Copyright Notice

The Genetic Data Environment (GDE) software and documentation are not in the public domain. Portions of this code are owned and copyrighted by the The Board of Trustees of the University of Illinois and by Steven Smith at the Harvard Genome Laboratory. This release of the GDE program and documentation may not be sold, or incorporated into a commercial product, in whole or in part without the expressed written consent of the University of Illinois and of its author, Steven Smith.

All interested parties may redistribute the GDE as long as all copies are accompanied by this documentation, and all copyright notices remain intact. Parties interested in redistribution must do so on a non-profit basis, charging only for cost of media. Modifications to the GDE core editor should be forwarded to the author Steven Smith at the address given below for inclusion into future releases. External programs used by the GDE are copyrighted by, and are the property of their respective authors unless otherwise stated.

While all attempts have been made to insure the integrity of these programs:

Disclaimer

THE UNIVERSITY OF ILLINOIS, HARVARD UNIVERSITY AND THE AUTHOR, STEVEN SMITH GIVE NO WARRANTIES, EXPRESSED OR IMPLIED FOR THE SOFTWARE AND DOCUMENTATION PROVIDED, INCLUDING, BUT NOT LIMITED TO WARRANTY OF MERCHANTABILITY AND WARRANTY OF FITNESS FOR A PARTICULAR PURPOSE. User understands the software is a research tool for which no warranties as to capabilities or accuracy are made, and user accepts the software "as is." User assumes the entire risk as to the results and performance of the software and documentation. The above parties cannot be held liable for any direct, indirect, consequential or incidental damages with respect to any claim by user or any third party on account of, or arising from the use of software and associated materials. This disclaimer covers both the GDE code editor and all external programs used by the GDE.

Genetic Data Environment version 1.0

Table of Contents

Introduction....3
History and Design Overview....3
What can the GDE do?....4
System Requirements....5
Installing the GDE....5
Using the GDE....6
Data Types....9
Menu Functions
    File menu....10
    Edit menu....12
    DNA/RNA menu....12
External Functions....13
Bug reports/extensions....15
Acknowledgments....15
Appendix A, File Formats....16
Appendix B, Adding Functions....19
Appendix C, External functions....23

Introduction

The Genetic Data Environment is part of a growing set of programs for manipulating and analyzing "genetic" data. It differs in design from other analysis programs in that it is intended to be an expandable and customizable system, while still being easy to use.

There are a tremendous number of publicly available programs for sequence analysis. Many of these programs have found their way into commercial packages which incorporate them into integrated, easy to use systems. Tremendous effort is put into rewriting analysis functions into these systems. At the same time, these commercial packages charge substantial license fees as well.

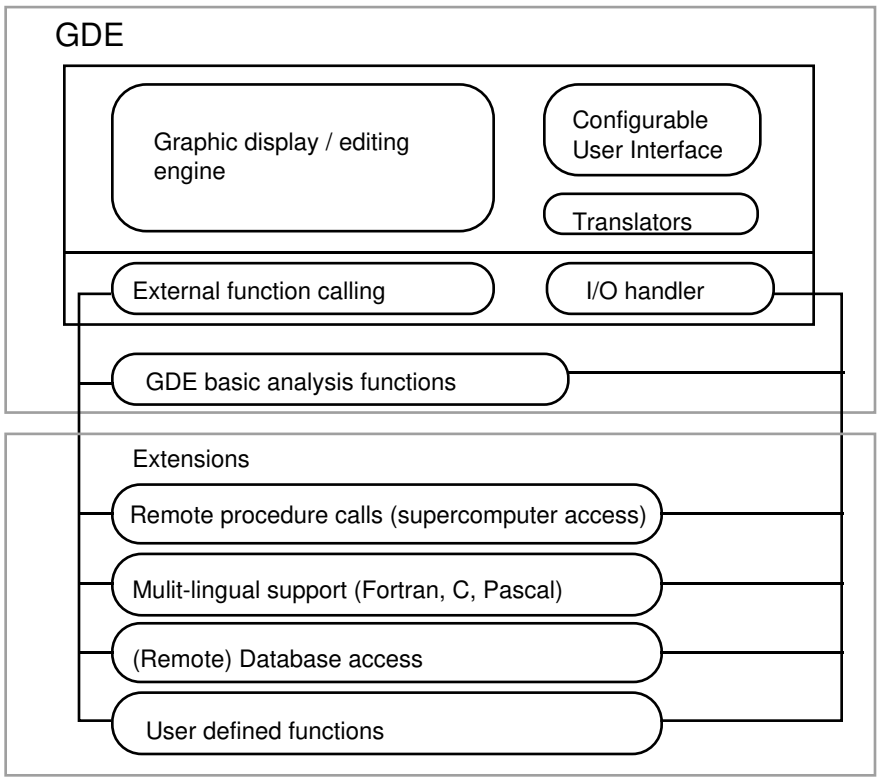
The goal of the GDE is to minimize the amount of effort required to integrate sequence analysis functions into a common environment. The GDE takes care of the user interface issues, and allows the programmer to concentrate on the analysis itself. Existing programs can be tied into the GDE in a matter of hors (or minutes) as apposed to days or weeks. Programs may be written in any language, and still seamlessly be incorporated into the GDE.

These programs are, and will continue to be, available at no charge. It is the hope that this system will grow in functionality as more and more people see the benefits of a modular analysis environment. Users are encouraged to make modifications to the system, and forward all changes and additions to Steven Smith at the address given below.

History and Design Overview

The basic ideas behind the GDE were formulated at the University of Illinois Department of Microbiology under the direction of Dr. Carl Woese. Woese, in collaboration with Dr. Gary Olsen of Urbana, and Dr. Ross Overbeek of Argonne National Laboratories embarked on the development of several software tools for comparative analysis of ribosomal RNA sequence data.

