CABIOS Vol. 10 no 6 1994
Pages 671–675

The genetic data environment an expandable GUI for multiple sequence analysis

S.W.Smith, R.Overbeek[1], C.R.Woese[2], W.Gilbert[3] and P.M.Gillevet[4,5]

Abstract

An X-Windows-based graphic user interface is presented which allows the seamless integration of numerous existing biomolecular programs into a single analysis environment. This environment is based on a core multiple sequence editor that is linked to external programs by a user-expandable menu system and is supported on Sun and DEC workstations. There is no limitation to the number of external functions that can be linked to the interface. The length and number of sequences that can be handled are limited only by the size of virtual memory present on the workstation. The sequence data itself is used as the reference point from which analysis is done, and scalable graphic views are supported. It is suggested that future software development utilizing this expandable, user-defined menu system and the I/O linkage of external programs will allow biologists to easily integrate expertise from disparate fields into a single environment.

Introduction

The field of computational molecular biology involves the analysis of a broad spectrum of biological data using various algorithms for molecular modeling, comparative and phylogenetic analysis, database management, genetic mapping and sequence analysis. New programs and algorithms for the analysis of these types of genetic data are constantly being developed using various computer languages, interface methods and file formats. As such, these programs require a wide range of computer expertise to use.

The best and most useful algorithms are often incorporated into commercial software packages for use by the biological community (see Roe 1988, and Ahern 1991, for reviews). These packages tend to add a defined, common user interface to a set of programs in order to make them more palatable to end-users. During this process, programs are often converted from one programming language to another. File formats may also require conversion. The end result of these conversions are user-friendly, although sometimes costly systems for genetic data analysis. The primary drawback to these commercial packages is the time it may take for novel analyses to be incorporated into the systems. Successful algorithms developed today take months if not years to find their way into commercial packages. Because of this, the average researcher does not immediately benefit from these novel algorithms. Furthermore, in many instances it is impossible to expand these systems with custom-built algorithms as the source code for these packages may not be freely distributed.

Approaches have been suggested to incorporate all types of genetic data analysis in a single generic environment for computational biology. In recent years, there has been an explosion of small public domain packages on both workstations and PC based systems, attempts have been made to produce comprehensive GUI system (Douthart *et al.*, 1986), and several DCL based navigational systems have been built (Kanehisa, 1982; Reisner and Bucholtz, 1986; B.Roe, pers.comm; Modelevsky and Akers, 1988). None of these systems have been developed to the extent that they are presently utilized universally by the general biological community.

DNA and protein sequence analysis are arguably the most common forms of computer analysis used by the biological community and most current methods involve the comparison of two or more sequences at one time. Several multiple-sequence editors have been developed (Dear and Staden, 1991; Clark, 1992; Jurka, 1987; George and Barker, 1986; Devereux *et al.*, 1984; Barbar and Maizel, 1990) that are used for searching for sequence motifs; others have been developed for phylogenetic analysis (G.Olsen and T. Macke pers. comm.). As they were tailored for a specific field of interest, they are difficult to expand without significant modification to the main program itself.

A versatile system for genetic data analysis should incorporate a simple to use interface as well as allow the addition of novel analytical tools. Thus the key goals to any general-purpose analysis environment should be ease of use and ease of expansion. We present an expandable

Millipore BioImage, Ann Arbor, MI, [1]Mathematics and Computer Science Division, Argonne National Laboratory, 9000 S. Cass Ave. Argonne, IL 60439, [2]Department of Microbiology, University of Illinois, 407 S.Goodwin Ave. Urbana IL. 61801 [3]Harvard Genome Laboratory, Harvard University, 16 Divinity Ave. Cambridge, MA 12138 [4]National Center for Human Genome Research, National Institutes of Health, Bethesda, MD 20892 USA

[5]*To whom reprint requests should be sent*

graphic user interface and demonstrate its utility with a multiple sequence editor called the Genetic Data Environment (GDE). The GDE is a modular software environment that will enable the user community to immediately benefit from the latest developments in computational biology and allow them to customize a system to meet their specific needs.

