

Flash GViewer

A customizable web tool for visualizing genome-wide annotation

See also [Flash GViewer](#).

Simon Twigger, Medical College of Wisconsin © 2005 Last Modified: February 16, 2006

The Rat Genome Database has an SVG-based genome viewer that shows the location of genes and QTLs against a backdrop of the rat cytogenetic map. For our proteomics projects we were interested in viewing the genomic locations of the proteins identified in a proteomics experiment and a GViewer-like display is well suited to this. To make a more stand-alone solution I rewrote much of the functionality into a Flash application that is more adaptable to other organisms and other types of data. It requires Macromedia Flash Plugin Version 7, you can download the plugin [here](#).

Current Features (v0.5)

Here are some of the Flash GViewer's current features:

- Uses a base map containing the chromosomes and banding pattern of the desired organism. Perl scripts are available to take the UCSC cytogenetic map files and convert them to the appropriate XML format. This can also be used to portray any other custom backdrop for a chromosome - colored segments corresponding to syntenic regions in other organisms, colored according to feature density, etc.
- Bands can link out to a URL defined in the base map (to allow users to click to view a region in a genome browser, for example).
- Annotations can come from a static XML file or can be dynamically generated by a server-side script
- **New in v0.5.** Annotations can be colored independently, allowing further information to be conveyed (eg. red/green/yellow for expression levels, peptide coverage or Xcorr for proteomic data).
- Features appear at the appropriate position on the chromosome and then dynamically move out to form a non-overlapping display.
- Placing the mouse over individual features brings up a tooltip displaying the feature's name, it also visually highlights the region of the chromosome that the feature is aligned with
- Clicking on a feature toggles the highlight display so it will stay on when you move the mouse away. This can be used to turn on the region highlights for multiple features to better visualize their overlapping regions. Click again to turn each feature's highlight off.
- Clicking on a feature can open up a new browser window with a customizable URL, eg a gene report page.
- Clicking on a chromosome will expand that chromosome and enlarge and label the features making them easier to view. Click the chromosome again to return the whole genome view.
- Turn on/off labels for zoom view and genome view (press i button in bottom left corner). Click redraw to refresh the view to show/hide labels.
- Javascript can be used to dynamically highlight displayed annotations by using triggering javascript functions in the web page containing the flash movie. Javascript can be used to query annotation data from the flash movie and handle the returned data.
- Customizable web interface and perl CGI script provided to allow users to upload annotations and display them on the GViewer against the base map of their choice.

- Region selection sliders in Zoom View allow the selection of a genomic region so you can then click to open up a genome browser window for the specified region.

Changes in v0.5

- Added the ability to color annotation features independently via addition of color tag to annotation XML format. Annotations without separate color information will revert to the default color for that particular feature type.

Changes in v0.4

- Improved Javascript functionality - added feedback from Flash to Javascript to allow Javascript to query Flash for the annotation data and have Flash pass this data back to a Javascript function for display or other use.
- Added ability to add a title to the GViewer movie.
- Reset zoom view to provide 2x zoom of chromosome and annotation.
- Added Region selection sliders in Zoom View to allow the selection of a genomic region. New configuration parameter to set external URL for browser link.

Changes in v0.3

- Added basic javascript to flash connectivity to allow javascript in a web page to control the feature highlighting in the flash movie

Changes in v0.2

- Bands can link out to a URL defined in the base map (to allow users to click to view a region in a genome browser, for example).
- Changed click/shift click of features and bands. Clicking on a feature turns highlight on/off, shift-clicking opens external link. Clicking on a band changes to zoomed view or back to genome view, shift clicking opens external link. This was altered because it was too easy to accidentally click on a feature and open a new browser window by accident.

Here's an example of the GViewer showing the rat genome with some of the rat genes and QTLs associated with blood pressure:

(Dont see anything? You need Flash Plugin version 7 for this to work. How can you gracefully handle browsers that dont have the plugin or an earlier version - I have had good success with Colin Moock's [FlashPlayerInspector](#). His page has more info on how this works (and when it doesnt), plus you can download the appropriate code (javascript) to test for the plugin and provide appropriate alternative text/content as required).

