AVIS

Web-based Tool for Visualizing Biological Networks

User Manual
Version 2.1

Seth Berger, Ravi Iyengar, Avi Ma'ayan

Department of Pharmacology and Systems Therapeutics
Mount Sinai School of Medicine, New York, NY

Table of Contents

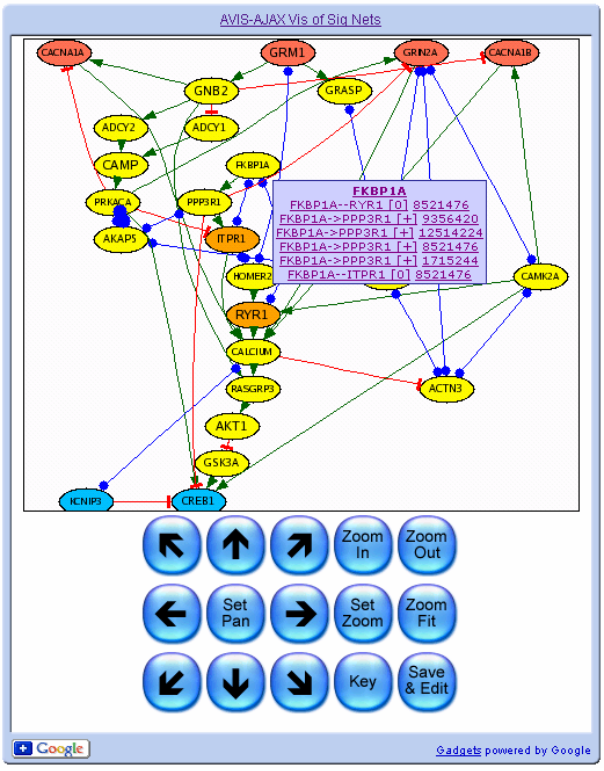
Introduction

AVIS is a visualization tool for viewing and sharing intracellular signaling, gene regulation and protein interaction networks. AVIS is implemented as an AJAX enabled syndicated Google gadget. It allows any webpage to render an image from a text file representation of signaling, gene regulatory, or protein interaction networks. Network diagram images can be zoomed and panned within webpages. As the view of the image changes, the view is re-rendered to ensure that all edges in the view are visible and all nodes in the view have readable labels. Labels with links can be displayed as users move the mouse over nodes.

Answers to questions and support can be provided by contacting Seth Berger at Seth.Berger@mssm.edu

Using AVIS

AVIS is designed to be embedded in any webpage to allow users to dynamically view cellular interaction networks without the webpage administrator having the need to have any programming knowledge.



End User

The end-user experience is designed to be light-weight and straightforward. A control panel is displayed beneath the image of the network. The control panel contains controls to zoom-in, zoom-out, and pan the image. The “Set Zoom” and “Set Pan” controls allow users to

jump directly to a view of choice. For example, the select node option in the Set Pan menu allows the view to be quickly centered on any node. The values entered for the "Pan X" and "Pan Y" are percents of the total span. For instance, setting both these values to 50% will center the view in the middle of the image as it will be set to 50% of the maximum pan. The "Save/Edit" button allow users to download the image of the current view, the entire network map, or edit a textual representation of the network in AVIS format (see section 3a). When the text is modified and the "submit" link at the top right corner is pressed, the network is re-rendered with the changes.

Web-site Administrator

Gadget Settings

Administrators of the webpages containing AVIS should select the methods and settings for running AVIS that best suits their needs. There are ten input settings that control the embedding of an AVIS object in a webpage:

1. "Network File" specifies the URL of a text file that AVIS loads. For example, specify http://actin.pharm.mssm.edu/AVIS2/sample.avis to load this sample.avis structured text file representing a cell signaling network.
2. "Initial Zoom" specifies the initial zoom when AVIS loads. This should be an integer greater than or equal to 100.
3. "Initial X – Pan" specifies the initial x-pan zoom when AVIS loads. This should be a number between 0 and 100.
4. "Initial Y – Pan" specifies the initial y-pan zoom when AVIS loads. This should be a number between 0 and 100.
5. Start in Key Mode specifies whether the viewer should load in key mode or standard mode. This allows implementers of avis to have multiple viewers of the same file initialized in different modes. Thus one can create a pop up window that displays the graph's key for side by side key and graph viewing.
6. "ImageView Scale" specifies what size view area to re-render when panning. This parameter should be adjusted for a trade-off between the cost of re-rendering, and sending large image files. Sending large images allows the ability to pan smoothly without the need to re-render the image. The "image view scale" parameter should be a number greater than or equal to 1. A value of 1 means that the image will be re-rendered every time a user tries to pan. Values greater than 1 will make the viewer render an image a multiple of its width and height and thus panning will be smooth over a larger range.
7. AVIS's "Graph Settings File" is a URL to an AVIS readable file (see section 3a). This field is optional and if left blank or set to default, appropriate guesses will be made. This file is appended to the AVIS file generated by loading any other network file. The file allows users to specify settings such as colors of node types, node sizes, and mapping of URL parameters.
8. "FileLoader" specifies which fileloader AVIS will use to parse the network text file into the AVIS format. This can be specified as default or the name of a Perl wrapped file loader of your choice.
    a. "Default Loader" can import the supported file types (see section 3.a) except SBML. SBML files can be loaded by specifying "loadSBML" as

the loader. It then uses a Graphviz layout algorithm to layout the nodes if a layout is provided from the SBML file. The options for the layout are specified by "Settings for Layout/Loader" (see section 2.b.i.8). Options can include the name of any of the Graphviz layout algorithms (see www.graphviz.org) such as dot, neato, fdp, circo, or twopi, and command line switches (see http://graphviz.org/doc/info/command.html) that adjust the algorithm settings.
For example, setting the layout settings to "dot -Grankdir=LR" will cause the graph to be displayed left to right as opposed to the default top to bottom.
Another example is setting the layout to "twopi". This will cause networks to be displayed with a radial layout.

b. A simple Perl wrapper can be written to allow any other file import or layout tools. (see 3.b)

9. Settings for "Layout/Loader" specify a settings-string that is passed to the file loader. This can be used to specify layout options or file import options depending on the file loader used. (see 2.b.i.7)
10. Graph Render Engine specifies the graph drawing tool AVIS will call to draw the graph. AVIS currently included 2 graph drawing engines. By default, AVIS will use graphviz to draw the image as this is faster and supports more options. Alternatively, one can use a simpler AVISGraphDraw engine by specifying 'AVIS' as the render engine. Finally, any other graph rendering engine can be wrapped in a Perl script to allow AVIS to use it.
    a. Default/graphviz: This rendering engine uses graphviz to draw the graph and calculate the locations of mouse events. Mouseover events are found by node bounding boxes.
    b. AVISGraphDraw is a simple rendering engine implemented in Perl with the PerlMagick library. It supports drawing nodes as circles and most of the AVIS drawing features. Mouseover events are resticted to circles around nodes.
    c. A simple Perl wrapper can be written to allow any graph layout engine to work with AVIS (see section 5).

Configuring AVIS

There are three main ways to configure AVIS. AVIS can be configured as a Google gadget, a syndicated Google gadget, or within an HTML iframe.

1. To use AVIS on your Google search page, log into your Google search page, and select "Add stuff". Then, select "add by URL", and enter the URL to your AVIS.xml file; for example you can use the xml file hosted on our server at http://actin.pharm.mssm.edu/AVIS2/AVIS.xml

2. To embed AVIS as a Google gadget in any other web page, use the Google "add gadget to your webpage" tool. This can be found at
http://gmodules.com/ig/creator?synd=open&url=PATH_TO_AVIS.xml
For example, go to use the gadget we provide use the following link:
http://gmodules.com/ig/creator?synd=open&url=http://actin.pharm.mssm.edu/AV

IS2/AVIS.xml .After adjusting the settings, use the “Get the Code” button to get the HTML code for embedding AVIS in your page.