Several technical problems arose in attempting to set up the framework for the project. Interfaces needed to be built in order to allow biologists to test and evaluate new algorithms, and existing programs needed to be merged with new ones. Work on this project was going on both at Argonne and at Illinois. Because of this, programs written in diverse languages needed to be merged into coherent, and useable applications. The project also had to face the task of software revision control. Changes in one groups code would need to be merged with changes in the other's in a controlled manner. In an attempt to overcome these problems, an expandable analysis environment was needed.



The first part of the system is the interactive editor. It knows how to display and edit sequence data. It understands and can manipulate amino acid, nucleic acid, color and numeric sequence data. It also handles protections and grouping functions as well as display attributes.

The second aspect of the system is an expandable user interface. The interface understands how to set up the menus and dialog boxes for the analysis functions by reading a configuration file in the users home directory. This file specifies what menus and menu items are available to the user. It also specifies how to run each program associated with that menu item.

Finally, the system knows how to pass data to, execute, and read data from external programs. All of this is specified in the .GDEmenus configuration file mentioned above. The net result is an easy to use, and expandable system for analysis.

This design makes it easy to incorporate analysis programs in any language, running locally or remotely. It allows the use of shell scripts and basic unix commands, or any mix there of, in a seamless manner. All external functions appear to be part of a single system driven by the menus and mouse interface. Details about adding functions are given in appendix B.

What can the GDE do?

Release 1.0 can do the following:

The GDE allows you to enter and edit DNA, RNA, amino acid, mask and text sequence data. It can manage several sequences at once in a multiple sequence alignment. It allows the user to protect sequence data from modification on a sequence by sequence basis. It allows editing operations to occur on a number of sequences simultaneously. It allows complementation and translation of DNA/RNA sequence, and supports selection masks for selecting specific columns in the alignment for analysis. It can work in a monochrome

mode, or can allow three forms of color high-lighting. Font size can be varied as well as other display and editing attributes. Data can be written out in Genbank format and in GDE flat file format as well.

Sequences can be selected using the mouse, and can be passed off for external analysis. GDE 1.0 supports a multiple sequence alignment algorithm (clustal). It also supports a distance matrix calculator for nucleic acid sequences, and a phylogeny generator by DeSoete (lsadt). Also included are a few functions to demonstrate the color highlighting abilities of the package.

System Requirements

GDE 1.0 currently runs on the Sun family of workstations. This includes the Sun3 and Sun4 (Sparcstation) systems. It was written in XView, and runs on Suns using OpenWindows 2.0 or MIT's X Windows. It runs in both monochrome, and color, and can be run remotely on any system capable of running X Windows Release 4. You should have at least 5 meg of free disk space available.

The source code release minimally requires Xwindows release 4 with the XView libraries, and the OpenLook window manager (olwm) which are available from MIT (ftp to export.lcs.mit.edu, and log in as 'ftp'). The XView libraries are also available from SUN for the Decstation, the HP workstation, and the IBM RISC/6000. The XView libraries are also available for Apple Unix, A/UX on the Macintosh, and for VAX VMS. No attempts have been made at this time to compile the programs on these platforms. If anyone is successful in doing so, please forward any code changes to the address given below.

Installing the GDE

GDE 1.0 is distributed in three forms, full source code, dynamically linked binaries for Sun4's, and statically linked binaries for Sun4's. The primary difference between static and dynamic linking is that statically linked binaries are larger, but more stable on machines running older versions of SunOS (earlier than 4.1).

Instructions for the source code release are included in the README.install file.

The binary installations consist of creating a GDE directory, such as /usr/local/GDE, and un-taring the installation tarfile into the directory. If you are installing the GDE for your own use, then you can simply make a GDE subdirectory. There is no need to be superuser (root) to do the installation in your own directory. For example:

```
demo% mkdir  /usr/local/GDE
demo% cp  GDE1.0.tar  /usr/local/GDE
demo% cd  /usr/local/GDE
demo% tar  -xf  GDE1.0.tar
```

After this, each new user will need to run the add two lines to their .cshrc file so that they can find the gde programs and files.

```
demo% cat >> ~/.cshrc
set path = ($path /usr/local/GDE/bin)
setenv GDE_HELP_DIR /usr/local/GDE/help/
^D
```

You may wish to make a copy of the .GDEmenus file from the help directory into your home directory. This is only necessary if you wish to modify your menus. Copy the demo files from /usr/local/GDE/demo into your local directory, and you are now ready to use the GDE.

