Construction of an Integrated Database to Support Genomic Sequence Analysis

Walter Gilbert and Ross Overbeek

GenoBase Developments

One central goal of our effort is to develop an integrated database to support comparative analysis of genomes. We now call this logic-programming-based system GenoBase (the previous acronym had to be changed because another project had already used it). In Phase I of the current proposal, the goal was to produce an initial integration of DNA sequence data, protein sequence data, available data on expression of genes within *Escherichia coli* (from the Eco2dbase project), and currently available data on metabolism. In fact, we have achieved a somewhat broader integration of available data, in large part because of the assistance from collaborators at NIH, George Mason University, members of the DBEMP project in Russia, and researchers at the Swedish Institute of Computer Science.

The central goal of an integration following the architecture that we have proposed is to make a wide variety of biological data available through convenient access for users. Two issues need to be directly addressed:

1. It must be possible to easily include new forms of data as they become available. For example, during the period since the original proposal was written, substantial amounts of data in the form of the Blocks database, the DBEMP data relating to metabolism, and newly developed phylogenetic trees have become available, and all are directly relevant to interpretation of sequence data. Anyone familiar with the effort normally required to integrate diverse categories of data, especially if a commitment is made to cast the data in a relational form (which we do not), will realize that most commonly-used technologies require substantial resources. We have explicitly attempted to develop a technology that reduces the cost of accessing and operating data, without expending the resources required to achieve a completely consistent, normalized representation of the diverse data items.
2. It must be possible to easily navigate through the ensemble of objects described within the database. In this respect, our effort is based on the same intellectual foundations that similar object-oriented systems utilize. Most of those systems have focused directly on producing extremely interactive, GUI-based navigation systems. Ours has focused on a complementary issue -- effective operation on the ensemble of data (rather than just display and maintenance of objects). We feel that this is an important capability and that we have unique resources at our disposal to address this issue. For example, it should be possible to rapidly answer questions like

"What patterns occur an unusually large number of times in upstream regions of genes expressed under heat shock?"

"Given a new class of promoters (such as those recently described in *Science* for the *E. coli* genome), what genes include instances in their -40 to -60 upstream regions?"

"Given a pathway under study, which of the enzymes in the pathway correspond to known protein sequences? Do any of these sequences have a known crystal structure? Which of the sequences correspond to known Block Groups?"

MASTER

se

DISCLAIMER

This report was prepared as an account of work sponsored by an agency of the United States Government. Neither the United States Government nor any agency thereof, nor any of their employees, make any warranty, express or implied, or assumes any legal liability or responsibility for the accuracy, completeness, or usefulness of any information, apparatus, product, or process disclosed, or represents that its use would not infringe privately owned rights. Reference herein to any specific commercial product, process, or service by trade name, trademark, manufacturer, or otherwise does not necessarily constitute or imply its endorsement, recommendation, or favoring by the United States Government or any agency thereof. The views and opinions of authors expressed herein do not necessarily state or reflect those of the United States Government or any agency thereof.

DISCLAIMER

Portions of this document may be illegible in electronic image products. Images are produced from the best available original document.

To effectively answer such questions, one requires not only the ability to navigate between collections of diverse objects, but also to be able to apply a rich set of operators to extract the relevant information.

The current GenoBase system has been enhanced as follows:

1. We have worked on increasing our ability to handle larger volumes of data within the flexible framework demonstrated on our earlier prototypes. This effort has culminated in our use of a standard set of database routines developed by the Swedish Institute of Computer Science and supported under the Sicstus Prolog Project at SICS. We have created a single large database at NIH that includes all of the data from EMBL, the Swiss Protein Data Bank, the Enzyme Data Bank, and some limited data on metabolism from the DBEMP. Our intent is to work on developing a stable system including a richer set of objects, and then to make the data available on an object-server accessible through the World Wide Web. This effort has been done in collaboration with Ron Taylor at NIH and is progressing rapidly.
2. We have worked closely with E. Selkov of the DBEMP to encode his data on metabolism into objects that can be used to integrate many forms of data in existing databases. Indeed, the wealth of data provided within the DBEMP will almost certainly play a central role in any effective integration in the future, and our initial efforts should be of use to many international groups in their efforts to achieve higher levels of integration.
3. We have developed a number of Windows-based user interfaces that allow relatively convenient access to the objects within GenoBase. The first effort was based on technology developed within the GDE effort and allowed us to explore generalized navigation tools, as well as a framework for applying operators and collecting the generated results (Appendix A). A second effort has built upon this experience and has produced an initial interface based on TCL/TK. We plan on making this version available at a number of sites; it will be used to gain experience in what new operators are required and what new categories of objects will be needed to address the needs of practicing biologists.

Achievements of the Mycoplasma Project

At the termination of the project at Harvard, we had accumulated over a million raw bases of *Mycoplasma capricolum* sequence (1,039,095 bp) with a total of over a quarter of a million linear bases (267,686 bp). We sequenced 1,505 random clones from the organism, producing 187,309 raw bases of sequence. These assembled into 1,032 unique starting points of 137,372 linear bases.

We are currently "walking on" 381 contigs into which 1198 of the original random clones have assembled. During the last year of the project we accumulated 901,723 bases of raw walking data, which has assembled into 215,236 bases of linear walking sequence with an average of 4.3 fold coverage. We have some 52,450 bases comprising the remaining 308 unused starting points.

We have done some preliminary analysis of the random sequences using the Blastx search algorithm against the Non-Redundant Genbank Database available at the National Library of Medicine. Blastx results indicate that 14% of the random sequences have similarity to proteins in the database (P value $< 10^{-6}$). On the other hand, of the 292 open reading frames longer than 30 amino acids found in the 381 contigs, fully 53% of them have similarities to proteins with P value $< 10^{-6}$ (see Appendix A). These results indicate that random one-pass sequencing may

have little identifiable information content; thus, obtaining high-quality, accurate sequences becomes extremely important.

We have roughly classified the types of similarities found using Blast. In summary, we found the following:

1. Similarities to proteins involved the **replication apparatus** and DNA repair enzymes, such DNA binding proteins, gyrase, ligase, and polymerase. These similarities were expected.
2. Similarities to proteins of the **translational** and **transcriptional apparatus**, such as many of the tRNA synthetases, RNA polymerase, and initiation/elongation factors. We note that over half of the tRNA synthetases have been identified. This work suggests that a significant portion of the genome has already been sequenced (see below).
3. Several examples of what appear to be **regulatory** proteins. Many similarities are to higher organisms, however; thus, their function in *Mycoplasma* is unknown.
4. Similarities to **transport** proteins. These are to be expected because this organism is an extracellular parasite and must import most of its metabolic precursors.
5. Similarities to both **catabolic** enzymes and **anabolic** enzymes. Most of these are predictable a priori knowing the biochemistry of the organism.
6. Similarities to proteins involved in pathogenesis. Some, like the P1 adhesion protein, are expected. Others, like hemolysin, are unexpected.
7. Numerous anomalous similarities, many to higher eukaryotes. These are totally unexpected and need further investigation to determine the validity of the similarity score and the validity of the alignment.

We analyzed the DNA of the organism on CHEF gels, exploring the rare cutting pattern to identify large regions of the organism. The *Mycoplasma* DNA has an apparent genome size of 1 megabase on pulse field gels using yeast chromosomes as molecular weight makers, but we believe these DNA sizes to be exaggerated because of the high AT content of the organism. We have recently enumerated the rare restriction cuts that have been sequenced (Table I). Interestingly, we appear to have identified about 35% of the known rare restriction sites in the organism in 215,000 bases; this extrapolates to a genome size of only 765 kb, much smaller than the estimates determined from the pulse field gels.