Demos

Flash/Javascript interaction MouseOver Demos

There is a real live demo at [flashGViewer doc page on the GMOD blog site](#).

Installation

Flash GViewer is available at [Sourceforge](#) and all source code is available via the [CVS repository](#).

Basic installation is very simple, copy these files/directories to a directory on your webserver:

- GViewer directory and contents
- sample.html

Load up sample.html in a browser and you should see a working version of the GViewer. The sample.html is a very basic HTML page that contains the tags that display the same version of GViewer as you see above. The data directory contains the sample rat annotation file and the rat, human and mouse baseMaps. You can change the baseMapURL to point to one of the other organisms and see what their maps look like. Not a great fan of the background color - change the bgcolor parameter to something more appropriate, see the Input Parameters section below. If everything is working as expected, you can modify the HTML page or cut out the object tag and insert it on one of your own pages. You can create your own static annotation XML files or create a CGI/JSP script that retrieves the data from a database. Details of the various file formats are given below.

Input Parameters

These values are passed into the flash application (referred to as a SWF or "swiff" file, as it has a .swf suffix) via the `<object>` tag used to place the flash SWF on an HTML page. An example tag that you might place on an HTML page is given below. In this hypothetical example we are retrieving the mouse chromosome and banding pattern (the baseMap) from a static XML file on a server and the annotations are retrieved from a CGI script which returns the annotation XML for QTLs associated with the phenotype of *blood pressure*. Some of the longer lines have been broken to keep the page width down, the sample.html page provided with the download has a correctly formatted set of tags for you to copy.

```
<object classid="clsid:d27c6b6e-ae6d-11cf-96b8-444553540000"
codebase="http://fpdownload.macromedia.com/pub/shockwave/cabs/flash/swflash.cab#version=7,0,0,0"
<param name="allowScriptAccess" value="sameDomain" />
<param name="movie" value="GViewer2.swf" />
<param name="quality" value="high" />
<param name="bgcolor" value="#FFFFFF" />
<param FlashVars="&baseMapURL=http://server.mcw.edu/~simont/
mouse_ideo.xml&annotationURL=http://server.mcw.edu/cgi-bin/getDiseaseQTLAnnotation.cgi
?disease=blood_pressure&titleBarText=Rat Genome with sample data
&browserURL=http://genome.ucsc.edu/cgi-bin/hgTracks?org=Rat%26position=Chr%">
<embed src="GViewer2.swf" quality="high" bgcolor="#FFFFFF" width="300"
height="250" name="GViewer2" align="middle" allowScriptAccess="sameDomain" type="application/x-shockwave-flash"
FlashVars="&baseMapURL=http://server.mcw.edu/~simont/mouse_ideo.xml&
annotationURL=http://server.mcw.edu/cgi-bin/getDiseaseQTLAnnotation.cgi?disease=blood_pressure&
&browserURL=http://genome.ucsc.edu/cgi-bin/hgTracks?org=Rat%26position=Chr%" pluginspage="http://
</object>
```

The critical sections are `FlashVars` and the name of the Flash application (e.g. **GViewer2.swf**). The other parts are standard and you are unlikely to have to change any of them, though you can change the background color of the GViewer using the `bgcolor` parameter. The `movie` parameter is the path to the `GViewer2.swf` file on your server/filesystem. The `FlashVars` parameter contains the information that is passed into the GViewer to enable

it to find the base map and appropriate annotations. Due to differences between the windows and Mac platforms, you have to put these parameters in two places, in the <param> tags of the <object> tag, and also inside the <embed> tag. The windows flash plugin seems to use the object tags, the mac plugin uses the embed tags. If you modify these values and don't see a change, check to make sure you changed it in both locations as this is usually the reason for nothing happening!

