MuGeN User Manual

Mark Hoebeke

Mark.Hoebeke@jouy.inra.fr

MuGeN User Manual

by Mark Hoebeke

Revision History

Revision 1.3 August 2003
Revisited documentation and updated it to latest release.
Revision 1.2 2003-01-21 Revised by: mh
Fixed bugs in table of Perl module dependencies.
Revision 1.1 2002-12-12 Revised by: mh
Added sections on preferences file format. Added detailed description of analysis result DTD. Added appendix on secondary depen
Revision 1.0 2002-06-19 Revised by: mh
Initial release of user manual

Table of Contents

List of Tables

List of Figures

List of Examples

MuGeN User Manual

What is MuGeN ?

The *Multi-Genome Navigator*, or MuGeN, is a bioinformatics software package providing tools for exploring multiple annotated genomes along with *in silico* analysis results. It offers two distinct programs, one for interactive vizualization and navigation and another for the generation of images in various formats. Both programs can load annotated sequence data from a local file or retrieve it from databases across the network. Most of the parameters governing the way annotations and analysis results are displayed are customizable, either through the graphical user interface or with command-line parameters. The following sections show how to install and to use MuGeN before describing how to format home-made analysis results in order to integrate them in MuGeN.

What's New ?

There have been quite a few changes since the latest release of MuGeN. The most important ones are summarized below :

-

> **Warning**
>
> The format of MuGeN's preferences file has undergone significant changes. This makes the format of the previous file incompatible with the current one. Hence, any existing preferences file has to be renamed before running the latest release of MuGeN. A fresh preferences file can then be generated by using the Save Preferences item of the Preferences menu in the Map Drawing Window. The way preferences are handled had to be changed to enable persistence of new preferences (screen width, number of displayed map lines, base pairs per map line, view mode thresholds).

- MuGeN's interactive navigation tool can be run with the `mugen` command as well as with the usual `mugenv` command.
- Computer analysis results can be extracted from sources other than local files. As an example of live result computation, a pattern search feature is provided. Another example, only available to users residing on the INRA campus at Jouy-en-Josas is analysis result retrieval from a remote database.
- Two new computer analysis result formats have been developed. A histogram-type result for drawing bar charts under annotated maps, and a link-type result for drawing lines linking positions in two consecutive annotated maps.
- Screen updates are much faster after a window resize due to a rewrite of the way "reactive" features are handled.
- Support for XEMBL access and EMBL access through CORBA have been removed : EMBL access is now available through vanilla BioPerl modules (much more reliable !).

Installing MuGeN

System Requirements

MuGeN has been used successfully on Intel/Linux and Sparc/Solaris platforms. It is not intended for use on Windows machines. Disk space requirements are minimal (less than 7 Mb for the programs/modules and documentation, and an additionnal 25 Mb for the example data) but memory footprint can grow as more feature-rich genomes and/or complex analysis results are loaded. For example, the display of two reasonably-sized microbial genomes (about 4 Mb each) and a box plot of the conserved portions between the two of them consumes about 70 Mb of RAM on an Intel/Linux workstation. Line plots having one or more data points per base can be especially memory-consuming.

Software dependencies

MuGeN's programs and modules are all written in Perl. They rely on a set of more or less specialized third-party modules whose list is given in Table 1> (required modules) and Table 2> (optional modules). Notice that these tables only list modules not commonly found in Perl distributions. For a more extensive list of dependencies, see Appendix A>. All of these components are freely available, mostly from CPAN.

Table 1. Modules Needed by MuGeN

> >

Component	Release	Available at
bioperl	1.0	bio.perl.org (http://bio.perl.org)
Error	0.15	CPAN (http://www.cpan.org/modules/by-module/Error/)
GD	1.41	CPAN (http://www.cpan.org/modules/by-module/GD/)
Gtk	0.7008	CPAN (http://www.cpan.org/modules/by-module/Gtk/)
IO-String	1.01	CPAN (http://www.cpan.org/modules/by-module/IO/)
libxml	0.07	CPAN (http://www.cpan.org/modules/by-module/XML/)
PodParser	1.18	CPAN (http://www.cpan.org/modules/by-module/Pod/)
Usage	0.10	CPAN (http://www.cpan.org/modules/by-module/Usage/)
XML-DOM	1.42	CPAN (http://www.cpan.org/modules/by-module/XML/)
XML-Writer	0.4	CPAN (http://www.cpan.org/modules/by-module/XML/)

Table 2. Optional Modules for MuGeN

> > > > >

Component	Release	Available at
SeqDB and Micado data retrieval		

Component	Release	Available at
DBD-Pg	1.01	CPAN (http://www.cpan.org/modules/by-module/DBD/)
DBI	1.20	CPAN (http://www.cpan.org/modules/by-module/DBI/)
SeqDB analysis results retrieval		
SOAP-Lite	0.55	CPAN (http://www.cpan.org/modules/by-module/DBI/)

Installation procedure

MuGeN is available as a gzipped archive. Download the archive in an installation directory and expand its contents by issuing:

```
gzip -dc mugen-XXXXXXXX.tgz | tar xf -
```

where `XXXXXXXX` stands for the release number. This will create a subdirectory called `mugen-XXXXXXXX` containing all of MuGeN's programs, modules and documentation. `cd` in this directory and type:

```
perl install.pl
```

This will check for the required and optional Perl modules and configure MuGeN's scripts for execution. The absence of one or more optional modules will not prevent the program's installation, only a warning will be issued.

