2" color="0x333333">

<edge id="2" name="Regulation" type="arrow"
thickness="4"url="http://stke.sciencemag.org/cgi/cm/stkecm;CMR_21000"
source="56"target="4" color="0x333333">
```

id – Number used to identify the edge.

name – (optional) Used to describe the type of edge.

type –Specifies the type of edge to draw between the source and target nodes.

Options:

bidirectional– A solid line attached to two nodes.



Figure 6. A bidirectional line signifying a non-covalent interaction between two proteins CRK and C3G.

arrow – A line that points from the source node to the target node.

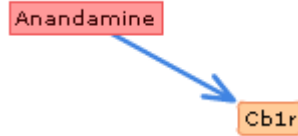


Figure 7. An arrow signifying an interaction between a ligand and a protein.

ortholine – A line that represents suppression of the target node.



Figure 8. A protein-protein interaction in which G_alpha_o suppresses Rap1GAP.

thickness – A positive whole number used to specify the width of the edge.

url – (optional) A link that can be displayed when clicking the edge.

source – The source node's id number (IMPORTANT: if this is omitted or incorrect a random source node is selected).

target – The target node's id number (IMPORTANT: if omitted or incorrect edge will not appear).

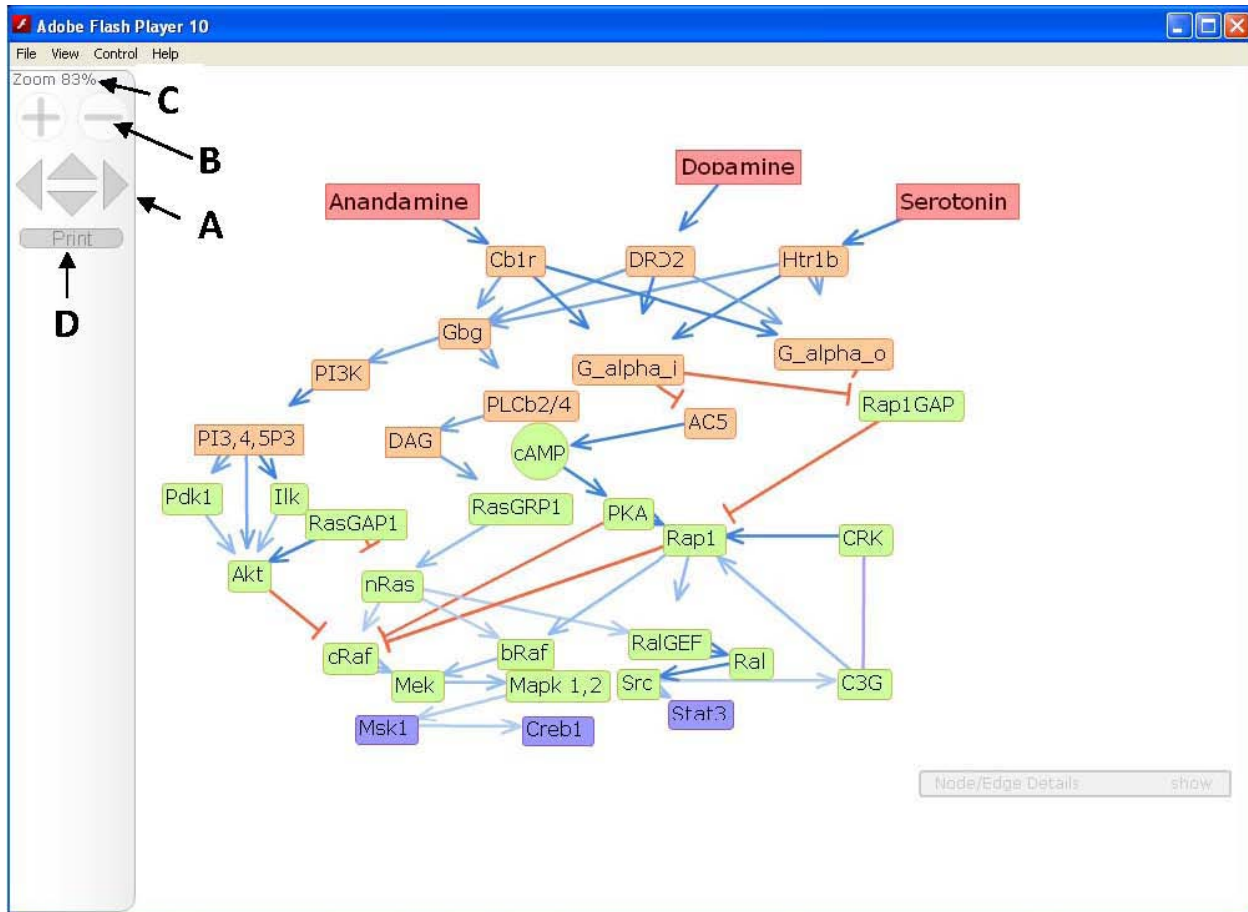
color – A hex code used to specify the edge color.