Table I Rare Cutting Sites

Recognition Site			Expected	Observed
Fsp I	TGCGCA	5	2	
Bgl I	GCCNNNNGCC	6	0	
Apa I	GGGCCC	2	1	
BssH II	GCGCGC	1	0	
Sal I	GTCGAC	2		
Sma I	CCCGGG	2	1	
Xho I	CTCGAG	2	1	
TOTAL			20	7

Graphic User Interface to Support GenoBase Queries

One of the major obstacles to the routine use of Logic programming environments by biologists is the difficulty in understanding the underlining data structures of the environment. GenoBase is a prolog based environment that links many different databases using the concept of typed objects. We found that we need a versatile querying environment to efficiently integrate the data from the Mycoplasma Genome Project at Harvard University into GenoBase. We have proceeded to develop a graphic user interface at Harvard University based on EZshelltool to generate prolog predicates to query the integrated database environment. We will describe the basic format of the GUI and give several examples of typical GenoBase queries. The overview that is described consists of opening a local shelltool and running the GenoBase environment on a remote machine.

EZshelltool

EZshelltool is an X Windows-based graphic user interface which allows the seamless integration of functions into a shelltool. This environment is based on the linkage of external programs into the shelltool by a user-expandable menu system and is supported on SunTM and DECTM workstations. There is no limitation to the number of external functions that can be linked to the interface. This user-defined menu system allows the customization of an environment with very little effort. We have used this development tool to prototype a graphic user interface to generate prolog predicates that query GenoBase.

Overview of GenoBase Interface

One initially starts ezshelltool on their local machine and obtains a window with various menus and a standard Xwindows based shell (figure 1). One then telnetts to the remote site and uses the **Startup** menu to start prolog (figure 2) and load GenoBase (figure 3). In all instances, clicking on the menu items writes a prolog predicate to standard out (the shelltool) and the predicate is evaluated.

The main window to start a query is evoked by clicking on the **Object** menu (figure 4). In the first example we will search E.coli for all coding sequences. One selects the **By Name, Type, Genome** button and then selects the **Type** of object from the pop up list (figure 5) as **cds** (coding sequence) and the **Genome** from another pop up list as **E.coli** (figure 6). One now has all the information to generate a valid GenoBase query and clicks **OK** at the top of the window (figure 7). As stated previously, this then generates a prolog query and writes it to standard out to be interpreted by the prolog process running on the remote machine. The resulting output is directed back to standard in which is the ezshelltool display.

In the second example, we select **Special Objects** as the method to pick objects and select which from the pop up list (figure 8). We also choose to save the set of objects retrieved by saving the results into the **HeatShock** variable (figure 9). As before the predicate is generated and evaluated by the prolog process.

The system contains a full **GenoBase Help** function, an example of which is given on figure 10. We finally demonstrate the utility of both the GUI environment and the GenoBase system itself. Here we have asked to enumerate all the **sequence_fragments** from **M. capricolum** that have a link from **DNA to peptide to enzyme to enzyme pathway** (figure 11). The example shows the results of the first item in the return list that has a link to the Electron Transport Chain pathway (**ETC_1**). It can be seen that it would be difficult to remember the syntax of the predicate as well as difficult to type it in without any errors. The GUI completely

alleviates these problems and make the system relatively trivial to use. We also include a printout of the entire menu system (Appendix A).

New GUI Interface

The main problem with the EZshell interface is the lack of running menus, that is the pop up lists are too large to be accommodated on the screen. This is especially true when attempting to load in all genomes in GenBank. It is in this light that the GUI prototyped here is being converted to TCL/TK interface. This should address the major problem with the present system and result in a very facile tool that will allow biologist to generate ad hoc GenoBase queries.

Future Analysis and Annotation of Raw Sequence

We have developed the GenoBase system to support interpretation of genomic sequence data. To effectively use the system to analyze data like that produced in the *Mycoplasma capricolum* sequencing effort, one must now produce an initial set of annotations that identify putative CDSs, regulatory signals, and so forth.

To this end, we have begun an extensive effort to create a system that would function as follows:

1. First, the sequence is automatically submitted to a suite of available tools (such as Blast, Fasta, Blocks, Genmark, and Blaize). This process involves a combination of locally maintained tools and access to available servers over the network; it is all achieved without manual intervention. The results from the tools are translated into Prolog facts asserting specific properties (such as similarities to known sequence and putative CDSs from tools like Genmark).

This effort clearly requires building on the rich set of tools that have been developed by other researchers to address precisely this problem. We have had contact with a number of the groups offering such services and have received several useful suggestions. In the case of Genmark, we have formed a collaboration in which we exchange initial analysis in order to gain more insight into the capabilities of each available tool.

2. The encoded output from the tools, much of which is quite irrelevant, must then be analyzed and used to construct a coherent set of annotations. This work, we believe, is best done within the context of high-level tools and requires direct access to the capabilities offered by a system such as GenoBase. Specifically, not only does such an annotation system produce input for storage and analysis within GenoBase, it also depends on the flexible access provided by GenoBase to develop an effective integration of the output of the available suite of tools.

We expect that this system will be fully operational by the end of Phase II of the proposal. We believe that it effectively complements many aspects of our efforts in developing GenoBase, and directly supports the interpretation of sequence produced by the *Mycoplasma capricolum* sequencing effort.

FIGURE 1

FIGURE 2

FIGURE 3

FIGURE 4

<input type="button" value="HELP"/> <input type="button" value="OK"/> <input type="button" value="Cancel"/>		Objects	
Saving Result <input type="button" value="no"/> <input type="button" value="yes"/>		<input type="button" value="Set of Objects"/> Explanation of Picking Objects	
Save Into # _____			
Howto Pick Object? <input type="button" value="By Name, Type, Genome"/> <input type="button" value="Saved Variable"/> <input type="button" value="Special Objects"/>			
Special Objects <input type="button" value="v"/> Sequenced Heat Shock Gene (E.coli)			
Saved Variable # _____			
Type <input type="button" value="v"/> No			
Select By Name? <input type="button" value="no"/> <input type="button" value="yes"/>			
Name _____			
Select By Genome? <input type="button" value="no"/> <input type="button" value="Yes"/>			
Which Genome? <input type="button" value="v"/> No			
Condition(s) <input type="button" value="none"/> <input type="button" value="require attribute"/> <input type="button" value="require structure (of peptide)"/> <input type="button" value="restrict by condition"/>			
Must Have Attribute _____			
Restricting Condition <input type="button" value="A"/> <input type="button" value="A and B"/> <input type="button" value="A or B"/>			
Condition A _____			
Condition B _____			
Cross? <input type="button" value="v"/> No			
Cross Again? <input type="button" value="No"/> <input type="button" value="Yes"/>			
Relationship <input type="button" value="v"/> No			
Cross Again? <input type="button" value="No"/> <input type="button" value="Yes"/>			
Relationship? <input type="button" value="v"/> No			

FIGURE 5

Saving Result

Save Into

Objects

Explanation of Picking Objects

How to Pick Object?

