ATLAS - User's Guide

Order Number: ATPD-1295

ATLAS

The Atlas Multidatabase Information Retrieval System (ATLAS), developed by the National Biomedical Research Foundation (NBRF), is a retrieval program specifically designed to access macromolecular sequence databases. This version of the program has been configured to provide simultaneous retrieval from all (or a subset) of the databases on this CD-ROM. Full support is provided for VAX/VMS, OpenVMS AXP, OSF/1 AXP, ULTRIX (RISC), SunOS, IRIX, and PC/DOS systems. A preliminary version for the Macintosh is also included.

The ATLAS CD-ROM contains the Atlas retrieval system, the FASTA database searching program, the PIR®-International Protein Sequence Database, the MIPS PATCHX Merged Protein Sequence Database, the NRL_3D Sequence-Structure Database, the ALN Protein Sequence Alignment Database, the RESID database of protein structure modifications, the JIPID *E. coli* Database and indexes to the GenBank Genetic Sequence Databank.

Document Date: January 30, 1996

Document Version: 14.0

Operating Systems: VAX/VMS Version 5.5

Alpha OpenVMS & OSF/1

ULTRIX Version 4.3

SunOS Version 4.1.3

SGI IRIX Version V.4

DOS Version 3.0 or higher

Macintosh Operating System Version 7.5

National Biomedical Research Foundation
Georgetown University Medical Center
3900 Reservoir Road, N.W.
Washington, D.C. 20007 USA
Phone: (202) 687-2121
FAX: (202) 687-1662

E-mail addresses:
PIRMAIL@NBRF.GEORGETOWN.EDU

Contents

INDEX

EXAMPLES

FIGURES

TABLES

Preface

The ATLAS multidatabase retrieval program is designed to retrieve and manipulate information contained in the PIR-International Protein Sequence Database as well as other databases that have been reformatted to either the NBRF, GenBank, or EMBL formats.

The ATLAS program is an outgrowth of two programs that were developed by NBRF: the Protein Sequence Query (PSQ) program and the Nucleic Acid Query (NAQ) program. These programs have been distributed since about 1980 and are in use at a large number of sites around the world. Users of these programs will be familiar with many of the concepts in the ATLAS program.

Some of the major new features of the ATLAS program are:

- operates on amino acid sequence, nucleotide sequence, and structured text databases;
- the identifiable fields in the annotation (e.g., title, author name, species name, keywords) have been indexed; therefore, searching for terms that occur in these fields is much faster than in the PSQ and NAQ programs;
- the index of terms mentioned above is constructed from many databases and the search commands search all of these databases simultaneously, thereby eliminating the need to repeat the same query in each database of interest.

The ATLAS program was developed by the NBRF with the cooperation of the following: William A. Gilbert of the University of New Hampshire in Durham, New Hampshire; Alex Reisner and Carolyn Bucholtz of Sydney University in Australia; and Jack London of the Thomas Jefferson University in Philadelphia, Pennsylvania. Program development was partially supported by NLM LM05206, by NSF BIR-9107540, and by Digital Equipment Corporation.

The PIR-International Protein Sequence Database is produced cooperatively by the Protein Information Resource (PIR), the Martinsried Institute for Protein Sequences (MIPS), and the Japan International Protein Information Database (JIPID). This database may be referred to as the PIR database or the Protein Sequence Database in the manual.

The FASTA program package was supplied by William R. Pearson, Department of Biochemistry, University of Virginia, Charlottesville, Virginia.

Intended Audience

This manual is intended for all users of the ATLAS program; however, it is not intended to be a tutorial.

Examples

The examples in this document were generated using the database configuration at PIR-NBRF as of the date of this document. The same examples run using other database configurations may not produce the same results. Most examples were run using the PIR1 dataset of the PIR-International Protein Sequence Database, Release 47.xx, as the active database.

Part I The Databases

This part of the manual contains material to familiarize the user with the terminology that relates to the sequence databases and descriptions of each database included on the CD-ROM

1 Database Terminology

This chapter defines terminology that relates to the structure of the databases and the information contained therein.

1.1 Entry

An *entry* is a collection of information pertaining to one particular amino acid or nucleotide sequence. This information is organized into three main sections:

- *Title* - contains the sequence name and the species name
- *Text* - contains the annotation that is associated with the sequence
- *Sequence* - contains the amino acid or nucleotide sequence. In the alignment database this section is absent.

In addition, each entry has assigned to it an *entry-code*. The entry-code is used to uniquely identify an entry within a database.

Example 1–1 Typical PIR-International Protein Sequence Database Entry

```
1  PIR1:CCHU
2  cytochrome c - human

3  Species: Homo sapiens (man)

   Date: #sequence_revision 30-Sep-1991 #text_change 11-Dec-1993

   Accession: A31764; A05676; A00001

   Evans, M.J.; Scarpulla, R.C.
       Proc. Natl. Acad. Sci. U.S.A. 85, 9625-9629, 1988
       Title: The human somatic cytochrome c gene: two classes of processed
       pseudogenes demarcate a period of rapid molecular evolution.
       Reference number: A31764; MUID:89071748
       Accession: A31764
       Molecule type: DNA
       Residues: 1-105 <EVA>
       Cross-references: GB:M22877

   Matsubara, H.; Smith, E.L.
       J. Biol. Chem. 238, 2732-2753, 1963
       Title: Human heart cytochrome c. Chymotryptic peptides, tryptic
       peptides, and the complete amino acid sequence.
       Reference number: A05676
       Accession: A05676
       Molecule type: protein
       Residues: 2-28;29-46;47-100;101-105 <MATS>
```

Example 1–1 Cont'd on next page

Example 1–1 (Cont.) Typical PIR-International Protein Sequence Database Entry

```
Matsubara, H.; Smith, E.L.
    J. Biol. Chem. 237, 3575-3576, 1962
    Title: The amino acid sequence of human heart cytochrome c.
    Reference number: A00001
    Note: 66-Leu is found in 10% of the molecules in pooled protein.

Genetics:
    Introns: 57/1

Superfamily: cytochrome c

Keywords: acetylation; electron transfer; heme; mitochondrion; oxidative
    phosphorylation; polymorphism; respiratory chain

Residues            Feature
2-105               Protein: cytochrome c #status experimental <MAT>
2                   Modified site: acetylated amino end (Gly) (in mature
                      form) #status experimental
15,18               Binding site: heme (Cys) (covalent) #status
                      experimental
19,81               Binding site: heme iron (His, Met) (axial ligands)
                      #status predicted

                            Composition
          6 Ala  A      2 Gln  Q      6 Leu  L      2 Ser  S
          2 Arg  R      8 Glu  E     18 Lys  K      7 Thr  T
          5 Asn  N     13 Gly  G      4 Met  M      1 Trp  W
          3 Asp  D      3 His  H      3 Phe  F      5 Tyr  Y
          2 Cys  C      8 Ile  I      4 Pro  P      3 Val  V
     Mol. wt. unmod. chain = 11,749      Number of residues = 105

4              5        10        15        20        25        30
     1 M G D V E K G K K I F I M K C S Q C H T V E K G G K H K T G
    31 P N L H G L F G R K T G Q A P G Y S Y T A A N K N K G I I W
    61 G E D T L M E Y L E N P K K Y I P G T K M I F V G I K K K E
    91 E R A D L I A Y L K K A T N E
```

In this example:

1 PIR1:CCHU is the entry code.

2 is the title section.

3 is the text section containing the annotation.

4 is the sequence section containing the amino acid sequence.

1.2 Database

A *database* is a collection of entries. Each database has assigned to it a *database-code.* Table 1–1 shows the databases and the database-codes that are currently on the ATLAS CD-ROM.

Table 1–1 Databases and Database-Codes

Database-code[1]	Database
PIR1	Section 1. Annotated and Classified Entries
PIR2	Section 2. Annotated Entries
PIR3	Section 3. Unverified Entries
PATCHX	MIPSX Merged Sequence Database (minus PIR 1+2+3)
NRL_3D	NRL Sequence Structure Database
ALN	Database of Protein Family Alignments
RESID	Residues Database
ECOLI	Escherichia coli DNA Database
GBBCT	GenBank Bacterial (Locus and Title)
GBEST	GenBank EST (Locus and Title)
GBINV	GenBank Invertebrate (Locus and Title)
GBMAM	GenBank Other Mammalian (Locus and Title)
GBPAT	GenBank Patent (Locus and Title)
GBPHG	GenBank Phage (Locus and Title)
GBPLN	GenBank Plant (Locus and Title)
GBPRI	GenBank Primate (Locus and Title)
GBRNA	GenBank Structural RNA (Locus and Title)
GBROD	GenBank Rodent (Locus and Title)
GBSYN	GenBank Synthetic and Chimeric (Locus and Title)
GBUNA	GenBank Unannotated (Locus and Title)
GBVRL	GenBank Viral (Locus and Title)
GBVRT	GenBank Other Vertebrate (Locus and Title)
GBNEW	GenBank NEW Sequences

[1]Database-codes are not fixed and can be changed; therefore, the database-codes in use at your site may differ from those shown.

1.3 Identifying an Entry

An *entry-identifier* specifies the entry to be retrieved for processing using one of the many commands of ATLAS that process individual entries. This information is provided to the program by supplying the entry-code for the sequence entry and the database-code for the database containing the entry.

The correct format for specifying an entry in a database is: database-code, colon, entry-code with no intervening blanks.

The program always maintains a set of default databases, called the *active databases* (see Section 1.5). If an entry-code is given without a database-code, the program looks for the entry in the active databases. If two, or more, entries in different databases have the specified entry-code, they will all be found.

When the database-code is specified, the entry can be retrieved from either an active database or an inactive database (see Section 1.4).

The following entry-identifier

```
PIR1:CCHU
```

identifies entry CCHU in database PIR1.

The command line to display entry PIR1:CCHU is shown below.

```
ATLAS> TYPE PIR1:CCHU
```

1.4 The Term Index

In addition to accessing individual entries as described in the previous section, one must be able to search the entire database for entries. The ATLAS program facilitates this via an index containing all retrievable terms.

Upon examination of the entry shown in Example 1–1, it will be noticed that the text section is formatted so certain categories of information, called *fields*, are easily identified. For example, the species name, the accession number, the superfamily name, and the keyword fields are labeled with the corresponding field name. Note that the fields may contain multiple values: each value is indexed separately.

Because of this special formatting it is possible to create indexes of the *terms* that occur in these and other fields. The indexes can then be searched to generate a list of entries that contain a particular term.

For example, a user could:

- Search the author index to generate a list of entries that reference author "Adelman, J P"
- Search the superfamily index to generate a list of entries that reference superfamily "proenkephalin"
- Search the keyword index to generate a list of entries that reference keyword "cell adhesion"

Table 1–2 lists all the fields that are currently indexed for the PIR-International databases. Whenever possible, these same fields are indexed for the non-PIR-International databases.

Table 1–2 Indexed Text Fields

Field Name	Field Contents	Search Command
ACCESSION	accession number	ACCESSION
AUTHOR	author name	AUTHOR
CROSS_REF	cross-reference number	CROSS
FEATURE	feature name	FEATURE
GENE_NAME	gene name	GENE
JOURNAL	journal citation	JOURNAL
KEYWORD	keyword	KEYWORD
MEMBERS	alignment member	MEMBERS
REFERENCE	reference number	REFERENCE
SPECIES	species name	SPECIES
SUPERFAMILY	superfamily name	SUPERFAMILY
SUPFAM_NUM	superfamily number	SFNUM
TITLE	entry title	FIND[1]

[1]FIND in the menu mode of the PC version is the command to search any of the indexed text fields. The fields are listed in a submenu under FIND.

The indexes for all the fields[1] and for all databases are combined to form the *term index*.

For each field that occurs in the term index system, there is a corresponding command in ATLAS to search that index. These commands are collectively called the *text searching commands*. The text searching commands can only search the databases that are included in the term index.

1.5 Active Databases

Any or all of the databases included in the term index can be processed by the text searching commands. Initially all databases are active. To selectively activate databases, use the database-list parameter of the BASES command to specify the list of databases to become active.

1.6 The Current List

The *current list* is a subset of the entries in the active databases that has been selected by the operation of a command or series of commands. The current list can be acted upon by many of the other commands and provides a facility to isolate and manipulate selected portions of the databases.

Text and data searching commands accept four modifiers that facilitate logical manipulation of the current list. These are the /CURRENT, /SUBTRACT, /ADD, and /KEEP modifiers (see Section 3.6).

The command LIST/RESTORE allows the restoration of the current list that existed before the last command operation. This allows for easy recovery when the user is not satisfied with the result of the command operation or when the current list is truncated as a result of a CTRL-C operation (see Section 3.7).

The current list can also be saved to a file. When the entry codes for the current list are saved with the LIST/OUTPUT=file-spec, it can be generated again quickly with the GET command.

[1] Not all the fields shown in Table 1–2 occur in all databases. If a databases does not have a particular field, then it does not occur in the term index. If the database is active when searching this index a message will appear warning the user that not all active databases are included in the index.

2 Database Descriptions

This chapter provides a description of each of the databases contained on the CD-ROM.

2.1 The PIR-International Protein Sequence Database

The Protein Sequence Database was initiated at the NBRF in the early 1960's by the late Margaret O. Dayhoff as a collection of sequences for the study of evolutionary relationships among proteins. The database is now an international collaboration of three data centers: the NBRF, the Martinsried Institute for Protein Sequences (MIPS), and the Japan International Protein Information Database (JIPID). The three centers cooperate to produce and distribute a single database of 'wild-type' protein sequences. Currently the NBRF effort is supported as the Protein Information Resource (PIR) funded in part by the National Library of Medicine.

The database contains information concerning all naturally occurring, wild-type proteins whose primary structure (the sequence) is known. A major goal of the database project is to provide comprehensive, nonredundant data uniquely organized by homology and taxonomy. In addition to sequence data, the database contains information (called annotation) concerning: (1) the name and classification of the protein and the organism in which it naturally occurs; (2) references to the primary literature, including information concerning the sequence determination; (3) the function and general characteristics of the protein, including gene expression, post-translational processing, and activation; and (4) sites and regions of biological interest within the sequence. The database is also unique in maintaining consistency of annotation with restricted vocabularies employed for features and keywords. Data are accumulated from the published literature, by submissions to PIR-International, and by translation of nucleic acid sequences submitted to GenBank, the European Molecular Biology Laboratory (EMBL) Nucleotide Sequence Database, and the DNA Data Base of Japan (DDBJ). These data include those deposited with the Genome Sequence Data Base of the National Center for Genome Resources. Entries in the database are cross-referenced to these source databases. In addition cross-references are included to the Genome Data Base (GDB), the yeast gene name LISTA database, and MEDLINE. Work is currently underway to cross-reference to the Drosophila genome database (FlyBase), the Brookhaven Protein Data Bank (PDB), and the Complex Carbohydrate Structure Database (CCSD) of the international CarbBank project.

Conceptually the database consists of three primary components: the Literature, Source Sequence, and Canonical Sequence Components. The Literature component contains full citation information for all sources of information in the database and is linked to MEDLINE abstracts

via MEDLINE MUIDs. Each citation is uniquely identified by PIR-International Reference Number. Each reported sequence is stored in its originally published form in the Source Sequence Component and is assigned a PIR-International Accession Number that uniquely identifies it. These unmerged source sequences are cross-referenced to the corresponding nucleic acid sequence when present in the GenBank/EMBL/DDBJ nucleic acids sequence databases. In the near future, cross-references will be added to link directly to CDS IDs, providing a stable link between the conceptual translation and the corresponding coding region specification. The Canonical Sequence Component (which encompasses the PIR1, PIR2, and PIR3 data sections) is constructed by assembling source sequences that represent the same molecule into merged entries, which display a single canonical sequence and instructions for regeneration of each original source sequence. Hence, all information concerning the various reports of the sequence is stored in a compacted, nonredundant form, while remaining directly accessible to users of the database. This architecture allows the competing goals of completeness and nonredundancy to be addressed seamlessly.

Canonical sequences within the database are organized by placement numbers reflecting their similarity to other sequences in the database known to be homologous. Secondarily, the data are organized by species and by protein type (proteins having the same name). Originally, the separation of the database into data sections PIR1, PIR2, and PIR3 reflected the level of data processing and classification. Incompletely processed data are now designated by the status preliminary, listed within the reference portion of the entry, and may occur in any section of the database. In a sense no entry is ever completely processed: if new information becomes available, it is merged into the entry. Further, entries are continually monitored and revised as appropriate to reflect the most current biological understanding of the data. The status preliminary generally indicates that the paper reporting the sequence has not been fully analyzed.

The partitioning of the entries into sections PIR1, PIR2, and PIR3 has been retained for ease of physical access to the data in the file distribution form (each file is limited to 64K entries) but this has no other significance. Entry codes are unique across all sections (PIR1-PIR4). For convenience, classified entries, found in PIR1, are ordered by placement number (and by species and protein type within placement classes); nonclassifed entries are ordered by species and taxonomy. These partitionings will be adjusted as appropriate in each release; therefore, section location is not a stable attribute of the entries.

The PIR4 Section has been recently introduced. It is not a regular component of the database but has been created to make available sequences that are not naturally occurring and/or naturally expressed. This includes conceptual translations of pseudogenes and other nonexpressed potential genomic coding regions, engineered and chemically synthesized sequences, and sequences of natural polypeptides that are not ribosomally synthesized. Effort will not be made to collect these data; however, they are often accumulated during routine data processing, in which case they will be stored and made available in PIR4. Sequences

of these types that occur within PDB entries will be accumulated comprehensively and cross-referenced to PDB.

These and all subsequent changes in the database format are described in the PIR Technical Development Bulletin, sent to an E-mail distribution list. More information regarding the format and content of PIR entries can be found in Chapter 1, Database Terminology and in Appendix D, PIR-International Protein Sequence Database Entry Format.

Correspondence regarding the PIR database should be directed to:

PIR Technical Services Coordinator
National Biomedical Research Foundation
3900 Reservoir Road, NW
Washington, DC 20007 USA
E-mail: pirmail@nbrf.georgetown.edu

2.2 The PATCHX Merged Sequence Database

The PATCHX database is produced by MIPS and includes all protein sequences not identical with or contained in sequences from PIR1, PIR2 and PIR3. It was created to supplement the PIR database in providing a comprehensive dataset for searching and to help in identifying sequences that need to be added to the PIR database. PATCHX is generated by sequentially merging entries in a manner that eliminates an entry whenever there is already an entry containing exactly the same sequence.

Table 2–1 Non-PIR Databases used to generate PATCHX

Database	Codes[1]	Description
MIPSOwn (parts 1 and 2)	Original	MIPS preliminary entries
PIRMOD (parts 1 and 2)	Original	MIPS/PIR preliminary entries
MIPSTrn	Original	MIPS preliminary translations
NRL_3D	Orignal	Brookhaven Data Bank sequences
SwissProt	Original	SwissProt entries
EMTrans (parts 1, 2, and 3)	Original	EMBL automatic translations
GBTrans (parts 1, 2, and 3)	Original	GenBank automatic translations
Kabat	Original	Kabat entries
PSeqIP	L...	NEWAT
	M...	PSD

[1]The original entry codes are used whenever possible. For EMTrans and GBTrans, the first six characters of the entry code correspond to the primary accession number of the EMBL or GenBank entry. Code conflicts, of which a few hundred occur, require the assignment of special codes; these start with ‘‘ZZ_’’. For technical reasons, MIPS-assigned entry codes are still used for PSeqiP.

PATCHX provides a good dataset to supplement the PIR database for sequence searching but, due to the manner in which it is constructed, the user should take note of the following:

- All sequences that are IDENTICAL within or between databases are present ONCE. Duplicate sequences and sequences that were completely contained within others (subsequences) have been eliminated according to the priority (top to bottom) in the table above. Because entries with identical sequences have been eliminated, independent reports of the same correct sequences may be eliminated; therefore, this database may not yield a complete bibliography for a given sequence.
- A black list of invalid sequences is kept. Black list sequences are removed from PATCHX to reduce noninformative redundancy. The EST sections of EMBL and GenBank are not considered for EMTrans and GBTrans. In addition, sequences with greater than 94% sequence identity to an entry in the PIR-International Protein Sequence Database (PIR1, PIR2, PIR3) (data supplied by K. Heumann, S. Liebl, W. Peng, and H.W. Mewes) have been removed from PATCHX.
- The MIPSOwn, PIRMOD, and MIPSTrn databases contain preliminary data that should be used with extreme caution.

