Genetic Data Environment version 2.0

Table of Contents

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. 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 hours (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 smith@nucleus.harvard.edu.

What's New for this Release

There have been several changes made for this 2.0 release of GDE. The most notable changes involve new editing capabilities. For most functions, the user can either select the sequences by their name, or by dragging over sections of the sequence text. See editing for more details. Other added features include a new checking mode, reduced scale viewing for sequence assembly, improved cut/copy/paste, and many new external functions. There is also a new file format, GDE format, which retains more of the all of the available information about a sequence. This format (unlike Genbank format in GDE1.0) will not loose fields such as group numbers, etc.

There is one other change that should be noted. The old "flat" file format has changed. The format now compresses leading gaps by allowing an offset number in parentheses after the name. GDE2.0 will still read old "flat" files, but will write them out in the new format. This might cause difficulties for some external functions written using this format. Because of the change, it is VERY important to keep the GDE1.0 support programs separate from GDE2.0, as some are now incompatible. The original flatfile format can be duplicated using ReadSeq/FastA format and sed.

System Requirements

GDE 2.0 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 15 meg of free disk space available.

We are also supporting a DECStation version of GDE. This is running under XView 3.0/X11R5. We encourage interested people to port the programs to their favorite Unix platform. We hope to support the SGI and Cray line of Supercomputers.

Note to Motif users

GDE2.0 can be run using different window managers. The most common alternative to olwm is the Motif window manager (mwm). The only problem in using another window manager is that the status line is not displayed. We have added a "Message panel" as an option under "File->Properties" which displays all of the information contained on the status line.

People using other window managers may also prefer using xterm, and xedit as default terminals and file editors. This can be accomplished by replacing all occurrences of 'shelltool' and 'textedit' with 'xterm -e' and 'xedit' in the $GDE_HELP_DIR/.GDEmenus file.