Special Objects

☒ Sequenced Heat Shock Gene (E.coli)

Saved Variable

*

No	
	Mycoplasma gallisepticum
Archaeoglobus fulgidus	Mycoplasma genitalium
Bac_subtilis	Mycoplasma hominis
Bacillus	Mycoplasma hyopneumoniae
Bacillus pasteurii	Mycoplasma hyorhinis
Clostridium botulinum	Mycoplasma incognitus
Clostridium tetani	Mycoplasma mycoides
Cyanobacterium nostoc	Mycoplasma neurolyticum
Cyanobacterium synochocystis	Mycoplasma orale
Cytophaga lytica	Mycoplasma pneumoniae
Desulfurococcus mobilis	Mycoplasma pulmonis
E.coli	Mycoplasma salivarium
Halobacterium cutirubrum	Mycoplasma sp.
Halobacterium halobium	Mycoplasma synoviae
Klebsiella pneumoniae	Mycoplasma-like organism
Legionella pneumophila	Mycoplasma-like sp.
Methanobacterium bryantii	Sulfolobus acidocaldarius
Methanobacterium formicicum	Sulfolobus shibatae
Methanococcus vannielii	Sulfolobus solfataricus
Methanococcus voltae	Thermococcus celer
Mycobacterium leprae	Thermococcus litoralis
Mycobacterium tuberculosis	Thermoplasma acidophilum
Mycoplasma capricolum	Thermotoga maritima
Mycoplasma collis	Vaccinia
Mycoplasma fermentans	phage_T7
Mycoplasma flocculare	phage_lambda

FIGURE 6

HELP) OK) Cancel)	
No	modified_base
-10_signal	mutation
-35_signal	nucleotide_alignment
3_prime_UTR	old_sequence
5_prime_UTR	operon
CAAT_signal	pdb
LTR	peptide
abstract_cds	peptide_alignment
allele	phylo_tree
attenuator	polyA_signal
cds	precursor_RNA
cellular	prim_transcript
chromosome	promoter
clone	prosite
compound	prosite_doc
conflict	protein_bind
eco2dbase	rRNA
enhancer	rbs
enzyme	reaction
exon	rep_origin
genome	repeat_region
IDNA	repeat_unit
insertion_seq	scrRNA
Intron	sequence_fragment
mRNA	sig_peptide
map	site
mat_peptide	stem_loop
misc_RNA	tRNA
misc_binding	tata_signal
misc_difference	terminator
misc_feature	transit_peptide
misc_recomb	transposon
misc_signal	unsure
misc_structure	variation

Objects

Set of Objects

Explanation of Picking Objects

Saved Variable	Special Objects
structure (of peptide)	restrict by condition

FIGURE 7

HELP OK Cancel

Saving Result ☐ no ☐ yes

Save Into

Howto Pick Object? ☐ By Name ☐ Type ☐ Genome ☐ Saved Variable ☐ Special Objects

Special Objects
☒ Sequenced Heat Shock Gene (E.coli)

Saved Variable

Type
☒ cds

Select By Name? ☐ no ☐ yes

Name

Select By Genome? ☐ no ☐ Yes

Which Genome?
☒ E.coli

Condition(s) ☐ none ☐ require attribute ☐ require structure (of peptide) ☐ restrict by condition

Must Have Attribute

Restricting Condition ☐ A ☐ A and B ☐ A or B

Condition A

Condition B

Cross?
☒ No

Cross Again? ☐ No ☐ Yes

Relationship
☒ No

Cross Again? ☐ No ☐ Yes

Relationship?
☒ No

Objects

Set of Objects

Explanation of Picking Objects

FIGURE 8

Saving Result

Save Info
 #HeatShock

Objects

Set of Objects

Explanation of Picking Objects

How to Pick Object?

Special Objects

☒ Sequenced Heat Shock Gene (E.coli)

Sequenced Heat Shock Gene (E.coli)	Pathway Cysteine-Synthesis_2
Sequenced Gene Ind. by phosphate starvation (E.coli)	Pathway Glutamate-Synthesis_1
Sequenced Gene Ind. by cadmium chloride (E.coli)	Pathway Histidine-Synthesis_1
Sequenced Gene Ind. by quinone ACDO (E.coli)	Pathway Homoserine-Synthesis_1
Sequenced Gene Ind. by hydrogen peroxide (E.coli)	Pathway Isoleucine-Synthesis_1
Sequenced Gene Ind. by Isoleucine starvation (E.coli)	Pathway Lysine-Synthesis_1
Sequenced Gene Ind. by cold shock (E.coli)	Pathway Methionine-Synthesis_1
Sequenced Gene Ind. by shift to anaerobic (E.coli)	Pathway Methionine-Synthesis_2
Sequenced Gene Ind. by shift to aerobic (E.coli)	Pathway Phenylalanine-Synthesis_1
Pathway ETC_1	Pathway Proline-Synthesis_1
Pathway ETC_2	Pathway Serine_1
Pathway PAD-Synthesis_1	Pathway Threonine-Synthesis_1
Pathway THF-Synthesis_1	Pathway Tryptophan-Synthesis_1
Pathway NAD-Synthesis_1	Pathway Tyrosine-Synthesis
Pathway NAD-Synthesis_2	Pathway Valine-Synthesis_1
Pathway CoA-Synthesis_1	Pathway Valine-Synthesis_2
Pathway TCA_1	Pathway Creatine-Synthesis_1
Pathway Fat-Degradation_1	Pathway Gluconeogenesis_1
Pathway Palm/acyl-ACP-Synthesis_1	Pathway ATP-Synthesis_1
Pathway Glycolysis_1	Pathway Purine-Synthesis_1
Pathway Pentose-Shunt_1	Pathway GTP-Synthesis_1
Pathway Arginine-Synthesis_1	Pathway IMP-Synthesis_1
Pathway Chorismate-Synthesis_1	Pathway Pyrimidine-Synthesis_1
Pathway Cysteine-Synthesis-1	

Cross Again?

Relationship?
☒ No

FIGURE 9

<input type="button" value="HELP"/> <input type="button" value="OK"/> <input type="button" value="Cancel"/>		<div style="display: inline-block; text-align: center;"> Objects <input type="button" value="Set of Objects"/> Explanation of Picking Objects </div>	
Saving Result <input type="button" value="no"/> <input type="button" value="yes"/>			
Save Into <input type="text" value="#HeatShock"/>			
How to Pick Object? <input type="button" value="By Name, Type, Genome"/> <input type="button" value="Saved Variable"/> <input type="button" value="Special Objects"/>			
Special Objects ☒ Sequenced Heat Shock Gene (E.coli)			
Saved Variable <input type="text" value="#"/>			
Type ☒ No			
Select By Name? <input type="button" value="no"/> <input type="button" value="yes"/>			
Name <input type="text"/>			
Select By Genome? <input type="button" value="no"/> <input type="button" value="Yes"/>			
Which Genome? ☒ No			
Condition(s) <input type="button" value="none"/> <input type="button" value="require attribute"/> <input type="button" value="require structure (of peptide)"/> <input type="button" value="restrict by condition"/>			
Must Have Attribute <input type="text"/>			
Restricting Condition <input type="button" value="A"/> <input type="button" value="A and B"/> <input type="button" value="A or B"/>			
Condition A <input type="text"/>			
Condition B <input type="text"/>			
Cross? ☒ No			
Cross Again? <input type="button" value="No"/> <input type="button" value="Yes"/>			
Relationship ☒ No			
Cross Again? <input type="button" value="No"/> <input type="button" value="Yes"/>			
Relationship? ☒ No			

Objects ▾ Save Last ▾ Clear Variable ▾ Bindings ▾ DNA sequence ▾
 Peptide sequence ▾ Structure ▾ Similarity ▾ StartUp ▾ Set Operations ▾
 GenoBase Help ▾ Done ▾ next ▾ Remaining ▾

```
|: help(sim_pep).
```

```
----- Help on sim_pep -----
```

```
Examples:
```

```
id(c000)*dna_to_simpep with X in X*peptide_to_enzyme
  produces a peptide similar to c000 annotate with a
  corresponding enzyme.
```

```
align_to_similarity(similarity(id(c000))).
  produces an alignment annotated with the objects involved.
  The alignment corresponds to a region of similarity identified
  by blast and refined by a local similarity program. Sometimes
  blast hits are merged into a longer region of similarity.
```

```
gde(export(AnnotatatedAlignment))
  uses gde to display an annotated alignment such as that
  produced by align_to_similarity/1.
```

```
pathway(obj(sequence_fragment,'M.capricolum')
  *dna_to_simpep*peptide_to_enzyme*enzyme_to_reaction).
```

```
  produces one by one pathways in which an enzyme shows
  similarity to a M.capricolum sequence fragment.
```

```
|: align_to_similarity(similarity(id(c000))).
```

```
! Syntax error
```

```
! between lines 317 and 318
```

```
! align_to_similarity(similarity(id(c000)))
```

```
! <<here>>
```

```
! )
```

```
|: align_to_similarity(similarity(id(c000))).
```

```
! Syntax error
```

```
! between lines 323 and 324
```

```
! align_to_similarity(similarity(id(c000)))
```

```
! <<here>>
```

