lwgv → light weight genome viewer

Jeremiah Faith Ravi Sachidanadam

July 9, 2008

Contents

1 Introduction: why another genome browser?

With *Ensembl[2], Golden Path [4], SGD[3], flybase[1], Generic Genome Browser, NCBI[5], etc...* all on the open source scene, you might be wondering why we need yet another genome browser. Here are some situations where lwgv excels:

- you have a reasonably small amount of data you would like to visualize
- you don't want to spend half your life messing with mysql and trying to get 50 perl modules to install [1]
- you want something *fast*
- your annotations change frequently
- you don't need/want a large relational database system
- you want to create annotation files "on the fly"

2 getting started

I assume the reader is fairly well-versed in CGI programming, web servers, and unix. If you have little experience with any of these, you'll likely need to read some web tutorials to get up to speed before reading this document.

2.1 learning the language

The best way to start is to type out the examples from this manual and get them running on your machine (the examples are towards the back of the manual). Then you can use the the language description sections to modify the examples to your liking.

2.2 running lwgv

lwgv uses the executable `lwgv.cgi`. This executable *must* be in the cgi executable path of your web server. To make sure the program compiled properly, run by hand:

```
./lwgv.cgi
```

If the binary compiled ok, it will print some text to the terminal beginning with:

```
Content-type: text/html

<HTML><HEAD> etc.....
```

To run the program in your browser using your local web server enter, for example, the url:

```
http://127.0.0.1/cgi-bin/lwgv.cgi
```

This should list any file with the suffix `.ann` in the same directory as `lwgv.cgi`. You need to change 127.0.0.1 and cgi-bin according to your web server configuration.

For more info about the parameters lwgv.cgi will take in the url see section 3.3 on page 8.

[1] but you will have to install a number of C libraries (sorry)

3 basic language definition

The language lwgv uses has three sections.

1. configurations
2. genome information (tracks, graphs, sequences, additional configuration changes)
3. commands

The configurations define basic global properties such as how large an image to draw and how much white space to include around the tracks. Genome information defines all of the annotation features and their locations and represents the bulk of must lwgv annotation files. And the commands represent track based customizations like track color and track order.

The language is case sensitive, and whitespace is ignored. Additionally, the language provides the statements `#include` and `#insertFile` to allow the inclusion of additional annotation information from external files and to allow arbitrary text to be printed to the screen respectively.

3.1 file inclusion

File inclusion statements can be included anywhere in the annotation file, as they are used in a preprocessing stage[2].

3.1.1 #include

includes the specified file into the annotation file (similar to the `#include` statement in C); files can be nested up to 5 levels (i.e. an `#include` inside an included file, inside an included file, etc...). For example,

`#include` *`filename`*

takes the entire contents of *`filename`* and places it in the position where the `#include` statement was made. The inserted file is then parsed and rendered just like all the other annotations in the annotation file. This inclusion statement allows for simplified annotation files in situations where part of your data is constant (e.g. the locations of genes on a genome) and part of your data is dynamic (e.g. the locations of custom designed RNAi oligos on those genes). In this case, the gene location annotations could be stored in a static file and included in the appropriate place in each dynamic file.

3.1.2 #insertFile

prints the specified file to stdout (typically the web browser), allowing any arbitrary html to be added. For example,

`#insertFile` *`filename`*

[2] they are dealt with in the lexical analyser only, similar to the way `#include <header.h>` statements work with the C preprocessor

prints the contents of *filename* to stdout. The contents are *not* parsed for visualization. `#insertFile` allows you place the genome browser interface in a custom html template. Just include the top of the html template at the top of the annotation file and the bottom of the html template at the bottom of the annotation file to 'sandwich' the genome browser in your interface (see Figure 1 for an example showing lwgv.cgi within the *Many Microbe Microarrays Database* (m3d.bu.edu) web interface.