Installing the GDE

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 GDE2.0.tar /usr/local/GDE
demo% cd /usr/local/GDE
demo% tar -xf GDE2.0.tar
```

After this, each new user will need to 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.

FastA and Blast need to have the properly formatted databases installed in the $GDE_HELP_DIR under the directories FASTA/PIR, FASTA/GENBANK, BLAST/pir BLAST/genbank. For FASTA, simply copy a version of PIR and Genbank into the proper directory. Alternately, the PIR and GENBANK files can be symbolic links to copies of Genbank held elsewhere on your system.

Blast installation involves converting PIR and GENBANK to a temporary FASTA format (using pir2fasta and gb2fasta) and then using pressdb for nucleic acid, and setdb for amino acid to reformat the databases again into blast format. The .GDEmenus file is currently set up to search with blast using the following databases: pir, genpept, genupdate, and genbank. If you wish to divide these into subdivisions, then the .GDEmenus file will have to be edited.

The most up to date release of blast can be obtained via anonymous ftp to ncbi.nlm.nih.gov. The most recent release of FASTA can be obtained via anonymous ftp to uvaarpa.virginia.edu.

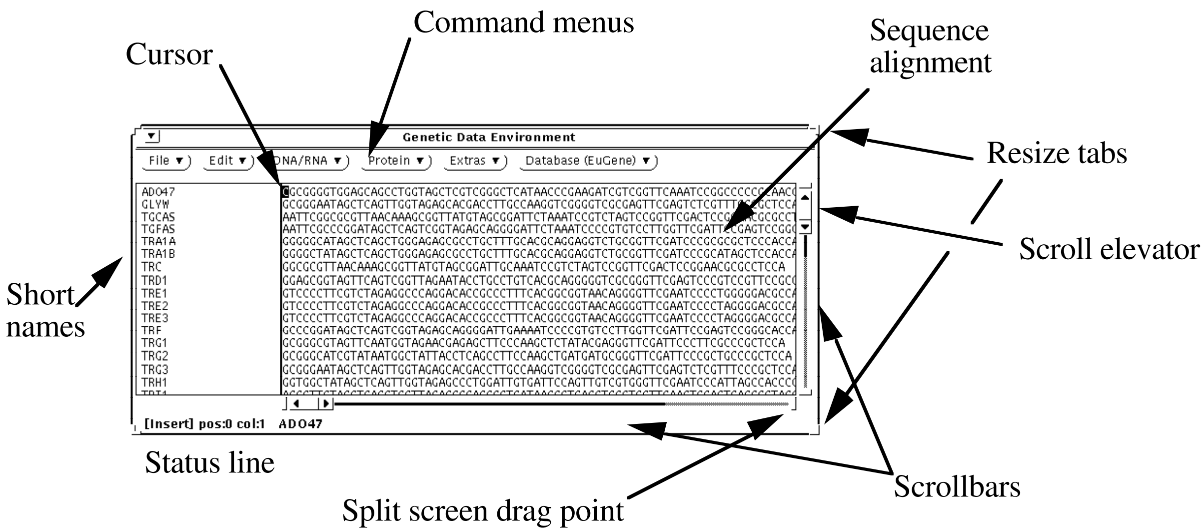
Using the GDE

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**

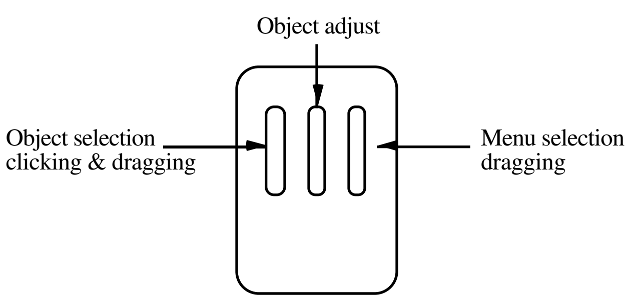
This should load the sample data set tRNAs into GDE, and the following window should appear:



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 is used for extending a text selection.

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. 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.

Selecting Text

Selecting text is accomplished in much the same way as selecting entire sequences. In the editing window, you can drag the mouse pointer over a rectangular region the select a block of text. By using the shift key (or the middle mouse button) you can adjust the selection to include other sequences, or other columns of text. If groups are enabled, GDE will automatically select all sequences in a group if any one sequence in a group is selected (See Sequence Editing).

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. The Sun function keys, cut, copy and paste are used to edit selected text. Text selections work in rectangular (possibly disjointed) regions. You can cut or copy a block of sequence text, and paste it to a new cursor location using these three keys.

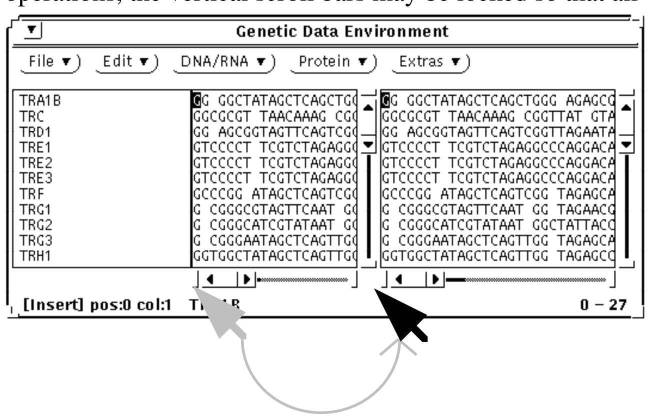
Repeat Counts

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

Split screen editing allows the viewing one region while editing another. This is very useful for aligning "downstream" regions by editing "upstream".

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 scroll bars 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 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.

Cut text, copy text and paste text 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 2.0 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.

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...

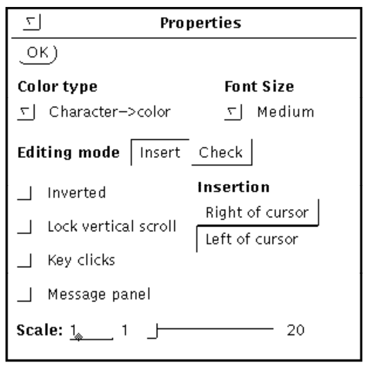
Selecting this will bring up the open file dialog box. Users can scroll through a list of files in the current directory, move up and down the directory tree, and open any individual data file. The sequence data in that file is loaded into the current editing window below any existing sequences. The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

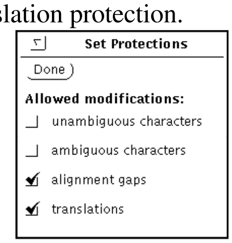
Properties...

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.



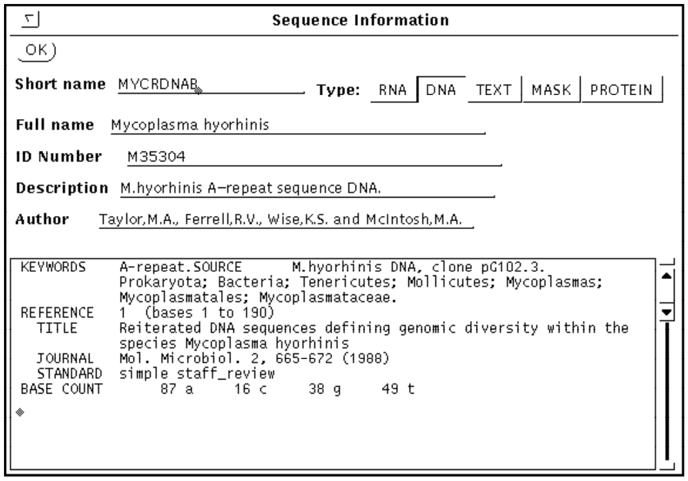
Protections...

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. 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 sequences
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.

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 gap characters from the selected sequences. The user has the option of removing all gaps, or simply all columns containing nothing but gaps. This is useful for minimizing the length of a subalignment.

Reverse Sequence
Reverses the selected sequences. Alignment gaps are reversed as well. The selected sequences will remain aligned after reversal.

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).

External Functions

See appendix C for a full description of functions supported in GDE 2.0 All external functions are described in the configuration file .GDEmenus. Here is a brief description of some of the basic functions included.

File menu

New Sequence <meta n>	Create a new sequence. Prompts for sequence type, and short name.
Import foreign format Export foreign format	Load and save sequences using Readseq by Don Gilbert (see Appendix C).
Save Selection	Save the currently selected sequences in a specified file.
Print Selection	Print the selected sequences to the chosen printer. This function supports the Unix command enscript as well as lpr. The .GDEmenus file may need to be modified to add the names of local printers to the printer list.

Edit menu

sort...	Sort the selected sequences by a primary and secondary key. Pass the new order to a new GDE window.
Extract	Extract the selected sequences into a new window.

DNA/RNA Menu

Translate	Translate the selected sequences from DNA/RNA to Amino acid. The user can specify the desired reading frame, and the minimum open reading frame (stop codon to stop codon) to translate. The user can also choose between single letter code and triple letter codes.