Correspondence regarding PATCHX should be directed to:

Dr. Friedhelm Pfeiffer
MIPS at the Max Planck Inst. for Biochemistry
8033 Martinsried, Germany
E-mail: pfeiffer@ehpmic.mips.biochem.mpg.de

2.3 The NRL_3D Sequence–Structure Database

The NRL_3D database is produced by PIR from sequence and annotation information extracted from the Brookhaven Protein Databank (PDB) of crystallographic structures. This database makes the sequence information in PDB available for similarity searches and retrieval, and provides cross-reference information for use with the PIR. The titles and biological sources of the entries have been changed from PDB to conform to the nomenclature standards used in the PIR. The bibliographic references are included and some appear with the reference numbers and MEDLINE cross-references they have in corresponding PIR entries. Secondary structure, active site, binding site, and modified site annotations in PDB appear as in the corresponding PIR features. Information on experimental method, resolution and R-factor are included along with keywords.

The ATLAS program is used for the retrieval and display of the entries in the NRL_3D database. Please see the AUTHOR, CROSS, FEATURE, FIND, JOURNAL, KEYWORD, LIST, MATCH, REFERENCE, SCAN, SPECIES, and TYPE commands in Chapter 4, The ATLAS Commands, Detailed Descriptions.

Correspondence regarding NRL_3D should be directed to:

Dr. John S. Garavelli
PIR Database Coordinator
National Biomedical Research Foundation
3900 Reservoir Road, NW
Washington, DC 20007 USA
E-mail: Garavelli@NBRF.Georgetown.EDU

2.4 The PIR-ALN Protein Sequence Alignment Database

PIR-ALN is a database of protein alignments produced by L.-S. Yeh and G.Y. Srinivasarao of PIR. Alignments are of sequences in the same family (less than 55% different from each other), or of sequences representing various families within a superfamily, or of sequence segments corresponding to the same homology domain in different proteins. As of September 1995, PIR-ALN contained 257 homology domain alignments. These effectively define the homology domains so that they can be consistently represented in the entry annotations.

The information contained in an ALN entry is divided into seven sections. The sections are listed below in the order in which they occur in the entry.

1 *Header* - Information that marks the first line of an entry

2 *Title* - The title of the alignment

3 *Date* - Creation and revision dates

4 *Members* - The entry codes of sequences used in the alignment

5 *Members Titles* - Titles of sequences used in the alignment

6 *Alignment* - The alignment of sequences

7 *Matrix* - The matrix of percent differences

In the alignment, the completely conserved residues are marked by ' * ' and partially conserved residues are marked by ' . ' under the alignment. In the matrix of percent differences, the upper portion of the matrix gives the number of differences between the sequences; the lower portion represents the percent differences.

The ATLAS program is used for the retrieval and display of the alignment entries in the ALN database. Please see the FIND, MEMBERS, and TYPE commands in Chapter 4, The ATLAS Commands, Detailed Descriptions.

Example 2–1 Sample ALN Entry

```
>TX;FA0003
serum albumin - family  1.0
Date: 08-Feb-1995; Revision: 08-Feb-1995
Members: ABBOS; ABHUS; ABRTS; ABPGS; ABSHS; ABHOS
ABBOS   serum albumin precursor - bovine
ABHUS   serum albumin precursor - human
ABRTS   serum albumin precursor - rat
ABPGS   serum albumin precursor - pig (fragment)
ABSHS   serum albumin precursor - sheep
ABHOS   serum albumin precursor - horse

Alignment:

ABBOS   MKWVTFISLLLLFSSAYSRGVFRRDTHKSEIAHRFKDLGEEQFKGLVLIAFSQYLQQCPF
ABHUS   MKWVTFISLLFLFSSAYSRGVFRRDAHKSEVAHRFKDLGEENFKALVLIAFAQYLQQCPF
ABRTS   MKWVTFLLLLFISGSAFSRGVFRREAHKSEIAHRFKDLGEQHFKGLVLIAFSQYLQKCPY
ABPGS   --WVTFISLLFLFSSAYSRGVFRRDTYKSEIAHRFKDLGEQYFKGLVLIAFSQHLQQCPY
ABSHS   MKWVTFISLLLLFSSAYSRGVFRRDTHKSEIAHRFNDLGEENFQGLVLIAFSQYLQQCPF
ABHOS   MKWVTFVSLLFLFSSAYSRGVLRRDTHKSEIAHRFNDLGEKHFKGLVLVAFSQYLQQCPF
          ****..**....**.****.**...***.****.****..*..***.**.*.**.**.

ABBOS   DEHVKLVNELTEFAKTCVADESHAGCEKSLHTLFGDELCKVASLRETYGDMADCCEKQEP
ABHUS   EDHVKLVNEVTEFAKTCVADESAENCDKSLHTLFGDKLCTVATLRETYGEMADCCAKQEP
ABRTS   EEHIKLVQEVTDFAKTCVADENAENCDKSIHTLFGDKLCAIPKLRDNYGELADCCAKQEP
ABPGS   EEHVKLVREVTEFAKTCVADESAENCDKSIHTLFGDKLCAIPSLREHYGDLADCCEKEEP
ABSHS   DEHVKLVKELTEFAKTCVADESHAGCDKSLHTLFGDELCKVATLRETYGDMADCCEKQEP
ABHOS   EDHVKLVNEVTEFAKKCAADESAENCDKSLHTLFGDKLCTVATLRATYGELADCCEKQEP
        ..*.***.*.*.***.*.***....*.**.******.**....**..**..****.*.**
.
.
.

ABBOS   PDTEKQIKKQTALVELLKHKPKATEEQLKTVMENFVAFVDKCCAADDKEACFAVEGPKLV
ABHUS   SEKERQIKKQTALVELVKHKPKATKEQLKAVMDDFAAFVEKCCKADDKETCFAEEGKKLV
ABRTS   PDKEKQIKKQTALAELVKHKPKATEDQLKTVMGDFAQFVDKCCKAADKDNCFATEGPNLV
ABPGS   PEDEKQIKKQTALVELLKHKPHATEEQLRTVLGNFAAFVQKCCAAPDHEACFAVEGPKFV
ABSHS   PDTEKQIKKQTALVELLKHKPKATDEQLKTVMENFVAFVDKCCAADDKEGCFVLEGPKLV
ABHOS   PEDEKQIKKQSALAELVKHKPKATKEQLKTVLGNFSAFVAKCCGREDKEACFAEEGPKLV
        ...*.*****.**.**.****.**..**..*...*..**.***...*...**..**...*

ABBOS   VSTQTALA-
ABHUS   AASQAALGL
ABRTS   ARSKEALA-
ABPGS   IEIRGILA-
ABSHS   ASTQAALA-
ABHOS   ASSQLALA-
        ......*.

Matrix of percent differences:

            Number of differences
            1   2   3   4   5   6

1 ABBOS      . 146 183 122  47 157

2 ABHUS     24   . 164 147 150 143

3 ABRTS     30  27   . 167 185 168

4 ABPGS     20  24  28   . 131 142

5 ABSHS      8  25  30  22   . 146

6 ABHOS     26  23  28  23  24   .
              Percent difference
```

Correspondence regarding ALN should be directed to:

Dr. Lai-Su L. Yeh or Dr. Geetha Y. Srinivasarao
Protein Information Resource
National Biomedical Research Foundation
3900 Reservoir Road, NW
Washington, DC 20007 USA
E-mail: yeh@nbrf.georgetown.edu or geetha@nbrf.georgetown.edu

2.5 The RESID Database of Amino Acids Residues

The RESID is a database of protein structure modifications produced by PIR. Due the large and steadily increasing number of protein structure modifications that require standardized annotation in the PIR-International Protein Sequence Database, the RESID database was introduced to assist users and annotators in interpreting features annotations for covalent binding sites, modified sites, and cross-links. The RESID database describes features annotated in the Protein Sequence Database and provides information on systematic chemical names, frequently observed alternate names, Chemical Abstracts Service registry numbers, atomic formulas and weights, and original amino acids that may have the modification.

Entries in the RESID database contain this information in the following order.

1 *Header* - the code number of the entry in the RESID database, these consist of the letters 'AA' followed by four digits

2 *Title* - the name of the amino acid residue

3 *Alternate names* - alternative names this residue may have in the chemical literature

4 *Systematic name* - an IUPAC systematic name

5 *CAS Registry Number* - the Chemical Abstracts Registry Number for the compounds corresponding to the free amino acids (CAS Registry Numbers are copyrighted by the American Chemical Society)

6 *Formula* - the atomic formula of the residue in the peptide chain

7 *Formula weight* - the chemical average isotope formula weight and the physical most common isotope formula weight

8 *Correction formula* - the difference between the residue atomic formula and the atomic formula for the encoded amino acid presented in a protein sequence

9 *Correction weight* - the differences between the residue chemical average isotope weight and the physical most common isotope weight and those for the encoded amino acid presented in a protein sequence; these correction weight can be used to calculate chemical and mass-spectrographic molecular weights for modified peptides

10 *Date* - Creation, structure revision and text revision dates for an entry

11 *Reference* - a reference block as in PIR Protein Sequence Database inluding author names, journal citation, article title, reference number, MEDLINE number and notes on the experimental methods used to detect and identify the residue

12 *Comment* - notes on the residue including sequence motifs

13 *Generating enzyme* - the enzyme activities required to produce a post-translationally modified residue

14 *Sequence code* - the single letter codes for the amino acids which may give rise to the same post-translationally modified residue

15 *Conditions* - conditions for the occurrence of the residue including whether it is amino-terminal, carboxyl-terminal, secondary or incidental to other modifications, or the number of peptide chains the residue cross-links

16 *Abbreviation* - the standard IUPAC three-letter code for the encoded amino acids

17 *Keywords* - the keywords in appearing in the PIR Protein Sequence Database associated with the residue

18 *Feature* - the feature for this residue as it appears in the PIR Protein Sequence Database

The ATLAS program is used for the retrieval and display of the entries in the RESID database. Please see the AUTHOR, CROSS, FEATURE, FIND, JOURNAL, KEYWORD, LIST, REFERENCE, and TYPE commands in Chapter 4, The ATLAS Commands, Detailed Descriptions.

Example 2–2 Sample RESID Entry

```
RESID:AA0077
N6-palmitoyl-L-lysine

Alternate names: epsilon-palmitoyllysine; N(zeta)-palmitoyllysine;
    N6-(1-oxohexadecyl)-L-lysine

Systematic name: (S)-2-amino-6-(hexadecanoylamino)hexanoic acid
    Cross-references: CAS:559012-43-0

Formula: C 22 H 42 N 2 O 2
    Formula weight: #chem 366.59 #phys 366.3246

Correction formula: C 16 H 30 O 1
    Correction weight: #chem 238.42 #phys 238.2297

Date: 31-Mar-1995 #structure_revision 31-Mar-1995 #text_change 31-Mar-
    1995

Hackett, M.; Guo, L.; Shabanowitz, J.; Hunt, D.F.; Hewlett, E.L.
    Science 266, 433-435, 1994
    Title: Internal lysine palmitoylation in adenylate cyclase toxin
      from Bordetella pertussis.
    Reference number: A55167
    Note: mass spectrographic identification
```

Example 2–2 Cont'd on next page

Example 2–2 (Cont.) Sample RESID Entry

```
Stanley, P.; Packman, L.C.; Koronakis, V.; Hughes, C.
    Science 266, 1992-1996, 1994
    Title: Fatty acylation of two internal lysine residues required for
      the toxic activity of Escherichia coli hemolysin.
    Reference number: A55387
    Note: radioisotope labeling

Generating enzyme: peptidyl-lysine N6-palmitoyltransferase (EC 2.3.1.-)

Sequence code: K
    Conditions: combinable

Keywords: lipoprotein

Residues            Feature
                    Binding site: palmitate (Lys) (covalent)
```


Correspondence regarding RESID should be directed to:

Dr. John S. Garavelli
PIR Database Coordinator
National Biomedical Research Foundation
3900 Reservoir Road, NW
Washington, DC 20007 USA
E-mail: Garavelli@NBRF.Georgetown.EDU

2.6 The ECOLI *Escherichia coli* DNA Database

The ECOLI database was produced at JIPID by T. Kunisawa with the cooperation of L.-S. Yeh of PIR. The database consists of the available genomic nucleic acid sequence data of *E. coli* K12 compiled from the existing major data collections (GenBank, EMBL, and DDBJ) and from the literature. The sequence data are solely from strain K12, for which genetic map positions are known. The genome of *E. coli* K12 consists of about 4.7 million bp and the goal of this effort is to merge information obtained from the data collections and literature with data obtained from genome sequencing projects. The number of base pairs in the ECOLI database has doubled since it was first announced (Protein Seq. Data Anal. 3:157-162, 1990) and now corresponds to about 40% of the entire genome.

Sequence redundancy in the database is eliminated by merging overlapping sequences. Each entry represents one sequence segment and all the entries in the database are ordered by genetic map position. Unlocalized genes will be included in the database once their chromosomal locations are known. Discrepancies among various reports of the same sequence are described in the "Residues" line following each Accession line. A tag, enclosed in angle brackets at the end of each Residues line, serves to identify the sequence reported in the publication.

The gene name (and its alternate gene symbols), map position, and strand of the sequence segments, if known, are indicated within each entry. A plus or minus (+ or -) is used to specify on which of the two DNA strands the segment exists; the "+" strand is the strand transcribed clockwise in the usual genetic map. All known protein coding regions are annotated in the Feature table, as well as additional features such as promoter region, Shine-Dalgarno sequence, and inverted repeats.

Correspondence regarding ECOLI should be directed to:

Dr. T. Kunisawa
JIPID, Research Institute for Biosciences
Science University of Tokyo
Noda 278, Japan
E-mail: kunisawa@jpnsut31.bitnet

2.7 The GenBank Nucleic Acid Sequence Databank

The GenBank database is a nucleic acid sequence database produced and distributed by the National Center for Biotechnology Information (NCBI). The complete database is not available on the ATLAS CD-ROM; however, retrieval of entry codes is available by GenBank accession number, author name, citation, species, sequence title and keyword indexing. Information available with each hit is limited to the LOCUS and DEFINITION fields of each entry.

Complete GenBank entries can be retrieved from the PIR Network Request Server using the following electronic addresses:

fileserv@nbrf.georgetown.edu

Alternatively, if supplemental CD-ROM readers or several hundred megabytes of disk space are available then the ATLAS software system is capable of retrieving full GenBank entries in the native form as they appear in the multi-volume NCBI-GenBank flat file format CD-ROM package. Follow instructions in the Installation section for details on how to use the GenBank data as distributed on CD-ROM.

Contact NCBI at the address below with regard to the NCBI-GenBank CD-ROM package.

NCBI-GenBank
National Center for Biotechnology Information
National Library of Medicine, 38A, 8N805
8600 Rockville Pike
Bethesda, MD 20894 USA
E-mail: info@ncbi.nlm.nih.gov

Part II The ATLAS program

This part of the manual contains instructions on the use of the commands and command modifiers of the ATLAS program.

3 Overview of the ATLAS Program

The program responds to commands and modifiers typed at the ATLAS> prompt; PC users have the option to use a menu system. For convenience, all commands and modifiers may be abbreviated to their shortest unambiguous form. Commands may be typed in using upper- and/or lower-case letters; the program does not discriminate between them. Command operation does not begin until the RETURN key is pressed. The command names recognized by the ATLAS program can be modified by the user. A special DEFINE command is included to allow the definition of aliases for specific command and command modifier combinations.

The descriptions of commands and command modifiers assume the use of the program in command mode (i.e., issuing commands at the ATLAS> prompt) and that the user has not modified the native command definitions. See Section 3.8 later in this chapter for an introduction to the menu system. Differences in commands between command mode and menu mode are indicated.

For command mode, commands should be typed at the ATLAS> prompt in the format:

```
ATLAS> COMMAND/MODIFIER parameter
```

Character strings can be combined on the same command line to allow for searching logical combinations of strings. Strings are separated by space characters. The strings AND, or OR, and NOT are reserved as operators. When no operator is present, the AND operator is assumed. The OR operator instructs ATLAS to search for entries that contain either or both of the strings that immediately precede and follow it. The NOT operator is a binary operator. It instructs ATLAS to find entries that contain the first string and do not contain the second.

Lines containing more than one operator are evaluated according to the following order of precedence: the OR operator has the highest precedence, followed by the AND operator, and then the NOT operator; the implied AND operator (no operator) has the lowest precedence.

The open and closed parenthesis characters specify an alternate evaluation order. Strings enclosed in parentheses are evaluated separately. The double quote characters are used to designate literal search strings. Literal search strings are searched for exactly as they appear. Quotes are used when strings containing the reserved characters, space, and open and closed parentheses or the reserved operators, AND, OR, and NOT, are required. The /SHOW modifier of the text searching commands will display how the search strings are parsed by the ATLAS program.

The commands can be grouped into the following categories according to function:

- **Text Searching Commands**
- **Sequence Searching Commands**

- **Display Commands**
- **File Interface Commands**
- **Utility Commands**

3.1 Text Searching Commands

The commands in this group provide the primary retrieval capabilities of the ATLAS program by searching the term indexes. All text searching commands are predefined versions of the SEARCH command. When using command mode, the field name is usually used as the command, e.g. KEYWORD, FEATURE, etc. The command KEYWORD is a symbol for the predefined full command, SEARCH/FIELD=KEYWORD. When using the menu, all the text searching commands are listed under the FIND command.

Each term in the term index that matches the user-supplied search string is displayed on the screen. In addition, the database-code, entry-code, and title of each entry in which the term is found are displayed. Each word in the search string must be at least three characters in length. The /BRIEF modifier limits display to only the index terms found, omitting the codes and titles of the entries.

The commands generate a new current list containing all the entries found during the search.

Table 3–1 Text Searching Commands

Command	Action
ACCESSION	Search the accession index for an accession number
AUTHOR	Search the author index for an author name
CROSS	Search the cross-reference index for a cross-reference number
FEATURE	Search the feature index for a feature name
FIND[1]	Search the entry titles for a sequence name and/or an organism name
GENE	Search the gene name index for a gene name
JOURNAL	Search the journal index for a journal citation
KEYWORD	Search the keyword index for a keyword
MEMBERS	Search the members index of the alignment database (ALN)
REFERENCE	Search the reference number index for a reference number
SPECIES	Search the species index for a species
SUPERFAMILY	Search the superfamily index for a superfamily name
SFNUM	Search the superfamily index for a superfamily number

[1]The FIND command in the menu system searches an index selected from those listed in the submenu.

3.2 Sequence Searching Commands

These commands allow limited sequence searching within the ATLAS program. They search the protein sequence data for the occurrence of short amino acid segments. The search strings are typed directly into the terminal in the one-letter amino acid code (see Appendix B).

Table 3–2 Sequence Searching Commands

Command	Action
SCAN	Rapid search for identically matching segments
MATCH	Search for protein segments allowing mismatches

3.3 Display Commands

These commands are designed to display database information.

Table 3–3 Display Commands

Command	Action
LIST	Display titles of all entries on the current list
TYPE	Display an entry
EXTRACT	Constructs and displays a modified sequence

3.4 File Interface Commands

These commands provide the interface between the ATLAS program and external files.

Table 3–4 File Interface Commands

Command	Action
COPY	Copy an entry into an external file
GET	Get current list from an external file
PRINT[1]	Display the contents of an external file

[1]The PRINT command is not available through the menu system.

3.5 Utility Commands

The utility commands set and display program defaults and perform other miscellaneous chores.