A description can also be placed in between the <edge> tags enclosed in <desc> tags. This description, similar to the node description, can contain plain text and images. It is displayed in the description box when the edge is moused over.

Using the Viewer

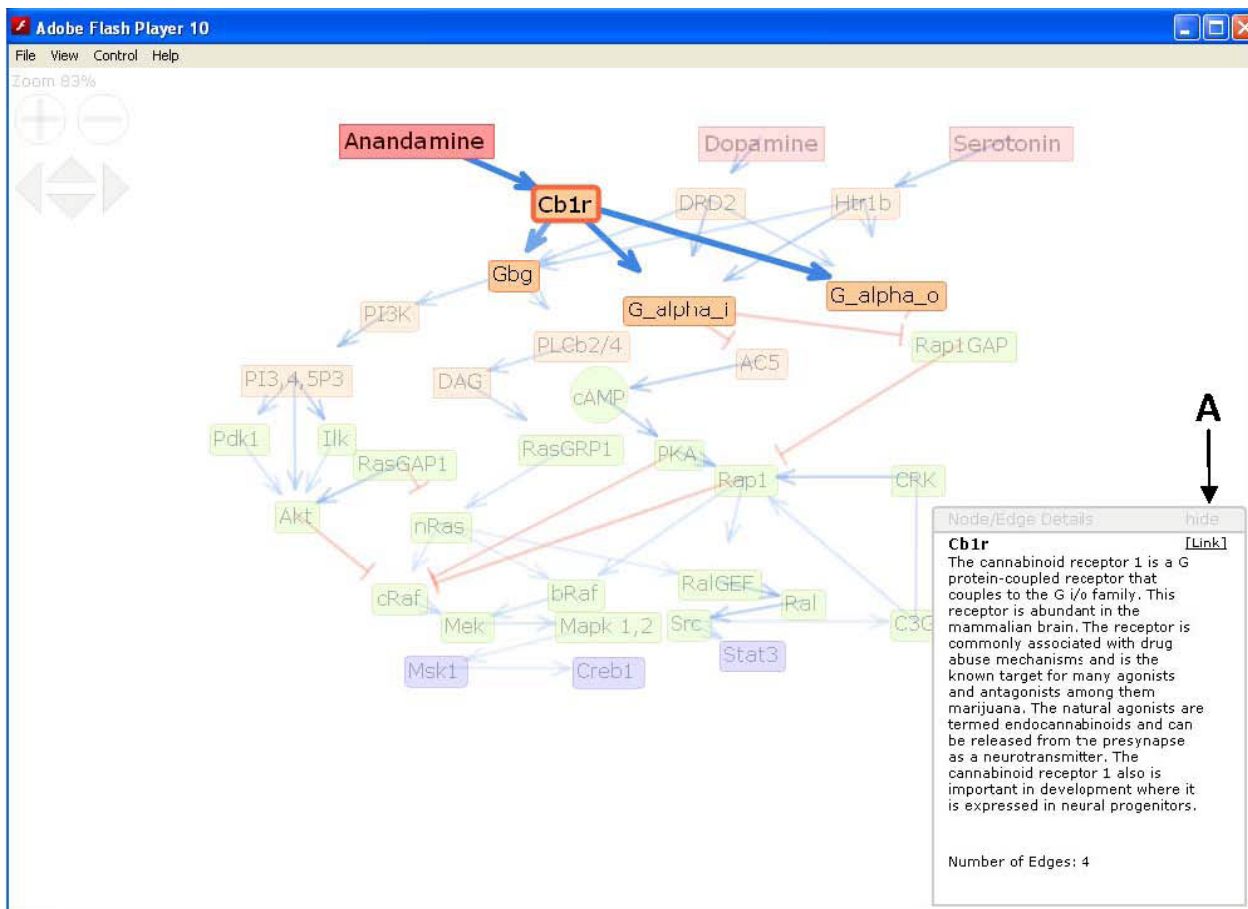
As discussed in the running and installation instructions the viewer can be run by double clicking on the graphviewer.swf or embedded within a webpage. Once loaded the network described by the file network.xml should be displayed. (If nothing is displayed, make sure that the xml file is named network.xml and is in the same directory as graphviewer.swf. Be sure to also check the syntax of the xml file and make sure that all elements are closed properly and that all edges reference valid nodes.)