Dot plot	Display a dotplot identity matrix for the selected sequence(s). If only one sequence is selected, then the dotplot is a self comparison. If two or more sequences are selected, then the first two sequences are compared.
Clustal Align	Align the selected sequences using the clustalv algorithm by Des Higgins. (See Appendix C)
Find All <meta f>	Search and highlight the selected sequences for a given substring. A specified percent of mismatching can also be allowed.
Variable Positions	The selected sequences are scored column by column for conservation. The result is returned as a grey scale alignment color mask. This can be useful in selecting PCR primers.
Sequence Consensus	Return the consensus for the selected sequences. This can either be a majority consensus, or an ambiguity consensus using IUPAC coding.
Distance Matrix	Calculate a distance matrix for the selected sequences. (See Appendix C)
MFOLD	Fold the selected sequences using MFOLD by Michael Zuker. The resulting structure is returned as a nested bracket ('[]') representation of the secondary structure.(See appendix C.)
Draw Secondary Structure	Draw the selected sequence using the proposed secondary structure. Both the secondary structure prediction, and the RNA sequence should be selected before

	calling this function. The drawing program is LoopTool. (See Appendix C)
Highlight Helix	Show all violations to a proposed RNA secondary structure. The secondary structure represented must be selected, as well as the aligned sequences to be tested. The selected sequences will then be colored according to whether or not they support the proposed 2° structure. Standard Watson/Crick paring will be colored dark blue, G-U paring will be colored light blue, mismatches will be colored gold, and pairng to gaps will be red.
Blastn/BlastX	Search the selected sequence (select only one) against a given database with the BLAST searching tool written by Altschul, Gish, Miller, Myers, and Lipman. Blastn searches DNA against DNA databases, blastx searches DNA against AA databases by translating the sequence in all six reading frames. (See Appendix C)
FastA	Search the selected sequence (select only one) against a given database using the FASTA similarity search program written by Pearson and Lipman. (See Appendix C)

Protein Menu

Clustal Align	Align the selected amino acid sequences using the clustal algorithm. (See Appendix C)
Blastp, Tblastn, Blast3	Search the selected sequence (select only one) against a given database with the BLAST searching tool written by Altschul, Gish, Miller, Myers, and Lipman. Blastp searches AA against AA databases, tblastn searches AA against DNA databases by translating the database in all six reading frames. Blast3 finds three way alignments that are could not be found with only pairwise comparisons. (See Appendix C)

Sequence Management Menu

Assemble contigs	Assemble the selected sequences into contigs using the program CAP (Contig Assemble Program) written by Xiaoqiu Huang. The resulting sequences are returned to the current GDE window, and they are grouped into contigs. The user can then sort the sequences by group, and offset to produce an ordered list of the contigs. (See Appendix C)
Strategy view	Pass out the selected sequences to StratView. This program will display contigs in a greatly reduced line drawing. This is very useful for large contigs.
Restriction sites	Search the selected sequences for the restriction enzymes specified in the given enzyme file. The restriction sites are then colored by enzyme.

Phylogeny menu

DeSoete Tree fit	Calculate a phylogenetic tree using a least squares fitting algorithm on a distance matrix calculated from the selected sequences. The results can then be passed on to treetool for display and manipulation. (See Appendix C)
Phylip 3.4	Pass the selected data to on of the treeing programs in Phylip, written by Joe Felsenstein. The chosen phylip program is started in it's own window, with the selected sequences already loaded. (See Appendix C)

Citation of work

We ask that any published work using any of the external functions in GDE cite the appropriate authors. Please see Appendix C for references.

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. 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, Chunwei Wang and Susan Russo at the Harvard Genome Laboratory. I would also like to personally thank the following people for their permission to include their software with this release of GDE.

Des Higgins
David Lipman and the group at NCBI
William Pearson
Don Gilbert
Xiaoqui Huang
Joe Felsenstein
Michael Zuker
Geert DeSoete
Michael Maciukenas and the group at the Ribosomal Database Project

It is only by the generosity of these people that GDE has been successful.

Appendix A, File Formats

The currently supported file formats include GDE data files, Genbank formatted files (with type extensions), a generic flat file format, and a color mask file.

GDE format

GDE format is a tagged field format used for storing all available information about a sequence. The format matches very closely the GDE internal structures for sequence data. The format consists of text records starting and ending with braces ('{}'). Between the open and close braces are several tagged field lines specifying different pieces of information about a given sequence. The tag values can be wrapped with double quote characters ('"') as needed. If quotes are not used, the first whitespace delimited string is taken as the value. The allowable fields are:

```
{
name            "Short name for sequence"
longname        "Long (more descriptive) name for sequence"
sequence-ID     "Unique ID number"
creation-date   "mm/dd/yy hh:mm:ss"
direction       [-1|1]
strandedness    [1|2]
type            [DNA|RNA||PROTEIN|TEXT|MASK]
offset          (-999999,999999)
group-ID        (0,999)
creator         "Author's name"
descrip         "Verbose description"
comments        "Lines of comments that can be fairly arbitrary
text about a sequence.  Return characters are allowed, but no internal
double quotes or brace characters.  Remember to close with a double
quote"
sequence        "gctagctagctagctagctcttagctgtagtcgtagctgatgctagct
gatgctagctagctagctagctgatcgatgctagctgatcgtagctgacg
gactgatgctagctagctagctagctgtctagtgtcgtagtgcttattgc"
}
```

Any fields that are not specified are assumed to be the default values. Offsets can be negative as well as positive. Genbank entries written out in this format will have all (") converted to ('), and all ({}) converted to ([]) to avoid confusion in the parser. Leading and trailing gaps are removed prior to writing each sequence. This format is deliberately verbose in order to be simple to duplicate.

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†

† Required field

`//` 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
        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. As of release 2.0, flat file format allows for an optional offset to be specified in parentheses after the sequence name. An offset represents how many leading gap characters should be placed before the start of a sequence. If this offset does not exist, then it is defined to be 0.

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, and the color mask is expected to start at the leftmost column on the screen.
`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
```