3. To embed AVIS in any other webpage you can use the iframe tag. First, create an iframe tag and set the height and width. The src attribute should point to an AVIS.php file with a series of get parameters for each of the settings: up_sigfile is the network file, up_zoom is the initial zoom, up_xpos and up_ypose are the initial pans, up_sik is either 1 or 0 for start in key mode, up_ivs is the imageviewscale, up_configFile is the URL of the preferences file, up_loader is the file loader, up_loaderprefs is the loader preferences, and up_renderer is the rendering engine. It is recommended for compatability with safari browsers that the iframe src is loaded dynamically during the body onload event.
For example, the following iframe includes the default settings of AVIS:

```
<iframe ID="theIframe" src="" style="display: block;" frameborder="0"
height="600" width="320"></iframe>

if the following javascript is placed in the BODY onload event of the page.

document.getElementById("theIframe").src="http://actin.pharm.mssm.edu/AVIS2/A
VIS.php?up_sigfile=http://actin.pharm.mssm.edu/AVIS2/sample.avis&up_zoom=100&
up_xpos=50&up_ypos=50&up_sik=0&up_ivs=1&up_configFile=default&up_loader=defau
lt&up_loaderprefs=dot&up_renderer=default";
```

File Formats

AVIS recognizes multiple input file formats distinguished by file extension.

AVIS Format

The AVIS file format allow users complete control to all of AVIS graph drawing features in a text based format. This format is designed to be parsed easily by the AVIS engine, while still being readable enough that small modifications can be added manually to customize visualization.

The file is divided into five main sections: *AVIS*, *edgetypedefs*, *nodetypedefs*, *nodes*, and *edges*. These sections can appear at any order and their presence is not required. Each section contains lines with the name of a property followed by a colon and its value. Each section, except for the *AVIS* section can be divided into blocks representing one entry in the section. A block begins with the definition of an ID property. All following properties, until the next ID property, are assigned to that block. Properties not understood by AVIS can be included to provide additional information about a block.

When drawing an object, AVIS first checks if the object’s properties are defined in the AVIS file. If no property is defined, AVIS checks if the section has a default block definition in the AVIS file. If no default block definition was found, AVIS checks the “type” property of the object, and looks up the property in the appropriate type definition

section in the AVIS file, resorting to the built-in default if necessary. If the property is not specified, AVIS selects an appropriate default value.

See table 1 for a list of properties and default built-in values.

Properties can be mapped to other properties automatically by using \AVIS_property\ notation. For example, if a node has a property called 'firstname' and another property called 'lastname', you can define the label of nodes of this type to be \AVIS_firstname\ \AVIS_lastname\. If a replacement can not be made, AVIS will return NA for the entire result. For example, the in the *nodetypedef* section, one can set the type for the default block to be \AVIS_compartment\, which will set the type for nodes to be mapped to their compartment property. Similarly, one can define the label for an edge to include the source and target name. For example, in the *edgetypedef* section, label can be set to \AVIS_source\ -- \AVIS_target\. Any user defined property can be stored in the AVIS format and can be mapped to other properties.

AVIS responds with mouseover events when the user moves the mouse over nodes when interacting with the viewer. The text that shows up contains the label of the node which is a link to the URL associated with the node. This is followed by a list of the labels representing the edges that connect to mouseover selected node with other nodes. Clicking on an edge label recenters the display to focus on the selected neighboring node involved in making up the edge. Each edge label is repeated for each of the reference properties it has. For example, it is convenient to use map PubMed ID (PMID) to references associated with edges to map a URL as a link to a PMID reference.

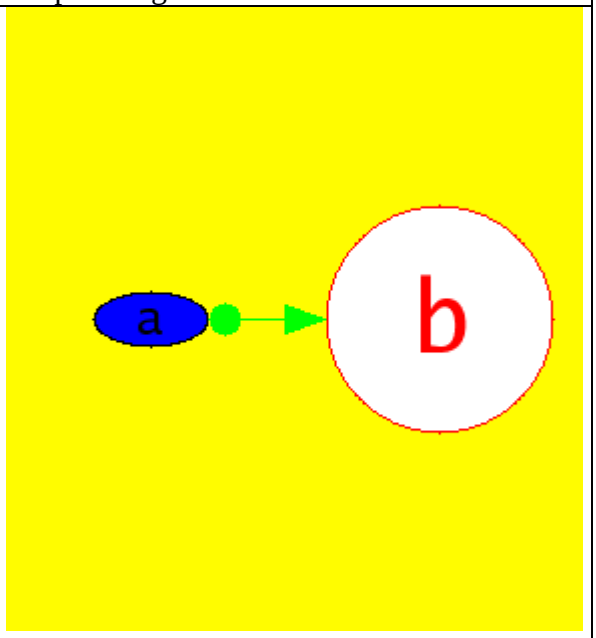
Several examples are listed below:

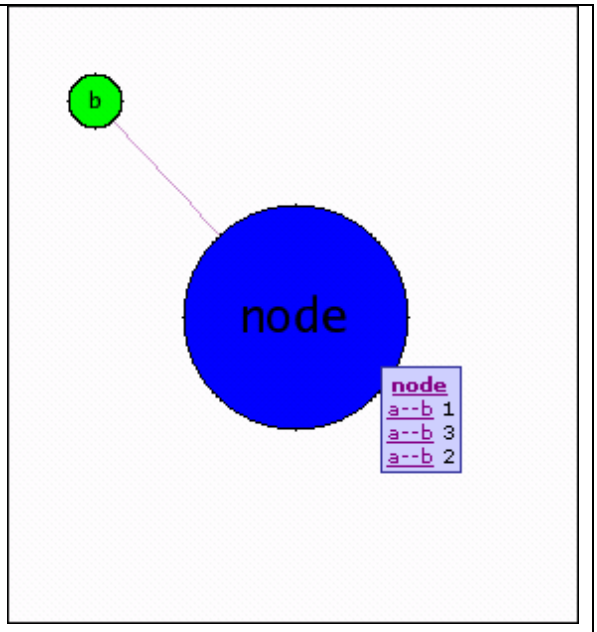
Table 1: List of AVIS format properties.

Property	**Section**	**Description**	**Default**
noderadius	AVIS	The base radius of all nodes in the graph as a percent of the width of the shorter dimension of the graph. (e.g. if set to 50, one node would fit in on the graph)	3
backgroundcolor	AVIS	Background of the graph	White
Label	nodes, edges	Text describing a node or Edge	\AVIS_id\
Id	nodes, edges	**Required field.** Unique identifier for edge, node, or typedef	
Type	nodes, edges	Defines the type of a node or edge.	default
fontcolor	nodes	Color of the font of a node label.	\AVIS_bordercolor\
X	nodes	X position of node (in percent units) (e.g. a value of 50 is centered)	0
Y	nodes	Y position of node (in percent units) (e.g. a value of 50 is centered)	0
reference	edges	Accession number for reference for an edge. Can have several definitions for multiple references.	NA

url	nodes, edges	Specifies the URL for an edge or node. For edges, this can contain replacements for multiple \AVIS_references\	NA
head	edges	Specifies target end of an edge, can be: ball, arrow, or tee	none
tail	edges	Specifies source end of an edge, can be: ball, arrow, or tee	None
scale, xscale, yscale	nodes	Multiple size of the base node radius. A scale of 2 draws a node twice as large as the base size. (xscale and yscale on strech in the x or y dimension)	1
headsize, tailsize	edges	Multiple size of the head or tail. A headsize of 2 draws a head twice as large as the base size.	1
thickness	nodes, edges	Width of edge stroke or node border.	1
color	nodes, edges	Edge color or node fill color	Edge: Black, Node:White
bordercolor	nodes	Edge border color	Black
Path	edges	Space separated list of coordinates that an edge goes though in percent coordinates. x1,y1 x2,y2 x3,y3	none
source, target	edges	Source and Target node of the edge	
Shape	node	Shape of the node	

AVIS File format examples:

Description	AVIS File	Output image
Two nodes with one edge between them.	*AVIS* noderadius:10 backgroundcolor:yellow *nodes* id:a x:25 y:50 yscale:.5 color:blue id:b x:75 y:50 bordercolor:red scale:2 *edges* id:edge1 source:a target: b head:arrow headsize:2 tail:ball color:green	

2 nodes, with edge to self. Types are mapped to user defined properties.	*AVIS* noderadius:20 *nodetypedef* id:default type:\AVIS_compartment\ color:green id:cytoplasm color:blue *nodes* id:a x:50 y:50 url:http://www.pubmed.com compartment:cytoplasm label:node id:b x:15 y:15 scale:.25 *edgetypedef* id:blah color:plum3 *edges* id:edge1 source:a target:b type:blah label:\AVIS_source\--\AVIS_target\ reference:1 reference:2 reference:3	

SIG File Format

SIG file format is another structured flat file format we developed. It consists of 13 space-separated columns. The first 5 columns define the source node. The next 5 columns define the target node. The last three define properties of the edge. Any of the fields can be replaced with NA if the information is not available. The five fields of a node definition, in order, are its Name, its Human Swiss-Prot ID, its Mouse Swiss-Prot ID, its function, and its sub-cellular compartment. The last three nodes for each edge is a sign for the effect, the class of interaction, and a PMID. The default AVIS file loader can recognize these.