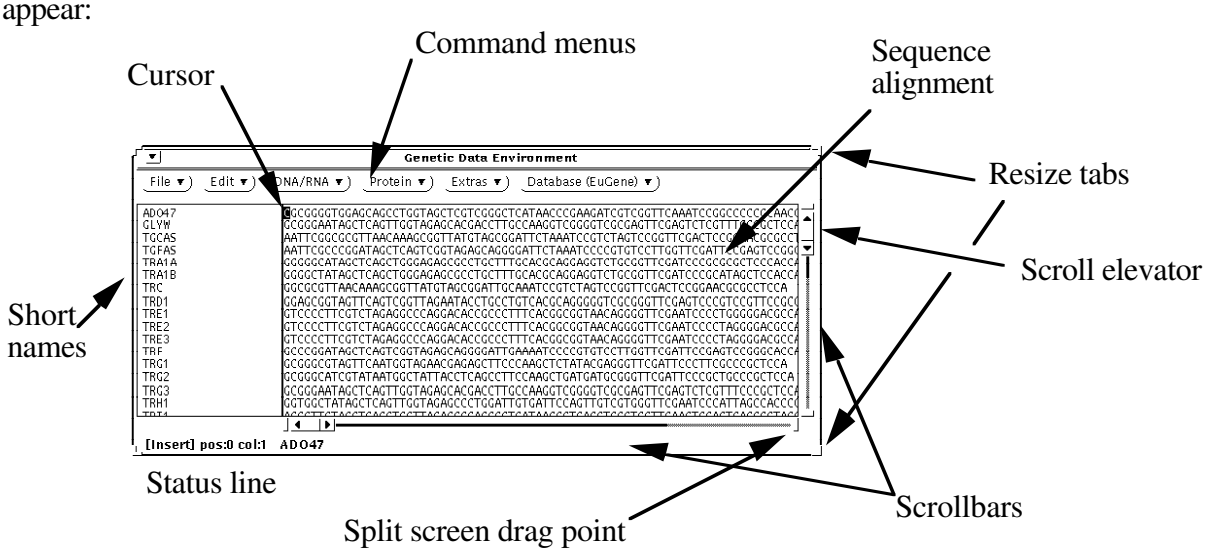
Using the GDE

It is assumed that the user is familiar with the Unix, and OpenWindows/Xwindows environments. It is also assumed that people running standard MIT X-Windows will be using the OpenLook window manager (olwm). Other window managers work with varied success. If you are not certain as to how your system is set up, please contact your systems administrator.

Once the window system has started, and a terminal window (xterm, shelltool etc.) you can start up the GDE by typing:

demo% **gde tRNAs**

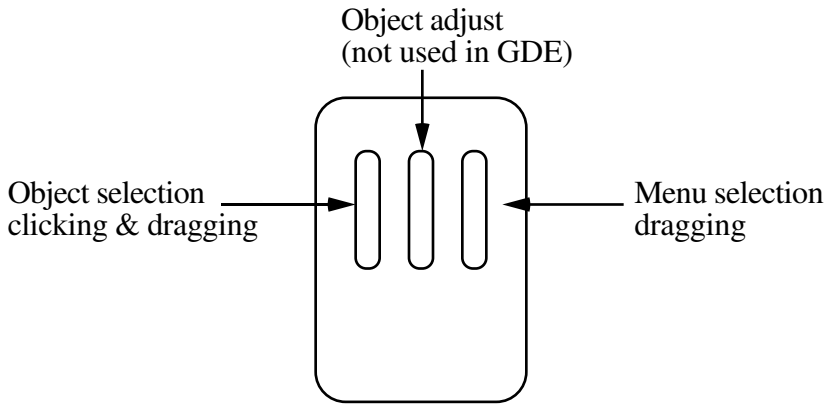
This should load the sample data set tRNAs into GDE, and the following window should appear:



This is the sequence alignment editor. It consists of a color alignment display, a set of command menus, horizontal and vertical scroll bars to navigate the alignment, a list of short sequence names (usually the LOCUS of a Genbank entry), and a status line. The cursor is located in the upper left corner.

Using the Mouse

The mouse follow OpenLook standards for operation. The functions for each button are:



The left mouse button is used for placing the cursor, selecting sequences by their short names, scrolling/paging, performing split screens, and resizing. The right button is used for pop up menus, and scrollbar menus. The middle button has no function within the GDE.

Cursor Movement

The cursor can be moved using the arrow keys, or by clicking the mouse within a sequence. The cursors position is displayed on the status line in both sequence position and alignment column number. The right hand side of the status line shows the left and right column positions of the currently active display.

Scrolling is controlled by the scrollbar elevator. By clicking (left mouse button) on one of the elevator arrows, the screen will scroll one character in that direction. By dragging the elevator center, the screen can be moved directly to any location. By clicking directly to one side of the elevator, the screen will scroll one full screen in that direction. And by clicking on the scrollbar anchor, the elevator will move to that anchor.

Scrollbars also have menus associated with them giving other scroll options. Use the right mouse button to activate the menu.

Selecting Sequences

Sequence selection is necessary before most functions can be performed. In the 1.0 release of the GDE, only entire sequences can be selected, and not sub-sections. This shortcoming will be fixed for the next release.

Selecting sequences is accomplished by clicking or dragging (left button) over the short name associated with the sequence(s). The name of the sequence should become highlighted on the release of the mouse button. By holding down the shift key, you can toggle the selection on or off for any set of sequences. By clicking just to the right of any sequence short name, you will deselect all of them.

Sequence Protection

All sequences can be individually protected against accidental modification. This is accomplished by selecting the set of sequences that you are interested in editing, and choosing the "Set protections" menu item under the File menu. Your choices are:

Unambiguous modification	Changing/adding/deleting regular characters
Ambiguous changes	Changing ambiguous codes ('N', 'X'...)
Alignment modifications	Changing alignment gaps ('-', '~')

Sequence Editing

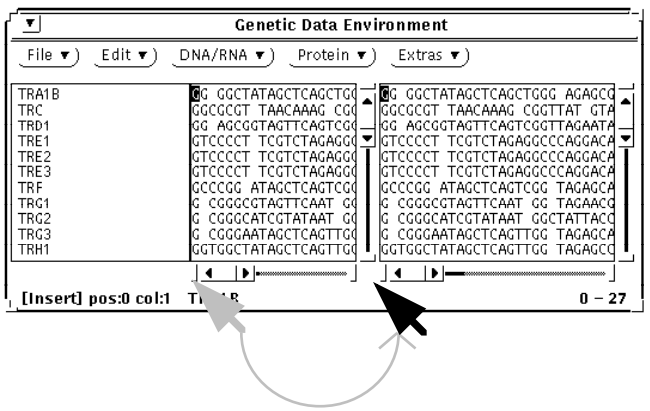
Sequences can be edited by simply typing to insert, and using the delete or backspace key to delete characters. Sequences must have the proper protections set to allow the type of modifications that you are attempting. The default protection level only allows modification to the alignment, but not to the sequences themselves.