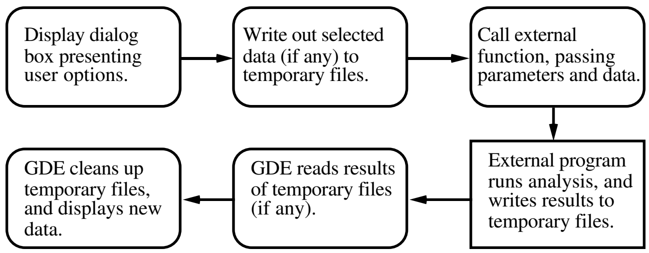
The colors in the default color lookup table are:

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:

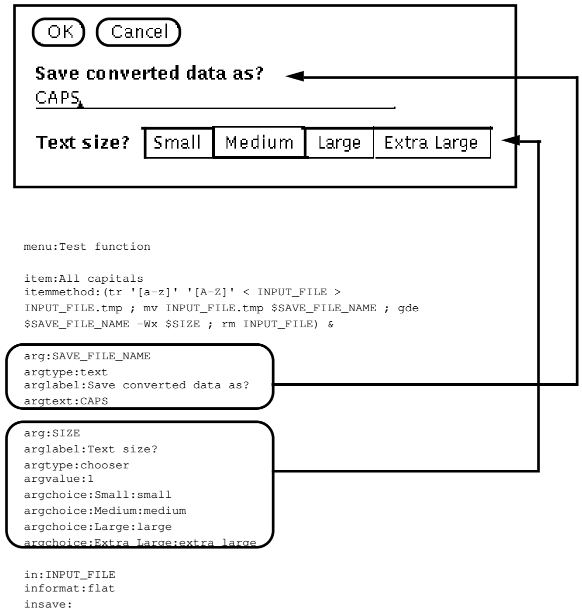
`arg:`*`argument_variable_name`*	Name of this variable. It will appear in the itemmethod: line with a dollar sign ($) in front of it.
`argtype:`*`slider,chooser,choice_menu or text`*	The type of graphic object representing this argument.
`arglabel:`*`descriptive label`*	A short description of what this 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.
`outoverwrite:`	Overwrite existing sequences in the current GDE window. Currently supported with "gde" format only.

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



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 (MFOLD 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: Higgins,D.G. Bleasby,A.J. and Fuchs,R. (1991) CLUSTAL V: improved software for multiple sequence alignment. ms. submitted to CABIOS

Parameters:

k-tuple pairwise search	Word size for pairwise comparisons
Window size	Smaller values give faster alignments, larger values are more sensitive.
Transitions weighted	Can weight transitions twice as high as transversions (DNA only).
Fixed gap penalty	Gap insertion penalty, lower value, more gaps
Floating gap penalty	Gap extension penalty, lower value, longer gaps

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.