The following line from a SIG file describes a positive interaction between CALCIUM and RASGRP3:

CALCIUM NA NA Messenger Cytosol RASGRP3 Q8IV61 NA GAP Cytosol + Binding 9582122

Pajek's NET File Format

Net files are used by Pajek (see http://vlado.fmf.uni-lj.si/pub/networks/pajek/). The default AVIS file loader can recognize these and maintains the layout described in the file.

SBML File Format

SBML files are described at www.sbml.org. They can be loaded into AVIS by specifying loadSBML as the loader. This does not currently support the draft SBML layout specification and performs a layout similar to the default AVIS file loader. Enzymes, small molecule, as well as reactions are represented as nodes with edges connecting reactants, products, and modifiers to their relevant reactions.

Writing a file import plug-in

Any file that can be translated to the AVIS format can be loaded by a script. The Perl script must define a subroutine called loadfile which takes as parameters the filename, text of the file, viewer width and height, URL of an AVIS options file, and a string of loader options. The script must then return a valid AVIS format file. This script can be placed in the cgi-bin/AVIS directory. If the name of the script is, for example scriptName.pl, it will be used if scriptName is entered as the name of the file loader.

Running / Installing your own AVIS server

If you do not wish to rely on our servers for hosting AVIS, and you are expecting to be generating a lot of traffic, or wish to modify AVIS to suit your needs, the source code is freely available for download and you can run your own AVIS Server.

You can install the server components of AVIS onto any webserver capable of hosting perl and PHP scripts. Instructions can be found below.

Installing AVIS on your server

The AVIS distribution List of Files included in the distribution can be found in includedFiles.txt

To prepare it for your server follow these steps:

i. Download the AVIS.zip file at: http://actin.pharm.mssm.edu/AVIS2/AVIS.zip

ii. Extract the zip file.

iii. Download and extract overlib (http://www.bosrup.com/web/overlib/) to the overlib folder in the AVIS-html directory.

iv. Make sure all the dependencies for AVIS are installed. These include:
Graphviz (www.graphviz.org)
ImageMagick and perlMagick
Several Perl modules available from cpan (www.cpan.org) such as:
LWP::UserAgent
URI:Escape
IPC::Run

Cache::FileCache
CGI
MD5

These can be installed following the instructions from the providing website, using the CPAN install perl module (perl -MCPAN -e "CPAN::install ModuleName"), or by installing the proper packages for your server. A shell script which automatically installs the required dependencies for a ubuntu linux 7.04 LAMP server is included (ubuntu_dependencies.sh). This will use aptitude to install graphviz, libmagick9, imagemagick, perlmagick, libwww-perl, libipc-run-perl, libmd5-perl, libCGI-perl, liburi-perl, libcache-perl, and libcache-cache-perl.

v. If you want SBML support, download and install libSBML 2.3.4 from http://sbml.org/software/libsbml/ Make sure you follow the instructions in both the main README and the README in the src/bindings/perl directory so that the perl bindings are installed into the perl include library path allowing perl programs to "use LibSBML;". You will have to install all dependencies for libSBML such as expat as well the tools for compiling the libSBML library.

vi. An installation script (configAvis.pl) has been included to automate the rest of the installation. Running this script will ask you several questions, check that the dependencies of AVIS are met, warning you which ones are lacking, then generate the AVIS configuration files, and copy the AVIS files to the web server's hostable directory. This script must be run as a user with sufficient priveledges to write files in the target directories. It can be run with the following command on most systems.
sudo perl configAvis.pl

This should complete your AVIS installation, alternatively one can manually install AVIS.