Once started the viewer should be displaying your network.xml file like the one seen below.

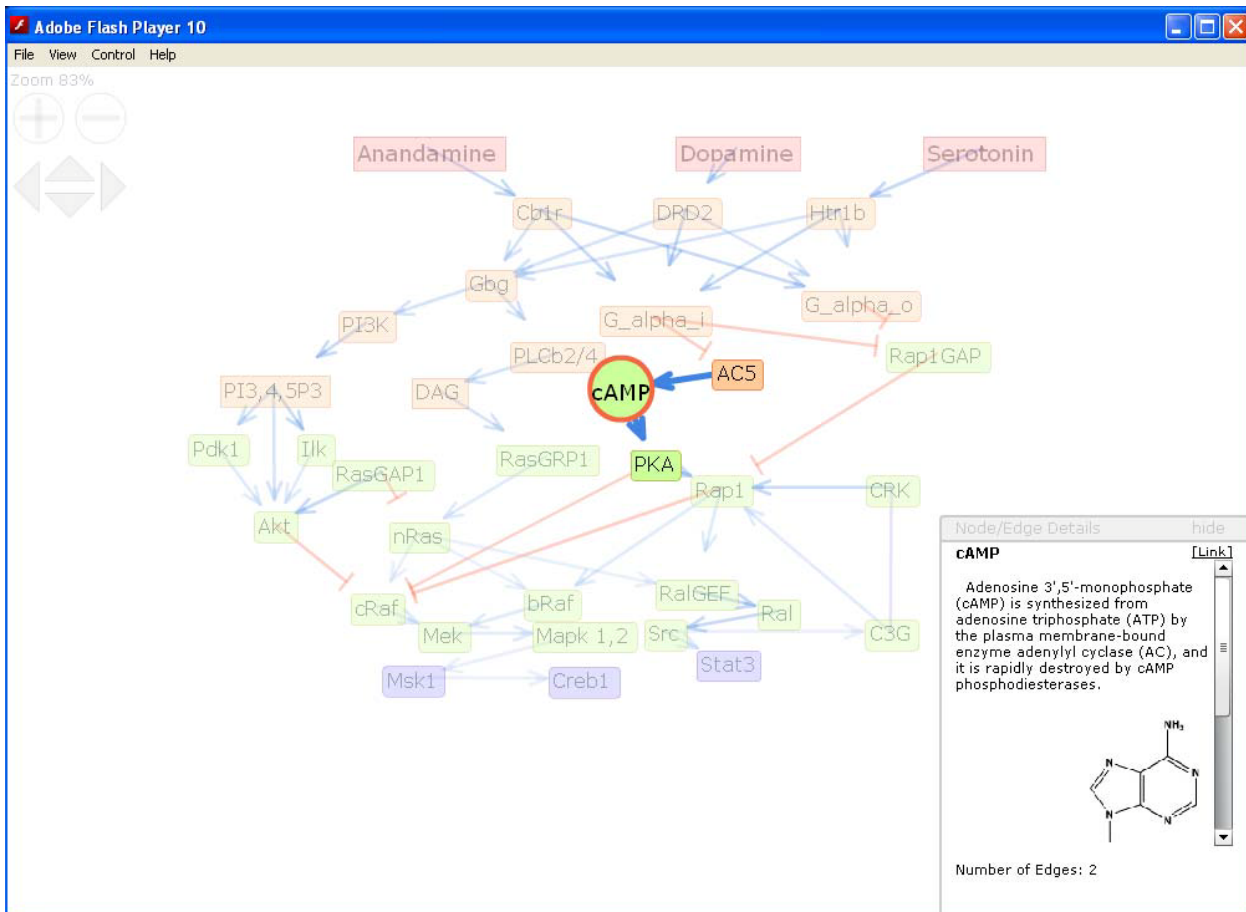


The network diagram can be dragged around by clicking and dragging the background. Alternatively the network diagram can be moved around the window by clicking the arrows located in the gray bar in on the left hand side (A). Clicking the plus and minus buttons located above the arrows allow for zooming in and out (B). Text above these buttons displays the current amount of zoom (C). Lastly, the print button on the bar allows the current network display to be printed (D). You can use the print option to export the network image to a PDF by using the PDF printer.