Design goals

GDE was designed to remove three bottlenecks encountered in the integration of sequence analysis software. The first goal was the development of a flexible system that would allow the incorporation of algorithms written in various programming languages. Existing programs in languages such as FORTRAN, Pascal, C, BASIC, Lisp, and PROLOG are generally rewritten in a common language before they are incorporated under a single user interface. This requires a significant amount of code writing, with the net result being a system that does little more than the components did. Any truly integrated system must accommodate the use of a mixture of languages in order to avoid constant rewrites of working software and allow this existing body of analytical software to be incorporated into the system with minimal effort.

The second goal to be addressed was the development of a graphical system for DNA and protein sequence display and manipulation. There is little argument over the fact that the menu and mouse style interface greatly decreases the learning curve associated with sequence analysis software. However, the building of such interfaces is of little interest to most programmers as these interfaces do not improve the quality of the actual algorithm, but merely simplifying its use by others. Thus, any integrated system for analysis must take responsibility for much of the graphic user interface and the representation of various views of the data.

The third goal addressed was the development of the hooks to accommodate end-user expandability. If an individual wishes to incorporate his own analysis into the system, he or she should not be required to modify, or even have access to the source code for the environment. It should instead be possible to describe to the environment how it should access a new program, and the environment should handle all details of executing this new function and the presentation of the analysis to the end-user. This would then allow a user to customize the environment to suit each particular user's needs.

By meeting these design goals, we hoped to minimize the work required to move a program from the programmer's test-bed to the end-users hands. In the process, we would produce an ever-expanding system for biomolecular sequence analysis in which all could benefit from other's expertise in various fields. It would benefit software developers to remain compatible with the system, because the requirements of compatibility are minimal, and the benefits of compatibility are great. It is hoped that this paradigm for user interfaces will prompt programmers to establish a common repository where external programs are accumulated and freely shared within the user community and that this repository be expanded to encompass all areas of computational biology.

Implementation

We divided the environment into three primary parts (see Figure 1.) to meet these goals. They are data representation and manipulation, an expandable user defined menu system, and external program execution. Data representation is the graphic display and editing of the data. In the case of protein and nucleic acid sequence data, the representation follows along the lines of a text editor for multiple sequences. The emphasis of sequence display is always on flexibility of representation and manipulation. Color highlighting is a modifiable trait, not a fixed scheme so that analysis functions can change color attributes as easily as they change actual sequence data. Alignment specific editing functions such as group insertion/deletion and data locking are needed.

User-interface issues are handled by a simple menu definition file which describes the external analysis

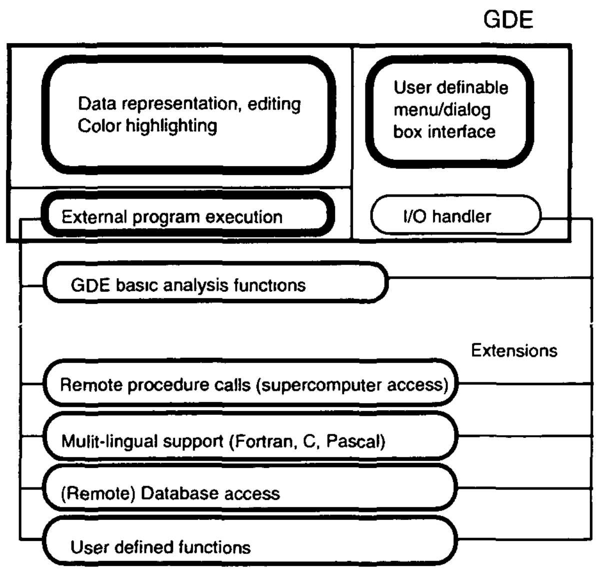
GDE
Data representation, editing
Color highlighting
User definable
menu/dialog
box interface
External program execution
I/O handler
GDE basic analysis functions
Extensions
Remote procedure calls (supercomputer access)
Mulit-lingual support (Fortran, C, Pascal)
(Remote) Database access
User defined functions

Fig. 1. Depicts the logical organization of the Genetic Data Environment The system consists of an internal data representation module, a module that interprets the expandable menu configuration file and a module the executes externally defined programs. These external extensions include basic analytical functions distributed with GDE and any function that that can be run with command line arguments.

function in terms of what raw data it requires, what parameters it uses, and what form the returned results will take. By interpreting this file at run time instead of compile time, each individual user at a site can have his or her own customized setup. This definition file uses a simple language that allows easy modification by end-users.