**** Manual Configuration / Installation ****

vii. Set linux file permissions on all .cgi files and .pl files to be executable and ensure all other files are readable.

viii. Edit AVISconfig.php in the AVIS-html directory. This sets 4 variables to the URLs on your server for the AVIS.cgi file, images directory, Avis manual file, and the sample.avis file. These are used by AVIS to find the files from the client side.

ix. Edit demoConfig.php in the AVIS-html directory. This sets 5 variables which point provide the relative or full URL path to the AVIS manual, relative or full URL path to the upload script, the full URL path to the AVIS.xml file, the full URL path to the AVIS.php file, and the relative

or full path to the node neighborhood script. These values are used by the index.php page and the index-gadget.php page.

x. Edit avisPrefs.txt in the AVIS-cgi directory. This file contains 1 line which should be the full URL to the AVIS.php file. This is used by the node neighborhood and upload scripts.

xi. Edit the AVIS.xml file in the AVIS-html directory, changing the second to last line (<Content type="url"...) to point to the AVIS.php file on your server.
This is the only reference to other URLs in this file. This tells the gmodules.com server to load and iframe pointing to your AVIS.php file when a page embeds AVIS as a google gadget.

xii. If you wish to maintain a log of AVIS activity, create a blank file where you want to save the log (for example, /var/log/avislog.txt). Next edit the avisLogPath.txt file in the AVIS-cgi directory so that it has only one line which is the path to the blank log file. Make sure the permissions of the log file allow it to be writable.

xiii.Copy the folder AVIS-cgi to your cgi-bin directory

xiv. Copy the folder AVIS-html to your html directory

xv: You can view your demo page now by viewing the index.php in the AVIS-html directory on your webserver.

::Troubleshooting Installation::

To run AVIS in gadget mode, the computer hosting the AVIS.xml file must not be behind a firewall preventing sites from accessing it as it must be visible to the gmodules.com server. This server downloads the xml file to generate the gadget interface. If the gmodule.com google gadget server can not see the XML file, the gadget will display "Information Temorarily Unavailable".

The robots.txt file on your server must allow permissions for automated crawlers to read the xml file or else the gadget server will not parse it. If you change the robot.txt file after its been cached by the gmodules.com server, it may take some time before the gmodules.com server recaches the file and allows downloading of the AVIS.xml file.

Verify that AVIS has all its dependencies installed by running the configAvis.pl script from the command line again. This will announce any unsatisfied dependencies.

You can purge the AVIS file cache ahead of the scheduled purge time by deleting the /tmp/FileCache folder.

If you are having any other issues, please contact seth.berger@mssm.edu.

Writing a graph render plugin

Alternative graph rendering tools can be used with AVIS. This requires a Perl wrapper that defines 2 subroutines: drawGraph takes a reference to an 'AVIS hash', the canvas x dimension, the canvas y dimension, the zoom percent, the pan x percent and the pan y percent. This must then return and imageMagick image with the canvas dimensions.
The second function that must be defined is loadBoxInfo which is called with an AVISHashRef, the canvas X dimension, and canvas Y dimension. This returns a string that contains valid javascript code which loads 2 arrays with values. One array is nodeLoc and is a list of structs where each element is {x: value, y: value, r: value, lb:value} where x and y specify the center of a node and lb specifies the node label. The other array is rectArray with contains a struct of {x1: value, y1: value, x2: value, y2:value, lb: value} where x1, y1 is the upper left right corner of the node and x2,y2 is the lower right corner. lb contains the HTML code to display when there is a mouseover event.

The 'AVIS hash' passed to these function contains all the info of an AVIS file so that $AVISHash{"section"}{"id"}{property}{value} = 1; The list of values for a given property can be found by calling &getPropertyValue($AVISHashRef, section, id, property);

Examples of AVIS use in other applications

Uploading local client machine files

We provide an example of how AVIS can be incorporated into a small script to allow users to upload files to a server and visualize these files using AVIS. The source of this script, upload2Avis, is included in the AVIS-cgi directory. This script allows you to browse your computer and upload a file temporarily to the server to visualize it using an AVIS gadget. You can use the script to upload files to visualize files by going to http://actin.pharm.mssm.edu/cgi-bin/AVIS2/upload2AVIS.cgi or the equivalent path for your installation.

Visualization of the Node Neighborhoods in databases

An example of using AVIS for visualizing large-scale interaction databases is provided. Three protein-protein interaction databases can be viewed by specifying a node and how many steps away from the node at http://actin.pharm.mssm.edu/cgi-bin/AVIS2/nodeNeighborhood.cgi or at the equivalent path in the installation package. The sig format data files are located in the sigs directory of the AVIS cgi folder.