The executables **mugen**, **mugenb** and **mugenv** are located in the MuGeN installation directory. By adding this directory to the $PATH variable the executables can be run from any directory.

MuGeN relies on a preferences file to fix display and database connection parameters. By default, it looks for a file named `.mugenrc` in the user's $HOME directory. A template preferences file, called `mugenrc_template.xml` and located in the `Data` subdirectory can be used for a start and copied to the $HOME directory.

A set of example files, used throughout this document, is available in the `mugen-data-XXXXXXXX.tgz` archive. This archive can be extracted anywhere and creates a `mugen-data-XXXXXXXX` directory containing several annotated genomes in GenBank format, as well as some computer anlysis results.

Using MuGeN

The following sections offer a guided tour of MuGeN's main features and of how to make them work[1]. The examples use the data files found in the MuGeN data archive. To run the commands given in these sections, make the `mugen-data-XXXXXXXX` directory your current directory, and make sure the **mugen** and **mugenb** commands are located in a direcory accessible through your $PATH.

Using *mugen* for Interactive Genome Exploration

To start a visual exploration session just run the **mugen** command. This opens MuGeN's graphical user interface consisting of the three windows detailed below.

The Map List Window

This window (Figure 1>) displays all loaded maps and computer analysis results. It also allows the manipulation of these maps and associated analysis results with the button row located below the map list.

To load a new map, select a data source from the available sources in the popup menu, then click on the Add button. Depending on the datasource, some additionnal information will be requested (typically a filename or an access number). For instance, to load a local GenBank file, make sure the current data source is Local File and use the Add button to select the GenBank file (try it out to load the genomes of *Bacillus subtilis* (resp. *B. halodurans*) genome contained in the `Bsub.gbk` (resp. `Bhal.gbk`) file of the data directory.

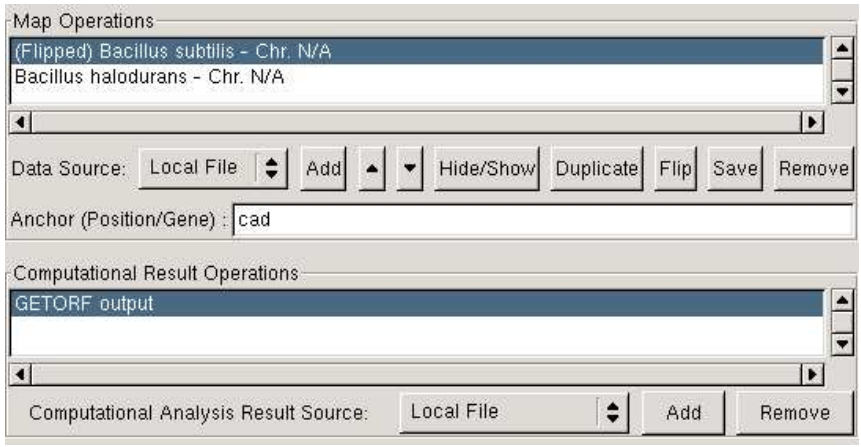
It may be useful to work with several copies of the same map (for instance to compare different portions of the same genome). To add a copy of the currently selected map, use the Duplicate (when duplicating a map, its associated analysis results are *not* duplicated).

The order in which the maps are displayed can be modified with the two arrow buttons. They shift the currently selected map up or down. A map can be hidden and redisplayed with the Hide/Show button. Any map can be "flipped" with the Flip button, meaning that the base positions decrease from left to right, instead of increasing, and that the strands of the features are switched: features ont the forward strand move to the reverse strand and vice versa. This feature is useful to compare genome portions which are conserved but whose directions are opposite. Finally a map can be removed using the Remove button. Notice that if there is only one map in the list, it cannot be removed.

Below the map operations panel, an Anchor textfield can be found. Each map can have it's own anchor which "fixes" its relative position. An anchor is either an integer value (positive or negative), representing a base position, or a gene name. In the latter case, the start position of this gene (if it exists in the selected map) will be used as anchor. Moreover, the map will be flipped if the gene is on the reverse strand. Anchors are useful to simultaneously display distant portions of genome maps. For example, after loading two genome maps of closely related organisms, the context of a gene bearing the same name in the two organisms can be examined by selecting each map in turn and entering the common gene name in the anchor textfield. In the case of *B. subtilis* and *B. halodurans*, a possible anchor for both genomes is the *cad* gene.

