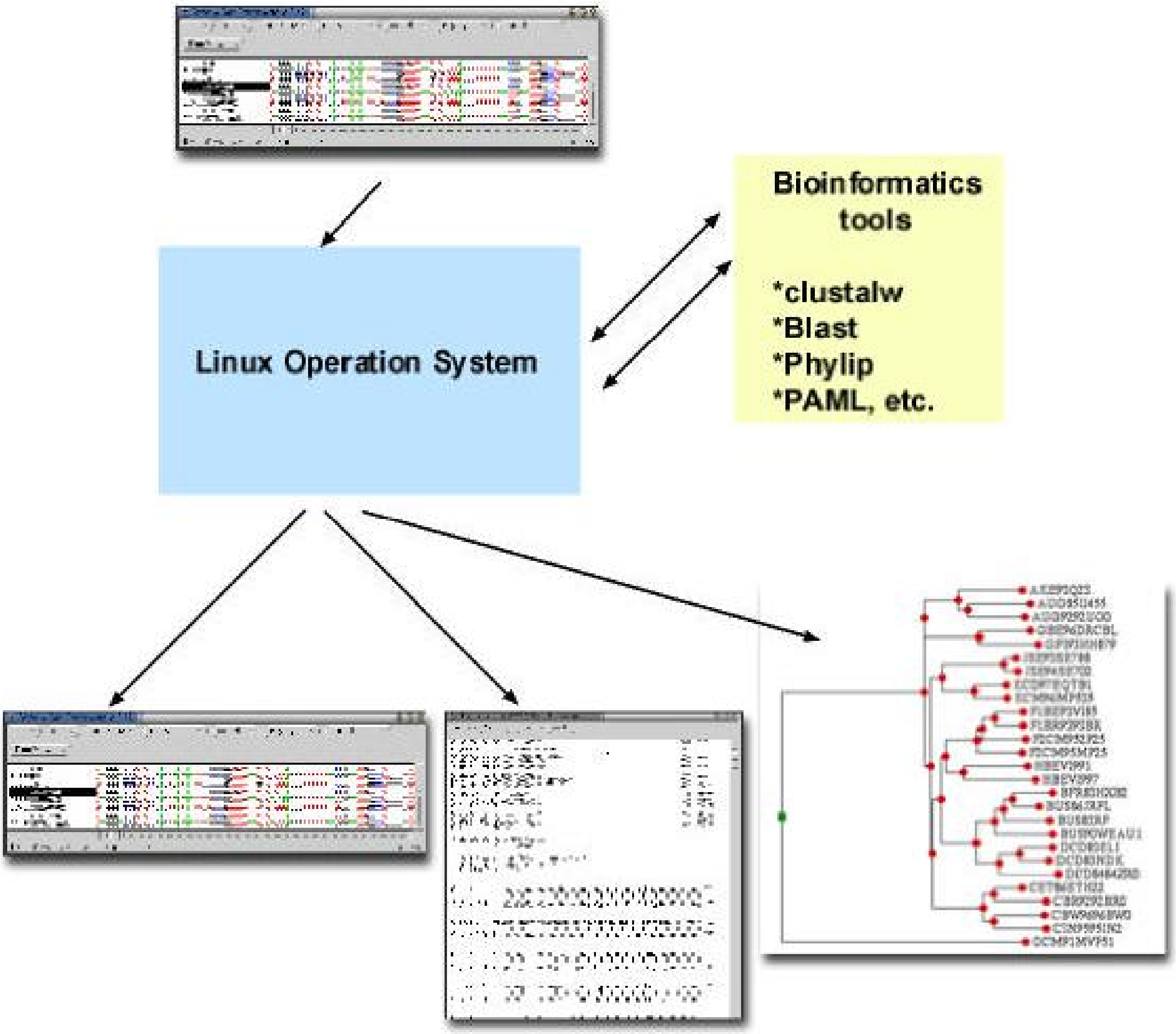
BIOINFORMATICS SOFTWARE INTEGRATION WITH GDE for LINUX



Version 1.0

Tulio de Oliveira

Bioinfotmatics software integration with GDE for Linux

GDE for linux tutorial number 3 – http://www.bioafrica.net/GDEtutorials.html
Tulio de Oliveira - 31/01/2002

How to set new software under GDE ?

Introduction:

GDE is just a front editor for sequence analysis; its massive power relies in editing the menu files allowing GDE run any program in the system.

Bioinformatics tools integration with GDE can be a simple and fast task. Most of the bioinformatics softwares running in a UNIX or Linux system can be integrated with GDE if they follow two basic rules:

1 Run in command line
2. Input/outup in known sequence formats by readseq (fasta, genbank, embl, phylip, nexus, GCG, PIR and others).