FIGURE 11

EzShellTool

```
l: obj([type=sequence_fragment,genome='M.capricolum']) with X in
pathway(X*dna_to_simpep*peptide_to_enzyme*enzyme_to_reaction).
```

```
ANNOTATE sequence_fragment c002 of M.capricolum
```

```
  dna_sequence: of length 1274
```

```
  Comments:
```

```
{
  type          "DNA"
  name          "c002."
  sequence-ID   "chloro:0:724399808"
  sequence      "ctaaattatttttttaataatttaagtgaactatcaatagtaattaaganctaaaaaan
nGTTATATTCACTTTTAATAGTTAATAATGGTAATAATTCATAATTACTCCTTATATTA
Attttagtgagttttttatagtttttaaaataataactagttataagttaaattaaatctaa
aattaatactaaaacattaaaatatactattaaaaattataagtaaagtgatataatttt
aacactgtaaaaataaaaagaaatttggtataaaaatgattaaaaaaacaaaaataaactt
actaattttatctactttttcatctatattatcttcattttatatttttttaaaaatattag
taattctagttcggatcaa-attacaaacttttataaaacaaaaacaagatgaaatagctg
aaataaaaattttcaataaacaagatctagaatattattctttaagagatgaatatattt
tatttaatacaa--tatcaagtaaacaggatTATGTTgAGATTtgcTTCAATTAAATC
ACTTgAAAGTGCtttaaatGcT-agcaAaTAATGAAATGTaCGACTtttctgaagcaagca
tAAGTGTtttAAatGAg-CAttgggTTgCTgatgggtggttt-tttagaacatttgataa
tttattanatagaaatggaattgcttttgaaatctgacttttagatttGGcgntcTTtACTA
ttttccaaatactggaatatattatgataagttatta-gatctttataaaaagtaaatata
ttactaattttatctaataaactagaagcgtatagtttttatagatatagtttttagata
atcttaaacaaatcaaagaacatatttcaaaacattctgctttatttggttggtattgatc
attgacaagtagttaaaaacaaagttaacaaacaagaaacttttcaaattacttctggat
aatgggagctcatgctggttagtataattgggttgagatgataattatattaatgatgata
attctaaaggtgctttttatagtttttaaatagtgatggattttatgataataatgatggg
ttaattatttaccttataattcctcaactatttatttagacctacaagggtataaattta
ttggtgataaattaattagttcaaaaggcattaattcttcaattaaaaacaaatatttta
attactacaatacaaaaatTAGAAAAAAATATACAgaaactnnncattaaatcaaaatgt
ttttagttgagatgatcaagt"
  contig        "c002"
  strandedness  1
  direction     1
  offset        -774
  comments      "
Used name: cons.CF130-2_8B.Ed
#left_Status(MSTtoolow) changed to [checked] by AddContig on 11/13/92 9:13:37
92091432 c002 FSP: A B C D E(++) BGL: A(+) B C D Control_Bands: 1(++) 10/14/92"
  group-ID      0
  walk          "FALSE"
  left_status   "checked"
  FSPfragment   "E"
  BGLfragment   "A"
}
```

```
=== BY reaction ETC_1
```

```
===
```

APPENDIX A

EZshelltool Menus for GenoBase Queries

menu:Objects

item:Set of Objects

#itemmethod:\$Save (O = require(\$Obj,X in \$Cond), [O \$RL]).

itemmethod:\$Save retrieve(\$Obj,X in \$Cond, [\$CR1 \$CR2 \$CR3])..

arg:Save

arglabel:Saving Result

argtype:chooser

argchoice:no:

argchoice:yes:\$SName :=

arg:SName

arglabel:Save Into

argtype:text

argvalue:#

arg:Obj

argtype:chooser

arglabel:Howto Pick Object?

argchoice:By Name, Type, Genome:obj([type=' \$Type', name=' \$Nm',genome=' \$Genome'])

argchoice:Saved Variable:last(\$SaveName2)

argchoice:Special Objects:\$Special

arg:Special

argtype:choice_menu

arglabel:Special Objects

argchoice:Sequenced Heat Shock Gene (E.coli):#exp_ecoli('50C')

argchoice:Sequenced Gene Ind. by phosphate starvation (E.coli):#exp_ecoli('PSI')

argchoice:Sequenced Gene Ind. by cadmium chloride (E.coli):#exp_ecoli('Cd')

argchoice:Sequenced Gene Ind. by quinone ACDQ (E.coli):#exp_ecoli('QN')

argchoice:Sequenced Gene Ind. by hydrogen peroxide (E.coli):#exp_ecoli('HP')

argchoice:Sequenced Gene Ind. by isoleucine starvation (E.coli):#exp_ecoli('ILE')

argchoice:Sequenced Gene Ind. by cold shock (E.coli):#exp_ecoli('10C')

argchoice:Sequenced Gene Ind. by shift to anaerobic (E.coli):#exp_ecoli('O2-')

argchoice:Sequenced Gene Ind. by shift to aerobic (E.coli):#exp_ecoli('O2')

argchoice:Pathway ETC_1:obj([type=reaction,name='ETC_1'])

argchoice:Pathway ETC_2:obj([type=reaction,name='ETC_2'])

argchoice:Pathway FAD-Synthesis_1:obj([type=reaction,name='FAD-Synthesis_1'])

argchoice:Pathway THF-Synthesis_1:obj([type=reaction,name='THF-Synthesis_1'])

argchoice:Pathway NAD-Synthesis_1:obj([type=reaction,name='NAD-Synthesis_1'])

argchoice:Pathway NAD-Synthesis_2:obj([type=reaction,name='NAD-Synthesis_2'])

argchoice:Pathway CoA-Synthesis_1:obj([type=reaction,name='CoA-Synthesis_1'])

argchoice:Pathway TCA_1:obj([type=reaction,name='TCA_1'])

argchoice:Pathway Fat-Degradation_1:obj([type=reaction,name='Fat-Degradation_1'])

argchoice:Pathway Palmitoyl-ACP-Synthesis_1:obj([type=reaction,name='Palmitoyl-ACP-Synthesis_1'])

argchoice:Pathway Glycolysis_1:obj([type=reaction,name='Glycolysis_1'])

argchoice:Pathway Pentose-Shunt_1:obj([type=reaction,name='Pentose-Shunt_1'])

argchoice:Pathway Arginine-Synthesis_1:obj([type=reaction,name='Arginine-Synthesis_1'])

argchoice:Pathway Chorismate-Synthesis_1:obj([type=reaction,name='Chorismate-Synthesis_1'])

argchoice:Pathway Cysteine-Synthesis_1:obj([type=reaction,name='Cysteine-Synthesis_1'])

argchoice:Pathway Cysteine-Synthesis_2:obj([type=reaction,name='Cysteine-Synthesis_2'])

argchoice:Pathway Glutamate-Synthesis_1:obj([type=reaction,name='Glutamate-Synthesis_1'])

argchoice:Pathway Histidine-Synthesis_1:obj([type=reaction,name='Histidine-Synthesis_1'])

argchoice:Pathway Homoserine-Synthesis_1:obj([type=reaction,name='Homoserine-Synthesis_1'])