Table 3–5 Utility Commands

Command	Action
BASES	Display or define the list of active databases
DEFINE[1]	Define abbreviations for commands or databases
HELP	Obtain help on commands and modifiers
QUIT	Terminate the program
SET	Set program operation parameters
SHOW	Show information about current program operation
SYSTEM	Transfer control to DOS (PC only)
EXIT	Return from DOS to ATLAS program (PC only)

[1]DEFINE as listed as an action under BASES in the menu system only allows the user to activate databases; command or database abbreviation definitions are not available through the menu.

3.6 Command Modifiers

The primary action of the commands can be modified by the addition of one or more command modifiers. Command modifiers immediately follow the command with no space between and must be preceded by a slash (/). Not all the command modifiers listed below work with every command. See the detailed descriptions of the commands and their command modifiers for more information.

Table 3–6 Command Modifiers

Modifiers of the ATLAS Commands	
Alternate Display Formats	**Modifier Function**
/BRIEF	Display only index terms found
/COUNTS	Display number of entries containing search term. This number is displayed after all the entry titles have been displayed. Use /COUNTS in conjunction with /BRIEF to display only the index terms and corresponding count.
/SHOW	Display how search terms are parsed by ATLAS
Current List Processing	**Modifier Function**
/CURRENT	Process entries on current list only. When /CURRENT is specified, the entry-identifier parameter on the command line should be omitted. If present, it is ignored.
/ADD	Entries found are added to current list
/SUBTRACT	Entries found are removed from current list
/KEEP	Command execution does not alter current list
/ALL	Process all entries in the active databases
/RESTORE	Restores previous current list (LIST command only)

Table 3–6 (Cont.) Command Modifiers

Modifiers of the ATLAS Commands	
Information Category Selection	**Modifier Function**
/TEXT	Only text portion of entry is processed
/SEQUENCE	Only sequence portion of entry is processed
Redirecting Screen Output	**Modifier Function**
/OUTPUT=file-spec	Directs normal output to a file (no screen display). Differs from /PRINTER=file-spec for some commands.
/PRINTER	Directs screen output to printer. Output is not immediately sent to the printer but accumulates until the QUIT command is issued to leave the program.
/PRINTER=file-spec	Directs screen output to disk file. The modifier value, *file-spec*, can be any valid file specification. The specified file is closed when the command terminates; it is not sent to the printer for printing, it is a permanent file that remains until explicitly deleted.
Search Manipulation	**Modifier Function**
/ANCHOR	Match terms beginning with character string
/NOANCHOR	Match character string anywhere in term
/ENTRY	Match each character string to separate terms

3.7 Special Control Characters

The ATLAS program responds to two control characters that interrupt program operation:

CTRL-C

The CTRL-C character terminates the command that is currently operating. It is generally used to recover from inadvertently issued commands.

CTRL-Y

The CTRL-Y character terminates the VAX/VMS version of the ATLAS program and returns the user to the system.

Note: CTRL-Y is *not* the proper way to exit the program; the QUIT command should be used for normal program termination.

3.8 The PC Menu System

Upon startup, the PC version of the ATLAS program is in a menu command system. This menu system can be toggled to command mode, in which the commands are the same as those for the VAX/VMS version described in Chapter 4.

This section describes how to get around in the menu and use some of the commands as seen through the menu system. The following main menu bar appears across the top of the screen:

```
Bases  Find  List  Get  Type  Copy  Extract  Scan  Option  Help  Quit
```

Initially "Bases" is highlighted. Move among commands with the left and right arrow keys. As each command is highlighted a message appears under the menu describing the command. The following table summarizes the descriptions of each highlighted command:

Table 3–7 Main Menu Selections

Bases:	Display or define the list of active databases
Find:	Search the specified text field
List:	Display or manipulate the current list
Get:	Get the current list from an external file or from the user
Type:	Display an entry
Copy:	Copy an entry to an external file
Extract:	Construct and display a modified sequence
Scan:	Search protein sequences
Option:	Set or show option
Help:	Help
Quit:	Terminate the ATLAS program

To select a command either type the first letter of the command or press the RETURN key or the down arrow key when the desired command is highlighted. The Esc key deselects a command and returns the user to the main menu. When a command from the main menu is selected, one or two pop-up submenus appear on the screen. The one on the left shows the actions (commands) available and the one on the right shows the options (command modifiers) available.

To make a selection in the action submenu, type the appropriate letter or use the up and down arrow keys until the desired selection is highlighted and press RETURN to initiate the action. When RETURN is pressed and there are no options highlighted, the menu command action is initiated without modification.

To make a selection from the option submenu, press the appropriate function key; pressing the key a second time deselects the option. More than one option may be selected if the menu box is divided into sections. One option may be selected from each section. If an attempt is made to select two items in the same section, the program simply toggles from one to the other.

Most of the commands work the same through the menu as they do in command mode. The greatest differences occur in the FIND and OPTION selections. The various text searching commands, which all work in the same manner, are listed in a submenu under FIND; the user simply selects the index to be searched.

OPTION has several commands that are unique to the PC version and particularly to the menu system. When OPTION has been selected, the following action submenu appears:

Table 3–8 OPTION Submenu

Ltr.	Action	Description
F	show fields	Display the indexed fields available for each database
A	ATLAS program header	Display the citation header for the ATLAS program
S	gateway to DOS	Temporarily leave ATLAS to use DOS
C	command mode	Toggle from the menu system to command mode
P	toggle pause	Switch to continuous scrolling
M	select colors	Change colors for main and submenus

The S (system) action allows the user to temporarily leave ATLAS and go to DOS. To return from DOS to ATLAS, type EXIT at the DOS prompt.

The C action allows the user to enter the command mode of ATLAS. Typing commands at the ATLAS> prompt is much more flexible than using the menu. To return from command mode to the menu type SET/MENU at the ATLAS> prompt.

The P action allows the user to change how information is scrolled on the screen. When the pause is active (default), only one screen full of information is displayed at a time. The user is prompted at the bottom of the screen to decide whether or not to continue viewing with the prompt:

```
More (Y/n):
```

When toggled, the information scrolls continuously on the screen and stops/starts in response to the Scroll Lock key.

The M action allows color to be changed in four regions of the screen: the main menu bar, the highlighting for the main menu bar, the submenus, and the highlighting for the submenus. When this action is selected, the lower part of the screen shows 12 numbers that determine the colors displayed.

The numbers are are divided into triplets; one triplet for each of the four regions. For example, in the triplet 7,1,1, the first represents the foreground color (0 to 7), the second represents the intensity (0 or 1), and the third the background color (0 to 7).

The space bar steps through the colors at the cursor and the arrow keys move the cursor. The colors on the screen change as each selection is made and can be saved, until you quit ATLAS, by pressing S.

4 The ATLAS Commands, Detailed Descriptions

This chapter provides a detailed description of each command, its parameters, and its modifiers. The command descriptions are arranged alphabetically.

ACCESSION

ACCESSION searches the accession index for all occurrences of a user-specified accession number. For each number found, the number, the entry identifier, and the title for each entry in which the number occurs are displayed. The list of these entries becomes the new current list. This command is an abbreviation for the full command SEARCH/FIELD=accession/ANCHOR.

FORMAT

ACCESSION *[accession-number]*

PARAMETERS

accession-number

The *accession-number* parameter on the command line is the number (or part of the number) to be selected. Accession numbers in PIR consist of 1 letter followed by 5 digits. The character string you enter must contain at least 3 characters. If this parameter is omitted, you will be prompted for it with the prompt:

prompts

Accession:

If you respond to the prompt by pressing the RETURN key, all accession numbers will be found.

DESCRIPTION

The accession index is constructed from fields that contain accession numbers in the databases included, e.g., the Accession: lines in PIR databases. An accession number is assigned to each sequence entry when it is entered into one of the PIR datasets. For entries that are merged, the accessions are listed for all individual entries included. Often in the PIR2 and PIR3 datasets the accession number is also used as an entry-code.

Normally the accession number search matches only accession numbers that begin with the character string you specified. Use the /NOANCHOR modifier (listed below) to alter this operation.

MODIFIERS

In the following table, modifiers accepted by the ACCESSION command are grouped according to their function. Multiple modifiers can be used together; a slash (/) precedes each modifier with no spaces before or after. A modifier designated as a default does not need to be typed in to be used.

Table 4–1 ACCESSION Command Modifiers

Alternate Display Format	Modifier Function
/BRIEF	Display only index terms found
/COUNTS	Display number of entries containing search term
/SHOW	Display how search string is parsed

Database List Processing	Modifier
/PIR	Ignore all databases except PIR1, PIR2, and PIR3

Current List Processing	Modifier Function
/CURRENT	Process only entries on current list
/ADD	Add entries found to current list
/SUBTRACT	Remove entries found from current list
/KEEP	Do not alter current list with command execution

Redirecting Screen Output	Modifier Function
/PRINTER	Direct screen output to printer
/PRINTER=file-spec	Direct screen output to disk file

Search Manipulation	Modifier Function
/NOANCHOR	Match search-string anywhere in term
/ANCHOR (default)	Match terms beginning with character string
/ENTRY	Match each character string to separate terms

EXAMPLES

The command line:

1
```
ATLAS> ACCESSION A000
```

will produce the list of entries containing accession numbers that begin with A000.

2
```
ATLAS> ACC/SUB A0009
```

This command will remove from the current list (produced by the command in the first example) those entries that contain accession numbers beginning with A0009. Codes and titles of the removed entries are displayed. To see the entries that remain, use the LIST command.

AUTHOR

AUTHOR searches the author index for all occurrences of a user-specified name or partial name. For each name found, the author, the entry-identifier, and the title for each entry referencing the author are displayed. The list of these entries becomes the new current list. This command is an abbreviation for the full command: SEARCH/FIELD=author/ANCHOR.

FORMAT

AUTHOR *[author-name]*

PARAMETERS

author-name

The *author-name* parameter on the command line is the name (or part of the name) of the author to be selected. The character string you enter must contain at least 3 characters. If this parameter is omitted on the command line, you will be prompted for it with the prompt:

prompts

Author:

DESCRIPTION

Normally the author name search matches only author names that begin with the character string you specified. Use the /NOANCHOR modifier (listed below) to alter this operation.

Displaying the List of Authors

If no Author-name is typed in at the prompt and you press RETURN, an alphabetical listing of all authors and the titles of their associated entries will be displayed. If the /BRIEF modifier (listed below) is used, only the author names will be displayed.

MODIFIERS

In the following table, modifiers accepted by the AUTHOR command are grouped according to their function. Multiple modifiers can be used together; a slash (/) precedes each modifier with no spaces before or after. A modifier designated as a default does not need to be typed in to be used.

Table 4–2 AUTHOR Command Modifiers

Alternate Display Formats	Modifier Function
/BRIEF	Display only index terms found
/COUNTS	Display number of entries containing search term
/SHOW	Display how search string is parsed

Table 4–2 (Cont.) AUTHOR Command Modifiers

Database List Processing	Modifier
/PIR	Ignore all databases except PIR1, PIR2, and PIR3

Current List Processing	Modifier Function
/CURRENT	Process only entries on current list
/ADD	Add entries found to current list
/SUBTRACT	Remove entries found from current list
/KEEP	Do not alter current list with command execution

Redirecting Screen Output	Modifier Function
/PRINTER	Direct screen output to printer
/PRINTER=file-spec	Direct screen output to disk file

Search Manipulation	Modifier Function
/ANCHOR (default)	Match terms beginning with character string
/NOANCHOR	Match character string anywhere in term
/ENTRY	Match each character string to separate terms

EXAMPLES

The following command line will produce the list of AUTHORS whose last names begin with Smith and the entry-identifiers and titles of entries containing those authors.

```
1   ATLAS> AUTHOR SMITH
```

When specifying an initial for an author, it should be in the format: *Lastname,I* with no space between the comma and initial. Example:

```
2   ATLAS> BASES PIR1
    ATLAS> AUTHOR ADELMAN,J
    Adelman,J
      PIR1:EQHUA   proenkephalin precursor - human
      PIR1:IVHU14  interferon alpha-I-14 precursor - human
      PIR1:IVHUA9  interferon alpha-9 precursor - human
      PIR1:IVHUB1  interferon beta-1 precursor - human
      PIR1:ABHUS   serum albumin precursor - human
    Adelman,J P
      PIR1:A34702  amphiregulin precursor - human
      PIR1:RHHUG   gonadoliberin precursor - human
      PIR1:RHRTG   gonadoliberin precursor - rat
      2 authors found
```

The above example found two authors; one with one initial and the other with two initials. To specify two initials, separate them with a single nonalphanumeric character (other than right or left parentheses). Alternatively, the search string may be enclosed in quotes. For example:

```
3   ATLAS> AUTHOR ADELMAN,J.P
```

or

```
4    ATLAS> AUTHOR "ADELMAN,J P"
```

BASES

The BASES command provides two functions. It can be used to either

- display a table showing all the accessible databases, or
- change the list of active databases.

FORMAT BASES *[database-list]*

PARAMETERS *database-list*

The *database-list* parameter allows the user to specify the list of databases to become active.

The *database-list* must be either:

- a list of database names separated by plus (+) signs or spaces,
- an abbreviation for a *database-list* the user created with the DEFINE command, or
- the wildcard character (*) which specifies all databases.

The order in which the databases are specified in the *database-list* is important because many commands that process entries from more than one database go through the active databases in the order specified.

DESCRIPTION

Displaying the Table of Databases

If no *database-list* parameter is specified on the command line, the BASES command displays a table showing all the databases included in the term index system. These are the only databases that are accessible through the text searching commands. The list of the *database-list* abbreviations defined by the user is shown below the table.

Changing the Active Databases

If a *database-list* is specified on the command line, the BASES command changes the list of active databases to the list specified.

MODIFIERS

The following table lists the modifiers accepted by the BASES command.

Table 4–3 BASES Command Modifiers

Database List Processing	Modifier Function
/ADD (database_list)	Add specified database(s) to current databases

Table 4–3 (Cont.) BASES Command Modifiers

Database List Processing	Modifier Function
/SUB (database_list)	Subtract specified database(s) from current databases

Alternate Display Formats	Modifier Function
/BRIEF	Display database list without indicating active databases

Redirecting Screen Output	Modifier Function
/PRINTER	Direct screen output to printer
/PRINTER=file-spec	Direct screen output to disk file

EXAMPLES

The following examples illustrate the use of the BASES command. The database configuration and *database-list* abbreviations shown in the first example are assumed in all the subsequent examples.

1

```
ATLAS> BASES
  Database  Entries  Type  Release
  --------  -------  ----  -------
  Database  Entries  Type   Rel.  Description
  --------  -------  ----   ----  -----------
* PIR1        12404  PROT  41.00  Section 1. Classified and Annotated Entries
* PIR2        35689  PROT  41.00  Section 2. Annotated Entries
* PIR3        22755  PROT  41.00  Section 3. Unverified Entries
  NRL_3D       3911  PROT  15.00  NRL Protein Sequences in Brookhaven PDB
  PATCHX      31883  PROT  41.00  Protein Seq DB PATCHX (subseq of MIPSX)
  ECOLI         556  NUCL   2.20  Escherichia coli DNA Database
  GBBCT       15107  NUCL  82.00  GenBank Bacterial (Locus and Title)
  GBEST       33727  NUCL  82.00  GenBank EST (Locus and Title)
  GBINV       11234  NUCL  82.00  GenBank Invertebrate (Locus and Title)
  GBMAM        5628  NUCL  82.00  GenBank Other Mammalian (Locus and Title)
  GBPHG         968  NUCL  82.00  GenBank Phage (Locus and Title)
  GBPAT        5281  NUCL  82.00  GenBank Patent (Locus and Title)
  GBPLN       16154  NUCL  82.00  GenBank Plant (Locus and Title)
  GBPRI       31972  NUCL  82.00  GenBank Primate (Locus and Title)
  GBRNA        3602  NUCL  82.00  GenBank Struct RNA (Locus and Title)
  GBROD       20581  NUCL  82.00  GenBank Rodent (Locus and Title)
  GBSYN        1717  NUCL  82.00  GenBank Synthetic (Locus and Title)
  GBUNA        1490  NUCL  82.00  GenBank Unannotated (Locus and Title)
  GBVRL       15876  NUCL  82.00  GenBank Viral (Locus and Title)
  GBVRT        6558  NUCL  82.00  GenBank Other Vertebrate (Locus and Title)
  GBNEW       18570  NUCL  82.06  GenBank(R) NEW (Locus and Title)
  ALN          1133  TEXT   5.10  Database of family alignments
* indicates an active database. Entries: 70848 active, 296796 total.

  PR*OTEIN = PIR1+PIR2+PIR3+PATCHX+NRL_3D
```

The above example shows the format of the display produced by the BASES command with no parameter.

2 `ATLAS> BASES  PIR1+PIR2+PIR3`

This command changes the list of active databases. After the command executes the new list will consist of three databases, PIR1, PIR2, and PIR3. An alternative way to activate the PIR datasets is shown below where any database code beginning with PIR will be activated:

3 `ATLAS> BASES PIR*`

This command is equivalent to the preceding command.

4 `ATLAS> BASES  PR`

Because the abbreviation PR has been defined (see the last line of example 1), this command is equivalent to "ATLAS> BASES PIR1+PIR2+PIR3+PATCHX+NRL_3D." To define abbreviations for the database-list, use the DEFINE command.

5 `ATLAS> BASES  PR+ALN`

This command will produce an error message. PR is a valid abbreviation, as shown in example 1, but abbreviations cannot be combined with the plus sign.

COPY

The COPY command copies entries into an output file. The output file is independent of the database files; the information in the file can be modified for any particular use.

FORMAT COPY *[entry-identifier]*

PARAMETERS *entry-identifier*

If the *entry-identifier* parameter is omitted from the command line, you will be prompted for it with the prompt:

prompts Code:

If you press RETURN at the code prompt, the program processes the entries on the current list. If there are no entries on the current list, pressing the return key terminates the COPY command.

On the first execution of the COPY command, if the file-specification is not specified on the command line the following prompt will appear:

prompts Output file:

On subsequent executions of COPY, if the file-specification is not given on the command line, then the output is added to the last file created with the COPY command. If the file-specification is given on the command line (using the /OUTPUT=file-spec modifier), then a new file is created.

MODIFIERS

In the following table, modifiers accepted by the COPY command are grouped according to their function. Multiple modifiers can be used together; a slash (/) precedes each modifier with no spaces before or after. A modifier designated as a default does not need to be typed in to be used.

Table 4–4 COPY Command Modifiers

Current List Processing	Modifier Function
/CURRENT	Process only entries on current list

Information Category Selection	Modifier Function
/TEXT	Process only text portion of entry
/SEQUENCE	Process only sequence portion of entry

Table 4–4 (Cont.) COPY Command Modifiers

Redirecting Output	Modifier Function
/OUTPUT=file-spec	Specify output file name
/OUTPUT=*	Append output to previous file
/NOFORMAT	Copy sequence exactly as it is in database (sequence lines up to 500 characters long)

EXAMPLES

The following example demonstrates how the sequence (with entry-identifier PIR1:CCHU) can be copied into an external file.

```
1   ATLAS> COPY CCHU
    Output file: TEST.SEQ
```

The following example shows the contents of the file (TEST.SEQ) just created:

```
2   >P1;CCHU
    cytochrome c - human
     M G D V E K G K K I F I M K C S Q C H T V E K G G K H K T G
     P N L H G L F G R K T G Q A P G Y S Y T A A N K N K G I I W
     G E D T L M E Y L E N P K K Y I P G T K M I F V G I K K K E
     E R A D L I A Y L K K A T N E *
    C;Species: Homo sapiens (man)
    C;Date: #sequence_revision 30-Sep-1991 #text_change 05-Aug-1994
    C;Accession: A31764; A05676; A00001
    R;Evans, M.J.; Scarpulla, R.C.
    Proc. Natl. Acad. Sci. U.S.A. 85, 9625-9629, 1988
    A;Title: The human somatic cytochrome c gene: two classes of processed...
    A;Reference number: A31764; MUID:89071748
    A;Accession: A31764
    A;Molecule type: DNA
    A;Residues: 1-105 <EVA>
    A;Cross-references: GB:M22877
    R;Matsubara, H.; Smith, E.L.
    J. Biol. Chem. 238, 2732-2753, 1963
    A;Title: Human heart cytochrome c. Chymotryptic peptides, tryptic pept...
    A;Reference number: A05676
    A;Accession: A05676
    A;Molecule type: protein
    A;Residues: 2-28;29-46;47-100;101-105 <MATS>
    R;Matsubara, H.; Smith, E.L.
    J. Biol. Chem. 237, 3575-3576, 1962
    A;Title: The amino acid sequence of human heart cytochrome c.
    A;Reference number: A00001
    A;Contents: annotation
    A;Note: 66-Leu is found in 10% of the molecules in pooled protein
    C;Genetics:
    A;Introns: 57/1
    C;Superfamily: cytochrome c; cytochrome c homology
    C;Keywords: acetylated amino end; electron transfer; heme; mitochondrion;...
    F;2-105/Protein: cytochrome c #status experimental <MAT>
    F;2/Modified site: acetylated amino end (Gly) (in mature form) #status...
    F;15,18/Binding site: heme (Cys) (covalent) #status experimental
    F;19,81/Binding site: heme iron (His, Met) (axial ligands) #status pre...
```

Note that the format of the entry looks much different from that shown in Example 1–1. Please see Appendix D for further information regarding the format for external user files.