Once the software follows these two rules, the basic steps to integrate new software are:

1. Install the software in the linux system.
2. Determine command line for using the software.
3. Integration by editing the GDE menu file.

Our tutorial will use two pratical examples for software integration. The user is encouraged to follow the instructions and construct a menu for accessing a sequence converter program (readseq) and a multiple alignment program (clustalw).

Step 1, Install the software:

The software needs to be installed in your system.

Readseq and Clustalw were downloaded in tar.gz format from their web sitesOfficial READSEQ Web Page. .

A number of bioinformatics packages are distributed in debian (.deb) and redhat (.rpm) linux file formats, and are easily installed . Redhat files for clustalw and readseq was downloaded.

Readseq and Clustalw were downloaded in tar.gz format, respectivily from the Official TreeView Web Page and Official READSEQ Web Page. Also both readseq and clustalw can be downloaded in RedHat (rpm) and Debian (deb) format from http://.....

GDE were installed in the /usr/local/bio/GDE following instruction from the How to Setup GDE tutorial also we suggest that bioinformatics tools be installed in the /usr/local/bio/ directory. Readseq was installed in /usr/local/bio/readseq/ and clustal at /usr/local/clustalw.

Note 1: Our examples and exercises were installed in a Red Hat 7.2 Linux distribution (freely downloadable: http://www.redhat.com), and minor changes will need to be performed to install the software in other Linux distributions..

Note 2: Installing bioinformatics software in the /usr/local/bio/ needs root account access.

Installing readseq and clustalw using the tar.gz format:

1. Download the software
2. Make a directory for the application in /usr/local/bio/ (mkdir /usr/local/bio/readseq/ ; mkdir /usr/local/clustalw/)
3. Move the software to the created directory.
4. Unpack the packages (tar -zxvf readseq.tar.gz)
5. Read instructions in the README file
6. Compile the packages following instructions (e.g. make; make install)
7. Link the software in the bin directory (e.g. ln -s /usr/local/bio/readseq/readseq /bin/readseq)

Installing readseq and clustalw using the redhat (.rpm) format:

1. Download the software in rpm format
2. Install using rpm (e.g. rpm -i clustalw-1.7-8.i386.rpm; rpm –i readseq-xx.rpm)

This will automatically unpack, install and setup the correct path.

2. Learning the Software.

GDE uses a menu file (/usr/local/bio/GDE/CORE/.GDEmenus) that control the commands(methods) to run bioinformatics programs (e.g. readseq and clustalw).

It is necessary to know how to run the bioinformatics software in command line before attempting to integrate with GDE. Also it is necessary to known all the software input/output formats options.

The better way (and normally the only way) to learn software commands is to read its documentation.

Let's begin the exercise trying to learn the code for readseq.

In the shell type the command:

```
>readseq --help
```

The following information will be displayed:

```
readSeq (1Feb93), multi-format molbio sequence reader.
usage: readseq [-options] in.seq > out.seq
 options
    -a[ll]         select All sequences
    -c[aselower]   change to lower case
    -C[ASEUPPER]   change to UPPER CASE
    -degap[=-]     remove gap symbols
    -i[tem=2,3,4]  select Item number(s) from several
    -l[ist]        List sequences only
    -o[utput=]out.seq redirect Output    -p[ipe]       Pipe (command
line, <stdin, >stdout)
    -r[everse]     change to Reversecomplement
    -v[erbose]     Verbose progress
    -f[ormat=]#    Format number for output, or
    -f[ormat=]Name Format name for output:
         1. IG/Stanford        10. Olsen (in-only)
         2. GenBank/GB         11. Phylip3.2
         3. NBRF               12. Phylip
         4. EMBL               13. Plain/Raw
         5. GCG                14. PIR/CODATA
         6. DNAStrider         15. MSF
         7. Fitch              16. ASN.1
         8. Pearson/Fasta      17. PAUP/NEXUS
         9. Zuker (in-only)    18. Pretty (out-only)

   Pretty format options:
    -wid[th]=#           sequence line width
    -tab=#               left indent
    -col[space]=#         column space within sequence line on output
    -gap[count]           count gap chars in sequence numbers
    -nameleft, -nameright[=#]  name on left/right side [=max width]
    -nametop              name at top/bottom
    -numleft, -numright   seq index on left/right side
    -numtop, -numbot      index on top/bottom
    -match[=.]            use match base for 2..n species
    -inter[line=#]        blank line(s) between sequence blocks
```

The usage determine the synthax of the program.
E.g. :

```
>readseq -a -f 8 seq.phy > seq.fasta
```

This command will read **all (-a)** the sequences in the file seq.phy (phylip format) and creates a **Fasta (-f8)** format file named seq.fasta.