The FlashVars parameters that you can set are as follows:

Parameter	description
	The URL of a file or server script that will return the XML description of the base chromosomal map and banding patterns.
baseMapURL	longestChromosomeLength The length of the longest chromosome (in base pairs). This value is only needed if the longest chromosome isnt Chromosome 1 (i.e. the first chromosome in the baseMapFile) as by default the application assumes this is the longest chromosome and scales every other chromosome accordingly. If another chromosome is actually longer (eg. in C. elegans) then the scaling will be off unless this value is set.
annotationURL	A URL of a file or server script that will provide the annotation XML
annotationXML	An embedded annotation XML string that defines the annotation directly in the HTML page, rather than the SWF file having to call a remote script or file. If the annotationXML parameter is defined it will take precedence over the annotation URL - ie it will use the XML embedded in the page rather than the remote source.
dimmedChromosomeAlpha	The alpha value (transparency) of chromosomes with no annotations. Defaults to 100 (no transparency), set to value between 0 (invisible) and 100 to dim the chromosomes that have no annotation. Useful to highlight the chromosomes with annotation information, values between 20-40 work well for this effect.
bandDisplayColor	The color of features drawn as bands, eg QTL. Should be set to a hexadecimal color using the Flash nomenclature, eg. 0xFF0000 is red, as opposed to traditional HTML nomenclature of #FF0000
wedgeDisplayColor	The color of features drawn as wedges, eg Genes. Should be set to a hexadecimal color using the Flash nomenclature, eg. 0xFF0000 is red, as opposed to traditional HTML nomenclature of #FF0000
titleBarText	Text to be shown in the top left right corner of the GViewer - can be used to put a title on the display
browserURL	Optional URL to create link out to genome browser for regions defined by region selection sliders in chromosome zoom view. The movie appends the chromosome number and start/stop coordinates to the end of this URL so it should be written with that fact in mind. The example above shows a valid URL for UCSC Rat genome. If the region sliders are used to select 1000 to 5000bp on chromosome 1, the final URL will look like:

<http://genome.ucsc.edu/cgi-bin/hgTracks?org=Rat%26position=Chr1:1000..5000>

The part in purple comes from the browserURL value, the part in bold is added to the end of the URL to complete the browser URL. %26 is used to escape the ampersand present in the UCSC URL otherwise it interferes with the apersands delimiting the

various FlashVars parameters.

The BaseMap

The baseMap file contains the data on the chromosome(s), their length, name, banding pattern and colors. A perl script is provided in the scripts directory that takes the cytoBandIdeo files produced by UCSC and converts them to the GViewer baseMap XML format. If this data is available you can find the appropriate UCSC file by going to <http://hgdownload.cse.ucsc.edu/downloads.html>, find your organism, find the annotation database and then see if there is a cytoBandIdeo.txt.gz file.

The Flash GViewer lays out the chromosomes by sorting them according to the index attribute of the chromosome and currently divides them over two rows with half the chromosomes on the top and half on the bottom. If there were 10 chromosomes, Chromosomes 1-5 would be on the top row with Chr 1 on the left, Chr 5 on the right, Chrs 6-10 would be on the bottom from left to right. Chromosomes are sized automatically based on the length of the first Chromosome, the assumption being that this is the longest. The scaling for all the other chromosomes and the mapping of genome coordinates (eg. basepairs) to flash coordinates (pixels) is all based off this value. The first chromosome is set to be 45% of the height of the Flash movie, you can set the *width* and *height* of the movie using the tags on the HTML page and the GViewer will resize the chromosomes accordingly. It doesn't scale the tooltip labels at this point so going really small makes the text illegible, going really large makes the text really large. It was developed with the movie having a size of 550px wide and 400px high so the text labels look good at this size, the further you stray from this the smaller/larger the text becomes. Try it and see how it looks, you can always zoom in using the Flash plugin's zoom feature.

The baseMap XML format looks like this:

```
<?xml version="1.0" standalone="yes"?>
  <genome>
    <chromosome index="1" number="1" length="268121971">
<band index="1" name="p13">
<start>0</start>
<end>10150087</end>
<stain>gneg</stain>
<color>0xFF0000</color>
</band>
<band index="2" name="p12">
<start>10150087</start>
<end>24291782</end>
<stain>gvar</stain>
</band>
<band index="3" name="p11">
<start>24291782</start>
<end>38547522</end>
<stain>gneg</stain>
</band>
</chromosome>
<chromosome index="2" number="10" length="110733352">
<band index="1" name="q11">
<start>0</start>
<end>6255594</end>
<stain>gpos</stain>
</band>
<band index="2" name="q12">
```

```

<start>6255594</start>
<end>26236148</end>
<stain>gneg</stain>
</band>
<band index="3" name="q21">
<start>26236148</start>
<end>34825919</end>
<stain>gpos</stain>
</band>
</chromosome>
</genome>