Individual nodes can be moved around while still maintaining the edges in the network. Simply click a node and drag it.



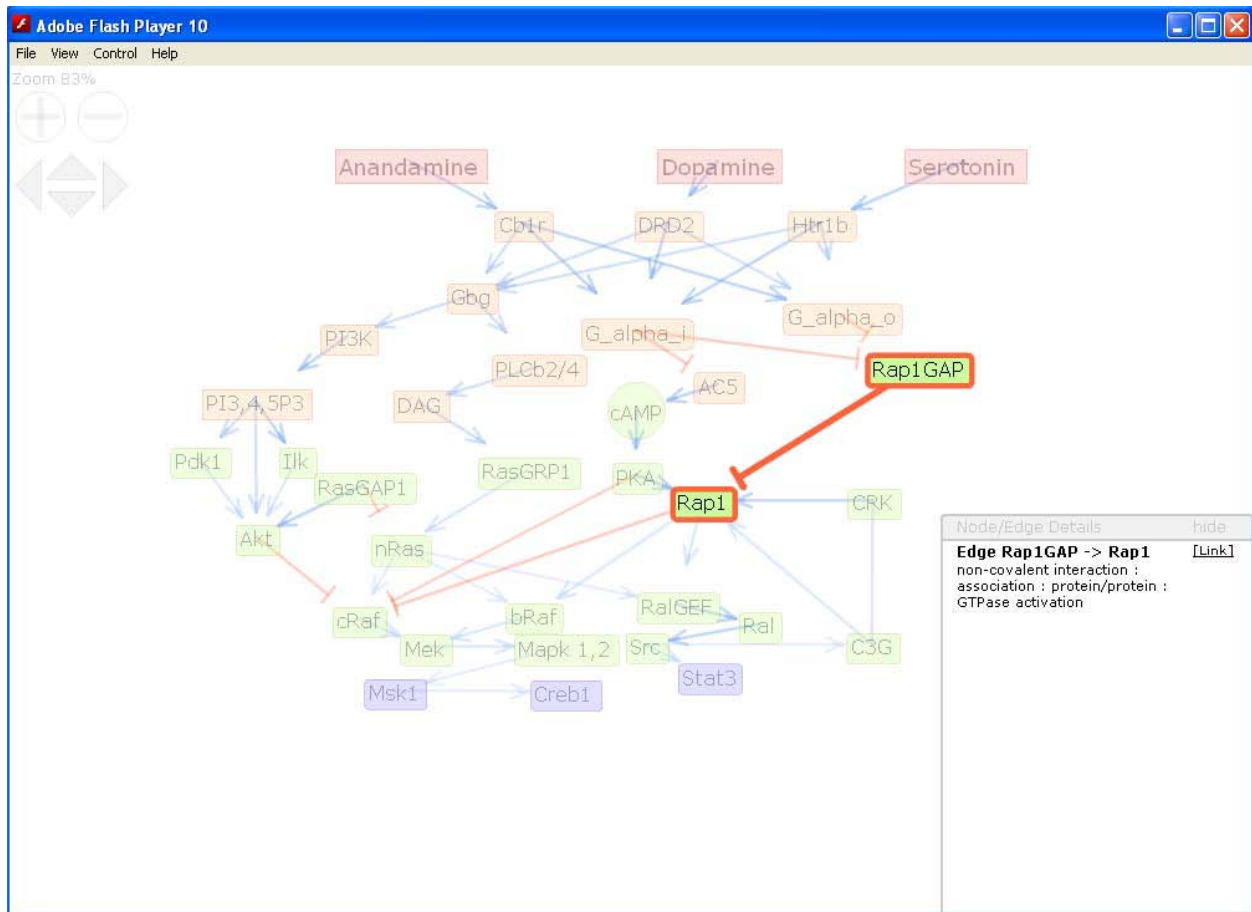
When subnetwork fading and the description box options are enabled in the xml, placing the cursor over a specific node highlights the node's edges and displays text in the description box for the node as defined in the xml file. To minimize the description box click the hide button (A) click it again to maximize the box.