Exercise 1:
In the tutorial directory (/usr/local/bio/GDE/tutorial/) change sequence format for the files: seq1.phy (to fasta and nexus format), seq2.fasta (to PIR and GCG format) and seq3.gb (for fasta and phylip formats). It is always good when naming a file to use the sequence format as extension (.gb for GenBank, .fasta for fasta, .phy for phylip, etc).

Learning Clustalw command line:

In the shell type the command:

```
>clustalw --help
```

```
 CLUSTAL W (1.7) Multiple Sequence Alignments



               DATA (sequences)

/INFILE=file.ext                          :input sequences.
/PROFILE1=file.ext and  /PROFILE2=file.ext  :profiles (old alignment).

               VERBS (do things)

/OPTIONS           :list the command line parameters
/HELP  or /CHECK   :outline the command line params.
/ALIGN             :do full multiple alignment
/TREE              :calculate NJ tree.
/BOOTSTRAP(=n)     :bootstrap a NJ tree (n= number of bootstraps; def.
= 1000).
/CONVERT           :output the input sequences in a different file
format.

               PARAMETERS (set things)

***General settings:****
/INTERACTIVE :read command line, then enter normal interactive menus
/QUICKTREE   :use FAST algorithm for the alignment guide tree
/NEGATIVE    :protein alignment with negative values in matrix
/OUTFILE=    :sequence alignment file name
/OUTPUT=     :GCG, GDE, PHYLIP or PIR
/OUTORDER=   :INPUT or ALIGNED
/CASE        :LOWER or UPPER (for GDE output only)
/SEQNOS=     :OFF or ON (for Clustal output only)

***Fast Pairwise Alignments:***
/KTUPLE=n      :word size              /TOPDIAGS=n  :number of best
diags.
/WINDOW=n    :window around best diags. /PAIRGAP=n   :gap penalty
/SCORE       :PERCENT or ABSOLUTE

***Slow Pairwise Alignments:***
/PWMATRIX=   :Protein weight matrix=BLOSUM, PAM, GONNET, ID or
filename
/PWDNAMATRIX= :DNA weight matrix=IUB, CLUSTALW or filename
/PWGAPOPEN=f :gap opening penalty      /PWGAPEXT=f  :gap opening
penalty

***Multiple Alignments:***
/NEWTREE=    :file for new guide tree
/USETREE=    :file for old guide tree
/MATRIX=     :Protein weight matrix=BLOSUM, PAM, GONNET, ID or filename
/DNAMATRIX=  :DNA weight matrix=IUB, CLUSTALW or filename
/GAPOPEN=f   :gap opening penalty      /GAPEXT=f  :gap extension
penalty
/ENDGAPS     :no end gap separation pen. /GAPDIST=n :gap separation
pen. range
/NOPGAP      :residue-specific gaps off  /NOHGAP    :hydrophilic gaps
```

```
off
/HGAPRESIDUES= :list hydrophilic res.  /MAXDIV=n    :% ident. for
delay
/TYPE=       :PROTEIN or DNA            /TRANSWEIGHT=f :transitions
weighting

***Profile Alignments:***
/PROFILE     :Merge two alignments by profile alignment
/NEWTREE1=    :file for new guide tree for profile1
/NEWTREE2=    :file for new guide tree for profile2
/USETREE1=    :file for old guide tree for profile1
/USETREE2=    :file for old guide tree for profile2

***Sequence to Profile Alignments:***
/SEQUENCES   :Sequentially add profile2 sequences to profile1 alignment
/NEWTREE=    :file for new guide tree
/USETREE=    :file for old guide tree

***Structure Alignments:***
/NOSECSTR1     :do not use secondary structure/gap penalty mask for
profile 1
/NOSECSTR2     :do not use secondary structure/gap penalty mask for
profile 2
/SECSTROUT=    :STRUCTURE or MASK or BOTH or NONEoutput in alignment
file
/HELIXGAP=n    :gap penalty for helix core residues
/STRANDGAP=n   :gap penalty for strand core residues
/LOOPGAP=n     :gap penalty for loop regions
/TERMINALGAP=n :gap penalty for structure termini
/HELIXENDIN=n  :number of residues inside helix to be treated as
terminal
/HELIXENDOUT=n :number of residues outside helix to be treated as
terminal
/STRANDENDIN=n :number of residues inside strand to be treated as
terminal
/STRANDENDOUT=n:number of residues outside strand to be treated as
terminal

***Trees:***
/OUTPUTTREE=nj OR phylip OR dist
/SEED=n    :seed number forbootstraps.
/KIMURA      :use Kimura's correction.  /TOSSGAPS  :ignore positions
with gaps.
```

The compulsory code to run clustalw is:
e.g:

>clustalw -infile=seq1.fasta -output=GDE -outfile=seq1.gde

The default paramethers in clustalw perform a fast alignment.