argchoice:Pathway Isoleucine-Synthesis_1:obj([type=reaction,name='Isoleucine-Synthesis_1'])

argchoice:Pathway Lysine-Synthesis_1:obj([type=reaction,name='Lysine-Synthesis_1'])

argchoice:Pathway Methionine-Synthesis_1:obj([type=reaction,name='Methionine-Synthesis_1'])

argchoice:Pathway Methionine-Synthesis_2:obj([type=reaction,name='Methionine-Synthesis_2'])

argchoice:Pathway Phenylalanine-Synthesis_1:obj([type=reaction,name='Phenylalanine-Synthesis_1'])

argchoice:Pathway Proline-Synthesis_1:obj([type=reaction,name='Proline-Synthesis_1'])

argchoice:Pathway Serine_1:obj([type=reaction,name='Serine_1'])

argchoice:Pathway Threonine-Synthesis_1:obj([type=reaction,name='Threonine-Synthesis_1'])

argchoice:Pathway Tryptophan-Synthesis_1:obj([type=reaction,name='Tryptophan-Synthesis_1'])

argchoice:Pathway Tyrosine-Synthesis:obj([type=reaction,name='Tyrosine-Synthesis'])

argchoice:Pathway Valine-Synthesis_1:obj([type=reaction,name='Valine-Synthesis_1'])

argchoice:Pathway Valine-Synthesis_2:obj([type=reaction,name='Valine-Synthesis_2'])
argchoice:Pathway Creatine-Synthesis_1:obj([type=reaction,name='Creatine-Synthesis_1'])
argchoice:Pathway Gluconeogenesis_1:obj([type=reaction,name='Gluconeogenesis_1'])
argchoice:Pathway ATP-Synthesis_1:obj([type=reaction,name='ATP-Synthesis_1'])
argchoice:Pathway Purine-Synthesis_1:obj([type=reaction,name='Purine-Synthesis_1'])
argchoice:Pathway GTP-Synthesis_1:obj([type=reaction,name='GTP-Synthesis_1'])
argchoice:Pathway IMP-Synthesis_1:obj([type=reaction,name='IMP-Synthesis_1'])
argchoice:Pathway Pyrimidine-Synthesis_1:obj([type=reaction,name='Pyrimidine-Synthesis_1'])

arg:SaveName2
arglabel:Saved Variable
argtype:text
argvalue:#

arg:Type
argtype:choice_menu
arglabel:Type
argchoice:No:unknown
argchoice:-10_signal:-10_signal
argchoice:-35_signal:-35_signal
argchoice:3_prime_UTR:3_prime_UTR
argchoice:5_prime_UTR:5_prime_UTR
argchoice:CAAT_signal:CAAT_signal
argchoice:LTR:LTR
argchoice:abstract_cds:abstract_cds
argchoice:allele:allele
argchoice:attenuator:attenuator
argchoice:cds:cds
argchoice:cellular:cellular
argchoice:chromosome:chromosome
argchoice:clone:clone
argchoice:compound:compound
argchoice:conflict:conflict
argchoice:eco2dbase:eco2dbase
argchoice:enhancer:enhancer
argchoice:enzyme:enzyme
argchoice:exon:exon
argchoice:genome:genome
argchoice:idNA:idNA
argchoice:insertion_seq:insertion_seq
argchoice:intron:intron
argchoice:mRNA:mRNA
argchoice:map:map
argchoice:mat_peptide:mat_peptide
argchoice:misc_RNA:misc_RNA
argchoice:misc_binding:misc_binding
argchoice:misc_difference:misc_difference
argchoice:misc_feature:misc_feature
argchoice:misc_recomb:misc_recomb
argchoice:misc_signal:misc_signal
argchoice:misc_structure:misc_structure
argchoice:modified_base:modified_base
argchoice:mutation:mutation
argchoice:nucleotide_alignment:nucleotide_alignment
argchoice:old_sequence:old_sequence
argchoice:operon:operon
argchoice:pdb:pdb
argchoice:peptide:peptide
argchoice:peptide_alignment:peptide_alignment
argchoice:phylo_tree:phylo_tree
argchoice:polyA_signal:polyA_signal
argchoice:precursor_RNA:precursor_RNA
argchoice:prim_transcript:prim_transcript
argchoice:promoter:promoter
argchoice:prosite:prosite
argchoice:prosite_doc:prosite_doc

argchoice:protein_bind:protein_bind
argchoice:rRNA:rRNA
argchoice:rbs:rbs
argchoice:reaction:reaction
argchoice:rep_origin:rep_origin
argchoice:repeat_region:repeat_region
argchoice:repeat_unit:repeat_unit
argchoice:scRNA:scRNA
argchoice:sequence_fragment:sequence_fragment
argchoice:sig_peptide:sig_peptide
argchoice:site:site
argchoice:stem_loop:stem_loop
argchoice:tRNA:tRNA
argchoice:tata_signal:tata_signal
argchoice:terminator:terminator
argchoice:transit_peptide:transit_peptide
argchoice:transposon:transposon
argchoice:unsure:unsure
argchoice:variation:variation

arg:Nm
argtype:chooser
arglabel:Select By Name?
argchoice:no:unknown
argchoice:yes:\$ObjName

arg:ObjName
argtype:text
arglabel:Name

arg:Genome
argtype:chooser
arglabel:Select By Genome?
argchoice:no:unknown
argchoice:Yes:\$OG1

arg:OG1
argtype:choice_menu
arglabel:Which Genome?
argchoice:No:unknown
argchoice:Archaeoglobus_fulgidus:Archaeoglobus_fulgidus
argchoice:Bac_subtilis:Bac_subtilis
argchoice:Bacillus:Bacillus
argchoice:Bacillus_pasteurii:Bacillus_pasteurii
argchoice:Clostridium_botulinum:Clostridium_botulinum
argchoice:Clostridium_tetani:Clostridium_tetani
argchoice:Cyanobacterium_nostoc:Cyanobacterium_nostoc
argchoice:Cyanobacterium_synechocystis:Cyanobacterium_synechocystis
argchoice:Cytophaga_lytica:Cytophaga_lytica
argchoice:Desulfurococcus_mobilis:Desulfurococcus_mobilis
argchoice:E.coli:E.coli
argchoice:Halobacterium_cutirubrum:Halobacterium_cutirubrum
argchoice:Halobacterium_halobium:Halobacterium_halobium
argchoice:Klebsiella_pneumoniae:Klebsiella_pneumoniae
argchoice:Legionella_pneumophila:Legionella_pneumophila
argchoice:Methanobacterium_bryantii:Methanobacterium_bryantii
argchoice:Methanobacterium_formicicum:Methanobacterium_formicicum
argchoice:Methanococcus_vannielii:Methanococcus_vannielii
argchoice:Methanococcus_voltae:Methanococcus_voltae
argchoice:Mycobacterium_leprae:Mycobacterium_leprae
argchoice:Mycobacterium_tuberculosis:Mycobacterium_tuberculosis
argchoice:Mycoplasma_capricolum:Mycoplasma_capricolum
argchoice:Mycoplasma_collis:Mycoplasma_collis
argchoice:Mycoplasma_fermentans:Mycoplasma_fermentans
argchoice:Mycoplasma_flocculare:Mycoplasma_flocculare
argchoice:Mycoplasma_gallisepticum:Mycoplasma_gallisepticum

