MuGeN: simultaneous exploration of multiple genomes and computer analysis results

Mark Hoebeke¹,*, Pierre Nicolas¹, ² and Philippe Bessières¹

¹Unité Mathématique, Informatique & Génome - INRA - Domaine de Vilvert - F 78352 Jouy en Josas CEDEX, France and ²Laboratoire de Statistique et Génome, CNRS - Tour Évry2, 523 place des terrasses de l’Agora - F 91034 Évry, France

Received on October 14, 2002; revised on December 18, 2002; accepted on January 10, 2003

ABSTRACT
Motivation: The availability of increasing amounts of sequence data about completely sequenced genomes spurs the development of new methods in the fields of automated annotation, and of comparative genomics. Tools allowing the visualization of results produced by analysis methods, superimposed on possibly annotated sequence data, and enabling synchronized navigation in multiple genomes, provide new means for interactive genome exploration. This kind of visual inspection can be used as a basis to assess the quality of new analysis algorithms, or to discover genome portions to be subjected to in-depth studies.

Results: We propose a software package, MuGeN, built for navigating through multiple annotated genomes. It is capable of retrieving annotated sequences in several formats, stored in local files, or available in databases over the network. From these, it then generates an interactive display, or an image file, in most common formats suitable for printing, further editing or integrating in Web pages. Genome maps may be mixed with computer analysis results loaded from XML files, whose format is generic enough to be adapted to a majority of sequence oriented analysis methods.

Availability: MuGeN is available at http://www-mig.jouy.inra.fr/bdsi/MuGeN

Contact: mark.hoebeke@jouy.inra.fr

Supplementary information: The supplementary information cited in this paper is available at http://www-mig.jouy.inra.fr/bdsi/MuGeN/Paper/supp.html

INTRODUCTION
Since the publication of the first complete microbial genome in 1995 (Fleischmann et al., 1995), sequencing programs have been producing complete genomes at a steady pace. As of July 2002, the NCBI genome repository (http://www.ncbi.nlm.nih.gov/PMGifs/Genomes/), offers access to 81 complete microbial genomes. Both sequence and annotation information are contained in text files well adapted to detailed examination of specific sequence features but ill-suited to rapidly compare two or more entries or to give a contextual view of a given genome portion.

Visual cross-examination of database entries is not only important when comparing genomes of more or less related organisms. It also allows some kind of information fusion when displaying several entries related to the same physical portion of DNA. For instance, figure S1 in the supplementary information section shows two GenBank entries covering an identical 7.5 kb portion of the genome of B.subtilis centered around the bnrdE and bnrdF genes belonging to the integrated bacteriophage SPBc2. Clearly, the entry describing SPBc2 carries remarkably more information than the one covering the whole genome of B.subtilis, which most notably lacks the introns for which experimental evidence is mentioned in the entry covering SPBc2.

Now, gene context could become a prevalent source of information (Tamames, 2001; Aravind, 2000; Huynen et al., 2000; Dandekar et al., 1998) as the sequencing of more closely related organisms is completed. However, strict conservation of gene order quickly diminishes with evolutionary time (Lathe,III et al., 2000) making it difficult to design algorithms determining conserved portions between a set of genomes. For any given gene, the visualization of its surrounding genes in several related genomes can aid to determine if it is part or not of a conserved portion.

Annotation of complete genomes is mainly performed with the help of bioinformatics tools, be it stand-alone packages like Glimmer (Delcher et al., 1999) or GeneMark.hmm (Lukashin and Borodovsky, 1998); or complete toolkits like the EMBOSS suite (Rice et al., 2000). The interpretation of results generated by these programs can be greatly enhanced if their graphical representation can be displayed directly superposed on the sequence, or on top of it.

*To whom correspondence should be addressed.
Finally, there is a need for the automatic generation of quality images representing annotated genomes mixed with analysis results. For large, feature-rich genomes it may be helpful to generate a series of pictures to be examined later on, each displaying a tractable portion of these genomes. Automatic generation also eases the design of Web-based genome displays, as it can be done by a CGI script based on a set of parameters defining the data to be visualized and entered in a fill-in form.