In summary, the environment retrieves user input, passes required data in required formats, executes all needed analysis programs, and returns all results to the user. Thus the external analysis programs appear to be part of one integrated system with all behind-the-scenes file conversion, parameter passing, and file cleanup being handled automatically without the intervention of the user.

The prototype system

The prototype implementation of the above paradigm is the Genetic Data Environment (GDE) which represents amino acid and DNA sequence data in a multiple-sequence alignment format. The primary display/editor and menu system was written using the MIT X Windows system under the XView user interface toolkit developed by Sun Microsystems. This greatly improved the esthetic quality of the interface, as well as speeding its development. We are currently running the system on Sun workstations using the Unix operating system, the flexibility of which was essential to the development of the system.

The display editor accepts five types of sequence information: nucleic acid sequence, amino acid sequence, text, sequence masks, and color mask information. It allows for sequence entry/editing under a four stage protection scheme which prevents accidental data corruption by allowing or restricting alignment gap modification, ambiguous character modification, standard character modification, and sequence translation/reversal. Once a set of sequences is aligned, they can be grouped so that modifications in one propagate to all, and sequences or sub-sequences can be duplicated, removed, moved, translated, complemented, and reversed.

The editor supports five coloring schemes: a monochrome mode, color by sequencing direction, a character to color mapping, an alignment wide color mask, and a sequence by sequence color mask. The character-to-color mapping aids in visual evaluation of alignments where each nucleic acid character is mapped to one color. Amino acid characters are grouped into seven categories based on size and charge. The alignment wide color mask gives each column of the alignment a specific color which is useful in representing position by position scoring of alignments. The sequence color mask will give a position by position color coding for a given sequence which is designed for representing position by position scoring on individual sequence. Other features of the primary display editor include the ability to do 'split screen' editing, tactile feedback, a checking mode, reduce scale view, variable font sizes, and extended sequence information annotation.

Expandable menu system

External functions are tied in by means of the menu configuration file 'GDEmenus'. This file resides in a shared help directory, or can also be placed in an individual user's home directory for easier customization. This configuration file defines how to access remote functions and how to present 'dialog boxes' to the user. It specifies the data format for external programs as well as the format for data that the functions might pass back to GDE.

The description language for the expandable menu file is fairly simple; an example of how one might tie in a hypothetical Unix program called db_search which performs a database search and retrieval follows. Assume the command line for this program is

```
db_search -base DataBankName -field Field Keyword
```

where DataBankName might be one of GENBANK, PIR, or EMBL; Field might be one of AUTHOR, PUB (for publication) or DESC (for description); Keyword would be some descriptive text for the chosen search field. Figure 2 depicts how such a function would be described in the .GDEmenus configuration file and this would create a new menu item labeled 'Data base search' (see Figure 3). Once the user clicks the OK button, program db_search is run with the user's chosen parameters, the results are written out to a temporary file, and the sequences contained in that file are loaded into the GDE editing window.

Sequence data in the editor may also be passed as input to an external function by adding a field to the configuration file describing the input format. The user selects sequences by highlighting the sequence names before selecting a menu function, and the selected data are then written out to a temporary file in the appropriate format before being passed as input to the external program. The menu file also allows for handing out data in specific regions by means of a selection mask, as well as controlling temporary file cleanup after a program executes.

The second release of GDE has several external analysis functions tied in through its expandable menu system and are distributed with the core editor. Most of these programs were incorporated with no changes made to the original source code and include programs for multiple sequence alignment (Higgins, 1992), similarity search

```
item:Data base search
itemmethod:db_search -bank $BANK -field $FIELD $KEYWORD > OutputFile

arg:BANK
argtype:chooser
arglabel:Which database to search
argchoice:Genbank:GENBANK
argchoice:PIR:PIR
argchoice:EMBL:EMBL

arg:FIELD
argtype:choice_menu
arglabel:What field to search on?
argchoice:Author:AUTHOR
argchoice:Publication:PUB
argchoice:Description:DESC

arg:KEYWORD
argtype:text
arglabel:Keyword to search for?

out:Outputfile
outformat:genbank
```

Fig. 2. is an example of a menu configuration definition hooking an external function into GDE. The pop up window label is defined by 'item:' and the external Unix command line that is executed is defined by 'itemmethod:' Command line arguments are defined in each of the 'arg:' sections and these values are substituted into the 'itemmethod:' upon its execution. The results of the database search are output in genbank format which is defined in the 'out:' section.