argchoice:Mycoplasma genitalium:Mycoplasma genitalium
argchoice:Mycoplasma hominis:Mycoplasma hominis
argchoice:Mycoplasma hyopneumoniae:Mycoplasma hyopneumoniae
argchoice:Mycoplasma hyorhinis:Mycoplasma hyorhinis
argchoice:Mycoplasma incognitus:Mycoplasma incognitus
argchoice:Mycoplasma mycoides:Mycoplasma mycoides
argchoice:Mycoplasma neurolyticum:Mycoplasma neurolyticum
argchoice:Mycoplasma orale:Mycoplasma orale
argchoice:Mycoplasma pneumoniae:Mycoplasma pneumoniae
argchoice:Mycoplasma pulmonis:Mycoplasma pulmonis
argchoice:Mycoplasma salivarium:Mycoplasma salivarium
argchoice:Mycoplasma sp.:Mycoplasma sp.
argchoice:Mycoplasma synoviae:Mycoplasma synoviae
argchoice:Mycoplasma-like organism:Mycoplasma-like organism
argchoice:Mycoplasma-like sp.:Mycoplasma-like sp.
argchoice:Sulfolobus acidocaldarius:Sulfolobus acidocaldarius
argchoice:Sulfolobus shibatae:Sulfolobus shibatae
argchoice:Sulfolobus solfataricus:Sulfolobus solfataricus
argchoice:Thermococcus celer:Thermococcus celer
argchoice:Thermococcus litoralis:Thermococcus litoralis
argchoice:Thermoplasma acidophilum:Thermoplasma acidophilum
argchoice:Thermotoga maritima:Thermotoga maritima
argchoice:Vaccinia:Vaccinia
argchoice:phage_T7:phage_T7
argchoice:phage_lambda:phage_lambda

arg:Cond
argtype:chooser
arglabel:Condition(s)
argchoice:none:1
argchoice:require attribute:has_attribute(X, '\$Attr')
argchoice:require structure (of peptide):has_direct_structure(X)
argchoice:restrict by condition:\$C1

arg:Attr
arglabel:Must Have Attribute
argtype:text

arg:C1
argtype:chooser
arglabel:Restricting Condition
argchoice:A:\$CA
argchoice:A and B:(((\$CA) /\ (\$CB))
argchoice:A or B:(((\$CA) \/ (\$CB))

arg:CA
argtype:text
arglabel:Condition A

arg:CB
argtype:text
arglabel:Condition B

arg:CR1
arglabel:Cross?
argtype:choice_menu
argchoice:No:
argchoice:abstract_to_cds:abstract_to_cds - 1
argchoice:alignment_to_peptide:alignment_to_peptide - 1
argchoice:cds_to_abstract:cds_to_abstract - 1
argchoice:cds_to_eco2dbase:cds_to_eco2dbase - 1
argchoice:cds_to_enzyme:cds_to_enzyme - 1
argchoice:cds_to_peptide:cds_to_peptide - 1
argchoice:chromosome_to_genome:chromosome_to_genome - 1
argchoice:chromosome_to_map:chromosome_to_map - 1
argchoice:class_to_enzyme:class_to_enzyme - 1

argchoice:cofactor_to_reaction:cofactor_to_reaction - 1
argchoice:doc_to_prosite:doc_to_prosite - 1
argchoice:eco2dbase_to_cds:eco2dbase_to_cds - 1
argchoice:eco2dbase_to_enzyme:eco2dbase_to_enzyme - 1
argchoice:eco2dbase_to_peptide:eco2dbase_to_peptide - 1
argchoice:enzyme_to_cds:enzyme_to_cds - 1
argchoice:enzyme_to_class:enzyme_to_class - 1
argchoice:enzyme_to_eco2dbase:enzyme_to_eco2dbase - 1
argchoice:enzyme_to_peptide:enzyme_to_peptide - 1
argchoice:enzyme_to_reaction:enzyme_to_reaction - 1
argchoice:gene_to_map:gene_to_map - 1
argchoice:genome_to_chromosome:genome_to_chromosome - 1
argchoice:map_to_chromosome:map_to_chromosome - 1
argchoice:map_to_gene:map_to_gene - 1
argchoice:object_to_piece:object_to_piece - 1
argchoice:object_to_region:object_to_region - 1
argchoice:pathway_to_reaction:pathway_to_reaction - 1
argchoice:pdb_to_swissprot:pdb_to_swissprot - 1
argchoice:peptide_to_alignment:peptide_to_alignment - 1
argchoice:peptide_to_cds:peptide_to_cds - 1
argchoice:peptide_to_eco2dbase:peptide_to_eco2dbase - 1
argchoice:peptide_to_enzyme:peptide_to_enzyme - 1
argchoice:peptide_to_prosite:peptide_to_prosite - 1
argchoice:piece_to_object:piece_to_object - 1
argchoice:product_to_reaction:product_to_reaction - 1
argchoice:prosite_to_doc:prosite_to_doc - 1
argchoice:prosite_to_peptide:prosite_to_peptide - 1
argchoice:reaction_to_cofactor:reaction_to_cofactor - 1
argchoice:reaction_to_enzyme:reaction_to_enzyme - 1
argchoice:reaction_to_pathway:reaction_to_pathway - 1
argchoice:reaction_to_product:reaction_to_product - 1
argchoice:reaction_to_substrate:reaction_to_substrate - 1
argchoice:region_to_object:region_to_object - 1
argchoice:substrate_to_reaction:substrate_to_reaction - 1
argchoice:swissprot_to_pdb:swissprot_to_pdb - 1
argvalue:No

arg:CR2
arglabel:Cross Again?
argtype:chooser
argchoice:No:
argchoice:Yes: , \$REL2

arg:REL2
arglabel:Relationship
argtype:choice_menu
argchoice:No:
argchoice:abstract_to_cds:abstract_to_cds - 1
argchoice:alignment_to_peptide:alignment_to_peptide - 1
argchoice:cds_to_abstract:cds_to_abstract - 1
argchoice:cds_to_eco2dbase:cds_to_eco2dbase - 1
argchoice:cds_to_enzyme:cds_to_enzyme - 1
argchoice:cds_to_peptide:cds_to_peptide - 1
argchoice:chromosome_to_genome:chromosome_to_genome - 1
argchoice:chromosome_to_map:chromosome_to_map - 1
argchoice:class_to_enzyme:class_to_enzyme - 1
argchoice:cofactor_to_reaction:cofactor_to_reaction - 1
argchoice:doc_to_prosite:doc_to_prosite - 1
argchoice:eco2dbase_to_cds:eco2dbase_to_cds - 1
argchoice:eco2dbase_to_enzyme:eco2dbase_to_enzyme - 1
argchoice:eco2dbase_to_peptide:eco2dbase_to_peptide - 1
argchoice:enzyme_to_cds:enzyme_to_cds - 1
argchoice:enzyme_to_class:enzyme_to_class - 1
argchoice:enzyme_to_eco2dbase:enzyme_to_eco2dbase - 1
argchoice:enzyme_to_peptide:enzyme_to_peptide - 1
argchoice:enzyme_to_reaction:enzyme_to_reaction - 1

argchoice:gene_to_map:gene_to_map - 1
argchoice:genome_to_chromosome:genome_to_chromosome - 1
argchoice:map_to_chromosome:map_to_chromosome - 1
argchoice:map_to_gene:map_to_gene - 1
argchoice:object_to_piece:object_to_piece - 1
argchoice:object_to_region:object_to_region - 1
argchoice:pathway_to_reaction:pathway_to_reaction - 1
argchoice:pdb_to_swissprot:pdb_to_swissprot - 1
argchoice:peptide_to_alignment:peptide_to_alignment - 1
argchoice:peptide_to_cds:peptide_to_cds - 1
argchoice:peptide_to_eco2dbase:peptide_to_eco2dbase - 1
argchoice:peptide_to_enzyme:peptide_to_enzyme - 1
argchoice:peptide_to_prosite:peptide_to_prosite - 1
argchoice:piece_to_object:piece_to_object - 1
argchoice:product_to_reaction:product_to_reaction - 1
argchoice:prosite_to_doc:prosite_to_doc - 1
argchoice:prosite_to_peptide:prosite_to_peptide - 1
argchoice:reaction_to_cofactor:reaction_to_cofactor - 1
argchoice:reaction_to_enzyme:reaction_to_enzyme - 1
argchoice:reaction_to_pathway:reaction_to_pathway - 1
argchoice:reaction_to_product:reaction_to_product - 1
argchoice:reaction_to_substrate:reaction_to_substrate - 1
argchoice:region_to_object:region_to_object - 1
argchoice:substrate_to_reaction:substrate_to_reaction - 1
argchoice:swissprot_to_pdb:swissprot_to_pdb - 1
argvalue:No