Repeat Counts

By typing a numeric value before an editing function you can insert, delete or move a number of characters at a time. The current repeat count is displayed on the status line, and can be cleared by clicking the left mouse button in the alignment window. In order to insert twenty gaps into a sequence, one would type "20-". In order to move down five sequences, one would type "5↓". This works with all sequence types, however the meta (diamond) key must be held down when the cursor is in a text or mask sequence. This is because numbers are valid characters in these sequences, and would otherwise be confused with repeat counts.

Split Screen

The alignment window can be split horizontally into two or more windows into the alignment. These windows scroll independently of each other both horizontally and vertically. The short names displayed to the left of the alignment correspond to the window that was last scrolled or edited. Care should be taken in any modifications done in this mode so that edits are performed on the correct sequence. To avoid confusion during split screen operations, the vertical scrollbars may be locked so that all windows scroll together.



In order to split a window into two views, grab (left button) the left or right anchor (small rectangle) at either end of the horizontal scrollbar and drag to the middle of the window.

This should split the window into two views. To join two views, place the mouse pointer on the horizontal scroll bar use the menu (right button) .

The views are NOT two copies of the alignment. Changes in one window are reflected in the other. Users should not be confused by this fact.

Sequence Grouping

Sequences can be grouped for editing functions. This is very helpful when trying to adjust several sub alignments. When grouped, all sequences within a group will be affected by editing in any member of the group. All sequences within a group must have protections set to allow modification before any one will be modified.

In order to group sequences, select the names of the sequences that should fall within a group, and select Group under the Edit menu. A number will be placed at the left of the sequence representing its assigned group number. To ungroup any sequence or sequences, the user selects those sequences and uses the Ungroup command under the Edit menu.

Special keys

There are also a few special function keys used in the GDE. Some functions have meta key equivalences so that they can be called from the keyboard, instead of by the menu system. The "meta" key is a standard property of X windows, and may be remapped to a different key symbol for different keyboards. For example, meta on Sun workstations is represented with a ◊, where on a Macintosh running MacX it might be the "apple" key. The operation of the key is the same as the control or shift key, it is held down while pressing the second key in the sequence.

Meta 'c' will toggle the case of a character from upper to lower. Meta <number> will trigger the repeat count inside of text or mask sequences. Cut copy and paste are mapped to the Openlook equivalent keys (L10, L6, and L8 on Sun keyboards). Other meta keys are defined in the .GDEmenus file, and may be changed to suit your preferences.

Data Types

The GDE supports several data types. The data types supported in 1.0 are DNA, RNA, protein (single letter codes), mask sequence, and text.

DNA and RNA

Nucleic acid sequences are tightly type cast, and can contain any IUPAC code (ACGTUM RSVWYHKDBN) as well as two alignment gap characters ('~' and '-'). Some keys are remapped to fit IUPAC codes. For example, 'X' is mapped to 'N'. All nonstandard characters get mapped to the alignment gap '-'. Upper and lower case are both supported, and the T/U characters are mapped based on whether you are working with DNA or RNA. The color coding for DNA and RNA is identical. The color for ambiguous characters, and for alignment gaps is grey.

Amino Acid Sequence

Amino acid sequences are loosely type cast, and can contain any valid ASCII character. The results of analysis on nonstandard characters is not guaranteed. The color for nonstandard amino acid characters, and for alignment gaps is grey.

Text Sequence

Any valid ASCII printable character can be entered into a text sequence. Care should be taken with using space characters, as these will only be saved properly in Genbank format, and not in flat file format. The characters @#% and ' should be avoided as well, as these can confuse the reading of flat files if saved in that format. For future type extensions, the characters !$^& and " should be avoided as well.

Mask Sequence

Mask sequence is identical to text sequence with the following exceptions. Mask sequence can have the ability (function dependent) of masking out positions in an alignment for analysis. If a mask sequence is selected along with some other sequence(s) for an analysis function that permits masking, then all columns that contain a '0' in the mask sequence will be ignored by the function. The mask itself would not be passed to the analysis function either. Some functions allow masking, some do not. Refer to the instruction page for each function to see whether or not it supports sequence masking.

Color Masks

Color masks give color to a sequence on a position by position basis. Individual sequences can have color masks attached to them, or one color mask can be used for an entire alignment. Color masks are generated externally by some analysis functions, and are then passed back to the GDE. The file format for a colormask is described in Appendix A.

Menu Functions: File menu

The GDE has several built-in menu functions under the File and Edit menus. These functions are unique in that they are part of the primary display editor, and are not described in the .GDEmenus file.

Open...

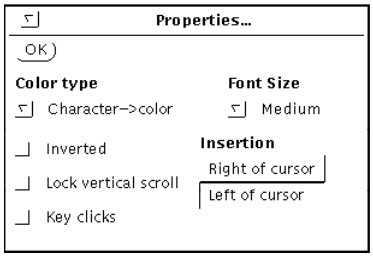
Selecting this will bring up the open file dialog box. Users can scroll through a list of files in the current directory, move up and down the directory tree, and open any individual data file. The sequence data in that file is loaded into the current editing window below any existing sequences. The open command will open any Genbank formatted file, or a GDE flat file.

Save as...

This function will save the entire alignment to a specified file in either Genbank or flat file format. The file will be saved in the local directory unless a relative or absolute path is specified.

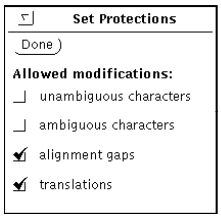
Properties...

Properties controls the display settings. Those settings include character size, color type, and insert direction. The screen can also be inverted, vertical scroll lock and keyboard clicks (tactile feedback) can be turned on or off. Vertical scrollbar lock will cause all split views to scroll together in the vertical direction.