CROSS

CROSS searches the cross-reference index for all occurrences of a user-specified cross-reference identifier. For each occurrence found, the cross-reference identifier, the entry identifier, and the title for each entry are displayed. The list of these entries becomes the new current list. Currently, the cross-reference index contains information from PIR databases only. This command is an abbreviation for the full command: SEARCH/FIELD=cross_ref.

FORMAT

CROSS *[cross-reference]*

PARAMETERS

cross-reference

The *cross-reference* parameter on the command line is the identifier (or part of the identifier) to be selected. The character string you enter must contain at least 3 characters. If this parameter is omitted, you will be prompted for it with the prompt:

prompts

Cross_ref:

DESCRIPTION

The cross-reference index is constructed from the PIR Cross_reference: lines. A cross-reference identifier enables the user to find the same or similar entry in another database. Each database has its own method of cross-referencing; PIR uses GenBank/EMBL/DDBJ accession numbers as the cross-reference identifiers for the nucleic acid sequence databases.

Normally the cross-reference search matches the specified character string anywhere it appears in the term. Use the /ANCHOR modifier (listed below) to alter this operation.

Displaying the List of Cross-Reference Identifiers

If no cross-reference identifier is typed in at the prompt and you press RETURN, a listing of all cross-reference identifiers and the titles of the associated entries will be displayed.

MODIFIERS

In the following table, modifiers accepted by the CROSS command are grouped according to their function. Multiple modifiers can be used together; a slash (/) precedes each modifier with no spaces before or after. A modifier designated as a default does not need to be typed in to be used.

Table 4–5 CROSS Command Modifiers

Table 4–5 (Cont.) CROSS Command Modifiers

Alternate Display Format	Modifier Function
/BRIEF	Display only index terms found
/COUNTS	Display number of entries containing search term

Current List Processing	Modifier Function
/CURRENT	Process only entries on current list
/ADD	Add entries found to current list
/SUBTRACT	Remove entries found from current list
/KEEP	Do not alter current list with command execution

Redirecting Screen Output	Modifier Function
/PRINTER	Direct screen output to printer
/PRINTER=file-spec	Direct screen output to disk file

Search Manipulation	Modifier Function
/ANCHOR	Match terms beginning with character string
/NOANCHOR (default)	Match character string anywhere in term

EXAMPLES

The following example demonstrates how to use the CROSS command to locate those entries in the PIR-International databases corresponding to accession number M32690 in GenBank entry GBVRL:BIM127.

```
1   ATLAS> B PIR*
    ATLAS> CROSS M32690
    GB:M32690
      PIR1:FOLJBT  gag polyprotein - bovine immunodeficiency virus (isolate 127)
      PIR1:GNLJBT  pol polyprotein - bovine immunodeficiency virus (isolate 127)
      PIR1:VCLJBT  env polyprotein precursor - bovine immunodeficiency virus
                   (isolate 127)
      PIR1:ASLJBT  vif protein - bovine immunodeficiency virus (isolate 127)
      PIR1:TNLJBT  trans-activating transcriptional regulatory protein - bovine
                   immunodeficiency virus (isolate 127)
      PIR1:VKLJBT  trans-regulatory splicing protein - bovine immunodeficiency
                   virus (isolate 127)
      PIR1:ASLJBW  orf-W protein - bovine immunodeficiency virus (isolate 127)
      PIR1:ASLJBY  orf-Y protein - bovine immunodeficiency virus (isolate 127)
      1 cross_ref found
```

DEFINE

The DEFINE command can be used to create abbreviations for:

- database lists
- ATLAS commands
- display layouts

FORMAT

DEFINE *[abbreviation] [text-string]*

PARAMETERS

abbreviation

Specifies the abbreviation.

text-string

Specifies the command or database list that will be referenced via the abbreviation.

DESCRIPTION

DEFINE allows the user to customize commands and database lists. It can also be used to specify a display layout that is used when an entry is displayed with the TYPE command.

Users can have these definitions available each time they use the ATLAS program. Please see the installation document on the CD-ROM for your system.

Defining a Display Layout

A display layout is a specification of the text lines to be displayed when an entry is typed. A text line is specified by giving the string that occurs at the beginning of the line (upper and lower case letters are treated as equal). The string usually consists of a tag (the first two characters on the line) optionally followed by a descriptor and is used to determine the kind of information contained on the line. The tags themselves are not visible with the TYPE command, but are described fully in Appendix D. These tags can also be seen when an entry is copied into an external file with the COPY command. The SHOW/DISPLAY command lists all the symbols that have been defined and the display layouts they represent.

MODIFIERS

The following table lists the modifiers accepted by the DEFINE command.

Table 4–6 DEFINE Command Modifiers

Modifier	Modifier Function
/BASE (default)	Define an abbreviation for a database list
/COMMAND	Define an abbreviation for an ATLAS command

Table 4–6 (Cont.) DEFINE Command Modifiers

Modifier	Modifier Function
/DISPLAY	Define a symbol for a display layout

EXAMPLES

The following examples illustrate the use of the DEFINE command.

1
```
ATLAS> DEFINE/COMMAND EX*IT QUIT
```

This example defines EXIT to be equivalent to the QUIT command. In fact, because of the placement of the asterisk, EX, EXI, and EXIT are all equivalent to the QUIT command.

Another useful customization of commands is to create abbreviations for the full version of the command. For example, to define a command that will cause both the keyword and title indexes to be searched:

2
```
ATLAS> DEFINE/COMMAND KT SEARCH/FIELD=KEYWORD+TITLE
```

To define a database-list abbreviation:

3
```
ATLAS> DEFINE PR PIR1+PIR2+PIR3
```

A database-list abbreviation can be used as the parameter with the BASES command. In the following example, the symbol definition is substituted for the symbol and the databases specified in the symbol-definition become the new active databases. Example:

4
```
ATLAS> BASES PR
```

A database-list abbreviation can be used to specify an entry in the form symbol:entry-code. In this case, the databases specified in the symbol-definition are searched for the entry-code. Example:

5
```
ATLAS> TYPE PR:CCHU
```

6
```
ATLAS> DEFINE/DISPLAY F F;
```

This command defines the symbol F to represent the display layout F;. Then to use the TYPE command with this display layout:

7
```
ATLAS> TYPE/TEXT=F code
```

Only the lines of text that begin with F; (the feature table) will be displayed. Caution must be used in the specification to be sure all appropriate entries are found as illustrated by the following example.

8
```
ATLAS> DEFINE/DISPLAY KEY C;KEYWORDS:
```

This command defines the symbol KEY to represent the display layout C;KEYWORDS:, however, as only the lines that match the specification exactly will be displayed, it will not match lines that begin with C;KEYWORD: (missing the s) or C;KEYWORDS (without the :). These lines will be displayed if KEY is defined as follows:

```
9    ATLAS>  DEFINE/DISPLAY KEY C;KEYWORD
```

In this next example, the display layout consists of a single string that contains a blank illustrating that the specification may contain blanks. A display layout can also specify more than one string, which must be separated by &. A text line will be displayed if it matches any of the strings given in the specification.

```
10   ATLAS> DEFINE/DISPLAY FN F;&N;
```

Here the symbol FN specifies the feature lines plus any lines that begin with N;. Note that if a blank space is placed before the ampersand it is considered part of the first specification and if placed after the ampersand it is considered part of the second specification. In the following example, the symbol REF specifies three line types:

```
11   ATLAS> DEFINE/DISPLAY REF R;&A;REFERENCE NUMBER:&A;RESIDUES:
```

The above discussion on display layouts applies to the PIR databases and may or may not apply to other databases. In particular, it does not work very well with the GenBank database. For example,

```
12   ATLAS> DEFINE/DISPLAY T Title
```

does not display the TITLE lines because the word TITLE begins in column 3 not at the beginning of the line. The following example inserts the two blanks before the word TITLE (the ampersand does not count as a blank space).

```
13   ATLAS> DEFINE/DISPLAY T &  TITLE
```

This correctly displays the TITLE lines but there is another problem. In GenBank, the TITLE line may be continued onto subsequent lines. These lines begin with 12 blanks so they will not be displayed with the above definition. Attempting to include a specification for the continuation lines will cause all continuation lines, not just those for TITLE, to be displayed.

EXTRACT

The EXTRACT command constructs a sequence from the sequence shown in the entry and a set of instructions. The instructions can be entered in response to the prompt for the sequence specification or they can be read from the annotation for the entry. They allow one to insert, delete, and replace residues, and to transpose and/or duplicate sequence elements. The EXTRACT command applies to amino acid sequences only. It is ignored for nucleotide sequences.

FORMAT

EXTRACT *[entry-identifier]*

PARAMETERS

entry-identifier

If the *entry-identifier* parameter is omitted from the command line, you will be prompted for it with the prompt:

prompts

Code:

If you press RETURN at the code prompt, the program processes the entries on the current list. If there are no entries on the current list, then pressing the return key terminates the EXTRACT command.

prompts

Sequence specification:

Enter the instructions for constructing the new sequence.

DESCRIPTION

In response to the prompt for the specification you can type any of the following.

Segment Specifications

These are the instructions for constructing the new sequence from the sequence shown in the entry. The segment specifications are separated by commas or semi-colons, for example,

```
1-7,'SCCF',9-18;35,'T',36-*
```

A comma indicates that the residues are adjacant on the same chain, whereas a semi-colon indicates the residues are on different chains. The valid forms for the segment specifications are:

- N - a single number.

  Residue N from the original sequence is copied into the new sequence.

- N1-N2 - a pair of numbers separated by a dash.

Residues N1 to N2 from the original sequence are copied into the new sequence. An asterisk (*) can be used in place of the second number, N2, to represent the last residue in the original sequence.

- 'xxx' - a string of amino acid symbols enclosed in quotes.

  The symbols within the quotes are copied into the new sequence.

A Code Extension

The code extensions appear in the annotation for the entry at the end of feature table lines or "Residues:" lines of the references. They are enclosed in <..>. You may type the code extension with or without the enclosing <..>, for example,

```
MAT                or                <MAT>
```

? RETURN

This displays the annotation lines of the entry which contain a code extension.

RETURN

Just type RETURN to terminate processing for this entry.

MODIFIERS

In the following table, modifiers accepted by the EXTRACT command are grouped according to their function. Multiple modifiers can be used together; a slash (/) precedes each modifier with no spaces before or after. A modifier designated as a default does not need to be typed in to be used.

Table 4–7 EXTRACT Command Modifiers

Alternate Display Formats	Modifier Function
/FULL_NUMBERING	Display numbering for each line of sequence
/ONE_LETTER (default)	Display one-letter amino acid abbreviations
/THREE_LETTER	Display three-letter amino acid abbreviations

Current List Processing	Modifier Function
/CURRENT[1]	Process only entries on current list

Location of Sequence Specification	Modifier Function
/TABLE	The Specification is read from the entry

Output to a file	Modifier Function
/OUTPUT	Output constructed sequence to a file
/OUTPUT=file-spec	Output to the indicated file
/EXTRACT	Output without verification

[1]When /CURRENT is specified, the *entry-identifier* parameter on the command line should be omitted. If present, it is ignored

Table 4–7 (Cont.) EXTRACT Command Modifiers

Redirecting Screen Output	Modifier Function
/PRINTER	Direct screen output to printer
/PRINTER=file-spec	Direct screen output to disk file

EXAMPLES

The *Klebsiella pneumoniae* entry with identification code ALKBG contains the following item in its feature table.

1

```
31-655              Product: cyclomaltodextrin glucanotransferase
                      #status experimental <MAT>
```

The following example demonstrates the use of the EXTRACT command to construct the product sequence given in the above item.

2

```
ATLAS> EXTRACT ALKBG

PIR1:ALKBG   655 residues
cyclomaltodextrin glucanotransferase (EC 2.4.1.19) precursor - Klebsiella
    pneumoniae
Sequence Specification: MAT

ALKBG->MAT
Product: cyclomaltodextrin glucanotransferase #status experimental
              5        10        15        20        25        30
    1 A E P E E T Y L D F R K E T I Y F L F L D R F S D G D P S N
   31 N A G F N S A T Y D P N N L K K Y T G G D L R G L I N K L P
                                  ...
  571 Q Y P Q W S A S L E L P S D L N V E W K C V K R N E T N P T
  601 A N V E W Q S G A N N Q F N S N D T Q T T N G S F

Residues 31-655 of ALKBG
```

The EXTRACT command was used to extract the product sequence from the sequence shown in the entry.

FEATURE

FEATURE searches the feature table index for all occurrences of a user-specified feature name or partial name. For each name found, the name, the entry-identifier, and the title for each entry in which the feature name occurs are displayed. The list of these entries becomes the new current list. Currently, the feature table index includes information from PIR entries only. This command is an abbreviation for the full command SEARCH/FIELD=feature.

FORMAT

FEATURE *[feature-name]*

PARAMETERS

feature-name

The *feature-name* parameter on the command line is the name (or part of the name) to be selected. The character string you enter must contain at least 3 characters. If this parameter is omitted, you will be prompted for it with the prompt:

prompts

Feature:

DESCRIPTION

The index is constructed from the feature table lines in PIR entries. Some examples of features are binding sites, active sites, modified sites, etc. Normally the search matches the specified string anywhere it appears in the feature. Use the /ANCHOR modifier (listed below) to alter this operation.

Displaying the List of Features

If no *feature name* is typed in at the prompt and you press RETURN, an alphabetical listing of all features and the titles of their associated entries will be displayed. If the /BRIEF modifier (listed below) is used only the feature names will be displayed.

MODIFIERS

In the following table, modifiers accepted by the FEATURE command are grouped according to their function. Multiple modifiers can be used together; a slash (/) precedes each modifier with no spaces before or after. A modifier designated as a default does not need to be typed in to be used.

Table 4–8 FEATURE Command Modifiers

Alternate Display Formats	Modifier Function
/BRIEF	Display only index terms found

Table 4–8 (Cont.) FEATURE Command Modifiers

Alternate Display Formats	Modifier Function
/COUNTS	Display number of entries containing search term
/SHOW	Display how search string is parsed

Database List Processing	Modifier
/PIR	Ignore all databases except PIR1, PIR2, and PIR3

Current List Processing	Modifier Function
/CURRENT	Process only entries on current list
/ADD	Add entries found to current list
/SUBTRACT	Remove entries found from current list
/KEEP	Do not alter current list with command execution

Redirecting Screen Output	Modifier Function
/PRINTER	Direct screen output to printer
/PRINTER=file-spec	Direct screen output to disk file

Search Manipulation	Modifier Function
/ANCHOR	Match terms beginning with character string
/NOANCHOR (default)	Match character string anywhere in term
/ENTRY	Match each character string to separate terms

EXAMPLES

The following example demonstrates the feature command:

1
```
ATLAS> FEATURE SELENOCYSTEINE
Modified site: selenocysteine #status experimental
  PIR1:OPBOE   glutathione peroxidase (EC 1.11.1.9) - bovine
  PIR1:OPRTE   glutathione peroxidase (EC 1.11.1.9) I - rat
  PIR1:OMRTSP  selenoprotein P precursor - rat
Modified site: selenocysteine #status predicted
  PIR1:DEECFS  formate dehydrogenase (EC 1.2.1.2) (benzylviologen-linked)
               selenocysteine-containing protein - Escherichia coli
  PIR1:OPHUE   glutathione peroxidase (EC 1.11.1.9) - human
  PIR1:OPMSE   glutathione peroxidase (EC 1.11.1.9) - mouse
  PIR1:HQDVLB  hydrogenase (EC 1.18.99.1) (NiFeSe) large chain - Desulfovibrio
               baculatus
  PIR1:A47327  selenoprotein P precursor - human
  PIR1:OMRTSP  selenoprotein P precursor - rat
  2 features found
```

The following example illustrates use of the RESID Database to help find appropriate features:

```
2   ATLAS> BASES PIR1
    ATLAS> FEATURE PYROGLUTAMIC ACID
    Feature: pyroglutamic acid
      No features found
```

The user cannot find a feature named "pyroglutamic acid" in the Protein Sequence Database. By using the RESID database, "pyroglutamic acid" is found to be an alternate name for "2-pyrrolidone-5-carboxylic acid" and that the feature appears as "Modified site: pyrrolidone carboxylic acid" in the Protein Sequence Database.

```
ATLAS> BASES RESID

ATLAS> TITLE PYROGLUTAMIC ACID
pyroglutamic acid
  RESID:AA0031  2-pyrrolidone-5-carboxylic acid
  1 title found

ATLAS> TYPE/CURRENT
RESID:AA0031
2-pyrrolidone-5-carboxylic acid

Alternate names: pyroglutamic acid; 5-oxoproline

Systematic name: (S)-5-oxo-2-pyrrolidinecarboxylic acid
    Cross-references: CAS:98-79-3

Formula: C 5 H 6 N 1 O 2
    Formula weight: #chem 112.11 #phys 112.0399

Correction formula: C 0 H -2 O -1
    Correction weight: #chem -18.02 #phys -18.0106

Date: 31-Mar-1995 #structure_revision 31-Mar-1995 #text_change 01-Sep-
    1995

Podell, D.N.; Abraham, G.N.
    Biochem. Biophys. Res. Commun. 81, 176-185, 1978
    Title: A technique for the removal of pyroglutamic acid from the
      amino terminus of proteins using calf liver pyroglutamate amino
      peptidase.
    Reference number: A44724

Comment: This modification can form non-enzymatically from amino-
    terminal glutamine.

Generating enzyme: glutaminyl-peptide cyclotransferase (EC 2.3.2.5)

Sequence code: Q; Z
    Conditions: amino-terminal

Keywords: pyroglutamic acid

Residues            Feature
                    Modified site: pyrrolidone carboxylic acid (Gln)
                    Modified site: pyrrolidone carboxylic acid (Glx)
```

```
ATLAS> BASES PIR1
ATLAS> FEATURE/BRIEF/COUNT PYRROLIDONE CARBOXYLIC ACID
     1  Modified site: blocked amino end (Gln) (in mature form) (probably
         pyrrolidone carboxylic acid) #status experimental
    20  Modified site: blocked amino end (Gln) (probably pyrrolidone
         carboxylic acid) #status experimental
     1  Modified site: blocked amino end (Glx) (probably pyrrolidone
         carboxylic acid) #status experimental
    97  Modified site: pyrrolidone carboxylic acid (Gln) #status
         experimental
     2  Modified site: pyrrolidone carboxylic acid (Gln) #status predicted
    47  Modified site: pyrrolidone carboxylic acid (Gln) (in mature form)
         #status experimental
    16  Modified site: pyrrolidone carboxylic acid (Gln) (in mature form)
         #status predicted
     2  Modified site: pyrrolidone carboxylic acid (Gln) (in mature form)
         (partial) #status experimental
     1  Modified site: pyrrolidone carboxylic acid (Gln) (partial) #status
         experimental
  9 features found
```

FIND

FIND searches the title index for all occurrences of a user-specified title or partial title. For each found, the entry-identifier and the title for each entry in which the search-string occurs are displayed. The list of these entries becomes the new current list. This command is an abbreviation for the full command: SEARCH/FIELD=title.