arg:CR3
arglabel:Cross Again?
argtype:chooser
argchoice:No:
argchoice:Yes:, \$REL3

arg:REL3
arglabel:Relationship?
argtype:choice_menu
argchoice:No:
argchoice:abstract_to_cds:abstract_to_cds - 1
argchoice:alignment_to_peptide:alignment_to_peptide - 1
argchoice:cds_to_abstract:cds_to_abstract - 1
argchoice:cds_to_eco2dbase:cds_to_eco2dbase - 1
argchoice:cds_to_enzyme:cds_to_enzyme - 1
argchoice:cds_to_peptide:cds_to_peptide - 1
argchoice:chromosome_to_genome:chromosome_to_genome - 1
argchoice:chromosome_to_map:chromosome_to_map - 1
argchoice:class_to_enzyme:class_to_enzyme - 1
argchoice:cofactor_to_reaction:cofactor_to_reaction - 1
argchoice:doc_to_prosite:doc_to_prosite - 1
argchoice:eco2dbase_to_cds:eco2dbase_to_cds - 1
argchoice:eco2dbase_to_enzyme:eco2dbase_to_enzyme - 1
argchoice:eco2dbase_to_peptide:eco2dbase_to_peptide - 1
argchoice:enzyme_to_cds:enzyme_to_cds - 1
argchoice:enzyme_to_class:enzyme_to_class - 1
argchoice:enzyme_to_eco2dbase:enzyme_to_eco2dbase - 1
argchoice:enzyme_to_peptide:enzyme_to_peptide - 1
argchoice:enzyme_to_reaction:enzyme_to_reaction - 1
argchoice:gene_to_map:gene_to_map - 1
argchoice:genome_to_chromosome:genome_to_chromosome - 1
argchoice:map_to_chromosome:map_to_chromosome - 1
argchoice:map_to_gene:map_to_gene - 1
argchoice:object_to_piece:object_to_piece - 1
argchoice:object_to_region:object_to_region - 1
argchoice:pathway_to_reaction:pathway_to_reaction - 1
argchoice:pdb_to_swissprot:pdb_to_swissprot - 1
argchoice:peptide_to_alignment:peptide_to_alignment - 1
argchoice:peptide_to_cds:peptide_to_cds - 1

```
argchoice:peptide_to_eco2dbase:peptide_to_eco2dbase - 1
argchoice:peptide_to_enzyme:peptide_to_enzyme - 1
argchoice:peptide_to_prosite:peptide_to_prosite - 1
argchoice:piece_to_object:piece_to_object - 1
argchoice:product_to_reaction:product_to_reaction - 1
argchoice:prosite_to_doc:prosite_to_doc - 1
argchoice:prosite_to_peptide:prosite_to_peptide - 1
argchoice:reaction_to_cofactor:reaction_to_cofactor - 1
argchoice:reaction_to_enzyme:reaction_to_enzyme - 1
argchoice:reaction_to_pathway:reaction_to_pathway - 1
argchoice:reaction_to_product:reaction_to_product - 1
argchoice:reaction_to_substrate:reaction_to_substrate - 1
argchoice:region_to_object:region_to_object - 1
argchoice:substrate_to_reaction:substrate_to_reaction - 1
argchoice:swissprot_to_pdb:swissprot_to_pdb - 1
argvalue:No
```

```
#
```

```
item:Explanation of Picking Objects
itemmethod:help(picking_objects).
```

```
menu:Save Last
item:Save As
itemmethod:$Save := !.
```

```
arg:Save
arglabel:Name of Save Variable?
argtype:text
argvalue:#
```

```
menu:Clear Variable
item:Clear Variable
itemmethod:clear($Save).
```

```
arg:Save
arglabel:Name of Save Variable?
argtype:text
argvalue:#
```

```
menu:Bindings
```

```
item:Features bound to an object
itemmethod:bound_to(last($SV), $Beg, $End).
```

```
arg:SV
arglabel:User Variable Containing Object (sequence frag or chromosome)
argtype:text
argvalue:#
```

```
arg:Beg
arglabel:Offset of start of region (from 0)
argtype:text
```

```
arg:End
arglabel:Offset of end of region (from 0)
argtype:text
```

```
menu:DNA sequence
```

```
item:Sequence of Objects
itemmethod:(O = last($SV), M=dna_sequence($INT), [O,M]).
```

```
arg:SV
arglabel:User Variable Containing Objects
argtype:text
argvalue:#
```

```

arg:INT
arglabel:What Part of Object?
argtype:chooser
argchoice:all:0
argchoice:upstream:interval(0,$LF,$RF)
argchoice:downstream:interval(0,(length(O)-1) + $LF,(length(O)-1) + $RF)

arg:LF
arglabel:Pick Left Boundary
argtype:slider
argmin:-200
argmax:200
argvalue:-50

arg:RF
arglabel:Pick Right Boundary
argtype:slider
argmin:0
argmax:200
argvalue:0

item:Matching Patterns
itemmethod:(O = last($SV),M=require(matches(dna_sequence($INT),'$PT'),X in X != []),[O,M]).

arg:SV
arglabel:User Variable Containing Objects
argtype:text
argvalue:#

arg:INT
arglabel:What Part of Object?
argtype:chooser
argchoice:all:0
argchoice:upstream:interval(0,$LF,$RF)
argchoice:downstream:interval(0,(length(O)-1) + $LF,(length(O)-1) + $RF)

arg:LF
arglabel:Pick Left Boundary
argtype:slider
argmin:-200
argmax:200
argvalue:-50

arg:RF
arglabel:Pick Right Boundary
argtype:slider
argmin:0
argmax:200
argvalue:0

arg:PT
arglabel:Pattern to Search for
argtype:text

item:Kmer Analysis
itemmethod:kmers(all(dna_sequence(last($SV))),$K).

arg:SV
arglabel:User Variable Containing Objects
argtype:text
argvalue:#

arg:K
arglabel:Size of kmers (keep it under 7, probably)
argtype:text

```

```
item:Common Subsequences
itemmethod:(O1 = last($SV1), O2 = last($SV2), (O1 == O2) = 0, C=common([[ $INT1,$N1]], [ $INT2,$N
arg:SV1
arglabel:Saved Variable
argtype:text
argvalue:#

arg:INT1
arglabel:What Part of Object?
argtype:chooser
argchoice:all:0
argchoice:upstream:interval(O1,$LF1,$RF1)
argchoice:downstream:interval(O1,(length(O1)-1) + $LF1,(length(O1)-1) + $RF1)

arg:LF1
arglabel:Pick Left Boundary
argtype:slider
argmin:-200
argmax:200
argvalue:-50

arg:RF1
arglabel:Pick Right Boundary
argtype:slider
argmin:0
argmax:200
argvalue:0

arg:N1
arglabel:Minimum Occurrences
argtype:slider
argmin:1
argmax:5
argvalue:1

arg:SV2
arglabel:Saved Variable
argtype:text
argvalue:#

arg:INT2
arglabel:What Part of Object?
argtype:chooser
argchoice:all:0
argchoice:upstream:interval(O2,$LF2,$RF2)
argchoice:downstream:interval(O2,(length(O2)-1) + $LF2,(length(O2)-1) + $RF2)

arg:LF2
arglabel:Pick Left Boundary
argtype:slider
argmin:-200
argmax:200
argvalue:-50

arg:RF2
arglabel:Pick Right Boundary
argtype:slider
argmin:0
argmax:200
argvalue:0

arg:N2
```


arglabel:Minimum Occurrences
argtype:slider
argmin:1
argmax:5
argvalue:1

arg:MIN
arglabel:Minimum Length
argtype:slider
argmin:4
argmax:30
argvalue:8

item:Codon Usage
itemmethod:codon_usage(all(last(\$SV1))).