The remaining part of the map list window contains a list of computer analysis results loaded for the currently selected genome map. Results can be added (respectively removed) through the Add (resp. Remove) button. As for annotated sequence data, analysis results may come from various sources which can be selected in the Computational Analysis Result Source menu. A sample analysis result file `Bsub_orfs.xml` contains all ORFS over 300 bp detected by the **getorf** program included in the EMBOSS package. It can be loaded when Local File as the current analysis result source.

Figure 1. The Map List Window



>

The map list window with two genome maps (*Bacillus subtilis* and *Bacillus halodurans*). The selected map (*B. subtilis*) is anchored on the *cad* gene which has caused the map to be flipped. A computer analysis result, *GETORF output* has been added to the *B. subtilis* map.

The Map Drawing Window

This window gives a graphical display of the annotated genome maps along with the computer analysis results. The main area is divided in "strips" or lines. Each strip represents a portion of an annotated genome with its associated computer analysis result. When several annotated maps are loaded, their strips are displayed one above the other (i.e. the first strip of the first map followed by the first strip of the second map followed by the second strip of the first map etc.). In that case, each map will have a different background color, ranging from white to light grey. When computer analysis results exist for a given map, they are either overlaid on the map features, or located immediately below the map they belong to.

Map displays can be obtained with three different detail levels: a brid’s eye view for viewing large genome portions (strips of a few hundred kb to several Mb) in which features are draw with simple boxes, an intermediate view valid for strips ranging from a few hundred bp to a few hundred kb) in which each feature is draw according to its type is sensitive to mouse clicks and/or movements, and a sequence view displaying the actual DNA sequence possibly with its translation in the six reading frames (usable for portions below a few hundred bp). The switch from one view mode to another is automatically performed when the strip size crosses user definable thresholds.

By default, six lines per strip are used to draw CDSs, on for each reading frame of each strand. Other features are drawn either on the axis, if they are positional features (promoters, terminators, RBSs), or on a separate line below the CDS lines if they extend more than a dozen bp. (different RNAs, miscellaneous features and others). Also by default, CDSses are colored according to the strand they are located on, and filled if they have a known function (meaning they have a function qualifier not containing "unknown", "putative" or "hypothetical"), and empty otherwise.

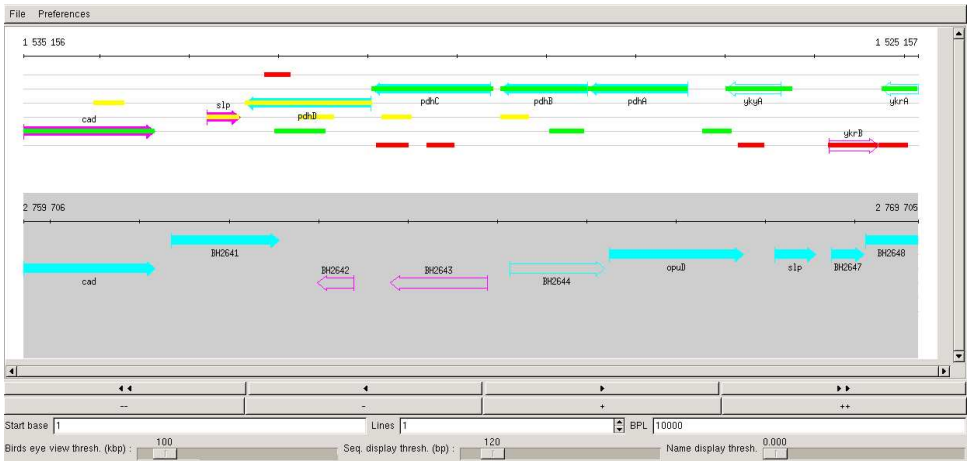
The majority of display settings can be modified with the user controls at the bottom of the Map Drawing Window or with the menu entries it offers. The topmost row of user controls contains arrow buttons to

move forward or backward along the maps. The row below allows them be to zoomed in or out. Precise starting points, number of lines and bases per line can be set with the text fields below the zoom buttons. Finally, the thresholds for switching between the different view modes can be fixed with the sliders at the bottom of the window. The rightmost slider defines the minimum relative size for features whose names will be displayed (for instance, a setting of 0.01 will not display feature names for features covering less than 1% of a strip).

The Preferences menu offers several items influencing the map display:

- Expand Strands: When checked, features belonging to different strands will be displayed on separate lines. Otherwise they will be displayed on the same line.
- Show Frames: When checked, CDSs are displayed on different lines acccording to their reading frame.
- Visible Features: This submenu offers one entry per feature type. Only the checked features are displayed on the map.
- Map Area Width: The width in pixels of the area on which the maps are drawn can be selected in this submenu.
- Save Preferences: The current settings of the Preferences menu are saved in the default preferences file (`$HOME/.mugenrc`).