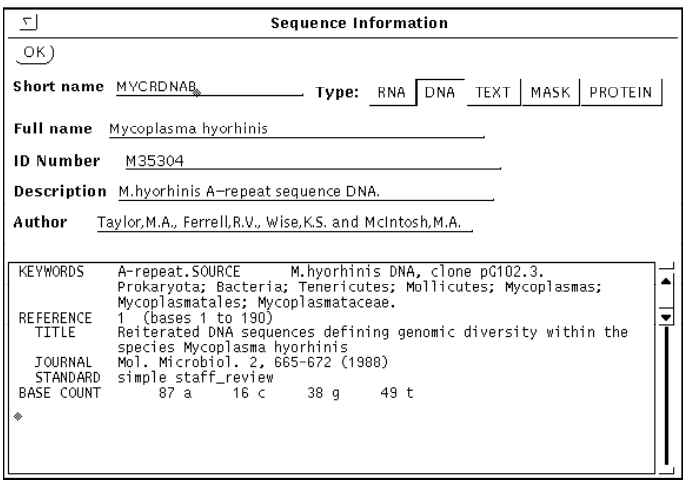
Set Protections

This will display, and then set the default protections for all selected sequences. If two or more of the sequences differ in their current protection settings, a warning message will appear in the protection dialog box. The protections currently available are alignment gap protection, ambiguous character protection, unambiguous character protection, and translation protection.



Get info...

This option allows the viewing and setting of attributes associated with each individual sequence. These attributes include short name, full name, description, author, comments, and the sequence type. Some sequence type conversions are not supported in the 1.0 release, and their use will have no effect. The attributes loosely correspond to fields in a Genbank entry. Comments can be included for each sequence in the comments field.



Edit menu

Select All

Selects all sequences. This is helpful when you have several dozen sequences.

Select by name...

Select all sequences containing a given string in their short names field. No wild cards are allowed, and only selecting is allowed, not de-selecting. The search is started when the Return key is pressed, and multiple searches can be accumulated. Press Done when finished.

Cut/Copy/Paste

Currently cut copy and paste are primarily useful for reordering sequences, and for making duplicate copies of a given sequence. They do not pass information to other programs. This capability will be implemented in a later release.

Cut and copy will place the selected sequences on an internal clipboard. They can then be pasted back into the top of editing window (default) or under the last selected sequence. The Cut, Copy and Paste keys on the SUN keyboard will also perform these functions.

Group/Ungroup

Assign a group number to the selected sequences. Edit operations in any one sequence within the group will be propagated to all within the group. Sequence protections from one group are also imposed upon all other sequence in the given group. If a given operation is illegal in one sequence in a group (i.e. alignment modification) then it will not work in any of the sequences in that group. Ungroup will remove the selected sequences from a given group.

Compress

Compress will remove all columns of dashes from the selected sequences. This function will leave the selected sequences aligned to each other, but may change their alignment to other nonselected sequences. The net result is that the selected sequences fall into a minimal alignment with no extra gaps. In order to remove all gaps from one sequence, select the sequence by itself and then Compress.

Reverse Sequence

Reverses the selected sequences. Alignment gaps are reversed as well. Care should be taken to pad the alignment out with gaps so that all sequences are the same length. If this is not done, then reversing the sequences can cause the alignment to be modified.

DNA/RNA menu

Complement Sequence

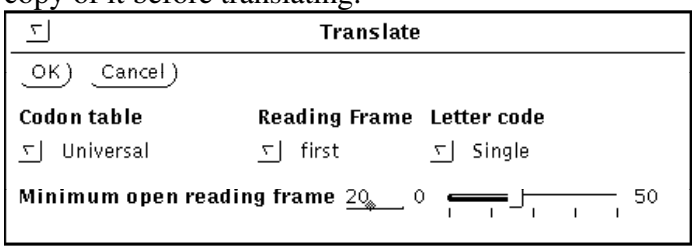
Converts DNA/RNA into its complement strand (keeping full IUPAC ambiguity). This function has no effect on text, protein, or mask sequence. Note that this function does not

produce the reverse strand of DNA but merely converts A<->T and G<->C. If the reverse strand is needed, remember to Complement and Reverse the sequence (Edit menu).

Translate...

Translate will convert the selected nucleic acid sequences to amino acid sequences in the chosen reading frame. You can choose between Universal, Mycoplasma, and Vertebrate mitocondrial codon tables, as well as specifying the minimum number of amino acids between stop codons. The translation can pass back single letter codes, as well as triple letter codes (in text).

The resulting translation will remain aligned to the original sequences, so you may want to use the Compress function (Edit menu) to remove extra alignment gaps. This function will overwrite your existing sequence with the translated sequence so you may want to make a copy of it before translating.



External Functions

See appendix C for a full description of functions supported in GDE 1.0 All external functions are described in the configuration file .GDEmenus. Here is a brief description of some of the basic functions included.

File menu

Read foreign format Save foreign format	Readseq (see Appendix C)
New Sequence <meta n>	Create a new sequence Prompts for sequence type, and short name.
Save Selection	Save the currently selected sequences in a specified file.
Print Selection	Print the selected sequences to the chosen printer.(.GDEmenus file may need to be modified to add the names of local printers to the printer list)

Edit menu

Extract	Extract the selected sequences into a new window.

DNA/RNA Menu

Show Helix	Show all violations to a proposed secondary structure.
Clustal Align	Align the selected sequences using the clustal algorithm. (See Appendix C)
Consensus	Return the consensus for the selected sequences.
Find All <meta f>	Search and highlight the selected sequences for a given substring.
Variable Positions	Show alignment columns that have variable regions.
DeSoete Tree <meta t>	Calculate a phylogenetic tree using a distance matrix tree fitting algorithm by DeSoete. (See Appendix C)
Distance Matrix <meta d>	Calculate a distance matrix for the selected sequences. (See Appendix C)
Blast D/RNA <meta b>	Blast algorithm developed by NCBI. Will do similarity searches of a selected sequence against databases. (See Appendix C)

Protein Menu

Blast Protein	Blast algorithm developed by NCBI. Will do similarity searches of a selected sequence against databases
Clustal Align	Align the selected sequences using the clustal algorithm. (See Appendix C)

Extras Menu

Remote Login	Connect to another machine via telnet.
Edit File	Bring up a text editor on a given file.