arg:SV1
arglabel:Saved Variable Containing cds Objects
argtype:text
argvalue:#

menu:Peptide sequence

item:Matching Patterns
itemmethod:(O = last(\$SV), M=require(matches(protein_sequence(O), '\$PT'), X in X =\= []), [O, val(

arg:SV
arglabel:User Variable Containing Peptides
argtype:text
argvalue:#

arg:PT
arglabel:Pattern to Search for
argtype:text

menu:Structure
item:Secondary Structure (of peptide)
itemmethod:(O=last(\$SV), [O, [protein_sequence(O), structure(O)]]).

arg:SV
arglabel:User Variable Containing Peptides
argtype:text
argvalue:#

item:Distance between amino Acids (peptide or alignment)
itemmethod:(O=last(\$SV), [O, dist(O, \$P1, \$P2)]).

arg:SV
arglabel:User Variable Containing Peptides
argtype:text
argvalue:#

arg:P1
arglabel:Position 1
argtype:text

arg:P2
arglabel:Position 2
argtype:text

menu:Similarity
item:local similarities for 2 Objects
itemmethod:(O1=last(\$SV1), O2=last(\$SV2), (O1==O2)=0, [O1,O2,local_similarity(\$I1,\$I2,\$N)]).

arg:I1
arglabel:What Part of Object?

```

argtype:chooser
argchoice:all:dna_sequence(O1)
argchoice:upstream:#dseqB(O1,$LF1,$RF1)
argchoice:downstream:#dseqB(O1,(length(O1)-1) + $LF1,(length(O1)-1) + $RF1)

arg:SV1
arglabel:Saved Variable Containing cds Objects
argtype:text
argvalue:#

arg:LF1
arglabel:Pick Left Boundary
argtype:slider
argmin:-200
argmax:200
argvalue:-50

arg:RF1
arglabel:Pick Right Boundary
argtype:slider
argmin:0
argmax:200
argvalue:0

arg:I2
arglabel:What Part of Object?
argtype:chooser
argchoice:all:dna_sequence(O2)
argchoice:upstream:#dseqB(O2,$LF2,$RF2)
argchoice:downstream:#dseqB(O2,(length(O2)-1) + $LF2,(length(O2)-1) + $RF2)

arg:SV2
arglabel:Saved Variable Containing cds Objects
argtype:text
argvalue:#

arg:LF2
arglabel:Pick Left Boundary
argtype:slider
argmin:-200
argmax:200
argvalue:-50

arg:RF2
arglabel:Pick Right Boundary
argtype:slider
argmin:0
argmax:200
argvalue:0

arg:N
arglabel:Max Number Similarities
argtype:slider
argmin:1
argmax:10
argvalue:3

item:Aligned Multiple Local Similarities
itemmethod:#lookatD(all($I1)).

arg:I1
arglabel:What Part of Objects?
argtype:chooser
argchoice:all:last($SV1) with dna_sequence(last($SV1))
argchoice:upstream:#dgseqB(last($SV1),$LF1,$RF1)
argchoice:downstream:#dgseqB(last($SV1),(length(last($SV1))-1) + $LF1,(length(last($SV1))-1)

```

arg:SV1
arglabel:Saved Variable Containing cds Objects
argtype:text
argvalue:#

arg:LF1
arglabel:Pick Left Boundary
argtype:slider
argmin:-200
argmax:200
argvalue:-50

arg:RF1
arglabel:Pick Right Boundary
argtype:slider
argmin:0
argmax:200
argvalue:0

item:Global Similarity of Peptides
itemmethod:global_similarity(all(protein_sequence(last(\$SV1)))).

arg:SV1
arglabel:Saved Variable Containing peptides
argtype:text
argvalue:#

item:Global Similarity of DNA Sequences
itemmethod:global_similarity(all(\$I1)).

arg:I1
arglabel:What Part of Objects?
argtype:chooser
argchoice:all:#dseqB(last(\$SV1))
argchoice:upstream:#dseqB(last(\$SV1), \$LF1, \$RF1)
argchoice:downstream:#dseqB(last(\$SV1), (length(last(\$SV1))-1) + \$LF1, (length(last(\$SV1))-1) +

arg:SV1
arglabel:Saved Variable Containing DNA Sequences
argtype:text
argvalue:#

arg:LF1
arglabel:Pick Left Boundary
argtype:slider
argmin:-200
argmax:200
argvalue:-50

arg:RF1
arglabel:Pick Right Boundary
argtype:slider
argmin:0
argmax:200
argvalue:0

menu:StartUp

item:Start Prolog
itemmethod:cd ~overbeek/Version4; prolog3

item:Start GenoBase
itemmethod:[toplevel], eval(macros(ross_macros),_), evalpp.

menu:Set Operations

item:Intersection
itemmethod:[member(intersection(sort(all(last(\$SV1))),sort(all(last(\$SV2)))))]].

arg:SV1
arglabel:Saved Variable For Set 1
argtype:text
argvalue:#

arg:SV2
arglabel:Saved Variable For Set 2
argtype:text
argvalue:#

item:Union
itemmethod:[member(union(sort(all(last(\$SV1))),sort(all(last(\$SV2)))))]].

arg:SV1
arglabel:Saved Variable For Set 1
argtype:text
argvalue:#

arg:SV2
arglabel:Saved Variable For Set 2
argtype:text
argvalue:#

item:Set Subtraction (X-Y)
itemmethod:[member(subtract(sort(all(last(\$SV1))),sort(all(last(\$SV2)))))]].

arg:SV1
arglabel:Saved Variable For Set 1
argtype:text
argvalue:#

arg:SV2
arglabel:Saved Variable For Set 2
argtype:text
argvalue:#

menu:GenoBase Help

item:Explanation of Help Utility
itemmethod:help(help).

item:List of Current Typed Objects
itemmethod:help(types).

item:Display Information
itemmethod:help(\$Value).

arg:Value
arglabel:Help on What?
argtype:choice_menu
argchoice:Help:help
argchoice:Object Types:types
argchoice>List Supported Relationships:relationships
argchoice:Neidhardt's eco2dbase Attributes:eco2dbase
argchoice:Valid Patterns used in match/2:patterns
argchoice:eval, evalpp, etc:using_evaluation
argchoice:Overview of "expression":semantics_of_expressions
argchoice:Structures Produced by Evaluation:structures_produced_by_eval
argchoice:Simple Operations:simple_operations
argchoice:Top Level Operations:toplevel
argchoice:Substitution of Variables:variables
argchoice:Operations on Lists:lists

argchoice:Operations on Vectors:vectors
argchoice:Operations on Annotated Objects:annotated_objects
argchoice:Operations on Objects:objects
argchoice:Operations on Points on Objects:points
argchoice:Intervals of Objects:intervals
argchoice:DNA Sequences:dna_sequence
argchoice:Common Patterns in Sequences:matching
argchoice:Adjacency on Objects:adjacent
argchoice:Structure of peptides:sec_struct
argchoice:Variants and Alignments:variants
argchoice:User Variables:uservars
argchoice:Other Object:\$OTHER

arg:OTHER
argtype:text
arglabel:Enter Other Query

menu:Done

item:Done
itemmethod:

menu:next
item:next value
itemmethod;;

menu:Remaining
item: all remaining values
itemmethod:a