FORMAT

FIND *[title]*

PARAMETERS

title

The *title* parameter on the command line is the name (or part of the name) to be selected. The character string you enter must contain at least 3 characters. If this parameter is omitted, you will be prompted for it with the prompt:

prompts

Title:

DESCRIPTION

The FIND command is the primary method for locating an entry in the databases by searching through the title index. For PIR entries, the title index is constructed from the title line, the alternate names and the contains fields.

Displaying the List of All Titles

If no title is typed in at the prompt and you press RETURN, a listing of all the titles in all the active databases will be displayed.

MODIFIERS

In the following table, modifiers accepted by the FIND command are grouped according to their function. Multiple modifiers can be used together; a slash (/) precedes each modifier with no spaces before or after. A modifier designated as a default does not need to be typed in to be used.

Table 4–9 FIND Command Modifiers

Alternate Display Formats	Modifier Function
/BRIEF	Display only index terms found
/COUNTS	Display number of entries containing search term
/SHOW	Display how search string is parsed

Table 4–9 (Cont.) FIND Command Modifiers

Database List Processing	Modifier
/PIR	Ignore all databases except PIR1, PIR2, and PIR3

Current List Processing	Modifier Function
/CURRENT	Process only entries on current list
/ADD	Add entries found to current list
/SUBTRACT	Remove entries found from current list
/KEEP	Do not alter current list with command execution

Redirecting Screen Output	Modifier Function
/PRINTER	Direct screen output to printer
/PRINTER=file-spec	Direct screen output to disk file

Search Manipulation	Modifier Function
/ANCHOR	Match terms beginning with character string
/NOANCHOR (default)	Match anywhere character string appears
/ENTRY	Match each character string to separate terms
/MAIN	Search TITLE field only NOT ALTERNATE NAMES or CONTAINS fields
/TEXT	Perform string search of titles (very slow)

EXAMPLES

When using the FIND command, type words (or parts of words) that characterize the entry. In general, it is best to type short words (each word must be at least 3 characters in length) that would occur regardless of alternate spellings of a name. The following examples demonstrate typical executions of the FIND command:

1
```
ATLAS> BASES PIR1
ATLAS> FIND HUMAN INSULIN GROWTH FACTOR
insulin-like growth factor IA precursor - human
    PIR1:IGHU1   insulin-like growth factor IA precursor - human
  insulin-like growth factor IB precursor - human
    PIR1:IGHU1B  insulin-like growth factor IB precursor - human
  insulin-like growth factor II precursor - human
    PIR1:IGHU2   insulin-like growth factor II precursor - human
  insulin-like growth factor-binding protein 1 precursor - human
    PIR1:IOHU1   insulin-like growth factor-binding protein 1 precursor - human
  insulin-like growth factor-binding protein 2 precursor - human
    PIR1:A41927  insulin-like growth factor-binding protein 2 precursor - human
  insulin-like growth factor-binding protein 3 precursor - human
    PIR1:IOHU3   insulin-like growth factor-binding protein 3 precursor - human
  6 titles found
```

In the above example, six entries were found in the PIR1 database.

The following command sequence will generate a current list of methionine tRNA ligases and histidine tRNA ligases in PIR1 from species other than *E. coli*.

```
2   ATLAS> BASES PIR1
    ATLAS> FIND METHIO TRNA LIG
    methionine--tRNA ligase (EC 6.1.1.10) - Escherichia coli
      PIR1:SYECMT  methionine--tRNA ligase (EC 6.1.1.10) - Escherichia coli
    methionine--tRNA ligase (EC 6.1.1.10) - Thermus aquaticus
      PIR1:SYTWMT  methionine--tRNA ligase (EC 6.1.1.10) - Thermus aquaticus
    methionine--tRNA ligase (EC 6.1.1.10), cytosolic - yeast (Saccharomyces)
                cerevisiae
      PIR1:SYBYMT  methionine--tRNA ligase (EC 6.1.1.10), cytosolic - yeast
                   (Saccharomyces cerevisiae)
    methionine--tRNA ligase (EC 6.1.1.10), mitochondrial - yeast (Saccharomyces)
                cerevisiae
      PIR1:SYBYMM  methionine--tRNA ligase (EC 6.1.1.10), mitochondrial - yeast
                   (Saccharomyces cerevisiae)
      4 titles found


3   ATLAS> FIND/ADD HISTIDIN TRNA LIG
    histidine--tRNA ligase (EC 6.1.1.21) - Chinese hamster
      PIR1:SYHYHT  histidine--tRNA ligase (EC 6.1.1.21) - Chinese hamster
    histidine--tRNA ligase (EC 6.1.1.21) - Escherichia coli
      PIR1:SYECH   histidine--tRNA ligase (EC 6.1.1.21) - Escherichia coli
    histidine--tRNA ligase (EC 6.1.1.21) - human
      PIR1:SYHUHT  histidine--tRNA ligase (EC 6.1.1.21) - human
    histidine--tRNA ligase (EC 6.1.1.21), cytosolic - yeast (Saccharomyces)
                cerevisiae
      PIR1:SYBYHC  histidine--tRNA ligase (EC 6.1.1.21), cytosolic - yeast
                   (Saccharomyces cerevisiae)
    histidine--tRNA ligase (EC 6.1.1.21), mitochondrial - yeast (Saccharomyces)
                cerevisiae
      PIR1:SYBYHM  histidine--tRNA ligase (EC 6.1.1.21), mitochondrial - yeast
                   (Saccharomyces cerevisiae)
      5 titles found


4   ATLAS> FIND/SUB ESCHER COLI
    histidine--tRNA ligase (EC 6.1.1.21) - Escherichia coli
      PIR1:SYECH   histidine--tRNA ligase (EC 6.1.1.21) - Escherichia coli
    methionine--tRNA ligase (EC 6.1.1.10) - Escherichia coli
      PIR1:SYECMT  methionine--tRNA ligase (EC 6.1.1.10) - Escherichia coli
      2 titles found


5   ATLAS> LIST
      7 entries on the current list

    PIR1:SYBYMT  methionine--tRNA ligase (EC 6.1.1.10), cytosolic - yeast
                     (Saccharomyces cerevisiae)
    PIR1:SYBYMM  methionine--tRNA ligase (EC 6.1.1.10), mitochondrial - yeast
                     (Saccharomyces cerevisiae)
    PIR1:SYTWMT  methionine--tRNA ligase (EC 6.1.1.10) - Thermus aquaticus
    PIR1:SYHUHT  histidine--tRNA ligase (EC 6.1.1.21) - human
    PIR1:SYHYHT  histidine--tRNA ligase (EC 6.1.1.21) - Chinese hamster
    PIR1:SYBYHM  histidine--tRNA ligase (EC 6.1.1.21), mitochondrial - yeast
                     (Saccharomyces cerevisiae)
    PIR1:SYBYHC  histidine--tRNA ligase (EC 6.1.1.21), cytosolic - yeast
                     (Saccharomyces cerevisiae)
```

The FIND METHIO TRNA LIG command produced a current list of methionine tRNA ligases. The FIND/ADD HISTIDIN TRNA LIG command resulted in the addition of histidine tRNA ligases to the list, and the FIND/SUB ESCHER COLI command resulted in the removal of *E. coli* sequences from this list. The current list generated as a result of these

operations is displayed using the LIST command described later in this chapter.

GENE

GENE searches the gene name index for all occurrences of a user-specified gene name. For each name found, the name, the entry identifier, and the title for each entry in which the name occurs are displayed. The list of these entries becomes the new current list. This command is an abbreviation for the full command: SEARCH/FIELD=gene_name.

FORMAT

GENE *[gene-name]*

PARAMETERS

gene-name

The *gene-name* parameter on the command line is the name (or part of the name) to be selected. The character string you enter must contain at least 3 characters. If this parameter is omitted, you will be prompted for it with the prompt:

prompts

Gene:

DESCRIPTION

The gene index is constructed from the Gene name: lines in PIR entries.

Normally the gene name search matches the specified character string anywhere it appears within the gene name. Use the /ANCHOR modifier (listed below) to alter this operation.

Displaying the List of Gene Names

If no gene-name is typed in at the prompt and you press RETURN, an alphabetical listing of all gene names and the titles of their associated entries will be displayed. If the /BRIEF modifier (listed below) is used, only the gene names will be displayed.

MODIFIERS

In the following table, modifiers accepted by the GENE command are grouped according to their function. Multiple modifiers can be used together; a slash (/) precedes each modifier with no spaces before or after. A modifier designated as a default does not need to be typed in to be used.

Table 4–10 GENE Command Modifiers

Alternate Display Format	Modifier Function
/BRIEF	Display only index terms found
/COUNTS	Display number of entries containing search term
/SHOW	Display how search string is parsed

Table 4–10 (Cont.) GENE Command Modifiers

Current List Processing	Modifier Function
/CURRENT	Process only entries on current list
/ADD	Add entries found to current list
/SUBTRACT	Remove entries found from current list
/KEEP	Do not alter current list with command execution

Redirecting Screen Output	Modifier Function
/PRINTER	Direct screen output to printer
/PRINTER=file-spec	Direct screen output to disk file

Search Manipulation	Modifier Function
/ANCHOR	Match terms beginning with character string
/NOANCHOR (default)	Match character string anywhere in term
/ENTRY	Match each character string to separate terms

EXAMPLES

The command line:

1
```
ATLAS> GENE ERA
```

will produce the following output:

2
```
At-ERabp
  PIR1:S31584  auxin-binding protein precursor - Arabidopsis thaliana
cytokeratin VIb
  PIR1:KRBOVI  keratin, 54K type I cytoskeletal - bovine
era
  PIR1:RGECGT  transforming protein homolog (ras) - Escherichia coli
ERabp
  PIR1:S16262  auxin-binding protein precursor - maize
merA
  PIR1:RDPSHA  mercury(II) reductase (EC 1.16.1.1) - Pseudomonas aeruginosa
               transposon Tn501
  PIR1:RDEBHA  mercury(II) reductase (EC 1.16.1.1) - Shigella flexneri plasmid
               R100
serA
  PIR1:DEECPG  phosphoglycerate dehydrogenase (EC 1.1.1.95) - Escherichia coli
  6 gene_names found
```

From the results of the search, it is obvious that the GENE command is unanchored as the default. An anchored search narrows the list as in the following example:

3
```
ATLAS> GENE/ANCHOR ERA
era
  PIR1:RGECGT  Transforming protein homolog (ras) - Escherichia coli
ERabp
  PIR1:S16262  auxin-binding protein precursor - maize
  2 gene_names found
```

GET

GET generates or modifies a current list from entry-codes either in an external file or specified by the user.

FORMAT

GET *[file-name]*

PARAMETERS

file-name

The *file-name* parameter on the command line identifies the file that contains the list of entry codes. The default file type is .COD. If this parameter is omitted on the command line, you will be prompted for it with the prompt:

prompts

File:

DESCRIPTION

Although not a text or data searching command, the GET command also accepts the current list modifiers. These modifiers allow the current list to be generated or modified by reading a list of entry-identifiers from an external file or specified by the user.

The external file must contain one entry-code per line and the entry-code must begin in column 1 of each line. If column 1 is blank the entire line is ignored (this is useful for inserting comments into the file).

The /USER modifier allows the user to directly specify entry-codes and thereby edit the current list.

If the entry-code is preceded by the database-code and if the database is not an active database, then the entry is ignored.

If the entry-code is not preceded by the database-code, then the active databases are searched.

MODIFIERS

In the following table, modifiers accepted by the GET command are grouped according to their function. Multiple modifiers can be used together; a slash (/) precedes each modifier with no spaces before or after.

Table 4–11 GET Command Modifiers

Current List Processing	Modifier Function
/CURRENT	Process only entries on current list
/ADD	Add entries found to current list
/SUBTRACT	Remove entries found from current list
/USER	Entry-codes are specified by user

EXAMPLES

The following examples demonstrate how to bring an external file of entry-codes into the ATLAS program to become the new current list.

```
1   ATLAS> GET TEST.COD
```

In this example, TEST.COD is the file-name for the list of entry-codes. The next example shows the contents of TEST.COD:

```
2   PIR1:CCHU
    PIR1:IGHU1
    PIR1:IGHU1B
    PIR1:IGHU2
```

The list may be typed into a file using an editor or it can be one that is created using the LIST/OUTPUT=file-spec command described in detail later in this chapter.

The next example shows an easy way to subtract an unwanted entry from the current list:

```
3   ATLAS> GET/USER/SUB
    Type the codes one per line with no spaces.
    Type a null line to exit.
    CCHU

      1 entry found


4   ATLAS> LIST
      3 entries on the current list

     PIR1:IGHU1   insulin-like growth factor IA precursor - human
     PIR1:IGHU1B  insulin-like growth factor IB precursor - human
     PIR1:IGHU2   insulin-like growth factor II precursor - human
```

The LIST command shows the current list without entry-code CCHU.

HELP

HELP displays a list of the individual commands and command modifiers.

FORMAT HELP *[command-name]*

PARAMETERS *command-name*

The *command-name* can be any of the commands of the ATLAS program. If no command-name is typed in at the prompt and you press RETURN, a listing of all topics on which HELP is available is displayed.

MODIFIERS

Table 4–12 HELP Command Modifiers

Alternate Display Format	Modifier Function
/BRIEF	List ATLAS commands and their modifiers

Redirecting Screen Output	Modifier Function
/PRINTER	Direct screen output to printer
/PRINTER=file-spec	Direct screen output to disk file

JOURNAL

JOURNAL searches the index of literature citations for all occurrences of a user-specified citation or partial citation. For each citation found, the full citation, the entry identifier, and the title for each entry that contains the citation are displayed. The list of these entries becomes the new current list. This command is an abbreviation for the full command: SEARCH/FIELD=journal.

FORMAT

JOURNAL *[literature-citation]*

PARAMETERS

literature-citation

The *literature-citation* parameter on the command line is the citation (or part of the citation) to be selected. The character string(s) you enter must contain at least 3 characters. If this parameter is omitted on the command line, you will be prompted for it with the prompt:

prompts

Journal:

DESCRIPTION

Normally the literature citation search matches a citation whenever the character string specified occurs anywhere within a citation. Use the /ANCHOR modifier to cause the search to match citations only when the specified character string occurs at the beginning of the citation.

Displaying the List of Literature Citations

If no *literature-citation* is typed in at the prompt and you press RETURN, an alphabetical listing of all citations and the titles of their associated entries will be displayed. If the /BRIEF modifier (listed below) is used only the citations will be displayed.

MODIFIERS

In the following table, modifiers accepted by the JOURNAL command are grouped according to their function. Multiple modifiers can be used together; a slash (/) precedes each modifier with no spaces before or after. A modifier designated as a default does not need to be typed in to be used.

Table 4–13 JOURNAL Command Modifiers

Alternate Display Formats	Modifier Function
/BRIEF	Display only index terms found
/COUNTS	Display number of entries containing search term
/SHOW	Display how search string is parsed

Table 4–13 (Cont.) JOURNAL Command Modifiers

Database List Processing	Modifier
/PIR	Ignore all databases except PIR1, PIR2, and PIR3

Current List Processing	Modifier Function
/CURRENT	Process only entries on current list
/ADD	Add entries found to current list
/SUBTRACT	Remove entries found from current list
/KEEP	Do not alter current list with command execution

Redirecting Screen Output	Modifier Function
/PRINTER	Direct screen output to printer
/PRINTER=file-spec	Direct screen output to disk file

Search Manipulation	Modifier Function
/ANCHOR	Match terms beginning with character string
/NOANCHOR (default)	Match anywhere character string appears
/ENTRY	Match each character string to separate terms

EXAMPLES

The command line:

1
```
ATLAS> JOURNAL/BRIEF 1990
```

will produce the list of all literature citations in the active databases for the year 1990. The list of entries containing such citations becomes the current list.

Note: Citations in which 1990 occurs in a context other than as the year, such as a page number (or part of a page number), will also be found and will appear on the current list of citations.

To find all entries in PIR and GenBank that cite a given article, it is usually sufficient to give three character-strings: one representing part of the journal name abbreviation, one representing the page number(s), and one representing the volume number or year. (These elements need not be in the order in which they appear in the citation.) For example,

2
```
ATLAS> JOURNAL
Journal: acids 1105  1982
Nucleic Acids Res. 10, 1105-1112, 1982
  PIR1:KIBPD4  deoxynucleotide monophosphate kinase (EC 2.7.1.-) - Phage T4
  PIR1:ZABPT4  gene 57A protein - Phage T4
  1 journal found
```

Enclose low page numbers in quotes, including a space on either side as in the following example:

```
3    ATlAS> jou Acta 446 " 1-9 "
     Biochim. Biophys. Acta 446, 1-9, 1976
       PIR1:H3NJ1C  cytotoxin 1 - Cape cobra
       PIR1:H3NJ3C  cytotoxin 3 - Cape cobra
       PIR1:H3NJ2C  cytotoxin 2 - Cape cobra
       1 journal found
```

KEYWORD

KEYWORD searches the keyword index for all occurrences of a user-specified keyword or partial keyword. For each keyword found, the keyword, the entry-identifier, and the title for each entry in which the keyword occurs are displayed. The list of these entries becomes the new current list. This command is an abbreviation for the full command: SEARCH/FIELD=keyword.

FORMAT

KEYWORD *[keyword]*

PARAMETERS

keyword

The keyword parameter on the command line is the keyword (or part of the keyword) to be selected. If this parameter is omitted on the command line, you will be prompted for it with the prompt:

prompts

Keyword:

DESCRIPTION

The keyword index is constructed from the Keywords: line of each entry. Normally the keyword search matches a keyword whenever the character string specified occurs anywhere within the keyword. Use the /ANCHOR modifier (listed below) to cause the search to match keywords only when the specified character string occurs at the beginning of the keyword.

Displaying the List of Keywords

If no keyword is typed in at the prompt and you press RETURN, an alphabetical listing of all keywords and the titles of their associated entries will be displayed. If the /BRIEF modifier (listed below) is used only the keywords will be displayed.

MODIFIERS

In the following table, modifiers accepted by the KEYWORD command are grouped according to their function. Multiple modifiers can be used together; a slash (/) precedes each modifier with no spaces before or after. A modifier designated as a default does not need to be typed in to be used.

Table 4–14 KEYWORD Command Modifiers

Alternate Display Formats	Modifier Function
/BRIEF	Display only index terms found
/COUNTS	Display number of entries containing the keyword
/SHOW	Display how search string is parsed

Table 4–14 (Cont.) KEYWORD Command Modifiers

Database List Processing	Modifier
/PIR	Ignore all databases except PIR1, PIR2, and PIR3

Current List Processing	Modifier Function
/CURRENT	Process only entries on current list
/ADD	Add entries found to current list
/SUBTRACT	Remove entries found from current list
/KEEP	Do not alter current list with command execution

Redirecting Screen Output	Modifier Function
/PRINTER	Direct screen output to printer
/PRINTER=file-spec	Direct screen output to disk file

Search Manipulation	Modifier Function
/ANCHOR	Match terms beginning with character string
/NOANCHOR (default)	Match character string anywhere in term
/ENTRY	Match each character string to separate terms

EXAMPLES

The following examples demonstrate how to generate a list of terms to select from when a single keyword may not be specific enough. Then use the keyword command to retrieve a more specific list.