Figure 2. The Map Drawing Window



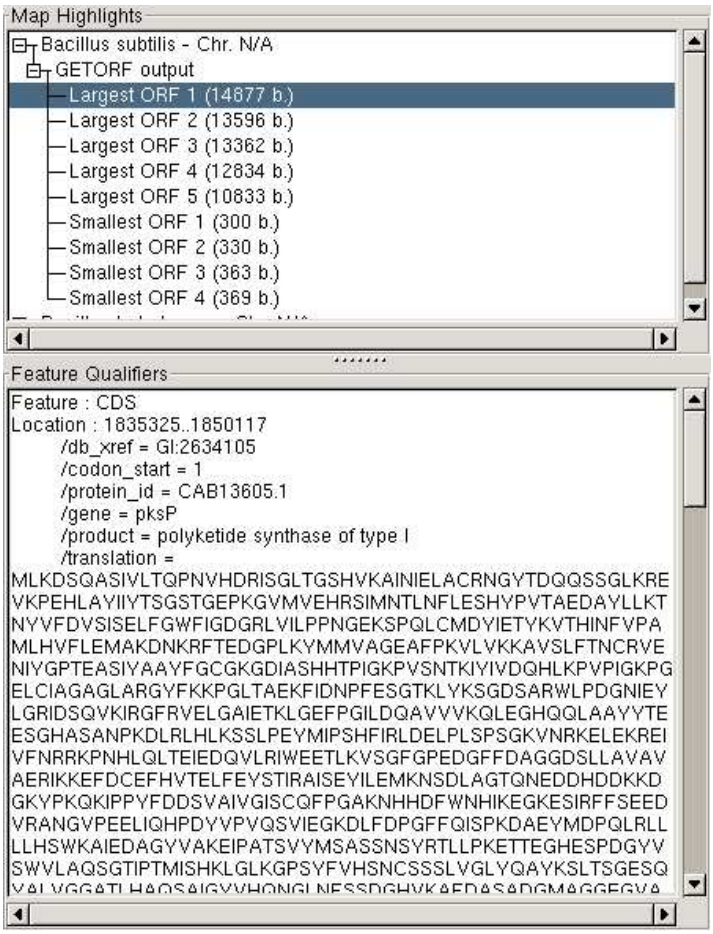
>

The Map Drawing Window showing portions of the genomes of *B. subtilis* and *B.halodurans* anchored on the *cad* gene. Results of the GETORF program are overlaid on the map of *B. subtilis*

The Information Window

The Information Window contains two panels. The top panel is dedicated to computer analysis result highlights. It displays the list of loaded maps, and for each of them, a sublist of loaded computer analysis results. These results in turn display a series of highlights. A highlight defines a portion of interest on the corresponding map. When selecting a highlight, the map automatically moves to display the start of this portion. The bottom panel gives "live" information about the feature currently under the mouse pointer (i.e. its location and the list of qualifiers with their associated values). Note that this live update functionality is not provided in bird's eye view mode.

Figure 3. The Information Window



>

The Information Window. Top panel: highlights of the GETORF results for *B. subtilis* consisting of the 5 largest ORFs and the 5 smallest ORFs. Selection of the "Largest ORF 1" moved the map to the start of the corresponding gene (*pksP*). Bottom panel : information associated with the *pksP* gene over which the mouse was moved.

Generating Annotated Genome Images with *mugenb*

Using MuGeN in batch mode allows for automatic creation of images containg annotated genome maps including computer analysis results. Most of the display parameters accessible in interactive mode have a command-line equivalent (see the Section called *MuGeN Option List*> for the list of options). Other parameters can be set using the preferences file (see the Section called *The MuGeN preferences file*>), which is also used by *mugenb*. The examples below illustrate some frequent use cases for MuGeN in batch mode, and can be run from MuGeN's data directory.

Example 1. PNG image generation

The following command line generates an image of the first 10 kb of the map of *B. subtilis* spread across 4 lines.

```
mugenb -d Bsub.gbk -f 1 -l 10000 -s 2500 -o PNG ex1.png
```

The `-d` argument defines which map to load.The `-f` and `-l` arguments define the first and last bases to display, the `-s` argument defines the "strip length" or the number of bases per line. Finally, the `-o` argument specifies the output file format. The last argument is the name of the output file.

Example 2. Loading multiple maps and analysis results

The following command line generates an image of the first 10 kb of the maps of *B. subtilis* and *B. halodurans*. Results of the GETORF program will be placed over the map of *B. subtilis*

```
mugenb -d Bsub.gbk -d Bhal.gbk -f 1 -l 10000 -s 2500 -c Bsub_orfs.xml,1 \
 -o PNG ex2.png
```

As can be seen, multiple `-d` options can be used to load multiple maps. Moreover, the `-c` argument references a results file to be used. The map it will be related to is defined by the number after the comma added after the filename (here 1 denotes the first map loaded, i.e. *B. subtilis*).