Several tools providing the various features listed above are already available, either as stand-alone browsing/editing tools like Artemis (Rutherford et al., 2000) and its companion tools ACT (http://www.sanger.ac.uk/Software/ACT), Apollo (http://www.ensembl.org/apollo) or the DAS BioJava client (Dowell et al., 2001), or integrated in Web sites like UCSC’s Human Genome Browser (Kent et al., 2002), Entrez’s Map Viewer (http://www.ncbi.nlm.nih.gov/PMGifs/Genomes/MapViewerHelp.html), Ensembl’s Genome Browser (http://www.ensembl.org/) or the Generic Genome Browser (Stein et al., 2002). However, none of these tools is usable both as a stand-alone program independent of any database and as a batch mode image generation tool easily embedded in Web pages capable of displaying multiple annotated genomes together with analysis result plots.

We have developed a software package, called MuGeN (for Multi Genome Navigator) that provides both an interactive annotated genome browser, mugen (for MuGeN visualization tool), and an image generator, mugenb (for MuGeN batch-mode tool). They share a major part of their functionalities, the former through a graphical user interface and the latter through command line options, and generate identical graphical representations. These representations can contain several annotated genome portions or whole genomes together with various analysis result plots.

MuGeN offers a range of features seemingly overlapping those of Genquire (Wilkinson et al., 2002). Both provide visualization modes with different levels of detail and retrieval of annotated genomes from diverse repositories. Genquire allows annotation editing, whereas MuGeN only allows visualization, but the former works on top of a database whereas the latter is independent of any underlying data repository. Moreover, MuGeN can be run in batch mode to generate series of annotated images, a feature not present in Genquire. Finally, mixing annotated genomes with plots resulting from analysis results, one of MuGeN’s strengths does not seem possible with the current release of Genquire (which does provide a plug-in mechanism allowing execution of analysis programs). Thus, although both tools may seem closely related they actually serve different purposes.

SYSTEM AND METHODS

Map Operations

MuGeN is capable of loading annotated genome data from local files in GenBank/EMBL, BsmI (Bioinformatic Sequence Markup Language: http://www.bsml.org) or even raw sequence data in FASTA format. It can also retrieve entries directly from GenBank or EMBL, given an access number, or from Mico (Biaudet et al., 1997), where whole genomes can be chosen from a list of available organisms. When several entries are used simultaneously, each can come from a different data source. A typical view of MuGeN’s graphical user interface is given in Figure 1.

The map list window (Fig. 1a) shows which maps are currently displayed and allows various map operations (i.e. loading of new maps, moving maps up and down, hiding/showing maps, flipping or duplicating maps, removing maps). The list of computer analysis results associated with the currently selected map is also shown. Moreover, the map list window enables the user to fix relative map positions by defining anchors, an anchor being either a gene name or an absolute base position. For example, if two maps contain supposedly related portions, these can be displayed on top of each other by setting their anchors to a common gene name. If the anchor is the name of a gene located on the reverse strand, the map will be automatically flipped. All anchored maps will thus have the same orientation relative to the anchor direction.

Map Display

The map display window (Fig. 1b) draws a graphical representation of all visible annotated maps with their computer analysis results. Each map is partitioned in strips of a user-defined length, and corresponding strips of each map are laid out consecutively from top to bottom (i.e. the first strip of the first map is located immediately above the first strip of the second map, and so forth). This favors easy comparison of related genome portions belonging to different maps, specially when each map has a different background color. By default, coding sequences are displayed in one of the six reading frames with other features located below, but frames and/or strands can be collapsed to generate more condensed views. Depending on the strip size, one of three view modes is used.

In the default view mode, most common feature types (CDSs, rRNAs, tRNAs, terminators, promoters, etc.) have a specific graphical representation. When the user positions the pointer inside a feature, the list of its qualifiers with the associated values are dynamically updated in the information window (see the next section). For features having qualifiers describing links to databases like SWISS-PROT (Boeckmann et al., 2003) or the Protein and Taxonomy databases at NCBI (Wheeler et al.,
Fig. 1. An overview of the different components of MuGeN’s graphical user interface. (a) The map list window lists the four loaded genome maps (Bacillus subtilis, B.halodurans, Staphylococcus aureus and Streptococcus pyogenes). For the currently selected map, B.halodurans, the list of computer analysis results is shown. All four maps are anchored on the secY gene, causing the flipping of the map of S.aureus, secY being located on the reverse strand in this genome. (b) The map drawing window. Each of the four maps is displayed on a horizontal strip with a distinct background. Colors for CDSs are determined by their functional categories defined for B.subtilis (Kunst et al., 1997). Below each of the genomes of B.halodurans, S.aureus and S.pyogenes, box plots show the corresponding portions of B.subtilis. Green stretches denote potential syntenies located on the same strand in both genomes, and red stretches denote potential syntenies of the opposite strand. Green and red boxes show orthologs, orange and blue boxes show homologs. Grey boxes stand for genes with no significant similarity. (c) Information window. The top half shows the list of highlights, in this case all conserved regions. The bottom half displays the qualifiers and their values for the feature currently under the mouse pointer. See Supplementary data for colour figure.
(2000), the user can click on the feature to choose a link in a popup-menu which will open the relevant page in a Web browser. When the strip size drops below a given threshold, the view mode changes to display the actual nucleotide sequence on both strands and their translations in all reading frames. A possible usage of this view mode is to check the correct positioning of start and stop codons, or other specific motifs. At the opposite, when the strip size increases above a second threshold, the default view mode becomes a bird’s eye view mode in which all features are drawn as simple boxes, and don’t react to mouse pointer movements or actions anymore. This mode is useful for generating comprehensive views of complete genomes and can give an idea of the distribution of its different features.