Clustalw can produce very different alignments using several different options, for a review on multiple alignment using clustalw go to

Example 2: change alignment variables

```
> clustalw -infile=seq1.fasta -ktuple=5 -window=10 -output=fasta -outfile=seq1.aln.fasta
```

Now the word size (ktuple) and window around best diags where changed to 5 and 10 respectively, the output format is fasta and the outfile was named seq1.aln.fasta.

Example 3: Running a more accurate alignment (slow).

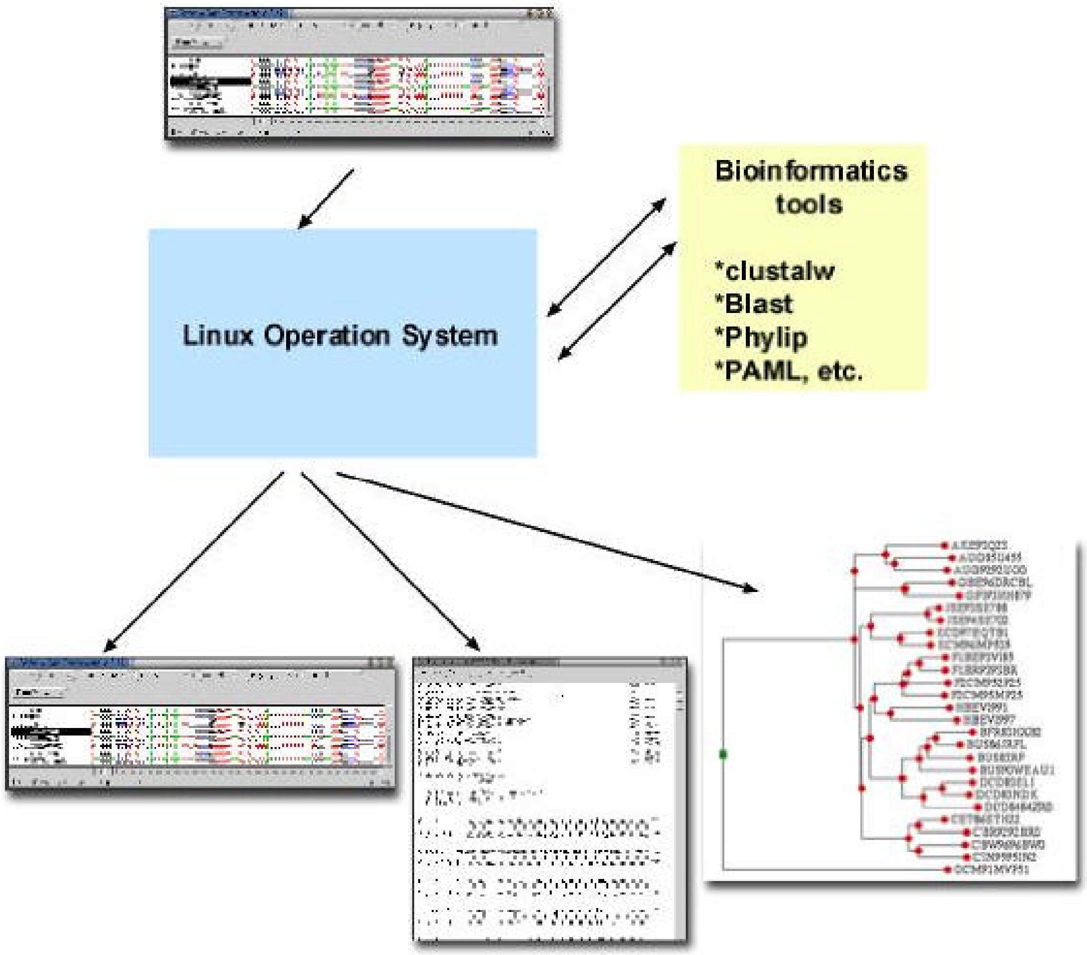
```
>clustalw -align -infile=seq1.fasta -output=GDE -outfile=seq1.gde
```

Now a slow and more accurate alignment is performed using the option –align.

Exercise 2:
- Align the seq2.fasta file created in the previous exercise in clustalw with the default options.
- perform a slow and accurate alignment with gap open penality of 4.