Example 3. Anchoring maps

The following example shows how to anchor each map to a specific position.

```
mugenb -d Bsub.gbk -d Bhal.gbk -f 1 -l 10000 -s 2500 -c Bsub_orfs.xml,1 \
-r rpmH -r cad -o PNG ex3.png
```

The `-r` flags specify anchor points for each map, there can be as many of them as there are loaded maps. The rank of the `-r` option determines which map it relates to (the first `-r` applies to the first map, the second `-r` to the second map and so on). Here, the map of *B. subtilis*will be anchored on the *rpmH* gene, and the map of *B. halodurans* on the *cad* gene. Anchor points can either be strings, standing for gene names, or numbers denoting base positions. When anchors are used, the `-f` and `-l` argument define the extents of the displayed portion relative to the starting point of the anchor. In the above example, the generated image will generate an image displaying 10 kb, on four lines, starting with the anchor point.

Example 4. Using remote data sources and flipping maps

The following example shows how to load remote maps and to flip them.

```
mugenb -d genbank:\!AY357726.1 -f 1 -l 8000 -s 2000 -o PNG ex4.png
```

This loads the entry with accession number AY357726.1 from GenBank across the network. The format of the argument following the `-d` switch is composed of a datasource, a colon and an identifier. When datasource and colon are omitted (as in the first examples) the identifier refers to a local file. Otherwise, it is specific to the datasource. Here, the identifier is preceded by an exclamation mark do indicate that the map is to be flipped (the exclamation mark is escaped by preceding it with a backslash, this is necessary because some shells give a special meaning to the exclamation mark).

Example 5. Generating clickable image maps

The following example shows how to generate image maps to be included in Web pages for generating clicable annotated genome maps.

```
mugenb -d genbank:AY357726.1 -o IMAP -f 1 -l 8000 -s 2000 \
-u http://localhost/htmldocs/viewmap.cgi? ex5.png
```

This generates the same PNG file as the previous example but also prints on the standard output a series of HTML instructions defining clickable areas corresponding to the map's features. The destination of links activated by a click is composed of a root URL which is specified with the `-u` argument to which specific information is appended (i.e. a tag attribute for the type of the feature, a name tag for CDS features representing the gene name and a start and end tag giving the start and end positions, in bp., of the feature in the map). Dynamic map pages can thus be built by including this generated output in CGI-generated Web pages.

The MuGeN preferences file

A major part of MuGeN's run-time parameters (like display options or remote feature retrieval settings) are held in MuGeN's preferences file. By default, MuGeN will use a file called `.mugenrc` located in the user's $HOME directory, but a different file can be loaded through the `-p` command option. The preferences are stored in XML format and the DTD is specified at the start of the file. The following sections detail the contents of the preferences file.

Feature data sources

Parameters used to connect to remote data sources are specified in the <featuredatasources>tags. The following lists explains which data sources are supported and how to configure them. Access to both data sources is essentially restricted to people working on the INRA campus at Jouy-en-Josas, but outside accesses may be opened on demand if required.

- The <micado> element has attributes specifying how to connect to the Micado database (see http://www-mig.jouy.inra.fr/bdsi/Micado/). By default, these parameters have dummy values. To be able retrieve data from Micado a valid username and password are required.
- The <seqdb> element has attributes specifying how to connect to the SeqDB database. By default, these parameters have dummy values. To be able retrieve data from SeqDB a valid username and password are required.

Analysis result data sources

As for annotated feature data, MuGeN is also capable of loading analysis results from remote locations. The <resultdatasources> element enumerates where these results can come from. For the time being, only one result source is available (and only to users on the INRA campus of Jouy-en-Josas) :

- The `seqdbresults` element defines how to reach the SeqDB database to extract results regarding atypical genes and gene context conservations in complete bacterial genomes through it's `url` attribute.

Display parameters

These parameters are attributes of the <mapdisplay> element.

- The `width` attribute fixes the preferred width for the map drawig area in pixels.
- The `bpl` attribute defines the number of bases per line of the map display.
- The `strands` attribute can take to values : `collapsed` which will display all CDSs, regardless of their orientation, on the same line, and `expanded` where the two strands are separate. The default value is `expanded`.
- The `frames`attribute can also take two values : `visible` which will draw each CDS in it's own reading frame, and `hidden` where all CDSs of a given strand are drawn on the same line. The default value is `visible`

Feature Display

Some features present in GenBank tend to clutter the graphical display (i.e. the `source` and `gene` features). Customization of which features to display and which features to hide is performed in the section delimited by the `features` tags. This section contains a set of `feature` element having a single `visible` attribute which can be set to `true` or `false`, the default being `true`.

Display Thresholds

The `thresholds` section contains the values of the various thresholds used by MuGeN. It's `namethresh` element has a `percent` attrinute to define the minimum size (in percent of the line width) for a feature whose name is to be displayed. The `seqthresh` sets the threshold (in bp) for swicthing between the default view mode and the sequence viewmode through it's `bp` attribute. Finally, the

`birdseyethresh` element has a `kbp` attribute defining the threshold for switching between the default view mode and the bird's eye view mode (in kbp).