XML enclosed within the <node> element that generated the above description:
 <desc>Adenosine 3',5'-monophosphate (cAMP) is synthesized from adenosine triphosphate (ATP) by the plasma membrane-bound enzyme adenylyl cyclase (AC), and it is rapidly destroyed by cAMP phosphodiesterases.

</desc>

Pictures can be displayed in this box in addition to text. However, the image must be less than 200 pixels wide to be displayed fully. Html
 tags can be used if the image height is being cut off. Other html tags (e.g.) can be added to format the text within the description box. Relevant hyperlinks can be included with the node and are presented as a link in the upper right corner of the box.



Similarly, placing the cursor over an edge shows a description in the box. Much like the node descriptions, the edge descriptions can have text, links, and images, associated with them.

Summary

The viewer (FNV) we developed allows for the dynamic representation of biological and molecular networks in a more simplistic and versatile manner than existing pathway viewers. Written in ActionScript 3.0, the viewer is delivered as a Flash .swf file that reads in an XML file. Layouts can be predefined through pixel mapping, or can be achieved using a variety of layout algorithms. Node colors and shapes can be used to designate component localization and entity information. Edges can be directed or undirected and can be used to define the effect of the interaction as it can be activation, inhibition, or neutral. Most importantly, molecule and interaction details can be accessed by simply hovering over nodes and edges, nodes can be made linkable, and can be moved around by the user.

Contact

For questions and comments about FNV email avi.maayan@mssm.edu.

Appendix

Network visualization tools table

Name	Description	URL	PMID
Bioconductor	Provides access to wide range of graphical methods of analyzing genomic data	http://www.bioconductor.org	15297301
BioLayout	Versatile Network Visualization of Structural and Functional Relationships	http://www.biobioinformatics.org	16000016
BiologicalNetworks/PathSys	A graph-based system for creating a combined database of biological pathways, gene regulatory networks and protein interaction maps	http://biologicalnetworks.net/PathSys/	16845051 & 16464251
BioMiner	Modeling, Analyzing, and Visualizing Biochemical Pathways and Networks	http://www.zbi.uni-saarland.de/chair/projects/BioMiner/index.shtml	12386006
BioPathAtMAPS	Visualization of pathways of Arabidopsis Thaliana	http://ibc.wsu.edu/research/lange/pdf/BioPathAt_Mar2005.pdf	15694451
BioPixie	System for biological data integration and visualization	http://pixie.princeton.edu/pixie/	16420673
BioPP	Web application that creates publishable networks	http://tsb.mssm.edu/pathwayPublisher/broadcast/	17519033
BioUML	Java framework for systems biology	http://www.biouml.org/	
CAIDA	Cooperative Association for Internet Data Analysis	http://www.caida.org/tools/	
CellDesigner	Java desktop tool for modeling biochem networks	http://www.celldesigner.org/	
Cytoscape & CytoscapeWeb	Visualizing molecular interactions	http://www.cytoscape.org/ http://cytoscapeweb.cytoscape.org	14597658
EPV	Java-based software enabling faster visualization and graph drawing with large number of nodes	http://www.cs.ucsb.edu/~tcan/fpv/	12761052
GeneInfoViz	Retrieves and constructs visualization of gene relation networks	http://www.genenet.org/geneinfoviz/	15724283
GenMAPP	Gene Map Annotator and Pathway Profiler	http://www.genmapp.org/	1984561
GINY	Graph INterface librarY	http://csbi.sourceforge.net/	
GnuPlot	Command-line interactive and function plotting utility	http://www.gnuplot.info	
GraphLet	GraphLet	http://infosun.fmi.uni-passau.de/Graphlet/index.html	
GraphViz	Open source graph drawing software	http://www.research.att.com/sw/tools/graphviz/	
GVF	The Graph Visualization Framework	http://gvf.sourceforge.net/	
Hippron	Dynamic signaling maps	http://www.hippron.com/hippron/index.html	
HyperGraph	Open source java code for hyperbolic geometry and trees	http://hypergraph.sourceforge.net/	