(Altschul *et al.*, 1992; Pearson and Lipman, 1988), contig assembly (X.Huang, submitted for publication), phylogenetic analysis (Felsenstein, 1989; De Soete, 1984; M.Maciukenas, pers. comm.), RNA secondary structure (Zuker, 1989), and other custom-built functions useful in comparative analysis.

Systems requirements

The latest release of the system, GDE 2.2a, currently runs on workstations from Sun Microsystems including Sun 3, Sun 4, and the SparcStation line of workstations. System software requirements are SunOS 4.1 or later, Openwindows 2.0, or the X11R4 window system from MIT along with the XView 2.0 toolkit. We have also implemented the system on the DEC station from Digital Equipment Corporation, and informal ports of GDE have been accomplished on the Silicon Graphics Iris as well as to the Cray X-MP. The bundled system for GDE 2.2a is

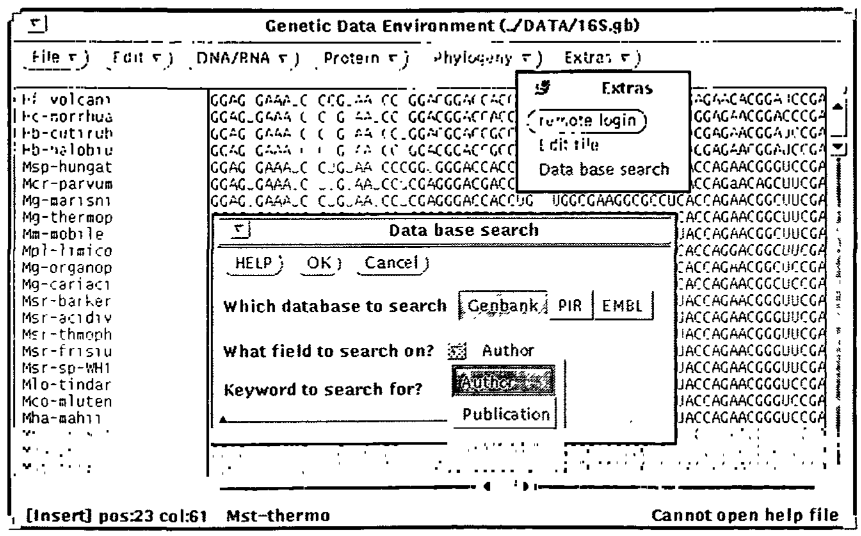

Fig. 3. shows the pop up menu created by GDE after interpreting the entry in the menu configuration that is described in Figure 2.

available to internet users via anonymous ftp from megasun.bch.umontreal.ca.

Discussion

The Genetic Data Environment was originally conceived as a phylogenetic tool at the University of Illinois and incorporated many salient features of previously developed multiple sequence editors (G.Olsen and T.Macke, pers comm.). The need for an expandable system became obvious very early in the effort and quickly evolved into GDE 1.0, first released in February 1991. This version of the editor had an expandable menu system and integrated external programs through input an output files. The base editor was subsequently expanded (GDE 2.0) and released in May 1992. This version included sparse matrix storage, cut and paste, and a flat database using tagged fields that is attached to the sequence entry. The latest release of GDE (2.2a) corrects the selection mask handling and I/O where leading characters in an alignment were sometimes lost. The XView version of GDE is presently being ported to Motif which will include Interprocess Communication support.

Although many software packages have been developed in the computational biology field, very few have considered the issue of user expandability. The Biotechnology Computing Environment (Modelevsky and Akers, 1988) was one of the first attempts to produce a system where the end-user can re-configure the menu system, in this case from within the command line environment using DCL commands.

We have recently extended the X-Windows based GDE system and built a terminal emulator (EZshelltool) using the same menu based paradigm. This latter system is used as a Navigator to lead investigators through numerous software packages and process control systems and is used extensively in the Mycoplasma Genome Project. We incorporated several database features and tools into GDE to handle data from the Mycoplasma project at the Harvard Genome Lab (Smith *et al.*, 1992). This model has proven extremely flexible allowing direct access to remote procedure calls on high-performance computers and network-based servers and can be easily modified on a daily basis as the need arises. The latter point is critical in when one is developing new technology and cannot predict *a priori* the needs of the system (Gillevet, 1993). Presently, there is no mechanism to build the menus automatically although this functionality will be implemented in the future.