Highlight visibility

In bird's eye view mode, regions defined in analysis result highlights are surrounded by boxes. This behaviour is inappropriate for results having multiple small highlights. Thus, the visibility of these boxes can be switched on or of with the `visible` attribute of the `highlights` element. This attribute can be set to `true` or `false` (`true` being the default).

Links to external resources

MuGeN is capable to communicate with Web browser to visualize resources related with currently displayed features. The following tags define which resources MuGeN should use and how the Web browser will be invoked.

- The <links> tags delimit a set of <links. Each <link> tag defines an external resource : the `name` attribute defines the name of the resource as it will be displayed in the resources menu, the `id` resource defines the prefix of the `/db_xref` qualifier. For instance, for a `/db_xref="taxon:71421"` qualifier, the prefix is `taxon`. The `url` attribute defines the URL leading to the resource. MuGeN will add the specific suffix when invoking the resource. (For instance, taking the same example, the string `71421` will be appended to the URL leading to the taxon database.
- The <browser> tag defines how to invoke an external browser from within MuGeN. It's `command` attribute specifies the exact command line necessary to launch the chosen browser. Inside the command line the string `_URL_` will be replaced by the actual URL of the external resource when this resource is activated.

Computer Analysis Result formats

MuGeN is capable of loading computer analysis results stored as XML files and conforming to the DTD defined in the `CompAnalResults.dtd` file located in the `Data` subdirectory of MuGeN's installation directory. Basically, anaylsis results come in 4 flavors, and each plot can define a set of highlights delimiting interesting regions. These highlights will be listed in the Information Window as selectable items. On selection, the Map Drawing window will automatically scroll to show the start of the highlight. Moreover, one result file can contain a mix of various result types (each with their own highlights) as well as a section defining specific colors.

- Line plots adapted to "curves" having one ore more data points per base. Example of line plots include GC% pots, state probabilities generated by Hidden Markov segmentation programs and others. Line plots can either be located on a separate strip below an annotated map, or drawn as overlays on the map. Each plot can be attached to one of the two strands and/or one of thre their three reading frames.
- Box plots listing a series of boxes defined by their start and end positions as well as optional parameters as thickness and color attributes. They can be used to single out regions of interest such as sets of unique genes, or trains of genes conserved among several organisms. As for line plots, box

plots can be drawn separately of annotated maps or on top of their features according to specific strands and/or reading frames.

- Histograms suited to represent a given quantity below an annotated map. They can be used to for instance to plot the number of genomes in which orthologs for a given gene occur. Contrary to the previous plot types, histograms are always drawn below annotated maps. The aspect of each bar of the histogram (starting point, width, color, filled or empty) is user-definable, as well as the overall height of the plot. This height is expressed in lines where a line represents the vertical space occupied text written in the standard font.
- Link plots for giving visual clues about relations between elements of different genomes. Link plots define a set of links consisting of a base position in the map the plot is associated with, and a base position in the next map of the display. On the scale of a genome, they can be used to link orthologous genes of two organisms. As for histograms, link plots are always drawn as separate plots below a map and their overall height is customizable.

The computer analysis results DTD

```
<!ELEMENT companalresults (colors|lineplots|boxplot|histogram|links)*>
<!ELEMENT colors (color)*>
<!ELEMENT color EMPTY>
<!ATTLIST color
    name   CDATA #REQUIRED
    red    CDATA #REQUIRED
    green  CDATA #REQUIRED
    blue   CDATA #REQUIRED>
<!ELEMENT highlights (highlight)*>
<!ELEMENT highlight EMPTY>
<!ATTLIST highlight
      label CDATA #REQUIRED
      begin CDATA #REQUIRED
      end   CDATA  #REQUIRED>
<!ELEMENT lineplots (lineplot|highlights)*>
<!ATTLIST lineplots
      type  (separate|overlay) "separate"
      comment CDATA #REQUIRED
      min   CDATA #IMPLIED
      max   CDATA #IMPLIED
      smoothing CDATA #IMPLIED>
<!ELEMENT lineplot (#PCDATA)>
<!ATTLIST lineplot
      frame (none|all|1|2|3) "none"
      strand (1|-1) "1"
      color CDATA #IMPLIED
      start CDATA #IMPLIED
      step  CDATA #IMPLIED>
<!ELEMENT boxplot (box|highlights)*>
<!ATTLIST boxplot
   type (separate|overlay) "separate"
   comment CDATA #REQUIRED>
<!ELEMENT box EMPTY>
```

```
   <!ATTLIST box
      begin CDATA #REQUIRED
      end   CDATA #REQUIRED
      thickness CDATA #IMPLIED
      label CDATA #IMPLIED
      halign (left|middle|right) "middle"
      valign (above|inside|below) "below"
      labelcolor CDATA #IMPLIED
      frame (none|all|1|2|3) "none"
      strand (1|-1) "1"
      color CDATA #IMPLIED
      filled (yes|no) "yes">
   <!ELEMENT histogram (bar|highlights)*>
   <!ATTLIST histogram
             comment CDATA #REQUIRED
             min CDATA #IMPLIED
             max CDATA #IMPLIED
             barwidth CDATA #IMPLIED
             barcolor CDATA #IMPLIED
             filledbars (yes|no) "yes">
   <!ELEMENT bar EMPTY>
<!ATTLIST bar
             start CDATA #IMPLIED
             width CDATA #IMPLIED
             height CDATA #REQUIRED
             color CDATA #IMPLIED
             filled (yes|no) "yes">
   <!ELEMENT links (link|highlights)*>
   <!ATTLIST links
             comment CDATA #REQUIRED>
   <!ELEMENT link EMPTY>
<!ATTLIST link
             from CDATA #REQUIRED
             to   CDATA #REQUIRED
             color CDATA #IMPLIED>
```