MFOLD - RNA secondary prediction

Author: Michael Zuker

Reference:

M. Zuker
On Finding All Suboptimal Foldings of an RNA Molecule.
Science, 244, 48-52, (1989)

J. A. Jaeger, D. H. Turner and M. Zuker
Improved Predictions of Secondary Structures for RNA.
Proc. Natl. Acad. Sci. USA, BIOCHEMISTRY, 86, 7706-7710, (1989)

J. A. Jaeger, D. H. Turner and M. Zuker
Predicting Optimal and Suboptimal Secondary Structure for RNA.
in "Molecular Evolution: Computer Analysis of Protein and Nucleic Acid Sequences", R. F. Doolittle ed.
Methods in Enzymology, 183, 281-306 (1989)

Parameters:

Linear/circular RNA fold
ct File to save results

Comments:

MFOLD passes it's output to a program Zuk_to_gen that translates the secondary structure prediction to a nested bracket ([]) notation. This notation can then be used in the Highlight Helix, and Draw Secondary structure (LoopTool) functions.

MFOLD currently does not support much in the way of additional parameters. We hope to have all additional parameters available soon.

Blast - Basic Local Alignment Search Tool

Reference:

Karlin, Samuel and Stephen F. Altschul (1990). Methods for assessing the statistical significance of molecular sequence features by using general scoring schemes, Proc. Natl. Acad. Sci. USA 87:2264-2268.

Altschul, Stephen F., Warren Gish, Webb Miller, Eugene W. Myers, and David J. Lipman (1990). Basic local alignment search tool, J. Mol. Biol. 215:403-410.

Altschul, Stephen F. (1991). Amino acid substitution matrices from an information theoretic perspective. J. Mol. Biol. 219:555-565.

Parameters:

	Which Database	Which nucleic or amino acid database to search.
	Word Size	Length of initial hit. after locating a match of this length, alignment extension is attempted.
Blastn		
	Match score	Score for matches in secondary alignment extension
	Mismatch score	Score for mismatches in secondary alignment extension
Blastx, tblastn, blastp, blast3		
	Substitution Matrix	PAM120 or PAM250

Comments: The report is loaded into a text editor. This should be saved as a new file as the default file is removed after execution. The latest version of blast can be obtained via anonymous ftp to ncbi.nlm.nih.gov.

FastA - Similarity search

Reference:

W. R. Pearson and D. J. Lipman (1988),
"Improved Tools for Biological Sequence Analysis", PNAS 85:2444-2448

W. R. Pearson (1990) "Rapid and Sensitive Sequence
Comparison with FASTP and FASTA" Methods in Enzymology 183:63-98

Parameters:

Database	Which database to search
Number of alignments to report	
SMATRIX	Which similarity matrix to use

Comments:

The FastA package includes several additional programs for pairwise alignment. We have only included a bare bones link to FastA. We hope to include a more complete setup for the actual 2.0 release.

Assemble Contigs - CAP Contig Assembly Program

Author - Xiaoqiu Huang
Department of Computer Science
Michigan Technological University
Houghton, MI 49931
E-mail: huang@cs.mtu.edu

Minor modifications for I/O by S. Smith

Reference -
"A Contig Assembly Program Based on Sensitive Detection of Fragment Overlaps" (submitted to Genomics, 1991)

Parameters:

Minimum overlap	Number of bases required for overlap
Percent match within overlap	Percentage match required in the overlap region before merge is alowwed.

Comments:

CAP returns the aligned sequences to the current editor window. The sequences are placed into contigs by setting the groupid. Cap does not change the order of the sequences, and so the results should be sorted by group and offset (see sort under the Edit menu).

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.

Phylip - Complete phylogenetic analysis package (3.4)

Author: Joe Felsenstein

Reference: Felsenstein, J. 1989. PHYLIP -- Phylogeny Inference Package (Version 3.2). Cladistics 5: 164-166

Comments:

Phylip is a very complete set of programs for phylogenetic analysis. GDE simply formats selected data (obeying any selection mask) to Phylip, and runs the chosen program. Although all programs from Phylip are included, only the programs pertaining to DNA/AA analysis are tied in to the menu system. The complete set of documentation is included, and can be viewed directly from GDE. It is strongly suggested that user read the documentation before attempting to interpret results.

The full source code for Phylip can be obtained from genetics.washington.edu using anonymous ftp.

Copyright Notice

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

Disclaimer