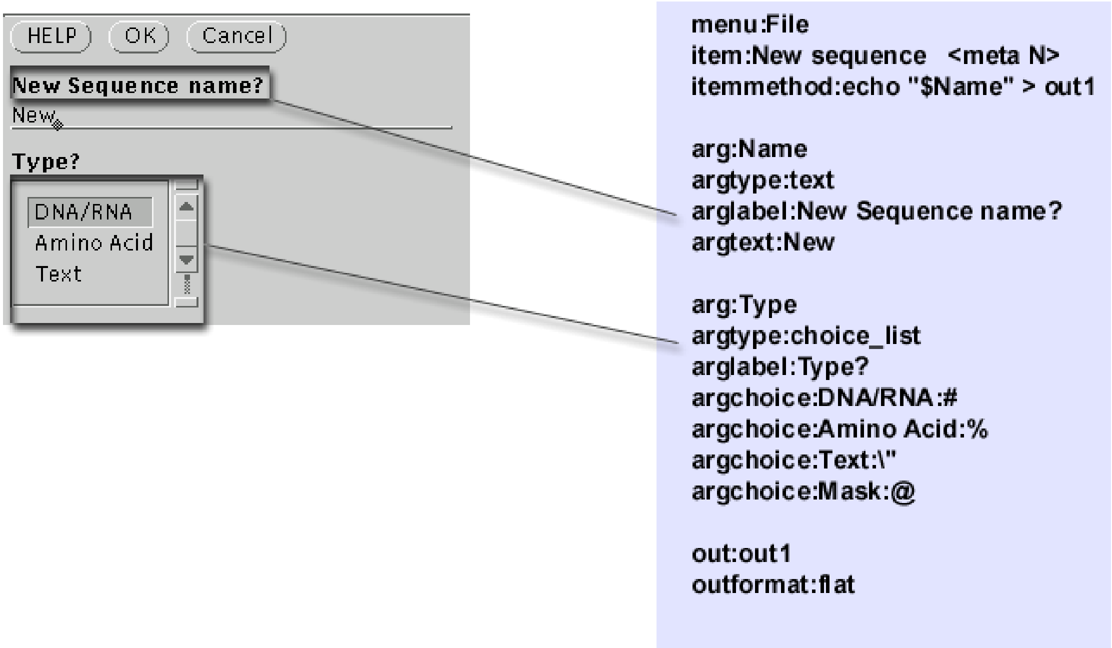
3. Integration by editing the GDE menu file.

The menus in GDE can be fully custumized and are controled by the **.GDEmenus** file located at (/usr/local/bio/GDE/CORE), the code used for setting the menus are very similar with shell scripting (running program in command line). The following diagram summarize how GDE works



The select sequences are stored in a temporary file. GDE menus run the chosen bioinformatics software. The output produced by the bioinformatics software is displayed. Basically GDE can run any software in your linux computer and display the output, doesn’ t matter if it is a sequence, a text, a graphic or a tree file.

Synthax of the menu file:



```
menu:File
item:New sequence   <meta N>
itemmethod:echo "$Name" > out1

arg:Name
argtype:text
arglabel:New Sequence name?
argtext:New

arg:Type
argtype:choice_list
arglabel:Type?
argchoice:DNA/RNA:#
argchoice:Amino Acid:%
argchoice:Text:\"
argchoice:Mask:@

out:out1
outformat:flat
```

The menu has the following hierarchal organization:

```
menu: name of the menu
item: option inside the menu
itemmethod: commands to be runned in the system

arg: variables to be used

in: infile
out: outfile
```

Itemmethod set the commands that run in the system

To test the commands in send to the system, open a shell and type the itemmethod commands:

e.g.

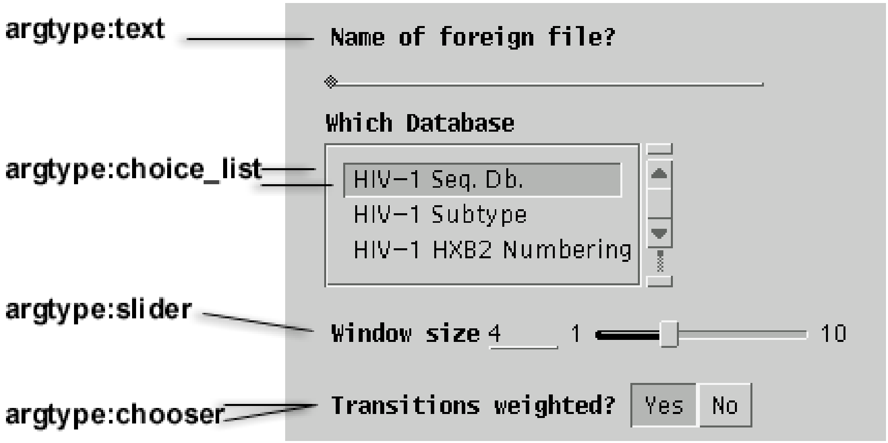
```
>echo "sequence" > out1
```

The word sequence will be writen to a file named out1.

* out1 is a GDE known variable and will add the sequence name for a new seq. in GDE.