Bug reports/extensions

Any bug reports, RFE's (request for enhancement), and useful extensions to the GDE should be forwarded by electronic mail to:

smith@nucleus.harvard.edu

Please include as much detail as possible in bug reports so that the bug can be reproduced. Serious bugs will be fixed, and patches supplied as as soon as possible. Less consequential bugs, and RFE's will be handed out in the next release.

Correspondence should be addressed to:

Steven Smith
Director of Computation
Harvard Genome Laboratory
16. Divinity Ave.
Cambridge, MA 02138

Acknowledgments

I would like to thank the following people for their input and assistance and code used in the development of the GDE:

Carl Woese, Gary Olsen and Mike Maciukenas at University of Illinois Dept of Microbiology, Ross Overbeek at Argonne National Laboratories,Walter Gilbert, Patrick Gillevet, and Chunwei Wang at the Harvard Genome Laboratory, and Mitch Sogin, my patient tester at the Marine Biological Laboratory at Woods Hole.

Appendix A, File Formats

The currently supported file formats include Genbank (with type extensions), a generic flat file format, and a color mask file. The final goal is to use the ASN.1 format as set forth by NCBI as the default file format for all data types used in the GDE.

Genbank format:

GDE can read a concatenated list of Genbank entries, and extract certain fields from such files. The default method for storing nucleic acid, amino acid, masking sequences or text is in Genbank format. The following fields are recognized:

`LOCUS:`	Short name for this sequence (Maximum of 32 characters)†
`DEFINITION:`	Definition of sequence (Maximum of 80 characters)
`ORGANISM:`	Full name of organism (Maximum of 80 characters)
`AUTHORS:`	Authors of this sequence (Maximum of 80 characters)
`ACCESSION:`	ID Number for this sequence (Maximum of 80 characters)
`ORIGIN:`	Beginning of sequence data†
`//`	End of sequence data†

All other lines are retained as comments. The LOCUS line also specifies what type of sequence follows. The form of this line is:

```
LOCUS       name            size bp       type    date
```

where *name* is the Genbank Locus name, *size* is total base count, *type* is one of DNA, RNA, PROTEIN, MASK, or TEXT and *date* is of the form dd-MON-yyyy. In this way, the standard Genbank format is extended to store all text, mask and protein data. The Genbank character set has also been extended in order to support these other data types. Valid characters are:

DNA/RNA:	Full IUPAC coding as well as '-' and '~' characters for alignment gaps
Protein:	All valid single letter codes plus '-' and '~'. Other ASCII characters may be inserted, however external functions may be confused by such characters.
Mask:	All legal printable ASCII characters. If used as a selection mask, all columns containing a '0' will be removed from any analysis.
Text:	All valid ASCII characters.

Here is a valid Genbank entry for two E.coli tRNA's:

```
LOCUS         ECOTRNT4      76 bp     RNA               28-JAN-1991
DEFINITION  E. coli (T4 infected) vulnerable tRNA (A).
  ORGANISM  Escherichia coli
   AUTHORS  Amitsur,M., Levitz,R. and Kaufmann,G.
FEATURES       From  To/Span     Description
    tRNA          1       76     vulnerable tRNA(A)
BASE COUNT  ?
ORIGIN
```

† Required field

```
        1 GGGUCGUUAG CUCAGUUGGU AGAGCAGUUG ACUUUUAAUC AAUUGGNCGC AGGUUCGAAU
       61 CCUGCACGAC CCACCA
//
LOCUS          ECOTRQ1      75 bp     RNA               28-JAN-1991
DEFINITION  E.coli Gln-tRNA-1.
  ORGANISM  Escherichia coli
   AUTHORS  Yaniv,M. and Folk,W.R.
SOURCE      -REFERENCE   [1]  JOURNAL   J. Biol. Chem. 250, 3243-3253 (1975)
FEATURES       From  To/Span     Description
    tRNA          1       75     Gln-tRNA-1 (NAR: 0510)
    refnumbr      1        1     sequence not numbered in [1]
BASE COUNT  ?
ORIGIN
        1 UGGGGUAUCG CCAAGCGGUA AGGCACCGGU UUUUGAUACC GGCAUUCCCU GGUUCGAAUC
       61 CAGGUACCCC AGCCA
//
```

Flat file format:

This is a simplified format for importing sequence data, and passing it out to analysis functions. Very little information is actually retained in this format, and should be used carefully so as not to lose attribute information. It is defined as follow:

```
type_character short_name
sequence_data
sequence_data
sequence_data
...
```

The type character is # for DNA/RNA, % for protein sequence, @ for mask sequence, and " for text. The short name is the same as the LOCUS line in Genbank.
This is followed by lines of sequence, each ending with a return character.These lines are read until the next type character is encountered, or until the end of the file is reached. Care should be taken in using this format with text as space characters are stripped automatically.

Here is a sample flat file for two Ecoli tRNA's:

```
#ECOTRNT4
GGGUCGUUAGCUCAGUUGGUAGAGCAGUUGACUUUUAAUCAAUUGGNCGCAGGUUCGAAU
CCUGCACGACCCACCA
#ECOTRQ1
UGGGGUAUCGCCAAGCGGUAAGGCACCGGUUUUUGAUACCGGCAUUCCCUGGUUCGAAUC
CAGGUACCCCAGCCA
```

Color mask:

The format for a color mask has been kept simple to make implementation of color functions easy. The format optionally defines which sequence to color, whether or not to color alignment gaps in the existing sequence, and how long the following mask will be. it is then followed by a list of decimal color codes (range 0 to 15) for each position in the sequence. There are four keywords used in the color mask file. Those keywords are:

`name:`*`short name`*	If short name matches a currently loaded sequence, then impose this color mask on that sequence. If this line is omitted, then color all sequences this color.
`length:`*`length`*	The following list in length long

`nodash:` Skip over dash characters when imposing this color mask on the named sequence. This allows an unaligned color mask to be placed over aligned sequence.