1
```
ATLAS> B PIR*
ATLAS> KEYWORD/BRIEF ADHESION
cell adhesion
membrane adhesion
pili adhesion
  3 keywords found
```

2

```
ATLAS> KEYWORD CELL ADHESION
cell adhesion
  PIR1:BNRT3   myelin-associated glycoprotein precursor, long form - rat
  PIR1:BNRT3S  myelin-associated glycoprotein precursor, short form - rat
  PIR1:RWHU1B  cell surface glycoprotein CD11b precursor - human
  PIR1:RWHU1C  cell surface glycoprotein CD11c precursor - human
  PIR1:FNHU    fibronectin precursor - human
  PIR1:IJFFTM  cadherin-related tumor suppressor precursor - fruit fly
               (Drosophila melanogaster)
  PIR1:IJHUCN  N-cadherin precursor, neuronal - human
  PIR1:IJBOCN  N-cadherin precursor - bovine (fragment)
  PIR1:IJMSCN  N-cadherin precursor, neuronal - mouse
  PIR1:IJCHCN  N-cadherin precursor, neuronal - chicken
  PIR1:IJXLC2  N-cadherin 2 precursor - African clawed frog
  PIR1:IJXLC1  N-cadherin 1 precursor - African clawed frog
  PIR1:A47543  R-cadherin precursor - mouse
  PIR1:IJCHCR  R-cadherin precursor - chicken
  PIR1:IJHUCE  E-cadherin precursor - human
  .
  .
  .
  PIR2:S19872  vascular cell adhesion molecule 1 precursor - rat
  PIR2:JS0675  vascular cell adhesion molecule-1 precursor - rat
  PIR2:B37057  integrin beta-6 chain - guinea pig (fragment)
  1 keyword found
```

LIST

The LIST command can be used to:

- Display the entry identifier and title of every entry on the current list
- Display the entry identifier and title of every entry in the active databases (/ALL)
- Save the entry-identifiers to a file (/OUTPUT=file-spec)
- Save the current list to a file (/PRINTER=file-spec)
- Restore the internally saved current list (/RESTORE)

FORMAT LIST

DESCRIPTION

The LIST command displays the entry-identifier and title of each entry on the current list.

Before each execution of a command that generates a new current list, the old current list is saved. If execution of a command does not produce the desired new current list, the old current list can be restored with the LIST/RESTORE command.

The LIST/OUTPUT=file-spec and LIST/PRINTER=file-spec differ in that the /OUTPUT modifier will save a list of only the database-codes and entry-codes for entries on the current list whereas the /PRINTER=file-spec modifier saves the screen output including both the entry-identifiers AND titles. The current list saved with a LIST/OUTPUT=file-spec command can be generated again very quickly with the GET command.

MODIFIERS

In the following table, modifiers accepted by the LIST command are grouped according to their function. Multiple modifiers can be used together; a slash (/) precedes each modifier with no spaces before or after. A modifier designated as a default does not need to be typed in to be used.

Table 4–15 LIST Command Modifiers

Current List Processing	Modifier Function
/CURRENT (default)	Process only entries on current list
/ALL	List all entries in the active databases
/RESTORE	Restore previous current list
/SET	Active databases become current list

Table 4–15 (Cont.) LIST Command Modifiers

Redirecting Output	Modifier Function
/OUTPUT=file-spec	Direct current list entry-codes to a file
/PRINTER	Direct current list codes and titles to printer
/PRINTER=file-spec	Direct current list codes and titles to disk file

MATCH

The MATCH command searches a sequence for all segments that match a user-specified peptide. A similar command, SCAN, searches a precomputed index of amino acid locations and is suitable when there is a large number of sequences. MATCH goes through the sequence and compares the unknown peptide with the sequence residue by residue; therefore it is suitable when there is a small number of sequences. MATCH can be used to search an entire database but it is quite slow (e.g., it takes several minutes to search just PIR1.)

FORMAT

MATCH *[entry-identifier]*

PARAMETERS

entry-identifier

If the *entry-identifier* parameter is omitted from the command line, you will be prompted for it with the prompt:

prompts

Code:

If you press RETURN at the code prompt, the program will process the entries on the current list. This is equivalent to using the modifier /CURRENT.

prompts

Peptide:

Enter a string of amino acid symbols (A,C,D,E,F,G,H,I,K,L,M,N,O,P,Q,R,S,T,V,W,Y or ambiguous amino acid symbols B = N or D, Z = E or Q, and X = any amino acid). The MATCH command ignores spaces and hyphens (-) in the search string entered by the user. The use of the special symbols + * ( ) [ ] is described below.

The last search string entered can be recalled by typing a period followed by RETURN. Just type RETURN to exit the MATCH command. The maximum length of the unknown peptide is 30 residues.

prompts

Number of mismatches allowed (e to exit):

Enter a number. The last number entered can be recalled by typing a period followed by RETURN. Just typing RETURN is equivalent to entering 0.

DESCRIPTION

For each segment found, the display shows:

the number of the first residue in the matching segment,
the ten residues preceding the matching segment,
the matching segment,

the ten residues following the matching segment.

In the case of nonexact matches, the display also shows the number of mismatches.

Use of + and * in the unknown peptide

The symbol + preceding the peptide limits the search to the amino end of the sequence. No other symbols can appear before the +. The symbol * following the peptide limits the search to the carboxyl end of the sequence. No other symbols can appear after the *.

Use of parentheses in the unknown peptide

One or more regions of the unknown peptide may be enclosed by parentheses. No mismatches are allowed for those regions within the parentheses. For example, if the peptide is KG(HPETL)EK(VQK), then mismatches can occur only in the KG and EK regions. The regions HPETL and VQK must match exactly. The MATCH command does not find insertion and/or deletion events.

Use of brackets in the unknown peptide

One or more regions of the unknown peptide may be enclosed by brackets. The residues within the brackets denote ambiguous residues at a position in the peptide. For example, if the peptide is KG[HP]EK[VQK], then there are 6 positions in the peptide. For a match, at position 3 either H or P can occur in the sequence and at position 6 either V, Q, or K can occur.

Note that positions 4 and 6 will also match Z because Z = E or Q. The ambiguous symbol B = D or N is treated similarly. However, an X in the unknown peptide will match any residue in the sequence, but an X in the sequence will only match an X in the peptide.

Both parentheses and brackets may be used in the unknown peptide, but a parenthesis cannot occur within a bracketed region.

When ambiguous positions are specified with the [..] notation the display shows a dot at those positions. The /SHOW modifier can be used to show the positions in the peptide and the residues in the sequence that will produce a match.

The two modifiers /PEPTIDE=peptide and /MISMATCHES=number inhibit prompting for the search parameters. With /PEPTIDE=peptide the peptide is entered on the command line and it is used in all subsequent searches until the command terminates. The same holds for /MISMATCHES=number. The number of mismatches is read from the command line and used on subsequent searches until the command terminates. When you use either modifier on the command line there is no prompt for the corresponding search parameter.

MODIFIERS

In the following table, modifiers accepted by the MATCH command are grouped according to their function. Multiple modifiers can be used together; a slash (/) precedes each modifier with no spaces before or after.

Table 4–16 MATCH Command Modifiers

Table 4–16 (Cont.) MATCH Command Modifiers

Search Parameters	Modifier Function
/PEPTIDE=peptide	Specify the peptide
/MISMATCHES=number	Specify the maximum number of mismatches allowed
/SHOW	Shows the positions in the peptide and the matching residues

Current List Processing	Modifier Function
/CURRENT	Process entries on the current list
/DEFINE	Affect the resultant current list
/DEFINE/ADD	Add entries found to the current list
/DEFINE/SUBTRACT	Remove entries found form the current list
/FULL	Report No matches found, normally these messages are suppressed

Redirecting Screen Output	Modifier Function
/PRINTER	Direct screen output to printer
/PRINTER=file-spec	Direct screen output to disk file

EXAMPLES

To be able to specify multiple possibilities for residue positions is a powerful feature of the MATCH command. In the following example, chymase sequences are searched for possible carbohydrate binding sites (Asn) and a current list is created.

```
1    ATLAS>B PIR*
     ATLAS>FIND CHYMASE
     .
     .
     .
     10 titles found
     ATLAS>MATCH/CURRENT/DEFINE
     Peptide:NX[ST]
     Number of mismatches allowed (e to exit):0

     PIR1:KYHUCM      247 residues
     chymase (EC 3.4.21.39) precursor - human
     Matching---------- NX.
     residue             !
          80  RSITVTLGAH NIT EEEDTWQKLE
         103  VIKQFRHPKY NTS TLHHDIMLLK

     PIR1:A46504      246 residues
     chymase (EC 3.4.21.39) precursor - mouse
         102  VEKQIIHKNY NVS FNLYDIMLLK

     PIR2:A35842      249 residues
     chymase (EC 3.4.21.39) precursor - dog
         121  IMLLKLKEKA NLT LAVGTLPLSP
         155  RVAGWGKRQV NGS GSDTLQEVKL

     PIR2:A41076      247 residues
     chymase (EC 3.4.21.39) 5 precursor - mouse
          80  RSITVLLGAH NKT SKEDTWQKLE
```

```
PIR2:A46721      244 residues
chymase (EC 3.4.21.39) precursor - mouse
     44  YMAYLKFTTK NGS KERCGGFLIA
     67  PQFVMTAAHC NGS EISVILGAHN

PIR2:S23505      177 residues
chymase (EC 3.4.21.39) 1 - mouse
     33  VEKYILPPNY NVS SKFNDIVLLK
     51  IVLLKLEKQA NLT SAVDVVPLPA

PIR2:S23504      247 residues
chymase (EC 3.4.21.39) 2 - mouse
     80  RSITVLLGAH NKT SKEDTWQKLE

PIR3:S33247      226 residues
chymase (EC 3.4.21.39) - human
     59  RSITVTLGAH NIT EEEDTWQKLE
     82  VIKQFRHPKY NTS TLHHDIMLLK

ATLAS>LIST
  8 entries on the current list

PIR1:KYHUCM  chymase (EC 3.4.21.39) precursor - human
PIR1:A46504  chymase (EC 3.4.21.39) precursor - mouse
PIR2:A35842  chymase (EC 3.4.21.39) precursor - dog
PIR2:A41076  chymase (EC 3.4.21.39) 5 precursor - mouse
PIR2:A46721  chymase (EC 3.4.21.39) precursor - mouse
PIR2:S23505  chymase (EC 3.4.21.39) 1 - mouse
PIR2:S23504  chymase (EC 3.4.21.39) 2 - mouse
PIR3:S33247  chymase (EC 3.4.21.39) - human
```

MEMBERS

MEMBERS searches the members index for all occurrences of a user-specified PIR entry-code. For each code found, the code, the alignment identifier, and the title for each alignment in which the code occurs are displayed. The list of these alignments becomes the new current list. This command is an abbreviation for the full command: SEARCH/FIELD=members.

FORMAT

MEMBER *[entry-code]*

PARAMETERS

entry-code

The *entry-code* parameter on the command line is the code (or part of the code) to be selected. The character string you enter must contain at least 3 characters. If this parameter is omitted, you will be prompted for it with the prompt:

prompts

Members:

DESCRIPTION

The members index is constructed from the Members: line in the ALN alignment database and consists of the PIR entry-codes for the sequences listed in the alignment.

Normally the members search matches anywhere the character string specified occurs in the entry-code. Use the /ANCHOR modifier (listed below) to alter this operation.

MODIFIERS

In the following table, modifiers accepted by the MEMBERS command are grouped according to their function. Multiple modifiers can be used together; a slash (/) precedes each modifier with no spaces before or after. A modifier designated as a default does not need to be typed in to be used.

Table 4–17 MEMBERS Command Modifiers

Alternate Display Format	Modifier Function
/BRIEF	Only the entry codes are displayed
/COUNTS	Display number of entries containing search term

Table 4–17 (Cont.) MEMBERS Command Modifiers

Current List Processing	Modifier Function
/CURRENT	Process only entries on current list
/ADD	Add entries found added to current list
/SUBTRACT	Remove entries from current list
/KEEP	Do not alter current list with command execution

Redirecting Screen Output	Modifier Function
/PRINTER	Direct screen output to printer
/PRINTER=file-spec	Direct screen output to disk file

Search Manipulation	Modifier Function
/ANCHOR	Match character string at beginning of term
/NOANCHOR (default)	Match character string anywhere in term
/ENTRY	Match each character string to separate terms

EXAMPLES

The following example demonstrates how to find out whether a PIR1 myosin entry with code MORTA1 is in an alignment in the ALN database:

```
1  ATLAS> MEM MORTA1
   MORTA1
     ALN:FA0247 Myosin L1 (A1) and L4 (A2) catalytic light chains
     1 members found
```

To display the alignment use the TYPE command:

```
2  ATLAS> TYPE FA0247
```

PRINT

PRINT displays the contents of an external file on the screen.

FORMAT

PRINT *[file-spec]*

PARAMETERS

file-spec

The *file-spec* parameter on the command line specifies the file to be displayed. If this parameter is omitted on the command line, you will be prompted for it with the prompt:

prompts

File:

The command aborts if no file-spec is given.

QUIT

QUIT terminates the execution of the program and returns control to the system.

FORMAT QUIT

REFERENCE

REFERENCE searches the reference number index for all occurrences of a user-specified number or partial number. For each number found, the number, the entry-identifier, and the title for each entry found are displayed. The list of these entries becomes the new current list. This command is an abbreviation for the full command: SEARCH/FIELD=reference/ANCHOR.

FORMAT

REFERENCE *[reference-number]*

PARAMETERS

reference-number

The *reference-number* parameter on the command line is the reference number (or part of the number) to be selected. The character string you enter must contain at least 3 characters. If this parameter is omitted on the command line, you will be prompted for it with the prompt:

prompts

Reference:

DESCRIPTION

Each journal article processed for the PIR databases is assigned a unique reference number.

Normally the reference number search matches only reference numbers that begin with the character string you specified. Use the /NOANCHOR modifier (listed below) to alter this operation.

Displaying the List of Reference Numbers

If no reference number is typed in at the prompt and you press RETURN, an alphabetical listing of all reference numbers and the titles of the entries will be displayed. If the /BRIEF modifier (listed below) is used only the reference numbers will be displayed.

MODIFIERS

In the following table, modifiers accepted by the REFERENCE command are grouped according to their function. Multiple modifiers can be used together; a slash (/) precedes each modifier with no spaces before or after. A modifier designated as a default does not need to be typed in to be used.

Table 4–18 REFERENCE Command Modifiers

Alternate Display Formats	Modifier Function
/BRIEF	Display only index terms found
/COUNTS	Display number of entries containing search term

Table 4–18 (Cont.) REFERENCE Command Modifiers

Current List Processing	Modifier Function
/CURRENT	Process only entries on current list
/ADD	Add entries found to current list
/SUBTRACT	Remove entries found from current list
/KEEP	Do not alter current list with command execution

Redirecting Screen Output	Modifier Function
/PRINTER	Direct screen output to printer
/PRINTER=file-spec	Direct screen output to disk file

Search Manipulation	Modifier Function
/ANCHOR (default)	Match terms beginning with character string
/NOANCHOR	Match character string anywhere in term
/ENTRY	Match each character string to separate terms

EXAMPLES

The command line:

1
```
ATLAS> REFERENCE A008
```

will produce the list of reference numbers that begin with A008 and the entry-codes and titles of the associated PIR entries.

Typing the full number will produce those PIR entries containing that particular reference. For example:

2
```
ATLAS> REFERENCE A00899
A00899
  PIR1:IFBY    beta-fructofuranosidase (EC 3.2.1.26) precursor - yeast
               (Saccharomyces cerevisiae)
  1 reference found
```

REPORT

REPORT displays the values for user-specified arithmetic expressions or field names. One to five expressions or fields may be specified. If more than one is specified, they must be separated by commas. Statistics are calculated for the first expression only. The list of these entries becomes the new current list.

restrictions

This command has been implemented for VAX/VMS and OpenVMS systems only!

FORMAT REPORT

PARAMETERS

When the REPORT command has been given and the RETURN has been pressed, you will be prompted for the following:

prompts

Expression:

Specify the expressions or field names described below.

prompts

Sort in ascending or descending order (A/D/N for no):

If the /CURRENT modifier is used, the display list may be sorted in the order of the specified expressions. Respond with RETURN if no sort is wanted, otherwise respond as directed with either A or D.

DESCRIPTION

The REPORT command tabulates ancillary data such as sequence length, molecular weight, amino acid composition, etc. Valid expressions may be single field names or arithmetic expressions containing numerical field names and numerical constants as arithmetic elements. Arithmetic expressions are strings of arithmetic elements separated by operators or parentheses. Expressions are evaluated according to the following operator precedence.

Table 4–19 Expression Operators

operator	function	precedence
*	multiplication	second
/	division	second
+	addition	third
-	subtraction	third

Operators are evaluated in the order of their precedence. Operators with the same precedence are evaluated from left to right. Subexpressions separated by parentheses are evaluated first irrespective of operator precedence. Numerical constants may be integers or real numbers.

Table 4–20 Expression Fields

Nonnumerical fields	Description
CODE	Entry code
GROUP	Taxonomic group
SUPERFAMILY	Superfamily number
FAMILY	Family number
SUBFAMILY	Subfamily number
ENTRY	Entry number
SUBENTRY	Subentry number
TYPE	Sequence type (P=complete, F=fragment)
TEXT_UPDATE	Date of last text update (YYMMDD)
SEQ_UPDATE	Date of last sequence update (YYMMDD)
ADDED_DATE	Date added to database (YYMMDD)

Numerical fields	Description
MW	Molecular weight
LEN	Sequence length
%Ala	% Alanine
%Arg	% Arginine
%Asn	% Asparagine
%Asp	% Aspartic acid
%Asx	% Asp or Asn
%Cys	% Cysteine
%Glu	% Glutamic acid
%Gln	% Glutamine
%Glx	% Glu or Gln
%Gly	% Glycine
%His	% Histidine
%Ile	% Isoleucine
%Leu	% Leucine
%Lys	% Lysine
%Met	% Methionine
%Phe	% Phenylalanine
%Pro	% Proline
%Ser	% Serine
%Thr	% Threonine
%Trp	% Tryptophan
%Tyr	% Tyrosine
%Val	% Valine
%X	% Unknown

MODIFIERS

In the following table, modifiers accepted by the REPORT command are grouped according to their function. Multiple modifiers can be used together; a slash (/) precedes each modifier with no spaces before or after. A modifier designated as a default does not need to be typed in to be used.

Table 4–21 REPORT Command Modifiers

Current List Processing	Modifier Function
/CURRENT	Process only entries on current list
/ADD	Add entries found to current list
/SUBTRACT	Remove entries found from current list
/KEEP	Do not alter current list with command execution

Redirecting Screen Output	Modifier Function
/PRINTER	Direct screen output to printer
/PRINTER=file-spec	Direct screen output to disk file

EXAMPLES

1

```
ATLAS> find bov thymosin alpha
thymosin alpha-1 - bovine
  PIR1:TNBOA1  thymosin alpha-1 - bovine
  1 title found
ATLAS> REPORT
Expression: (%ASP+%GLU)*LEN/100
      9.00  TNBOA1  thymosin alpha-1 - bovine
```

The above expression gives the number of acidic amino acids in a sequence. For bovine thymosin alpha-1, which has a length of 28, 10.71% Asp, and 21.43% Glu, the expression is equal to 9.0.

SCAN

SCAN is a fast sequence matching routine that locates exactly matching segments. The entries containing the test segment become the new current list.

FORMAT

SCAN *[peptide]*

PARAMETERS

peptide

The *peptide* parameter on the command line is the peptide, in one-letter amino acid notation, to be located. The peptide must be at least three residues but not greater than 30 residues long. If this parameter is omitted on the command line, you will be prompted for it with the prompt:

prompts

Peptide:

DESCRIPTION

The SCAN command searches a tripeptide index for nearly instantaneous matching of identical segments. Even though the string of amino acid residues must match identically, the SCAN command ignores spaces and hyphens (-) in the search string entered by the user.

MODIFIERS

In this summary the modifiers accepted by the *SCAN* command are grouped according to their function.