Setting Variables:
Variables in GDE are identified by the arg command (arg:variablename). Variables are normally software options, database names or files names. Variables are set as follow:

arg: variable name
argtype: Type of the variabe (text , choice_list, chooser, slider)
arglabel: Text before variable
argchoice: choice in a list

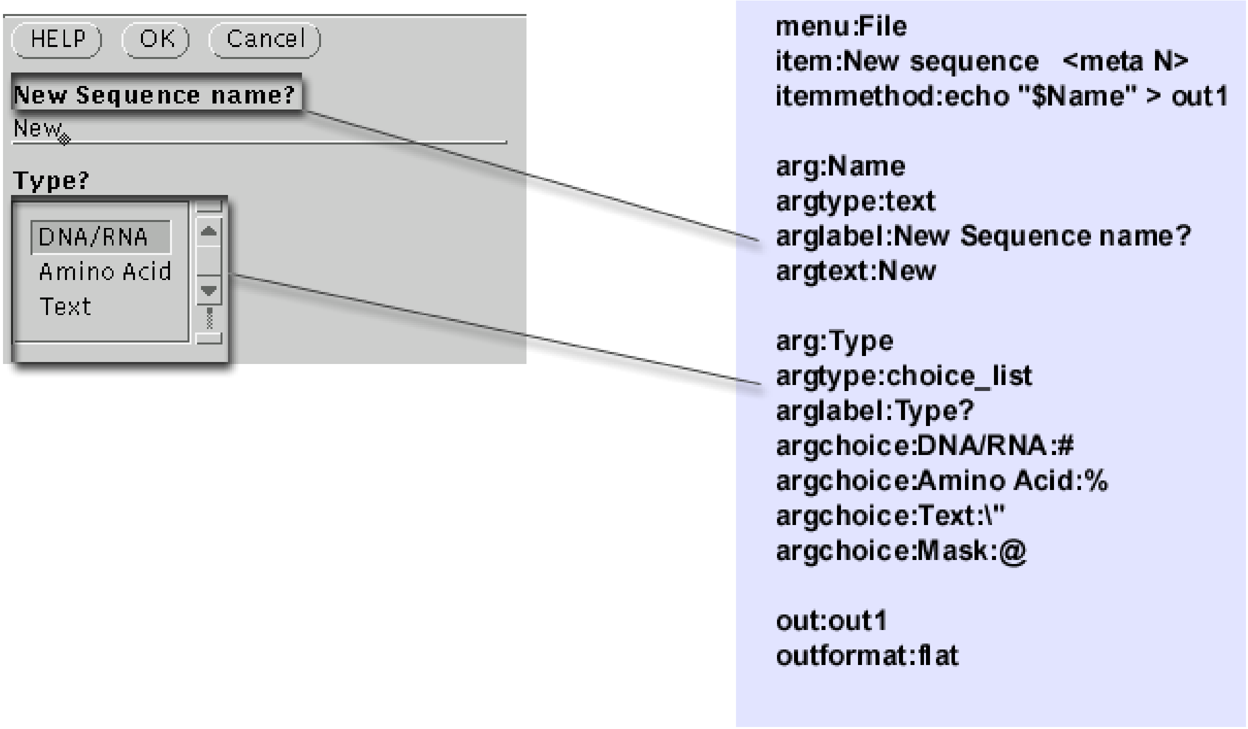


Ending a item menu:
The item menu finishes with the out and outformat variables. GDE is able to import the sequence outfile to the viewer. Several outfile formats can be produced for text or sequence visualisation.

```
out:out1
outformat:flat
```

Variabes:
out = output of the method
outformat = output format (flat, gde, genbank, colormask, text)

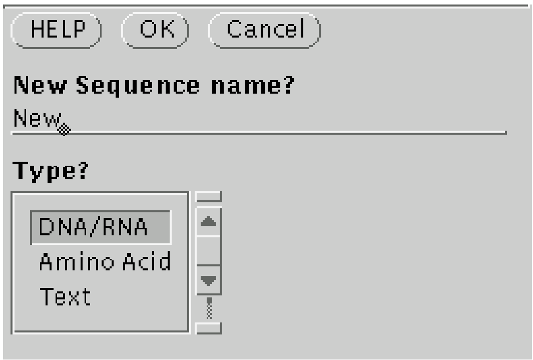
Picture of the menu in GDE:



New Sequence name?
New

Type?
DNA/RNA
Amino Acid
Text

```
menu:File
item:New sequence   <meta N>
itemmethod:echo "$Name" > out1

arg:Name
argtype:text
arglabel:New Sequence name?
argtext:New

arg:Type
argtype:choice_list
arglabel:Type?
argchoice:DNA/RNA:#
argchoice:Amino Acid:%
argchoice:Text:\"
argchoice:Mask:@

out:out1
outformat:flat
```