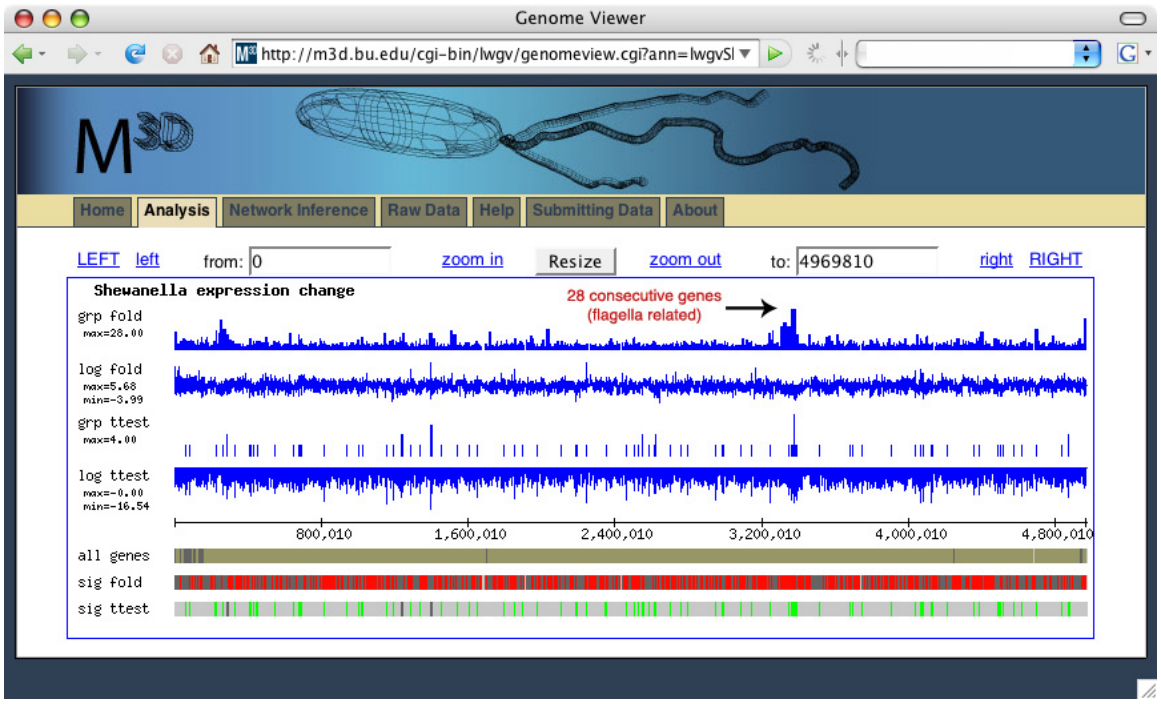

Figure 1: lwgv.cgi in the *Many Microbe Microarrays Website* html template.

3.2 configuration

all configuration options take the form:

```
#config PARAMETER_NAME PARAMETER_VALUE
```

3.2.1 boolean values

boolean values must take the form:

```
#config PARAMETER_NAME BOOLEAN_VALUE
```

where `BOOLEAN_VALUE` is either `true` or `false`. The boolean configuration options are:

Parameter name	**description**	**default**
DEBUG_MODE	prints error messages to the browser (e.g. if you are missing include files it warns you)	false
GRAPH_SMOOTH	performs spline smoothing on line graphs	false
SHOW_ZOOM_BAR	add toolbar to make the genome browseable (you usually want this *true* unless you are only showing a short sequence)	true
SHOW_RAINBOW	prints track names in color before printing sequence	true

3.2.2 integer values

integer values must take the form:

```
#config PARAMETER_NAME INTEGER_VALUE
```

where `INTEGER_VALUE` is a positive non-decimal number. The integer configuration options are:

Parameter name	description	default
IMAGE_WIDTH	width (in pixels) of the genome image; this image width can also be overriden from a url GET query string using `&image_width=INT`	600
TRACK_HEIGHT	height (in pixels) the track will be drawn	10
TRACK_PADDING	amount of whitespace to place between tracks	10
TRACK_SIDE_PADDING	amount of whitespace to add on the sides of the picture	5
SIDE_WIDTH	amount of space to allow on the left-side of the image for track names	75
GRAPH_HEIGHT	height of the graph(s), in pixels	30
GRAPH_PADDING	amount of whitespace to place between graphs	10
GRAPH_DENSITY	how much interpolation to do with the spline algorithm; higher numbers = more smoothing(only applies when GRAPH_SMOOTH is true)	0
MAX_SEQ_SHOW	maximal zoom range where the corresponding sequence will be shown (if set to zero [the default] or no sequence is given, sequence will never be shown). For example, if MAX_SEQ_SHOW is 1000, lwgv.cgi will print the browsers sequence to the screen when the region shown is less than 1000 base pairs. This feature only works if you've added the sequence to the annotation file using `sequence` *`mySeq`* `addSeq(ACGAT etc...)`	0
MAX_TABLE_SHOW	maximal zoom range where the corresponding feature table will be shown (if set to zero [the default], the table will never be shown).	0
SEQ_LINE_LENGTH	number of nucleotides to print per line (applies when MAX_SEQ_SHOW is set)	80
SUB_SEQ_LENGTH	length of sequence to print before printing a whitespace character, makes the sequence easier to read (applies when MAX_SEQ_SHOW is set)	10
COMPRESSION_LIMIT	number of pixels a feature must be separated by in order not to be joined (default is 1; when set to zero, all features within the specified range will be drawn with their corresponding hmap information; this can create LOTS of unnecessary I/O on large annotation files)	1