Table 4–22 SCAN Command Modifiers

Current List Processing	Modifier Function
/CURRENT	Process only entries on current list
/ADD	Add entries found to current list
/SUBTRACT	Remove entries found from current list
/KEEP	Do not alter current list with command execution

Information Category Selection	Modifier Function
/TITLE	Display only titles of entries found
/SEGMENT	Display only segments of entries found

Redirecting Screen Output	Modifier Function
/PRINTER	Direct screen output to printer
/PRINTER=file-spec	Direct screen output to disk file

EXAMPLES

In the following example, the SCAN command was used to search the PIR1 dataset for the string CIDCGA, which forms part of the active site of certain ferredoxins.

```
1   ATLAS> SCAN CIDCGA
      5 matches found
    Database: PIR1
    FECLCE  ferredoxin - Clostridium sp.
             37  DAVRVIDADK CIDCGA CANTCPVDAI
    FECLCU  ferredoxin - Clostridium acidiurici
             37  DSRYVIDADT CIDCGA CAGVCPVDAP
    FECLCT  ferredoxin - Clostridium thermosaccharolyticum
             37  TGKYEVDADT CIDCGA CEAVCPTGAV
    FEME    ferredoxin - Megasphaera elsdenii
             36  GETKYVVTDS CIDCGA CEAVCPTGAI
    FETWT   ferredoxin - Thermus aquaticus
             39  GDQFYIHPEE CIDCGA CVPACPVNAI
```

As indicated here, the peptide string can be entered on the command line; otherwise, a prompt for the string will be issued. The display of the results of the search show the number of matches found, the database searched, and the entry-code and title for each hit. The line underneath the title of each match shows the sequence position of the first matching residue, the ten residues preceding the matching segment, the matching residues, and the ten residues following them. The list of these five entries becomes the new current list, which can then be further manipulated.

SEARCH

SEARCH/FIELD=field-name searches the specified indexed field(s) for all occurrences of a user-specified parameter. The fields are those available under the text searching commands. For each parameter found, the parameter, the entry-identifier, and the title for each entry are displayed. The list of these entries becomes the new current list.

FORMAT SEARCH/FIELD=field-name[+field-name] *[parameter]*

PARAMETERS *parameter*

The *parameter* on the command line is the character string to be selected. Each word in the character string must contain at least 3 characters and must be appropriate for the field(s) to be searched. If this parameter is omitted, you will be prompted for it with a prompt for the field (i.e. Keyword:) or, if multiple fields are to be searched, the following prompt appears:

prompts Term:

DESCRIPTION

The SEARCH/FIELD command is the most flexible means of searching the indexed fields. When the DEFINE command is used to create an abbreviation for SEARCH/FIELD=field-name(s), the user can readily customize the program to meet specialized needs.

Note: The field-name may NOT be abbreviated.

MODIFIERS

In the following table, modifiers accepted by the SEARCH command are grouped according to their function. Multiple modifiers can be used together; a slash (/) precedes each modifier with no spaces before or after. A modifier designated as a default does not need to be typed in to be used.

Table 4–23 SEARCH Command Modifiers

Alternate Display Format	Modifier Function
/BRIEF	Display only index terms found
/COUNTS	Display number of entries containing search term
/SHOW	Display how search string is parsed

Table 4–23 (Cont.) SEARCH Command Modifiers

Current List Processing	Modifier Function
/CURRENT	Process only entries on current list
/ADD	Add entries found to current list
/SUBTRACT	Remove entries found from current list
/KEEP	Do not alter current list with command execution

Redirecting Screen Output	Modifier Function
/PRINTER	Direct screen output to printer
/PRINTER=file-spec	Direct screen output to disk file

Search Manipulation	Modifier Function
/NOANCHOR (default)	Matches character string anywhere in term
/ANCHOR	Matches terms beginning with character string
/ENTRY	Match each character string to separate terms
/MAIN (for /FIELD=title)	Search titles only NOT alternate names or contains fields
/TEXT (for /FIELD=title)	Perform string search on titles (does not use indexes and is slow for large numbers of entries)

EXAMPLES

The following example demonstrates the SEARCH command:

```
1   ATLAS> B PIR1
    ATLAS> SEARCH/FIELD=KEYWORD/ENTRY KINASE RECEPT
    PIR1:KIHUCA  protein kinase C (EC 2.7.1.-) alpha - human
    PIR1:KIRTC   protein kinase C (EC 2.7.1.-) alpha - rat
    PIR1:KIMSCA  protein kinase C (EC 2.7.1.-) alpha - mouse
    .
    .
    .
      47 entries found
```

In this example, the /ENTRY modifier was used to search the index for two separate terms: one containing KINASE and the other containing RECEPT. Without this modifier, the program assumes both character strings are part of the same search term. The entries found in example 1 had a keyword containing KINASE and a keyword containing RECEPT. The following example shows the results of the same search without the /ENTRY modifier:

```
2   ATLAS> SEARCH/FIELD=KEYWORD KINASE RECEPT
    Keyword: KINASE RECEPT
      No keywords found
```

SELECT

SELECT searches for entries on the basis of a user-specified arithmetic expressions or field names. One to five expressions or fields may be specified. If more than one is specified, they must be separated by commas. Statistics are calculated for the first expression only. The list of these entries becomes the new current list.

restrictions

This command has been implemented for VAX/VMS and OpenVMS sytems only!

FORMAT SELECT

PARAMETERS

When the SELECT command has been given and the RETURN has been pressed, you will be prompted for the following:

prompts

Expression:

Specify the expression to be searched for. Valid expressions may be arithmetic expressions containing numerical field names and numerical constants or single nonnumerical field names. Respond with a ? for more details.

prompts

Range:

A range or a single number may be specified. Valid range specifications are of the form R1-R2. The terms R1 and R2 are the lower and upper bounds of the inclusive search range. These terms may be character strings if the expression contains only a single nonnumerical field specification. R1-* specifies a range greater than or equal R1. *-R2 specifies a range less than or equal to R2. If a single number is specified, values equal to this number are searched for.

DESCRIPTION

Table 4–24 Expression Operators

operator	function	precedence
*	multiplication	second
/	division	second
+	addition	third
-	subtraction	third

Operators are evaluated in the order of their precedence. Operators with the same precedence are evaluated from left to right. Subexpressions

separated by parentheses are evaluated first irrespective of operator precedence. Numerical constants may be integers or real numbers.

Table 4–25 Expression Fields

Nonnumerical fields	Description
CODE	Entry code
GROUP	Taxonomic group
SUPERFAMILY	Superfamily number
FAMILY	Family number
SUBFAMILY	Subfamily number
ENTRY	Entry number
SUBENTRY	Subentry number
TYPE	Sequence type (P=complete, F=fragment)
TEXT_UPDATE	Date of last text update (YYMMDD)
SEQ_UPDATE	Date of last sequence update (YYMMDD)
ADDED_DATE	Date added to database (YYMMDD)

Numerical fields	Description
MW	Molecular weight
LEN	Sequence length
%Ala	% Alanine
%Arg	% Arginine
%Asn	% Asparagine
%Asp	% Aspartic acid
%Asx	% Asp or Asn
%Cys	% Cysteine
%Glu	% Glutamic acid
%Gln	% Glutamine
%Glx	% Glu or Gln
%Gly	% Glycine
%His	% Histidine
%Ile	% Isoleucine
%Leu	% Leucine
%Lys	% Lysine
%Met	% Methionine
%Phe	% Phenylalanine
%Pro	% Proline
%Ser	% Serine
%Thr	% Threonine
%Trp	% Tryptophan
%Tyr	% Tyrosine
%Val	% Valine
%X	% Unknown

MODIFIERS

In the following table, modifiers accepted by the SELECT command are grouped according to their function. Multiple modifiers can be used together; a slash (/) precedes each modifier with no spaces before or after. A modifier designated as a default does not need to be typed in to be used.

Table 4–26 SELECT Command Modifiers

Current List Processing	Modifier Function
/CURRENT	Process only entries on current list
/ADD	Add entries found to current list
/SUBTRACT	Remove entries found from current list
/KEEP	Do not alter current list with command execution
/EQUALS=entry-identifier	Searches for values equal to or within a range of the value of a user-specified sequence
/BRIEF	Suppresses warning messages that occur when an expression cannot be evaluated (e.g., when a division by zero is attempted)

Redirecting Screen Output	Modifier Function
/PRINTER	Direct screen output to printer
/PRINTER=file-spec	Direct screen output to disk file

EXAMPLES

1
```
ATLAS> SELECT
Expression: MW
Range: 95000-98000
```

The above example will produce a list of proteins with a molecular weight between 95 and 98 kD.

2
```
ATLAS> SELECT/EQUALS=kdhop
Expression:superfamily
```

This example will produce the list of entries in the same superfamily as the sequence with code KDHOP

SET

SET is used to change program operation parameters.

SYNTAX SET/MODIFIER

DESCRIPTION

The SET command changes the default parameters of the program. With the /CLOCK modifier, SET is used to set the command execution timer. The timer displays the CPU time and elapsed time for the execution of each command. The timer is initially off.

The /MENU modifier, for the PC version, will change from command mode to the menu system. The /SWITCH modifier allows the user to change the modifier character ' / ' to another specified character. This is particularly useful for UNIX users when the slash character interferes with slashes used in pathnames.

MODIFIERS

The following table summmarizes the modifiers accepted by the SET command.

Table 4–27 SET Command Modifiers

Modifier	Function
/CLOCK	Set internal timer
/MENU	Change from command mode to menu mode (PC version only)
/NOWRAP	Turns line wrap off. For subsequent text searching commands which use /PRINTER=filename to create an output file, the lines written are not broken up into multiple lines so that they fit into 80 columns. They are written on one line which may be up to 512 characters long. The action remains into effect until the SET/WRAP command is issued.
/SWITCH=switch character	Change modifier character from ' / '
/WRAP	Turns line wrap on. For subsequent text searching commands which use /PRINTER=filename to create an output file, the lines writtenn are broken into multiple lines so that they fit into 80 columns. The action remains into effect until the SET/NOWRAP command is issued.

SHOW

The SHOW command displays information about the program operation.

FORMAT SHOW

DESCRIPTION

Unmodified, or with the default /FIELDS modifier, the SHOW command displays the indexed fields currently available for the databases in the term index system. SHOW with the /COMMANDS modifier will also display the definitions for the commands to search these indexes. The /DISPLAYS modifier will list all the symbols that have been defined and the display layouts they represent. The /DATABASE and /INFORMATION modifiers will display the header information for the active databases and the Atlas program respectively. To display the date and time, use the /TIME modifier. The /TOTALS modifier will display the composition table generated by the previous execution of the TYPE command. See the discussion under the TYPE command.

MODIFIERS

In the following table, modifiers accepted by the SHOW command are grouped according to their function.

Table 4–28 SHOW Command Modifiers

Redirecting Screen Output	Modifier Function
/PRINTER	Direct screen output to printer
/PRINTER=file-spec	Direct screen output to disk file

Miscellaneous	Modifier Function
/COMMANDS	Display command definitions for each field
/DATABASE	Display headers for each active database
/DISPLAYS	List all the symbols defined for display layouts
/FIELDS (default)	Display indexed fields for each database
/INFORMATION	Display Atlas program header
/TIME	Display current date and time
/TOTALS	Display the composition table generated by the TYPE command

EXAMPLES

The following example illustrates the show command:

```
1
```

```
ATLAS> SHOW/COMMANDS
GENE          SEARCH/FIELD=GENE_NAME
CROSS         SEARCH/FIELD=CROSS_REF
MEMBERS       SEARCH/FIELD=MEMBERS
REFERENCE     SEARCH/FIELD=REFERENCE/ANCHOR
ACCESSION     SEARCH/FIELD=ACCESSION/ANCHOR
SPECIES       SEARCH/FIELD=SPECIES
SFNUM         SEARCH/FIELD=SUPFAM_NUM
SUPERFAMILY   SEARCH/FIELD=SUPERFAMILY
JOURNAL       SEARCH/FIELD=JOURNAL
FEATURE       SEARCH/FIELD=FEATURE
AUTHOR        SEARCH/FIELD=AUTHOR/ANCHOR
KEYWORD       SEARCH/FIELD=KEYWORD
FIND          SEARCH/FIELD=TITLE

KT  SEARCH/FIELD=KEYWORD+TITLE
```

The command definition for each of the text searching commands is listed; user-defined commands appear at the bottom of the list. In this case, KT was defined in example 2 of the DEFINE command.

SPECIES

SPECIES searches the species index for all occurrences of a user-specified species name or partial name. For each name found, the name, the entry-identifier, and the title for each entry in which the species name occurs are displayed. The list of these entries becomes the new current list. This command is an abbreviation for the full command: SEARCH/FIELD=species.

FORMAT

SPECIES *[species-name]*

PARAMETERS

name

The *species name* parameter on the command line is the name (or part of the name) to be selected. The character string you enter must contain at least 3 characters. If this parameter is omitted on the command line, you will be prompted for it with the prompt:

prompts

Species:

DESCRIPTION

Normally the species name search matches a name whenever the character string you specified occurs anywhere within the name. Use the /ANCHOR modifier to cause the search to match species names only when the specified character string occurs at the beginning of the name.

Displaying the List of Species

If no *species name* is typed in at the prompt and you press RETURN, an alphabetical listing of all species names and the titles of their associated entries will be displayed. If the /BRIEF modifier (listed below) is used only the names will be displayed.

MODIFIERS

In the following table, modifiers accepted by the SPECIES command are grouped according to their function. Multiple modifiers can be used together; a slash (/) precedes each modifier with no spaces before or after. A modifier designated as a default does not need to be typed in to be used.

Table 4–29 SPECIES Command Modifiers

Alternate Display Formats	Modifier Function
/BRIEF	Display only index terms found
/COUNTS	Display number of entries containing search term
/SHOW	Display how search string is parsed

Table 4–29 (Cont.) SPECIES Command Modifiers

Database List Processing	Modifier
/PIR	Ignore all databases except PIR1, PIR2, and PIR3

Current List Processing	Modifier Function
/CURRENT	Process only entries on current list
/ADD	Add entries found to current list
/SUBTRACT	Remove entries found from current list
/KEEP	Do not alter current list with command execution

Redirecting Screen Output	Modifier Function
/PRINTER	Direct screen output to printer
/PRINTER=file-spec	Direct screen output to disk file

Search Manipulation	Modifier Function
/ANCHOR	Match terms beginning with character string
/NOANCHOR (default)	Match character string anywhere in term
/ENTRY	Match each character string to separate terms

EXAMPLES

The following example demonstrates the SPECIES command:

1

```
ATLAS> BASES *
ATLAS> SPECIES ALCALIG DENIT
Warning: not all the requested indexes exist.
Alcaligenes denitrificans
  PIR1:AZALCD  azurin precursor - Alcaligenes denitrificans
  PIR2:PH1228  D-aminoacylase (EC 3.5.1.-) - Alcaligenes denitrificans
               (fragment)
  NRL_3D:1AZBA   azurin (copper removed), chain A - Alcaligenes denitrificans
  NRL_3D:1AZBB   azurin (copper removed), chain B - Alcaligenes denitrificans
  NRL_3D:1AZCA   azurin (apo), chain A - Alcaligenes denitrificans
  NRL_3D:1AZCB   azurin (apo), chain B - Alcaligenes denitrificans
  NRL_3D:1AIZA   azurin (with cadmium), chain A - Alcaligenes denitrificans
  NRL_3D:1AIZB   azurin (with cadmium), chain B - Alcaligenes denitrificans
  NRL_3D:2AZAA   azurin (oxidized), chain A - Alcaligenes denitrificans
  NRL_3D:2AZAB   azurin (oxidized), chain B - Alcaligenes denitrificans
Alcaligenes denitrificans subsp. xylosoxydans
  PIR2:A61148  cyanidase - Alcaligenes denitrificans subsp. xylosoxydans (strain)
               DF3 (fragment)
  PIR3:S17501  glutaminase - Alcaligenes denitrificans subsp. xylosoxydans
  2 species found
```

In this example, all the databases were activated by the command BASES *. When the SPECIES command was issued, a message appeared warning that not all the active databases are included in this index.

SUPERFAMILY

SUPERFAMILY searches the index of superfamily names for all occurrences of a user-specified name or partial name. For each name found, the superfamily, the database, the entry identifier, and the title for each entry belonging to the superfamily are displayed. The list of these entries becomes the new current list. This command is an abbreviation for the full command: SEARCH/FIELD=superfamily.

FORMAT

SUPERFAMILY *[superfamily-name]*

PARAMETERS

superfamily-name

The *superfamily-name* parameter on the command line is the name (or part of the name) of the superfamily to be selected. Each word you enter must contain at least 3 characters. If this parameter is omitted on the command line, you will be prompted for it with the prompt:

prompts

Superfamily:

Note: Classification of sequences into superfamilies is part of the annotation and verification process; therefore, not all sequences have been assigned to superfamilies.

DESCRIPTION

Superfamily Definition

A superfamily is a group of protein domains that share sequence similarity due to common ancestry. Although the protein domain often extends the entire length of protein, those proteins with more than one domain may belong to more than one superfamily.

Displaying the List of Superfamilies

If no *superfamily-name* is typed in at the prompt and you press RETURN, an alphabetical listing of all superfamilies and the titles of their associated entries will be displayed. If the /BRIEF modifier (listed below) is used only the superfamily names will be displayed.

MODIFIERS

In the following table, modifiers accepted by the SUPERFAMILY command are grouped according to their function. Multiple modifiers can be used together; a slash (/) precedes each modifier with no spaces before or after. A modifier designated as a default does not need to be typed in to be used.

Table 4–30 SUPERFAMILY Command Modifiers

Alternate Display Formats	Modifier Function
/BRIEF	Display only index terms found
/COUNTS	Display number of entries containing search term
/SHOW	Display how search string is parsed

Current List Processing	Modifier Function
/CURRENT	Process only entries on current list
/ADD	Add entries found to current list
/SUBTRACT	Remove entries found from current list
/KEEP	Do not alter current list with command execution

Redirecting Screen Output	Modifier Function
/PRINTER	Direct screen output to printer
/PRINTER=file-spec	Direct screen output to disk file

Search Manipulation	Modifier Function
/ANCHOR	Match terms beginning with character string
/NOANCHOR (default)	Match character string anywhere in term
/ENTRY	Match each character string to separate terms

EXAMPLES

The following command line demonstrates the SUPERFAMILY command:

```
1   ATLAS> SUPERFAMILY ENKEPHALIN
    proenkephalin
      PIR1:EQHUA   enkephalin precursor - human
      PIR1:EQBOA   enkephalin precursor - bovine
      PIR1:EQRTA   enkephalin A precursor - rat
      PIR1:EQXL    enkephalin precursor - African clawed frog
      PIR1:DFHU    beta-neoendorphin / dynorphin precursor - human
      PIR1:DFPG    beta-neoendorphin / dynorphin precursor - pig
      PIR1:DFRTP   beta-neoendorphin / dynorphin precursor - rat (fragment)
      PIR2:A18864  enkephalin-containing peptide E - guinea pig (fragment)
      PIR2:JL0067  Met-enkephalin-Arg-Phe - pig
      PIR2:JT0381  enkephalin precursor - pig (fragment)
      PIR2:A19448  enkephalin-containing protein - bovine
      PIR2:B35678  enkephalin precursor - mouse
      PIR2:A48541  proenkephalin - rat
      PIR2:A47589  enkephalin precursor - mouse (fragment)
      PIR2:A19145  dynorphin - pig
      PIR2:A41395  dynorphin precursor - rat
      PIR2:A60410  alpha-Neo-endorphin - guinea pig
      1 superfamily found
```

The user must be aware that the entries found in a particular superfamily may NOT be all the sequences of that type in the database; only some of the entries in PIR2 and few of the entries in PIR3 have been categorized by superfamily.

SFNUM

SFNUM searches the index of superfamily numbers contained in the .CDX database file for all occurrences of a user-specified number or partial number. For each number found, the number, database, entry identifier, and title for each entry belonging to the superfamily are displayed. The list of these entries becomes the new current list. This command is an abbreviation for the full command: SEARCH/FIELD=supfam_num.