A set of controls at the bottom of the window provide navigation operators to move through the maps, to zoom in or out, to go directly to a given position or to fix the thresholds for switching between the three view modes. In the same manner, menus allow to export the currently displayed maps in XFig (for editing with the XFig drawing program), Postscript (encapsulated or not, for quality printing) and PNG (for inclusion in Web pages) formats, to choose which features to hide or to display, to expand or to collapse strands and/or frames, and to fix the width of the graphics area.

Feature Qualifiers and Highlights

In default view mode, when the mouse is positioned over a feature, all of its qualifiers and their associated values are displayed in the information window (Fig. 1c). This window also provides a list of all ‘highlights’ defined for each loaded computational analysis result. Selecting an item in this list focuses the graphical window on the area defined by the highlight.

Computational Analysis Results

A major goal of MuGeN is to ease the task of drawing a large variety of computational analysis results mixed with annotated data. MuGeN’s design (see the IMPLEMENTATION section below) allows this by specializing existing result drawing packages, but it comes with built-in drawing functionalities for the most common of analysis types, namely box plots and line plots. Box plots cover functionalities for the most common of analysis types, such as ORF finding programs, operon detection algorithms and others. Line plots are adapted to methods yielding one or more data points per base, like GC content computations, or representations of hidden Markov model states (as illustrated in Nicolas et al., 2002). Samples of both types of plots are given in figure S2 in the supplementary information section.

To keep the data format as universal as possible, we adopted the XML standard for storing results (for XML applications in bioinformatics, see Achard et al., 2001). As XML is straightforward to produce, either directly, or with the help of widespread modules or libraries, parsing the output of analysis programs to generate well-formed XML files is a trivial task. Another benefit of using XML files instead of coupling MuGeN to specific analysis software formats is that any changes in these formats don’t require modifications in MuGeN itself, but only in the results parser.

Hence, we defined a Document Type Definition or DTD (see listing 1) adapted to the description of box- and line plots. Apart from the plot data itself, this DTD allows the definition of new colors and of highlights corresponding to specific portions of the plots. Note that one XML file is capable of storing multiple line and box plots. Precise positioning of plots is possible through the use of attributes: plots can be drawn on separate strips or mixed with features, with the type attribute. For box plots, every box can be precisely positioned in one or all reading frames of a given strand, and its graphical properties (color, thickness, filled or not) can be set separately (see listing S1, supplementary information). For line plots, a step value of \( n \) can be used for results yielding data points every \( n \) bases. Display speed can also be increased by setting a smoothing threshold: points having values closer than this threshold will be joined by horizontal lines instead of polygonal lines.

Batch Mode Operation

All functionalities described above, with the exception of the dynamical display of feature qualifiers and values, and
highlight focusing are available as well in the command line tool. It offers an easy method for automated creation of series of images in the above mentioned formats of annotated genome maps. Moreover, it is capable of generating images in IMAP format, for inclusion in Web pages. When this mode is chosen, MuGeN accepts an option specifying a root URL which will be used to generate a client side image map. The HTML code of this map is then printed on standard output and is ready for inclusion in Web pages. Each feature of the map will be a clickable element pointing to the root URL suffixed with MuGeN specific parameters: the type of the feature and its name (for CDSs), start and stop position. Listing 3 in the supplementary information gives an example of the HTML map information generated by MuGeN when given http://machine.org/bioinfo/genmap.cgi?access=Z99116 and as root URL.