3.2.3 string values

string values must take the form:

`#config` *`PARAMETER_NAME STRING_VALUE`*

Parameter name	description	default
GRAPH_TYPE	available styles are *histogram* and *line*	histogram
TMP_PIC_DIR	location for temporary genome pics (directory must have write permissions); if not specified the program calls itself again to draw the pic and does not use temporary images (a little slower, but usually no big deal)	NULL
TMP_PIC_URL	the url path to your TMP_PIC_DIR if the TMP_PIC_DIR itself does not represent the url path	NULL
IMAGES_PATH	directory containing the zoombar images; if not specified, the program uses text links rather than pictures for the zoom bar, but this allows you do use fancier ones or custom ones.	NULL
BIG_LEFT_PIC	large left scroll image	doubleLeft_01.gif
SMALL_LEFT_PIC	small left scroll image	left_01.gif
BIG_RIGHT_PIC	large right scroll image	doubleRight_01.gif
SMALL_RIGHT_PIC	small right scroll image	right_01.gif
ZOOM_IN_PIC	zoom in mage	plus_01.gif
ZOOM_OUT_PIC	zoom out image	minus_01.gif
RESIZE_PIC	resize image	resize_01.gif

3.3 url parameters

Annotation files are specified using the parameter *ann*. For example:

```
http://www.myHomePage/lwgv.cgi?ann=myAnnFile.ann
```

will show the annotation file `myAnnFile.ann`. If the *ann* parameter is not used, the program will display all the files ending in .ann in the current directory. By default the program looks in the current directory for annotation files; this can be changed using the parameter *anndir*. For example:

```
http://www.myHomePage/lwgv.cgi?anndir=/tmpAnn
```

will show all the annotation files in `/tmpAnn` and

```
http://www.myHomePage/lwgv.cgi?ann=myAnnFile.ann&anndir=/tmpAnn
```

will show the file `myAnnFile.ann` in the directory `/tmpAnn`.

4 genome information

lwgv currently accepts three different types of genomic feature data: tracks, graphs, and sequences. The order features (tracks, graphs, etc...) are added is not important. Genomes are started with the command

```
begin genome myGenomeName
```

and are finished with the command

```
end genome
```

4.1 tracks

Features are added to a track using

```
track myTrackName addPairs(start:stop)
```

where `start` and `stop` are integers specifying the beginning and end locations of the features in the genome. The first time *myTrackName* is used it creates a new track with that name; subsequent calls simply add the features to the existing track. Multiply pairs may be added at once by separating them with a comma. Thus:

```
track gene_locations addPairs(1234:3456)
track gene_locations addPairs(0:1000)
track gene_locations addPairs(4567:9879)
```

and

```
track gene_locations addPairs(1234:3456, 0:1000, 4567:9879)
```

are equivalent.

4.1.1 track links

Two types of links may be added to a track: a normal link which displays a small tooltip when the mouse is held over it on the image (in most browsers) and a super link, which uses an html layer and javascript to allow an arbitrary amount of information about a link to be displayed when the mouse is held of the link [3]. A normal link uses the key word `link` and takes the form:

```
track myTrack addPairs(start:stop:link("linkName","url"))
```

note that *linkname* and *url* must be in quotes.