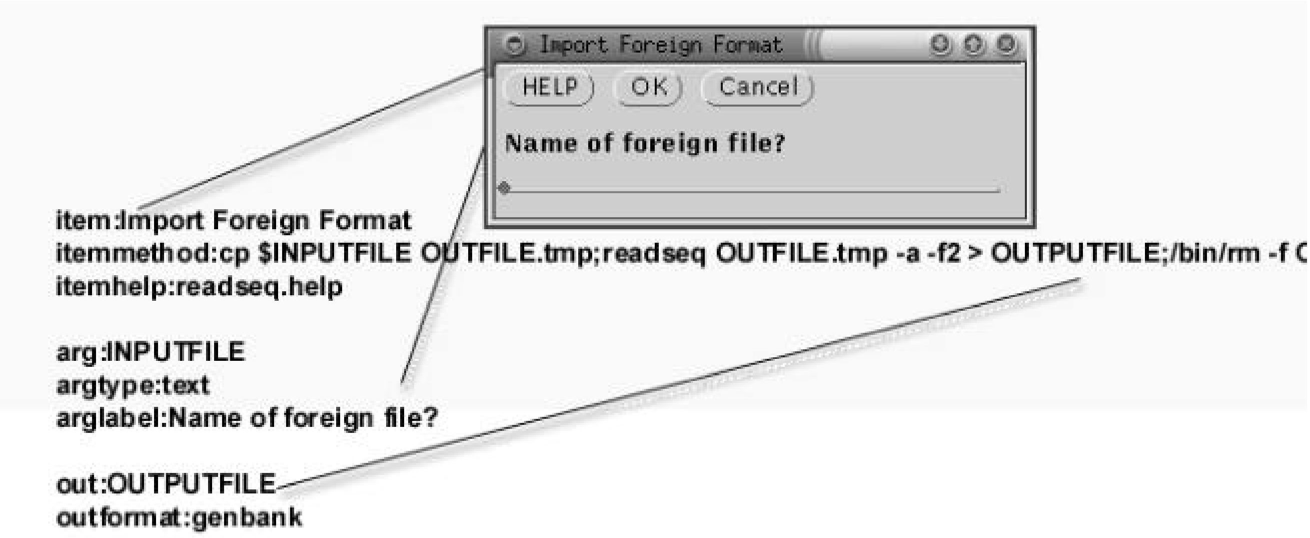
New Sequence name?
New

Type?
DNA/RNA
Amino Acid
Text

Readseq menu construction:

Readseq is able to transform sequence formats, and it will be integrated in the GDE for importing and exporting functions. We will create file name viariables and shell script to construct the itemmethod.

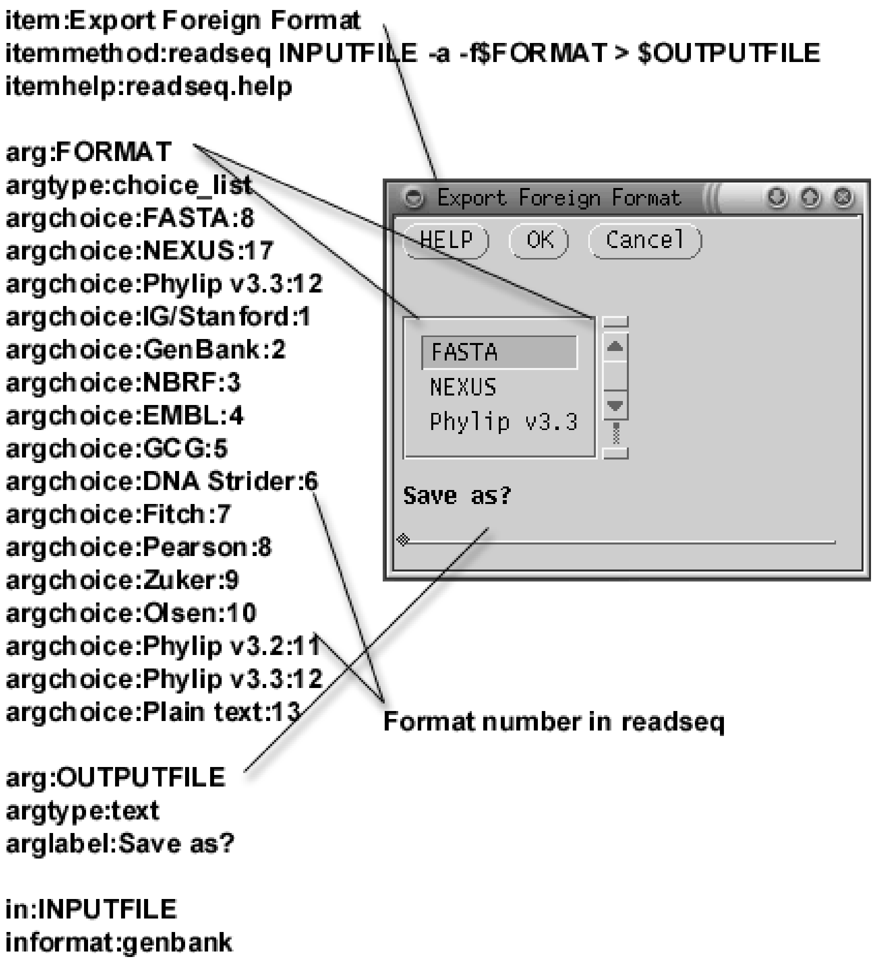
First type a text for the menu option



```
item:Import Foreign Format
itemmethod:cp $INPUTFILE OUTFILE.tmp;readseq OUTFILE.tmp -a -f2 > OUTPUTFILE;/bin/rm -f
itemhelp:readseq.help

arg:INPUTFILE
argtype:text
arglabel:Name of foreign file?

out:OUTPUTFILE
outformat:genbank
```

```
item:Import Foreign Format
```