DTD explanation

- Color definitions: colors must have a name and three color attributes defining the amount of color in each of the three color channels. These values range from 0 to 1 and conform to the RGB color model. Example:

  ```
  <color name="turquoise" red="0.25" green="0.88" blue="0.8">
  ```

- Plot highlights: regions of interest can be defined as highlights. Each highlight will have it's own entry in the highlight section of the Information Window. Highlights are defined by a label and begin and end points expressed in bases. Example:

  ```
  <highlight name="putative gene transfer" begin="1695413" end="1878744">
  ```

- Lineplots: this container tag groups a set of lineplots and their associated highlights. It defines how the lineplots will be positioned wrt. the features through the `type` attribute. If it's value is set to `"separate"`, the lineplots will be drawn on a separate line below the features they are related to; if set to `"overlay"` they will be mixed with the features. The `comment` attribute is used to set the name of the set of lineplots. This name will be displayed in the Computer Analysis Result panel of the Map List Window. For a given map, there can only be one result with a given name. The `min` and `max` attributes are optional and can be used to specify the extreme values of the plots. By default, these are computed automatically. Finally, the `smoothing` is meant to improve drawing speed by allowing a series of points whose values do not differ by more than the relative amount given, to be drawn as a horizontal line. For instance, if this parameter is set to `"0.1"` and the plot contains a series of consecutive values in the range 0.9 to 0.99, only the endpoints of the series will drawn and linked with a horizontal segment.
- Lineplot: this tag encloses the actual data values to be plotted. The position of this plot relative to the features is fixed with the `frame` and `strand` attributes. The former allows toposition a plot in a specific reading frame (values `"1", "2"` or `"3"`), to make a plot span all three reading frames (value `"all"` or position it below the CDSs with the other types of features (value `"none"`). Additionnally, the `strand` attribute defines over which of the two strands the plot will be drawn. The `color` defines the color of the plot. The `start` attribute sets the position of the base corresponding to the first data point, and the `step` attribute defines the number of bases separating each data point. Examples:

```
<lineplots type="separate" comment="Line plot example 1">
<lineplot frame="all" color="red" start="1000" step="10">
100
0
50
0
100
</lineplot>
</lineplots>
```

  This will draw a line plot on a separate line below the map features. The plot starts at base 1000 with value 100, drops to value 0 at base 1010, raises to value 50 at base 1020 etc.

```
<lineplots type="overlay" comment="Line plot example 2">
<lineplot frame="1" strand="1" color="green" start="1000" step="10">
100
0
50
0
100
</lineplot>
</lineplots>
```

  This will plot the same curve as above, except that it will be positioned over de CDSs of the leading strand and in reading frame 1.
- Boxplot: this container tag encloses a set of box descriptions. It has `type` and `comment` attributes identical to line plots.
- Box: this tag describes the precise characteristics of a boxplot component. It's `frame, strand` and `color` attributes have the same meaning as for line plots. The `begin` and `end` attributes define the range of base positions the box should cover. The `thickness` attribute defines the vertical width of the box and can take values in [0..1]. The `label` attribute allows the definition of a text string

accompanying the box. The position of this string wrt. the box is set with the `halign` and `valign` attributes and it's color with the `labelcolor` attribute. Example :

```
<boxplot type="separate" comment="Box plot example">
<box begin="1" end="1000" thickness="0.2" color="red" filled="yes"
label="First Kb" labelcolor="blue" valign="above"/>
<box begin="1001" end="2000" thickness="1" color="black" filled="no"
label="Left of 2nd Kb" labelcolor="green" valign="inside" halign="right"/>
</boxplot>
```