In conclusion, as the technical revolutions in DNA mapping and sequencing technology are setting the stage for the complete analysis of entire genomes, it is imperative that an evolutionary perspective be established to provide a systematic and coherent framework for the comparative analysis of such data. The initial tool required for comparative analysis is an expandable multiple sequence editor and, as such, GDE represents the first step in developing this framework. It is hoped that the paradigm outlined for the integration of multiple sequence analytical tools into a common graphic user interfaces can be extended to other views and perspectives of biology, allowing the development of an encompassing environment for computational biology.

Acknowledgments

The authors thank Jordan Konisky for the initial support of this effort at the Center for Prokaryotic Genome Analysis at the University of Illinois and to Gary Olsen for his initial suggestions on system requirements for Multiple Sequence Editors. We would also like to thank Don Gilbert of Indiana University, Mike Maciukenas at the University of Illinois, and Chunwei Wang at the Harvard Genome Laboratory for their programming efforts and contributions. Carl Woese is supported in part by a Ribosomal Database Project grant from NSF DIR 891–17863 at the University of Illinois. Further development of this interface is being supported by the Harvard Genome Laboratory, Harvard University which is funded by NIH grant R01 HG00124.

References

Ahern,K. (1991) *Genetic Engineering News* January.
Altschul,S.F., Gish,W., Miller,W., Myers,E.W. and Lipman,D. J. (1990) *J. Mol. Biol.* **215** 403–410.
Barbar,A.M. and Maizel,J.V. Jr (1990) *Gene Anal. Technol. Applic.* 7, 39–45.
Clark,S.P. (1992) *Comput Applic. Biosci.*, *8*: 535–538.
De Soete,G. (1983) A least squares algorithm for fitting additive trees to proximity data. *Psychometrika*, **48,** 621–626.
Dear,S. and Staden,R.. (1991) *Nucleic Acids Res.*, **19,** 3907–3911
Devereux,J., Haeberli,P. and Smithies,O. (1984) *Nucleic Acids Res.* **12,** 387–395.
Douthart,R.J., Thomas,J.J., Rossier,S.D., Schmatz,J.E. and West,J W. (1986) *Nucleic Acids Res* **14,** 285–297.
Felsenstein,J. (1989) *Cladistics*, **5,** 164–166.
George,D.G. and Barker,W.C. (1986) *Macromolecules, Genes and Computers.* MBCRR of the Dana Farber Cancer Institute, Harvard University and GENBANK-Los Alamos National Laboratory. Proceedings from the Los Alamos National Laboratory.
Gillevet,P.M. (1993) Integration of the wet lab and dataflow in multiplex genomic walking. In: Lim,H.A. (ed.) *Proceedings of the Second International Conference of Bioinformatics, Supercomputing and Complex Genome Analysis.* World Scientific Publishing Co. River Edge, NJ.
Higgins,G. (1992) *Comput. Applic. Biosci.*, **8,** 15–22.
Jurka,J. (1987) The prokaryotic-eukaryotic interface. In: Morowitz,H.J. and Smith,T. (eds) *Report of the Matrix of Biological Knowledge Workshop.* Santa Fe Institute, Santa Fe, NM.
Kanehisa,M. (1982) *Nucleic Acids Res.* **10,**183–96.
Modelevsky,J.L. and Akers,T.G. (1988) *Comput. Applic. Biosci.*, **4,** 253–257
Pearson,W.R. and Lipman,D.J. (1988) *Proc. Natl Acad. Sci. USA*, **85,** 2444–2448.
Reisner,A.H. and Bucholtz,C.A (1986) *Nucleic Acids Res.* **14,** 233–238.
Roe,B. (1988) *Biotechniques*, **6,** 560–562.
Smith,S.W., Wang,C., Gillevet,P M. and Gilbert,W. (1992) *Genetic Data Environment and the Harvard Genome Database. Genome Mapping and Sequencing.* Cold Spring Harbor Laboratory Press, Cold Spring Harbor.
Zuker,M. (1989) *Science*, **244,** 48–52.