`start:` Begin reading the color mask on the next line.

Here is a sample color mask file:

```
name:test_sequence
length:10
nodash:
start:
3
3
3
6
5
3
3
3
2
7
```

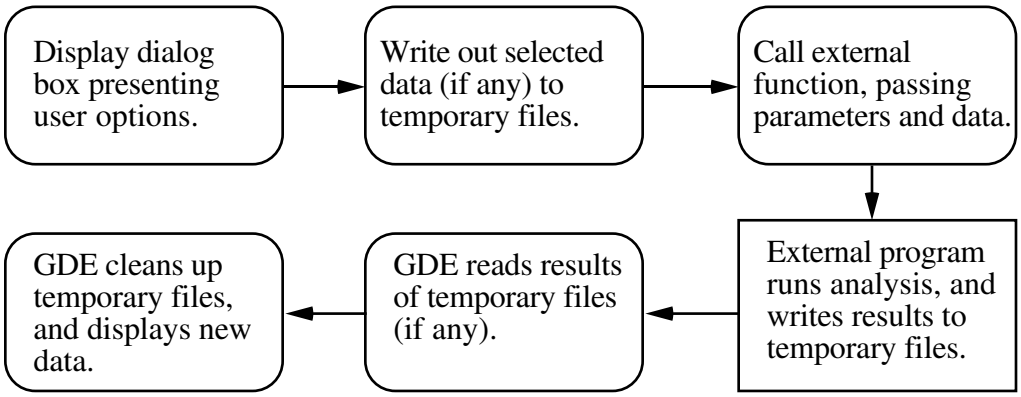
The colors in the default color lookup table are:

0	White	8	Black
1	Yellow	9	Grey 1
2	Violet	10	Grey 2
3	Red	11	Grey 3
4	Aqua	12	Grey 4
5	Lime Green	13	Grey 5
6	Blue	14	Grey 6
7	Purple	15	White

Appendix B, Adding Functions

The GDE uses a menu description language to define what external programs it can call, and what parameters and data to pass to each function. This language allows users to customize their own environment to suite individual needs.

The following is how the GDE handles external programs when selected from a menu:



Each step in this process is described in a file .GDEmenus in the user's current or home directory.

The language used in this file describes three phases to an external function call. The first phase describes the menu item as it will appear, and the Unix command line that is actually run when it is selected. The second phase describes how to prompt for the parameters needed by the function. The third phase describes what data needs to be passed as input to the external function, and what data (if any) needs to be read back from its output.

The form of the language is a simple keyword/value list delimited by the colon (:) character. The language retains old values until new ones are set. For example, setting the menu name is done once for all items in that menu, and is only reset when the next menu is reached.

The keywords for phase one are:

`menu:`*`menu name`*	Name of current menu
`item:`*`item name`*	Name of current menu item
`itemmeta:`*`meta_key`*	Meta key equivalence (quick keys)
`itemhelp:`*`help_file`*	Help file (either full path, or in GDE_HELP_DIR)
`itemmethod:`*`Unix command`*	

The item method command is a bit more involved, it is the Unix command that will actually run the external program intended. It is one line long, and can be up to 256 characters in length. It can have embedded variable names (starting with a '$') that will be replaced with appropriate values later on. It can consist of multiple Unix commands separated by semi-colons (;), and may contain shell scripts and background processes as well as simple command names. Examples will be given later.

The keywords for phase two are:

`arg:`*`argument_variable_name`*	Name of this variable. It will appear

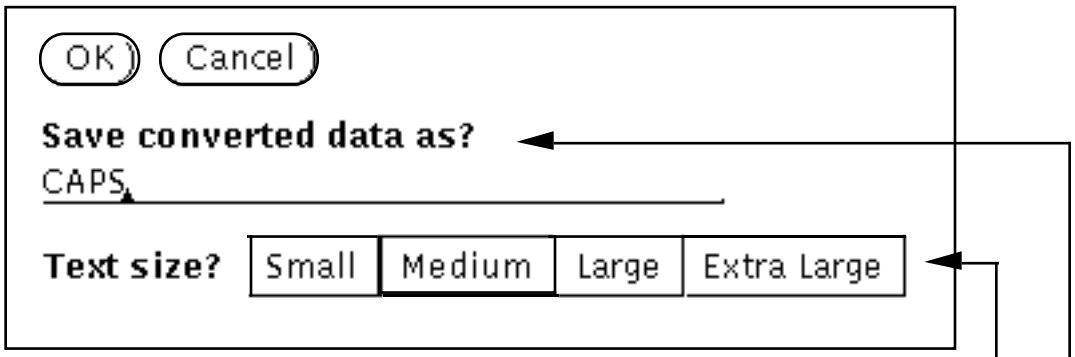
	in the itemmethod: line with a dollar sign ($) in front of it.
`argtype:`*`slider,chooser,choice_menu or text`*	The type of graphic object representing this argument.
`arglabel:`*`descriptive label`*	A short description of what this argument represents
`argmin:`*`minimum_value (integer)`*	Used for sliders.
`argmax:`*`maximum_value (integer)`*	Used for sliders.
`argvalue:`*`default_value (integer)`*	It is the numeric value associated with sliders or the default choice in choosers and choice_menus (the first choice is 0, the second is 1 etc.)
`argtext:`*`default value`*	Used for text fields.
`argchoice:`*`displayed value`*`:`*`passed value`*	Used for choosers and choice_menus. The first value is displayed on screen, and the second value is passed to the itemmethod line.