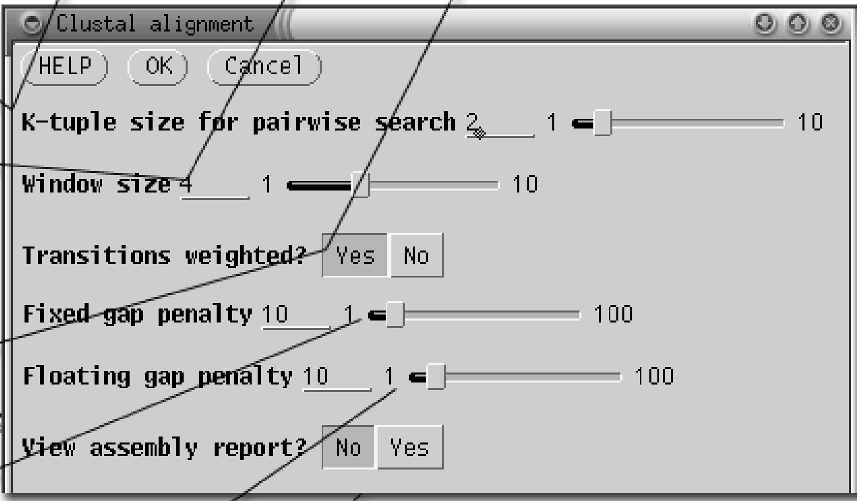
Export file:



```
item:Export Foreign Format
itemmethod:readseq INPUTFILE -a -f$FORMAT > $OUTPUTFILE
itemhelp:readseq.help

arg:FORMAT
argtype:choice_list
argchoice:FASTA:8
argchoice:NEXUS:17
argchoice:Phylip v3.3:12
argchoice:IG/Stanford:1
argchoice:GenBank:2
argchoice:NBRF:3
argchoice:EMBL:4
argchoice:GCG:5
argchoice:DNA Strider:6
argchoice:Fitch:7
argchoice:Pearson:8
argchoice:Zuker:9
argchoice:Olsen:10
argchoice:Phylip v3.2:11
argchoice:Phylip v3.3:12
argchoice:Plain text:13

arg:OUTPUTFILE
argtype:text
arglabel:Save as?

in:INPUTFILE
informat:genbank
```

Format number in readseq

```
item:Clustal alignment
itemmethod:(tr '%#' '>'<in1>clus_in;clustalw -ktup=$KTUP -win=$WIN -trans=$ TRANS
-output=PIR -infile=clus_in -align  > in1.rpt;sed "s/>DL;/#/g" < clus_in.pir> in1;
$REPORT gde in1;/bin/rm -f clus_in* in1*   )&

itemhelp:clustal_help

arg:KTUP
argtype:slider
arglabel:K-tuple size for pairwise search
argmin:1
argmax:10
argvalue:2

arg:WIN
argtype:slider
arglabel:Window size
argmin:1
argmax:10
argvalue:4

arg:Trans
argtype:chooser
arglabel:Transitions weighted?
argchoice:Yes:/TRANSIT
argchoice:No:

arg:FIXED
argtype:slider
arglabel:Fixed gap penalty
argmin:1
argmax:100
argvalue:10

arg:FLOAT
arglabel:Floating gap penalty
argtype:slider
argmin:1
argmax:100
argvalue:10

arg:REPORT
argtype:chooser
arglabel:View assembly report?
argchoice:No:
argchoice:Yes:kedit in1.rpt&

in:in1
informat:flat
insave:
```



K-tuple size for pairwise search 2 1 10

Window size 4 1 10

Transitions weighted? Yes No

Fixed gap penalty 10 1 100

Floating gap penalty 10 1 100

View assembly report? No Yes