```

The XML elements in this file are as follows, I've used @xxxx to indicate attributes of elements

XML Element	description
chromosome	Each chromosome's data is contained within a chromosome element
@index	the chromosome index - an integer value used solely to sort the chromosomes to get correct order on the image (otherwise Flash sorts them 1, 10, 11, 2, 3, 4, X, Y, etc).
@number	the actual chromosome number, eg. 1, X, Y, used to label the chromosome on the Flash GViewer
@length	the length of the chromosome in appropriate units (probably basepairs). I don't think there would be a problem using other units (cR, cM) as long as you were consistent throughout the baseMap file and annotation XML.
band	each band's data on the chromosome is contained within a band element
@index	index number for the band, used to sort the bands in an appropriate order
@name	name of band, eg q11, p12. Currently not used but could be used to display the band info when the mouse was placed over it
start	band start location in appropriate Units (probably basepairs)
end	band end location in appropriate Units (probably basepairs)
color	Flash hex representation of the band color (begins with 0x rather than a #), eg. 0xFF0000 would give a red band. Can be used to create a customized color scheme for the baseMap.
stain	Giemsa stain abbreviation, obtained from the UCSC cytogenetic banding files. This is used to color the bands appropriately. This value is overruled by the value in a color element (see below). Currently recognized stain abbreviations and the corresponding colors are shown below.

Stain abbreviations and colors

Stain Abbreviation	Hex Color	Swatch
gpos & gpos100	#000000	
gpos75	#444444	
gpos66	#666666	
gpos50	#888888	

gpos33	#AAAAAA
gpos25	#CCCCCC
gneg	#FFFFFF
gvar	#CCCCCC
[default]	#CCCCCC

The Base Maps

The following base maps are provided in the standard distribution.

Basic Genome Maps

Base Map File	Map Description	External Links
rat_ideo.xml	Rat genome with cytogenetic bands, created from UCSC cytoBandIdeo file	Yes - UCSC Genome Browser
rgd_rat_ideo.xml	Rat genome with cytogenetic bands, created from UCSC cytoBandIdeo file	Yes - RGD Genome Browser
mouse_ideo.xml	Mouse genome with cytogenetic bands, created from UCSC cytoBandIdeo file	Yes - UCSC Genome Browser
human_ideo.xml	Human genome with cytogenetic bands, created from UCSC cytoBandIdeo file	Yes - UCSC Genome Browser
celegans_ideo.xml	C. elegans genome, no bands.	No

Comparative Maps (Synteny data from RGD VCMMap SEQBASD v1.0)

Base Map File	Map Description	External Links
human-rat_synteny.xml	Human genome with bands corresponding to syntenic regions in Rat	No
human-mouse_synteny.xml	Human genome with bands corresponding to syntenic regions in Mouse	No
rat-human_synteny.xml	Rat genome with bands corresponding to syntenic regions in Human	No
rat-mouse_synteny.xml	Rat genome with bands corresponding to syntenic regions in Mouse	No
mouse-rat_synteny.xml	Mouse genome with bands corresponding to syntenic regions in Rat	No
mouse-human_synteny.xml	Mouse genome with bands corresponding to syntenic regions in Human	No

The Annotation Data

Flash GViewer can read annotation data in the XML format shown below from a static text file or dynamically from a server-side script. The annotation consists of multiple feature elements which provide the basic information for each feature to be displayed. The server-side script would probably query a database for this information and then return the XML data as its output.