The keywords for phase three are as follows:

`in:`*`input_file`*	GDE will replace this name with a randomly generated temporary file name. It will then write the selected data out to this file.
`informat:`*`file_format`*	Write data to this file for input to this function. Currently support values are Genbank, and flat.
`inmask:`	This data can be controlled by a selection mask.
`insave:`	Do not remove this file after running the external function. This is useful for functions put in the background.
`out:`*`output_file`*	GDE will replace this name with a randomly generated temporary file name. It is up to the external function to fill this file with any results that might be read back into the GDE.
`outformat:`*`file_format`*	The data in the output file will be in this format. Currently support values are colormask, Genbank, and flat.
`outsave:`	Do not remove this file after reading.

This is useful for background tasks.

Here is a sample dialog box, and it's entry in the .GDEmenus file:




```
menu:Test function

item:All capitals
itemmethod:(tr '[a-z]' '[A-Z]' < INPUT_FILE >
INPUT_FILE.tmp ; mv INPUT_FILE.tmp $SAVE_FILE_NAME ; gde
$SAVE_FILE_NAME -Wx $SIZE ; rm INPUT_FILE) &

arg:SAVE_FILE_NAME
argtype:text
arglabel:Save converted data as?
argtext:CAPS

arg:SIZE
arglabel:Text size?
argtype:chooser
argvalue:1
argchoice:Small:small
argchoice:Medium:medium
argchoice:Large:large
argchoice:Extra Large:extra_large

in:INPUT_FILE
informat:flat
insave:
```

Using the default parameters given in the dialog box, the executed Unix command line would be:

```
(tr '[a-z]' '[A-Z]' < .gde_001 >.gde_001.tmp ; mv .gde_001.tmp CAPS ; gde CAPS -Wx
medium ; rm .gde_001 ) &
```

where `.gde_001` is the name of the temporary file generated by the GDE which contains the selected sequences in flat file format. Since the GDE runs this command in the background ('&' at the end) it is necessary to specify the `insave:` line, and to remove all temporary files manually. There is no output file specific because the data is not loaded back into the

current GDE window, but rather a new GDE window is opened on the file. A simpler command that reloads the data after conversion might be:

```
item:All caps
itemmethod:tr '[a-z]' '[A-Z]' <INPUT > OUTPUT

in:INPUT
informat:flat

out:OUTPUT
outformat:flat
```

In this example, no arguments are specified, and so no dialog box will appear. The command is not run in the background, so the GDE can clean up after itself automatically. The converted sequence is automatically loaded back into the current GDE window.

In general, the easiest type of program to integrate into the GDE is a program completely driven from a Unix command line. Interactive programs can be tied in (clustal for example), however shell scripts must be used to drive the parameter entry for these programs. Programs of the form:

```
program_name -a1 argument1 -a2 arguement2 -f inputfile -er errorfile > outputfile
```

can be specified in the .GDEmenus file directly. As this is the general form of most one Unix commands, these tend to be simpler to implement under the GDE.

As functions grow in complexity, they may begin to need a user interface of their own. In these cases, the command line calling arguments are still necessary in order to allow the GDE to hand them the appropriate data, and possible retrieve results after some external manipulation.

Appendix C, External functions

ClustalV - Cluster multiple sequence alignment

Author: Des Higgins.

Reference:Vax clustal, April 1989 Genetics

Parameters:

k-tuple size	Word size for pairwise comparisons
filter sensitivity	Smaller values give faster alignments, larger values are more sensitive.
similarity scoring	Percent identity works well with equal length sequences.
2 gap penalty settings	
translation scoring	Can weight transitions twice as high as transversions (DNA only).

Comments:

ClustalV is a directed multiple sequence alignment algorithm that aligns a set of sequences based on their level of similarity. It first uses a Lipman Peasron pairwise similarity scoring to find "clusters" of similar sequences, and pre-aligns those sequences. It then adds other sequences to the alignment in the order of their similarity so as to produce the cleanest alignment.

Warning: ClustalV only uses unambiguous character codes. It will also convert all sequences to upper case in the process of aligning. Clustal does not pass back comments, author etc. Be sure to keep copies of your sequences if you do not wish to lose this information.

Lsadt - Least squares additive tree analysis

Author: Geert De Soete, 'C' implementation by Mike Maciukenas University of Illinois

Reference:LSADT, 1983 Psychometrika, 1984 Quality and Quantity

Parameters:

Distance correction to use in distance matrix calculations (see count below).

What should be used for initial parameters estimates

Random number seed

Display method (See TreeTool below)

Comments:

The program has been rewritten in 'C' and will be included with the rRNA Database phylogenetic package being written at the University of Illinois Department of Microbiology.

Count is a short program to calculate a distance matrix from a sequence alignment (see below).

Count - Distance matrix calculator

Author: Steven Smith

Parameters:

Correction method	Currently Jukes-Cantor or none
Include dashed columns	
Match upper case to lower	

Comments:

Passes back a distance matrix in a format readable by LSADT.

Treetool - Tree drawing/manipulation

Author: Michael Maciukenas, University of Illinois

Comments:

See included documentation for TreeTool usage.

Blast - High speed homology search algorithm

Reference:

Karlin & Altschul, "Methods for assessing the statistical significance of molecular sequence features by using general scoring schemes," Proc. Natl. Acad. Sci. USA 87 (1990) 2264-2268;
Altschul and Lipman, "Protein database searches for multiple alignments," Proc. Natl. Acad. Sci. USA 87 (1990) 5509-5513;
Altschul, Gish, Miller, Myers and Lipman, "Basic local alignment search tool," J. Mol. Biol. 214 (1990) in press.

Comments:

For this release of the GDE, the .GDEmenus file has been configured to use the blastp and blastn programs. They are not included with the 1.0 release, but can be obtained via anonymous ftp from ncbi.nlm.nih.gov along with many other useful tools. A network version of the blast program is under development at NCBI. Contact Warren Gish at NCBI for details.