IMPLEMENTATION

Design Guidelines

MuGeN was designed according to object-oriented principles. This resulted in a highly modular and component-based architecture. Figure S3 in the supplementary information section depicts the architecture of the application’s main classes. Each major functionality is provided by an abstract class, whose specialized subclasses implement an appropriate behavior. This is a classical approach for the graphical components, like the FeatureWidget class and its descendents, which allows easy addition of new classes. For instance, a specific representation for a new type of ‘scalable’ feature can be obtained by deriving a new class, either from ScalableFeatureWidget if a completely different look is chosen, or, even more easily, by deriving a new class from ArrowFeatureWidget or SpearFeatureWidget having new color and fill properties. A widget factory, based on the abstract factory design pattern (Gamma et al., 1995), then instantiates feature widgets as needed according to their precise type.

The support of a wide variety of output formats is also built by specializing classes tailored to each format from the generic PaintDevice class. This class provides methods to draw graphics primitives (i.e. lines, circles, polygons, text) on a normalized drawing surface. Each type of output is implemented by a corresponding subclass able to render these primitives following the specifications of the given format. The same holds for the FeatureDataSource class and its subclasses, responsible for feeding the application with features loaded locally or remotely. Regarding the CompAnalysisResults class hierarchy, an instance of the appropriate subclass is created, by a ‘computer analysis results factory’ when the XML parser encounters the tag describing the type of analysis result. Hence, adding new types of analysis results amounts to implementing a class handling the results and registering it in the results factory.

Software components

MuGeN is made of software modules written in the Perl programming language. This choice was made because of the availability of Bioperl (Stajich et al., 2002 http://bio.perl.org) a robust set of Perl modules dedicated to bioinformatics. Of paramount importance was the fact that Bioperl offered high-quality parsers for the main file formats used to represent annotated sequences (GenBank/EMBL, and later on BsmI). MuGeN relies on these components to build its internal feature representations. Moreover, the widely adopted Gtk widget set (http://www.gtk.org) could be accessed in Perl scripts through a Gtk-Perl module (http://www.gtkperl.org), providing all the functionality to build the graphical user interface.

MuGeN runs on all platforms supporting the Perl programming language and its additional modules. It has been successfully tested on the Linux and Solaris operating systems, and is available under an open source license allowing it to be extended and enhanced by any interested party as long as the modifications are also publicly released. Porting MuGeN to Windows or MacOS X is not being considered for now. Nevertheless, as MacOS X shares many features of a UNIX operating system, is capable of running an X Window server as well as the Gtk and Bioperl toolkits, porting MuGeN to MacOS X could be not too difficult.

CONCLUSION

With MuGeN, we provide a framework for the exploration of multiple annotated genomes completed by computer analysis results, both as a user friendly interactive application, and as a command-line driven batch script. To date, it has been used during the mining of the chromosome of B. subtilis (Nicolas et al., 2002) and is integrated in the upcoming release of the Web interface of the Micado database (http://www-mig.jouy.inra.fr/bdsi/Micado). It is also used to visualize syntenies in microbial species (unpublished work).

For the time being, computational results can only be loaded from the local filesystem, which is well adapted to large data sets and to analysis methods requiring an execution time incompatible with an interactive tool. However, for other analysis methods it seems desirable to extend MuGeN with the necessary functionality to retrieve results from remote servers, or even to launch analyses automatically. With the emergence of new protocols for inter-application data exchange like the XML Protocol (formerly known as SOAP, http://www.w3c.org/2000/xp/Group), the technicalities of remote execution and data
exchange should not be difficult to overcome. What is still needed is a standardized way of describing such analysis services and the semantics of the data they generate. Although promising initiatives are emerging (see for example BioMOBY at http://www.biomboby.org), they still need to mature before gaining wide acceptance. In the same way, retrieval of annotation data from specific databases (like Micado) needs some additional code to be added to MuGeN. As more and more sequence databases will comply with the BioSQL schema, or provide access through an Open Bioinformatics Database Access (http://obda.open-bio.org) layer, their integration into MuGeN will be made easier, as these resources will be directly addressable through Bioperl. Nevertheless, extending MuGeN to interface it with new databases or analysis result repositories will not tie the application to any one of them, keeping it completely stand-alone. By distributing MuGeN under license terms compliant with the Open Source Definition (http://www.opensource.org/docs/definition.php), we allow any interested developers to modify and/or extend the software, which, in our view, will contribute to improve its quality and usefulness.

**SUPPLEMENTARY DATA**

For Supplementary data, please refer to *Bioinformatics* online.

**REFERENCES**