The annotation XML format looks like this:

```
<?xml version="1.0" standalone="yes"?>
<genome>
  <feature>
    <chromosome>7</chromosome>
    <start>10150087</start>
    <end>10150087</end>
    <type>gene</type>
    <label>Abc1</label>
    <color>0xFF0000</color>
    <link>http://rgd.mcg.edu/showReport.cgi?id=45678</link>
  </feature>
  <feature>
    <chromosome>6</chromosome>
    <start>20150087</start>
    <end>40150087</end>
    <type>qtl</type>
    <label>Bp123</label>
    <color>0x33AA33</color>
    <link>http://rgd.mcg.edu/showReport.cgi?id=12345</link>
  </feature>
</genome>
```

XML Element	description
chromosome	The chromosome number that this feature belongs to
start	start location in appropriate Units (probably basepairs)
end	end location in appropriate Units (probably basepairs)
type	feature type, used to determine what glyph to use when rendering the feature - currently two types gene and qtl. Gene is displayed as a triangle with the point towards the chromosome at the location of the feature. QTL is displayed as a bar stretching from the start location to the end location. The tooltip text that is displayed when the mouse is placed over the feature, or when the feature is displayed on the larger scale single chromosome view
label	color Flash hex representation of the band color (begins with 0x rather than a #), eg. 0xFF0000 would give a red feature. Can be used to create a customized color scheme for each individual feature.
link	URL to link out to if the user clicks on the feature, eg. to go to a gene report when they click on a gene

Web Interface & CGI script

To allow users to upload their own features for display on GViewer, a simple web form is available that takes annotation from a text field or uploaded file and displays it against the base map selected by the user. This allows users to view their own data against a variety of different base maps (eg. Mouse-Human syntenic map, Mouse-Rat syntenic map). Available options include:

Upload annotations in a simple text format, either by pasting into the form or uploading a file. Select from one of the available base maps (or create your own and edit the list accordingly). Customize the movie's color scheme. The background color, bar color and wedge colors can be modified using a pop-up color selector. Select different movie sizes. Three size options are provided - normal, large and extra large. Enlarging the movie can be useful for taking better screenshots for publication, etc. Web Interface Installation

1. The HTML page for the web interface is part of the main file distribution and is `/GViewer/index.html`. This contains an HTML form that points to the perl cgi script (see 2, below) and is initially configured to point to `/cgi-bin/gviewer_annotation.cgi`. The action attribute of the form tag should be modified if the perl script is installed in a different location.

```
<form name="gviewer_form" method="post" action="/cgi-bin/gviewer_annotation.cgi"
enctype="multipart/form-data">
```

The form contains a list of the available base maps provided with GViewer. These should be edited to remove any that are inappropriate and to add in any custom base maps that you may have created. The value attribute of the option tag should be the basemap's file name and these basemaps are assumed to be in the `GViewer/data` directory.

```
<select name="baseMap">
<option value="rat_ideo.xml" selected>Rat Cytogenetic Map</option>
<option value="mouse_ideo.xml">Mouse Cytogenetic Map</option>
<option value="human_ideo.xml">Human Cytogenetic Map</option>
...
</select>
```

The web page can be customized to fit in with an existing web site by modifying the `.css` stylesheet, changing the header graphic and the `gradient.gif` file used to create the background for the form header rows. The form is the only part of the page that is required so the header and footer could be deleted and the form could be copied into an existing web page with no ill effects.

2. Move the contents of the `cgi-bin/` directory to your server's `cgi-bin` directory. The current script is a single perl script that relies only on the `CGI.pm` module. Check that the path to the perl executable is correct at the top of the script and that `CGI.pm` is available. The perl script has a couple of variables that may need to be modified to suit the location of the GViewer files on your server.

```
# absolute URL pointing to GViewer directory holding movie and data

my $pathToGViewerWebDir = "/Gviewer";

# absolute URL pointing to GViewer style sheet, probably the same as the webdir, above.
my $pathToStyleSheet = "/Gviewer";
```

3. Text Annotation file format. The format of the annotation data that can be uploaded is very simple and is designed to be easy to create from typical spreadsheet applications or by manual typing into a text document. It consists of the same data found in the XML file with a single tab-delimited row for each feature. The data is in the following order:

```
Chromosome [tab] start_position [tab] end_position [tab] type [tab] label_text [tab] link (option)
```

The data used in the example annotation XML file above would look like this and could be pasted into the form or uploaded as a text file:

7 10150087 10150087 gene Abc1 <http://rgd.mcw.edu/showReport.cgi?id=45678>
 6 20150087 40150087 qtl Bp123 <http://rgd.mcw.edu/showReport.cgi?id=12345>