JDesigner	Draw and manage SBML format pathways	http://members.lycos.co.uk/sauro/biotech.htm	

Jdigraph	Java library for directed graphs	https://jdigraph.dev.java.net/	
JGraphEd	Create graphs by adding/removing/modifying nodes or edges	http://www.jharris.ca/JGraphEd/	
JGraphT	Java graph-theory objects and algorithms	http://jgrapht.sourceforge.net/	
jSquid	Java applet that expands functionality of Medusa	http://jsquid.sbc.su.se/	18445606
JUNG	Java Universal Network/Graph Framework	http://jung.sourceforge.net/	
MAPPFinder	Gene relationships explored	http://www.GenMAPP.org	12540299
MAVisto	A tool for the exploration of motifs in biological networks	http://mavisto.ipk-gatersleben.de/	16020473
mDraw	Interactive network visualization tool	http://www.weizmann.ac.il/mcb/UriAlon/	
Medusa	A general graph visualization tool (Standalone and web-based versions)	http://coot.embl.de/medusa/index.shtml	16188923
MNV	Metabolic Networks Visualizer	http://www.patronov.net/sciencevr/mnv/indexmake.html	14668228
NetBuilder	An interactive standalone tool for representing and simulating genetic regulatory networks	http://strc.herts.ac.uk/bio/maria/NetBuilder/	
NetViz	Dynamic Visualization of Social Networks	http://www.netvis.org/	
Ondex	Data integration and visualization	http://www.ondex.org	16533819
Osprey	Visualization of complex interaction networks	http://biodata.mshri.on.ca/osprey/servlet/Index	12620107
Pajek	Program for large network analysis	http://vlado.fmf.uni-lj.si/pub/networks/pajek/	
PathFinder	Tool for the dynamic visualization of metabolic pathways based on annotation data	http://bibiserv.techfak.uni-bielefeld.de/pathfinder/	11836220
PathwayStudio (Formally Pathway Assist)	Commercial tool that allows for visualization and querying of a network	http://www.iobion.com/products/products_IOBIONLAB_PA.html	14594725
PATIKA	2D representations of single or directed graphs	http://www.patika.org	16287939 & 12117798
PaVESy	Pathway Visualization and Editing System	http://pavesy.mpimp-golm.mpg.de	15105280
Piccolo	Infrastructure for Java and C# that allows for easy graph visualization	http://www.cs.umd.edu/hcil/piccolo/	
PIVOT	Java based tool for visualizing protein-protein interaction	http://acgt.cs.tau.ac.il/pivot/	14960471
Prefuse	interactive information visualization	http://prefuse.sourceforge.net/	
ProViz	Protein-protein interaction graphs visualization tool	http://cbi.labri.fr/eng/proviz.htm	15347570
SocNetV	A standalone social networks visualization tool	http://socnetv.sourceforge.net/index.html	
Tom Sawyer Software	Graph analysis, layout, and visualization solutions	http://www.tomsawyer.com/	
TouchGraph	Graph Visualization using spring-layout	http://touchgraph.sourceforge.net/	

Tulip	An information visualization framework dedicated to the analysis and visualization of relational data	http://tulip.labri.fr	
UCINET	Software for Social Network Analysis	http://www.analytictech.com/ucinet_5_description.htm	
VGJ	Web-based Graph Drawing Tool	http://www.eng.auburn.edu/csse/research/graph_drawing/vgj.html	
VisANT	web-based software framework for visualizing and analyzing networks	http://visant.bu.edu/	15980487
VitaPad	Visualization tools for analysis of pathway data	http://bioinformatics.med.yale.edu	15564306
VUE: Visual Understanding Environment	A concept and content mapping application	http://vue.tufts.edu/	
Walrus	Tool for interactively visualizing large directed graphs in 3D	http://www.caida.org/tools/visualization/walrus/	
WebInterViewer	Visualizing molecular interactions networks	http://interviewer.inha.ac.kr	15215357
WilmaScope	Java3D application creates 3d animations of graphs	http://www.wilmascope.org/	
WinGraphViz	Windows COM based on GraphViz	http://wingraphviz.sourceforge.net/wingraphviz	
XmdvTool	Interactive exploration of multivariate datasets	http://davis.wpi.edu/~xmdv/	
yWorks	Diagramming company with several graph drawing products	http://www.yworks.com/en/products.htm	
Zoomgraph (now known as GUESS)	Visualizing Zoomable Data Driven Graphs	http://www.hpl.hp.com/research/idl/projects/graphs/	