A super link uses the key word `slink` and takes the form:

```
track myTrack addPairs(start:stop:slink("title" =>
"linkName","url","linkName2","url2",...))
```

An example of a normal link is:

```
track web_info addPairs(0:2345:link("Google","http://www.google.com"))
```

An example of a super link is:

```
track web_info addPairs(0:2345:slink("Journal Links" =>
"Science","http://www.sciencemag.org",
"Nature","http://www.nature.com", "Cell","http://www.cell.com"))
```

[3] this only works on javascript enabled browsers

4.1.2 track colors

Track colors are set in the **commands** section described later. However, besides making the entire track one color, it is also possible to specific colors for individual feature in a track by adding `setPairColor(R,G,B)` directly after the track positions. For example:

```
track NM_057228.3 addPairs(6438342:6438438:setPairColor(255,255,102))
track NM_169348.1 addPairs(6438446:6442508:setPairColor(255,153,0):slink("title"
=> "NCBI","http://www.ncbi.nlm.nih.gov/))
```

However, lwgv.cgi allows a maximum of 256 colors, so you should be careful with how many different colors you use. Also, when using coloring the raw DNA sequence below the viewer image (if MAX_SEQUENCE_SHOW is set), lwgv.cgi only uses the track colors to color the raw DNA sequence. It does *not* use the `setPairColor(R,G,B)` information.

4.2 graphs

Points are added to a graph using

```
graph myGraph addPoints(x:y)
```

where x and y are real numbers specifying a point in the graph.

4.3 sequences

Sequence(s) are added using:

```
sequence mySeq addSeq(ACTGATCTGCATATCTAGCTGATCGTAGCTAGCTGATCGATGCTAGC...)
```

5 commands

Commands all have the form:

```
type variable command(value)
```

For example

```
genome myGenome removeGraph(myGraph)
```

removes *myGraph* from *myGenome* As another example

```
sequence mySeq setSequencePairsFromTrack(myTrack)
```

associates the features from *myTrack* to *mySeq*; this causes the program to color the sequence at all the features annotated on *myTrack*, using the track colors.

If *type* and *variable* are not given, the last appropriate type and variable are used. So in a file with only one sequence, `sequence` *mySeq* could be left off leaving the shorter:

```
setSequencePairsFromTrack(myTrack)
```

Note that since the software currently only supports one genome per file (something we hope to change in the future, to allow scanning multiple sequence, genomes, etc...on the same screen), the current genome is always the same, so `genome myGenome` is superflous, but not harmful in any way. Lastly many commands allow a comma separated list to be passed to them so:

```
removeGraph(myGraph1)
removeGraph(myGraph2)
removeGraph(myGraph3)
```

can also be written:

```
removeGraph(myGraph1, myGraph2, myGraph3)
```

Table 1 shows all the commands currently available. Those with an asterick (*) in the `command` column, accept multiple values.

type	**command(*type*)**	**explanation**
none	`showGenome(genome)`	causes the webpage to be created for *genome*
none	`trackCorrelate(trackA:locationA <=> trackB:locationB*)`	draws a line from *locationA* on *trackA* to *locationB* on *trackB*
sequence	`setSequencePairsFromTrack(track*)`	colors the sequence using the features and colors in *track(s)*. this feature works best with 1-2 tracks, otherwise nesting problems with overlapping features makes coloring the tracks impractical.
genome	`orderTracks(track*)`	orders *track(s)* in the order specified in the list (those not specified will be added to the end in the original order)
genome	`removeTrack(track*)`	removes *track(s)* from the genome
genome	`removeGraph(graph*)`	removes *graph(s)* from the genome
track	`setTrackColor(R, G, B)`	sets the RGB values (integers from 0-255) for the track
graph	`setGraphColor(R, G, B)`	sets the RGB values (integers from 0-255) for the graph
genome	`setGenomeLength(integer)`	sets the length of the genome to the value of *integer*

Table 1: summary of commands in *lwgv*

6 examples

Trying and modifying these examples is probably the best way to figure out how to use lwgv. Then, you can go back and look at the full list of features in each sections to see if there are any other software features you want to use.

6.1 simple example

Here is an example annotation file. The comments delimited by % explain what the commands do.

```
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%                                                                              %
% this example is about the bare minimum needed to get running                %
%                                                                              %
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

% initialize a genome
begin genome Hello_World_Genome
     % add a feature to first_track from 150bp to 200bp
     track first_track addPairs(150:200)
end genome

% create the webpage and image for the genome
showGenome(Hello_World_Genome)
```

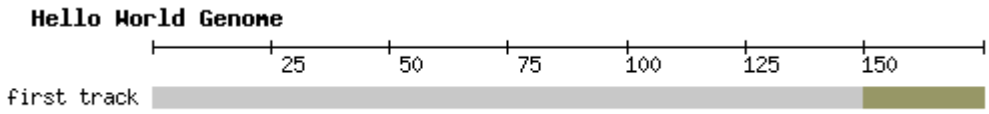

Figure 2: The image created created by lwgv using the simple example. Notice the underscores in the genome name and the track name become white space in the image.