Javascript - Flash Connectivity

It is possible to use javascript to control basic functions in the flash movie. This allows a higher degree of interactivity between the web content and the flash movie itself by allowing javascript to trigger actions in the movie without the User having to directly interact with the movie itself. This feature uses the [Flash/Javascript Integration Kit](#) and has been tested on Mac (Safari, Firefox) and windows (IE6) using Flash Player 7 and 8. A number of additional flash .swf and actionscript files and some javascript files are required, these are included in the GViewer distribution so no further installation is necessary. Currently the following functions are available for control via Javascript.

Flash function	Function description
setHighlight(featureName)	Turns on the highlight and label for the feature with the label 'featureName'
unsetHighlight(featureName)	Turns off the highlight and label for the feature with the label 'featureName'
getAnnotationData(format)	Can be used to get the raw annotation data from the Flash Movie in various formats (text and html). Data is passed back to a javascript function displayAnnotationData(String) as a string object containing tab-delimited (text) or table (html). The function can then do something with the data such as display it in a new window, etc.

To call these functions using Javascript, a few changes are required in the web page containing the flash movie:

Import the JavaScriptFlashGateway.js javascript file.

```
<script type="text/javascript" src="./GViewer/javascript/JavaScriptFlashGateway.js"></script>
```

Configure the flash proxy that acts as a bridge between the javascript and GViewer flash movie. you have more than one GViewer on the page, you should use different uids to communicate with t There is then a simple javascript function that takes the name of a feature and an appropriate

```
<script type="text/javascript">
```

```
var uid = 1234567890; // needs to be unique on page, gets passed to Flash movie too

var flashProxy = new FlashProxy(uid, "./GViewer/javascript/JavaScriptFlashGateway.swf");

function doHighlight(action, featureName) {

if (action == 'set') {
  flashProxy.call('setHighlight', featureName);
}
else {
  flashProxy.call('unsetHighlight', featureName);
}
}
```

Flash_GViewer_Documentation

```
// Function to trigger the Flash method to get the annotation data in html format
function getAnnotationData() {
    flashProxy.call('getAnnotationData', 'html');
}

// Function called by flash to display the annotation data
function displayAnnotationData(data) {
    var generator=window.open('', 'Annotation Data', 'height=400,width=600');

    generator.document.write('<html><head><title>Annotation Data</title>');
    generator.document.write('<link rel="stylesheet" href="style.css">');
    generator.document.write('</head><body>');
    generator.document.write('===Known Issues with the Flash/Javascript connection===');
```

Using `mouseover` or `mouseout` can cause the highlight to accidentally stay on if another `mouseover` event happens before the `mouseout` event (can happen if two `mouseover` links are very close together on the page and the user moves the mouse quickly from one to the other).

Current Solution - Moving the mouse on and off the link a few times should make the highlight go away.

Known Issues

- If you load many features (hundreds), Flash is probably going to start running slowly - I haven't done any testing on this as it is rather dependent on the hardware the user is using to view the webpage but each new feature creates a new `MovieClip` in Flash and ultimately this can start to bog it down.
- Similarly, if your `baseMap` has many bands, the same thing may start to happen, again I have no quantitative data on this but if you have hundreds of bands (or syntenic segments or whatever the 'bands' represent in your situation) be aware that things might start to slow down. This is not probably not going to be a great solution to viewing an entire set of genome annotations (25,000 genes, 600,000 ESTs, etc), they will need to be filtered somehow. If you try a really large dataset, let me know how it goes!
- If you have lots of features that overlap on the chromosome, when you view an individual chromosome the current layout manager displays them all so they don't overlap - the problem being this may spread outside of the viewable area. You can now turn the labels off in the zoomed view which will alleviate this problem to some extent. The features are compressed on the genome view so they will overlap but all will be visible
- Debugging - the SWF has limited debugging messages at this point, so if your XML files are incorrectly formatted, etc. it won't tell you what's going wrong, you just won't get past the 'Loading' stage, not too friendly I agree. It will tell you if no annotation file was provided or if it has no `baseFile`.