MuGeN Option List

Table 3. Options common to mugenb and mugenv

> >

Option	Multi [a]	Functionality
-d *source:id*	Yes	Specifies a resource from which to load annotated genome maps. Each resource consists of two parts, a *source* and an *id*. The source can be one of **file, genbank, embl, xembl** or **micado**. When no source is specified, **file** is taken as default. The id points to the specific map in the source. When the latter is a file, the id is simply the filename (in GenBank, EMBL, BSML or fasta format). When the source is a database (**genbank, embl, xembl, micado**) the id is the access number of the database entry. Maps will be displayed from top to bottom in the order they are entered on the command line. If the *id* start with a "!" the map will be flipped.
-f *firstbase*	No	Specifies the starting point of the image to build. In the absence of any reference points, this is the first base of the map that will be located in the upper left corner of the image. If a reference point is given, the upper left corner will be the reference point offset by the amount specified by this option.
-l *lastbase*	No	Specifies the ending point of the image to build. In the absence of any reference points, this is the last base of the map that will be located in the upper lower right corner of the image. If a reference point is given, the lower right corner will be the reference point offset by the amount specified by this option.
-s *step*	No	Specifies the number of bases per display line.

Option	Multi [a]	Functionality
-r *refpos*	Yes	Specifies a *reference position* or *anchor* for a genome map. If the reference position is an integer, the start of the displayed image will be computed by adding the value of the -f option to the integer. If the reference position is a string, MuGeN will look for a CDS feature having a gene qualifier whose value equals the given string. If such a CDS is found, it's start base will be used to compute the start of de displayed image as explained above. Moreover, if the gene is on the reverse strand, the map will be flipped. The genome map for which the reference position is defined is determined by the index of the *-r* option wrt. the *-d* option (i.e. the first *-r* option will be applied to the map defined by the first *-d*option, the second *-r* applies to the second *-d* and so on).
-c *filename[,index]*	Yes	Specifies a computational analysis results file to display with a genome map. If a comma and an *index* are appended to the filename, the result will be applied to the genome map of the corresponding index. Index 1 is the genome map loaded by the first *-d* option, index 2 the map corresponding to the second *-d* and so on.
-e *filename*	No	Specifes a file containing a color scheme to apply to displayed features.
-w *n*	No	Specifes the width in pixels of the drawing area
-p *filename*	No	Specifes the preferences file to load. If no *-p* option is given, the preferenes file will be set to `${HOME}/.mugenrc`.
Notes: a. Multi options are options that can be used several times on the command line.		

Table 4. Options specific to mugenb

> >

Option	Multi	Functionality
-o *format*	No	Specifies the output format of the image file to be generated. Valid formats are : PNG, IMAP, PS, EPS, XFIG.
-m *mediatype*	No	Specifies the media type, for PS or EPS output files. Valid types are : a7, a6, a5, a4, a3, a2, a1, a0, b7, b6, b7, b4, b3, b2, b1, b0, lettern legal, executive, ledger.

Option	Multi	Functionality
-u *urlprefix*	No	Specifies the root URL for client-side image maps in IMAP format. Parameters relative to dislayed features will be appended to this root URL. For instance, given a root URL of `http://www.somewhere.org/cgi-bin/myscript.pl?myid=xyz&`, and an image containing a CDS feature, whose name is abcX positioned from base 1234 to base 5678, the URL generated for it's clickable area will be `http://www.somewhere.org/cgi-bin/myscript.pl?myid=xyz&tag=CD`

Notes

1. Table 3> details all command line options supported by MuGeN.

Appendix A. Secondary dependencies

C libraries

The following list details a set of libraries upon which the modules used by MuGeN relie. They are part of all major Linux distributions, but may be needed for installing MuGeN on other platforms.

- Glib: Glib is a general purpose C library needed by the Gtk toolkit. It is available at ftp://ftp.gtk.org/pub/gtk/v1.2/. (ftp://ftp.gtk.org/pub/gtk/v1.)
- Gdk: Gdk is a low level graphics library also used by Gtk. It is also available at ftp://ftp.gtk.org/pub/gtk/v1.2/ (ftp://ftp.gtk.org/pub/gtk/v1.).
- Gtk: Gtk is the library providing all of the graphics interface widgets and controls. It is available at ftp://ftp.gtk.org/pub/gtk/v1.2/ (ftp://ftp.gtk.org/pub/gtk/v1.).
- Gd: GD is the library needed for generating PNG images with MuGeN. It is available at http://www.boutell.com/gd/

Perl modules

Below is the list of Perl modules needed by MuGeN. Most of them are part of standard Perl installations on Linux distributions, but again, installing MuGeN on Solaris may require to install some or all of them. They can all be found on CPAN (http://www.cpan.org).

- Carp: used for errors, warnings and information messages.
- File::Basename: used to separate a path into it’s components.
- GetOpt::Long: used to process command-line arguments.
- HTTP::Request: used to retrieve data from EMBL (optional).
- IO::File: used to write preferences with XML::Writer.
- LWP::UserAgent: used to retrieve data from EMBL (optional).