6.2 medium example

```
% make the image narrower (500 pixels)
#config IMAGE_WIDTH 500

% initialize a genome
begin genome Simple_Genome
     % add a feature to first_track from 150bp-200bp and 25bp-40bp
     track first_track addPairs(150:200, 25:40)

     % and create a hyperlink on same track from 300bp to 475bp
     track first_track addPairs(300:475:link("my_link_title",
                                          "http://www.somewhere.com"))

     % and create a graph
     graph myGraph addPoints(50.58 : 1397.7, 67.80 : 2000.0,
                             500.0 : 2222.2)
end genome

% make first_track red, second_track green
track first_track setTrackColor(255,0,0)
track second_track setTrackColor(0,255,0)

% make sure first_track is before second_track in the image
orderTracks(first_track, second_track)

% create the webpage and image for the genome
showGenome(Simple_Genome)
```

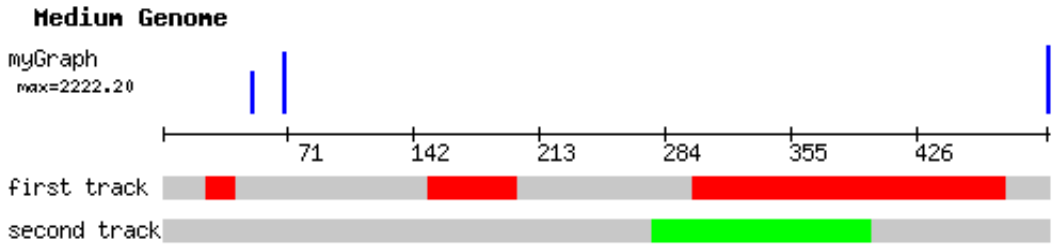

Figure 3: The image created created by lwgv using the medium example. Notice the colors and track order correspond to those we set.

References

[1] W. M. Gelbart, M. Crosby, B. Matthews, W. P. Rindone, J. Chillemi, S. Russo Twombly, D. Emmert, M. Ashburner, R. A. Drysdale, E. Whitfield, G. H. Millburn, A. de Grey, T. Kaufman, K. Matthews, D. Gilbert, V. Strelets, and C. Tolstoshev. Flybase: a drosophila database. the flybase consortium. *Nucleic Acids Res*, 25(1):63–66, Jan 1997. (eng).

[2] T. Hubbard, D. Barker, E. Birney, G. Cameron, Y. Chen, L. Clark, T. Cox, J. Cuff, V. Curwen, T. Down, R. Durbin, E. Eyras, J. Gilbert, M. Hammond, L. Huminiecki, A. Kasprzyk, H. Lehvaslaiho, P. Lijnzaad, C. Melsopp, E. Mongin, R. Pettett, M. Pocock, S. Potter, A. Rust, E. Schmidt, S. Searle, G. Slater, J. Smith, W. Spooner, A. Stabenau, J. Stalker, E. Stupka, A. Ureta-Vidal, I. Vastrik, and M. Clamp. The ensembl genome database project. *Nucleic Acids Res*, 30(1):38–41, Jan 2002. (eng).

[3] L. Issel-Tarver, K. R. Christie, K. Dolinski, R. Andrada, R. Balakrishnan, C. A. Ball, G. Binkley, S. Dong, S. S. Dwight, D. G. Fisk, M. Harris, M. Schroeder, A. Sethuraman, K. Tse, S. Weng, D. Botstein, and J. M. Cherry. Saccharomyces genome database. *Methods Enzymol*, 350:329–346, 2002. (eng).

[4] W. J. Kent, C. W. Sugnet, T. S. Furey, K. M. Roskin, T. H. Pringle, A. M. Zahler, and D. Haussler. The human genome browser at ucsc. *Genome Res*, 12(6):996–1006, Jun 2002. (eng).

[5] D. L. Wheeler, D. M. Church, A. E. Lash, D. D. Leipe, T. L. Madden, J. U. Pontius, G. D. Schuler, L. M. Schriml, T. A. Tatusova, L. Wagner, and B. A. Rapp. Database resources of the national center for biotechnology information: 2002 update. *Nucleic Acids Res*, 30(1):13–16, Jan 2002. (eng).