FORMAT

SFNUM *[superfamily-number]*

PARAMETERS

superfamily-number

The *superfamily-number* parameter on the command line is the number which groups a set of entries into a superfamily. Each number entered must contain at least 3 characters; if the number contains less than three characters, precede the search string with "-" (dashes). If this parameter is omitted on the command line, you will be prompted for it with the prompt:

prompts

Supfam_num:

Note: Classification of sequences into superfamilies is part of the annotation and verification process; therefore, not all sequences have been assigned to superfamilies.

DESCRIPTION

Superfamily number Definition

A superfamily number denotes a classification described by five numbers although the first is used most commonly. Each number denotes the level of sequence similarity. The first number, referred to as the *superfamily number*, groups related entries. The second number, referred to as the *family number*, groups entries of a superfamily into families whose sequences are less that 50% different. The third number, referred to as the *subfamily number*, groups entries of a family into subfamilies whose sequences are less than 20% different. The fourth number, referred to as the *entry number*, groups entries of a subfamily into entries whose sequences are less than 10% different. The fifth number allows for unique subentry classification.

Displaying the List of Superfamily numbers

If no *superfamily-number* is typed in at the prompt and you press RETURN, a numerical listing of all superfamily numbers and associated entries will be displayed. If the /BRIEF modifier (listed below) is used only the superfamily numbers will be displayed.

MODIFIERS

In the following table, modifiers accepted by the SFNUM command are grouped according to their function. Multiple modifiers can be used together; a slash (/) precedes each modifier with no spaces before or after. A modifier designated as a default does not need to be typed in to be used.

Table 4–31 SFNUM Command Modifiers

Alternate Display Formats	Modifier Function
/BRIEF	Display only index terms found
/COUNTS	Display number of entries containing search term
/SHOW	Display how search string is parsed

Current List Processing	Modifier Function
/CURRENT	Process only entries on current list
/ADD	Add entries found to current list
/SUBTRACT	Remove entries found from current list
/KEEP	Do not alter current list with command execution

Redirecting Screen Output	Modifier Function
/PRINTER	Direct screen output to printer
/PRINTER=file-spec	Direct screen output to disk file

Search Manipulation	Modifier Function
/ANCHOR	Match terms beginning with character string
/NOANCHOR (default)	Match character string anywhere in term
/ENTRY	Match each character string to separate terms

EXAMPLES

The following command line demonstrates the SFNUM command:

1
```
ATLAS> SFNUM -59.0
--59.0 1.0 0.0 0.0 0.0
  PIR2:A54756  isocitrate dehydrogenase (NADP+) (EC 1.1.1.42), cytosolic - rat
  PIR2:S33859  isocitrate dehydrogenase (NADP+) (EC 1.1.1.42) - bovine
  PIR2:S51419  hypothetical protein L9470.12 - yeast (Saccharomyces cerevisiae)
--59.0 1.0 1.0 1.0 1.0
  PIR1:DCBYIS  isocitrate dehydrogenase (NADP+) (EC 1.1.1.42) precursor,
               mitochondrial - yeast (Saccharomyces cerevisiae)
--59.0 1.0 2.0 1.0 1.0
  PIR2:A43294  isocitrate dehydrogenase (NADP+) (EC 1.1.1.42), mitochondrial -
               pig
  3 supfam_nums found
```

The user must be aware that the entries found in a particular superfamily number may NOT be all the sequences of that type in the database; only some of the entries in PIR2 and none of the entries in PIR3 have been categorized by superfamily number.

TAXONOMY

The TAXONOMY command searches for entries on the basis of taxonomic classification. The list of entries found becomes the new current list.

restrictions

This command has been implemented for VAX/VMS and OpenVMS sytems only!

FORMAT

TAXONOMY *taxonomic-class*

PARAMETERS

taxonomic-class

The *taxonomic-class* parameter specifies the desired taxonomic classification. If you enter ?, the list of valid taxonomic classes and the classification scheme will appear.

DESCRIPTION

The TAXONOMY command searches for entries on the basis of taxonomic classification. The list of entries found becomes the new current list.

MODIFIERS

In the following table, modifiers accepted by the SELECT command are grouped according to their function. Multiple modifiers can be used together; a slash (/) precedes each modifier with no spaces before or after. A modifier designated as a default does not need to be typed in to be used.

Table 4–32 SELECT Command Modifiers

Current List Processing	Modifier Function
/CURRENT	Process only entries on current list
/ADD	Add entries found to current list
/SUBTRACT	Remove entries found from current list
/KEEP	Do not alter current list with command execution
/EQUALS=entry-identifier	Selects sequences with the same taxonomic classification as the specified sequence
/MAIN	Searches only the main species of the entry; that is, the species of the sequence shown in the entry. Normally the TAXONOMY command searches the main species plus the species of other sequences mentioned in the text but not explicitly shown in the entry

Table 4–32 (Cont.) SELECT Command Modifiers

Redirecting Screen Output	Modifier Function
/PRINTER	Direct screen output to printer
/PRINTER=file-spec	Direct screen output to disk file

TAXONOMIC CLASSES

Tables 4–33, 4–34, and 4–35 below list the valid taxonomic class codes.

Table 4–33 Taxonomic Class Codes (Viruses)

Code	Taxonomic classification
VIRU	Viruses
BP	. Bacterial viruses
VP	. Plant viruses
VA	. Animal viruses

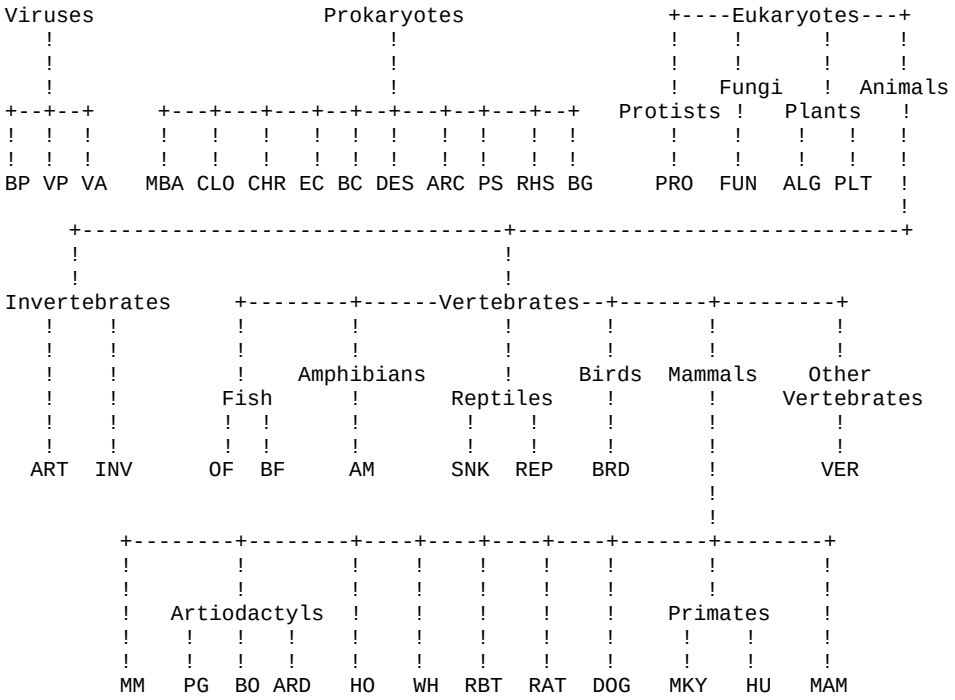
Table 4–34 Taxonomic Class Codes (Prokaryotes)

Code	Taxonomic classification
PROK	Prokaryotes
MBA	. Miscellaneous bacteria: Propionibacterium; Streptococcus; Thermus; theromophilic bacteria; Staphylococcus; Acinetobacter; Myxobacter
CLO	. Clostridia; Megasphaera; Peptococcus (nonphotosynthetic anaerobes)
CHR	. Chromatium; Chlorobium; other photosynthetic anaerobes
EC	. Escherichia coli; other Enterobacteriaceae; Vibrionaceae
BC	. Bacillus; Lactobacillus
DES	. Desulfovibrio; Desulfuromonas
ARC	. Thermoplasma; Halobacterium; Streptomyces; Micrococcus
PS	. Pseudomonas; Alcaligenes; Bordetella; Azotobacter; Mycobacterium
RHS	. Rhodospirillaceae (except Rhodospirillum tenue and Rhodopseudomonas gelatinosa); Paracoccus; Agrobacterium
BG	. Blue-green algae

Table 4–35 Taxonomic Class Codes (Eukaryotes)

Code	Taxonomic classification
EUKA	Eukaryotes
PRO	. Protists, including slime molds
FUN	. Fungi
PLAN	. Plants
ALG	. . Eukaryotic algae
PLT	. . Higher plants
ANIM	. Animals
INVE	. . Invertebrates
ART	. . . Arthropods
INV	. . . Other invertebrates
VERT	. . Vertebrates
FISH	. . . Fish
OF	. . . . Jawless fish; cartilaginous fish
BF	. . . . Bony fishes
AM	. . . Amphibians
REPT	. . . Reptiles
SNK	. . . . Squamata (snakes, etc.)
REP	. . . . Other reptiles
BRD	. . . Birds
MAMM	. . . Mammals
MM	. . . . Monotremes; marsupials
ARTI	. . . . Artiodactyls
PG	. . . . . Artiodactyls; pigs; hippopotamus
BO	. . . . . Artiodactyls; Bovidae (cattle, etc.)
ARD	. . . . . Other artiodactyls
HO	. . . . Perissodactyls
WH	. . . . Cetaceae
RBT	. . . . Lagomorphs
RAT	. . . . Rodents
DOG	. . . . Carnivores; pinnipeds
PRIM	. . . . Primates
MKY	. . . . . Monkeys
HU	. . . . . Human; great apes
MAM	. . . . Other mammals, including primitive primates
VER	. . . Other vertebrates

Figure 4–1 Taxonomic Classification Scheme



TYPE

The TYPE command displays all, or selected categories, of the information contained in an entry. This is the basic command for displaying a sequence and/or the annotation associated with the sequence.

FORMAT

TYPE *[entry-identifier]*

PARAMETERS

entry-identifier

If the *entry-identifier* parameter is omitted from the command line, you will be prompted for it with the prompt:

prompts

Code:

If you press RETURN at the code prompt, the program processes the entries on the current list. If there are no entries on the current list, pressing the return key terminates the TYPE command.

DESCRIPTION

The display of an entry can be separated into four components:

1. header - entry code and entry title
2. text - annotation associated with the entry
3. composition - composition of the sequence
4. sequence - sequence

Displaying a Complete Entry

Unmodified, the TYPE command displays all of the information in an entry in the order listed above. The /OLD modifier also displays all of the information, but in the order: header, composition, sequence, and text.

Displaying Selected Components

The header component of an entry is always displayed. Any combination of the three remaining components can be selected for display with the modifiers: /TEXT, /COMPOSITION, and /SEQUENCE. (See Appendix D for a description of the entry format).

Displaying the Sequence with Three-letter Amino Acid Abbreviations

An alternate display format is available for the sequence portion of a protein entry, which may be displayed with the one-letter (default) or three-letter amino acid abbreviations.

Displaying Selected Text Lines

Selected lines of text can be displayed, but first a display layout must be defined with the DEFINE command (see the description of the DEFINE command). The SHOW/DISPLAY command shows all the symbols that have been defined and the display layouts they represent.

Computing Total Sequence Composition

Each time the TYPE command is executed with the composition component indicated either implicitly, no information category modifier, or explicitly, using the /COMPOSITION modifier, the total composition of all the protein sequences displayed is computed and stored. The SHOW/TOTALS command shows this stored total composition table. The SHOW/TOTALS command can be executed at any time after the total composition table is generated by the TYPE command but before the TYPE command is executed again because it will generate a new total composition table.

Printing Entries in PostScript Format (VMS and OpendVMS only)

Protein entries can be printed on a PostScript printer using the /POSTSCRIPT modifier (VMS and OpenVMS only). To use this modifier, define a symbol called POSTSCRIPT_QUEUE to be xxx. The files will be sent to queue xxx.

The command creates a file in your home directory called PROTEIN.PS for the entry specified (or for each entry on the current list when /CURRENT is also used). The command then sends the file to the defined queue and then deletes it. If you are using /CURRENT and there are 500 entries on the current list, 500 files will be created and printed, each file will have the same name but a different version number.

The entries are sent to the printer immediately; you do not need to terminate the program to have the entries printed.

The entry must be a protein sequence entry and the entry must be in the NBRF format. The /POSTSCRIPT modifier will work with these other modifiers: /CURRENT, /SEQUENCE, /TEXT (but not /TEXT=display_ layout), /COMPOSITION, /OLD, and /THREE_LETTER.

MODIFIERS

In the following table, modifiers accepted by the TYPE command are grouped according to their function. Multiple modifiers can be used together; a slash (/) precedes each modifier with no spaces before or after. A modifier designated as a default does not need to be typed in to be used.

Table 4–36 TYPE Command Modifiers

Alternate Display Formats	Modifier Function
/EXCHANGE	Display entry in CODATA format
/FULL_NUMBERING	Display numbering for each line of sequence
/ONE_LETTER (default)	Display one-letter amino acid abbreviations
/THREE_LETTER	Display three-letter amino acid abbreviations
/OLD	Display the entry in this order: header, composition, sequence, and text

Table 4–36 (Cont.) TYPE Command Modifiers

Current List Processing	Modifier Function
/CURRENT[1]	Process only entries on current list
Information Category Selection	**Modifier Function**
/COMPOSITION	Display the composition of the sequence
/COMPOSITION=QUIET	Compute the composition but suppress its display
/SEQUENCE	Display sequence portion of entry
/TEXT	Display text portion of entry
/TEXT=display-layout	Display only selected text lines of entry
Redirecting Screen Output	**Modifier Function**
/PRINTER	Direct screen output to printer
/PRINTER=file-spec	Direct screen output to disk file
/POSTSCRIPT	Create-print-delete a PostScript output file for a protein entry

[1]When /CURRENT is specified, the *entry-identifier* parameter on the command line should be omitted. If present, it is ignored

EXAMPLES

The following example demonstrates the use of the TYPE command:

```
1   ATLAS> TYPE/THREE_LETTER FEME
    PIR1:FEME
    ferredoxin 2[4Fe-4S] - Megasphaera elsdenii

    Species: Megasphaera elsdenii

    Date: #sequence_revision 13-Jul-1981 #text_change 03-Mar-1995

    Accession: A00203

    Azari, P.; Glantz, M.; Tsunoda, J.; Yasunobu, K.T.
        unpublished results, cited by Yasunobu, K.T., and Tanaka, M., Syst.
          Zool. 22, 570-589, 1973
        Reference number: A00203
        Accession: A00203
        Molecule type: protein
        Residues: 1-54 <AZA>

    Superfamily: ferredoxin 2[4Fe-4S]; ferredoxin 2[4Fe-4S] homology

    Keywords: 4Fe-4S; electron transfer; iron-sulfur protein

    Residues            Feature
    1-54                Domain: ferredoxin 2[4Fe-4S] homology <FER>
    8,11,14,46          Binding site: 4Fe-4S cluster (Cys) (covalent)
                          #status predicted
    18,36,39,42         Binding site: 4Fe-4S cluster (Cys) (covalent)
                          #status predicted

                                Composition
              7 Ala  A      0 Gln  Q      0 Leu  L      4 Ser  S
              0 Arg  R      6 Glu  E      2 Lys  K      5 Thr  T
              0 Asn  N      5 Gly  G      1 Met  M      0 Trp  W
              3 Asp  D      1 His  H      0 Phe  F      1 Tyr  Y
              8 Cys  C      4 Ile  I      2 Pro  P      5 Val  V
         Mol. wt. unmod. chain = 5,430        Number of residues = 54
```

```
                      5                  10                  15
  1 Met His Val Ile Ser Asp Glu Cys Val Lys Cys Gly Ala Cys Ala
 16 Ser Thr Cys Pro Thr Gly Ala Ile Glu Glu Gly Glu Thr Lys Tyr
 31 Val Val Thr Asp Ser Cys Ile Asp Cys Gly Ala Cys Glu Ala Val
 46 Cys Pro Thr Gly Ala Ile Ser Ala Glu
```

The TYPE command was used to display the *Megasphaera elsdenii* ferredoxin entry (identification code FEME) in the three-letter amino acid code.

```
2   ATLAS> TYPE/CURRENT/COMPOSITION=QUIET
    PIR1:LUHU
    annexin I - human

    PIR1:LUHU36
    annexin II - human

    PIR1:LUHU3
    annexin III - human
```

```
3   ATLAS> SHOW/TOTALS
               Composition: 3 protein sequences, 1008 residues

            78 Ala  A     37 Gln  Q    103 Leu  L     69 Ser  S
            59 Arg  R     70 Glu  E     90 Lys  K     60 Thr  T
            30 Asn  N     61 Gly  G     24 Met  M      4 Trp  W
            86 Asp  D     16 His  H     29 Phe  F     42 Tyr  Y
            11 Cys  C     68 Ile  I     21 Pro  P     50 Val  V
```

The TYPE command was used to generate the stored total composition table for the three sequences on the current list. The /COMPOSITION=QUIET modifier was used to produce the minimum display for the TYPE command. After the table was generated by the TYPE command, the SHOW/TOTALS command was used to display it.

Part III The FASTA Database Searching Program

This part of the manual contains a description of the FASTA program.

As a convenience to the end-user, version 1.6c2 of the FASTA program package was supplied to NBRF for inclusion on the ATLAS CD-ROM by:

William R. Pearson
Department of Biochemistry
Box 440, Jordan Medical Education Building
University of Virginia
Charlottesville, VA 22908

5 Overview of the FASTA Database Searching Program

This summary of the FASTA program was derived from the following references which offer a complete description of the FASTA program:

W.R. Pearson (1990) "Rapid and Sensitive Sequence Comparison with FASTP and FASTA", Methods in Enzymology 193:63-98

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

D.J. Lipman and W.R. Pearson (1985) "Rapid and Sensitive Protein Similarity Searches", Science 22:1435-1441

FASTA.DOC Release 1.6. The original document supplied with the FASTA program package and provided on this CD-ROM.

5.1 Introduction

FASTA is a universal sequence comparison program that defaults to protein comparisons unless the query sequence is > 85% A+C+G+T, whereupon a DNA sequence is assumed. It compares the query sequence in the first file with all the sequences (there need only be one) in the second or library file, reporting the one best similarity score and alignment for each pairwise comparison. The score is not affected by poorly aligned portions of the sequence outside the best region.

Terms to know

DIAGONAL - The term diagonal refers to the diagonal line that is seen on a dot matrix plot when a sequence is compared with itself and denotes an alignment between two sequences without gaps.

ktup - The ktup parameter determines how many consecutive identities in a match. For proteins a ktup of 1 or 2 is used; ktup=1 is more sensitive but is about 3 times slower than ktup=2. DNA sequences can be searched with a ktup value of 1 to 6.

PAM250 MATRIX - The PAM250 matrix was derived from the analysis of the amino acid replacements occurring among related proteins, and it specifies a range of positive scores for replacements that commonly occur among related proteins and negative scores for unlikely replacements.

5.2 Steps FASTA Uses in a Comparison

FASTA uses four steps in calculating similarity between two sequences; the results are given with these three scores:

- *initn* - The initial score, which may include several similar regions, is used to rank the sequences.
- *init1* - The initial score from the best initial region.

- *opt* - The optimized score allowing gaps in a band 32 residues wide.

Step 1. Find the ten best local regions of identities

In the first step, the 10 best diagonal regions are found using a simple formula based on the number of ktup matches and the distance between the matches without considering shorter runs of identities, conservative replacements, insertions, or deletions. These alignments are scored by incrementing for every matching pair and decrementing for every mismatching pair. The ktup parameter determines how many consecutive identities are required in a match. FASTA saves the ten best local regions, regardless of whether they are on the same or different diagonals.

Step 2. Rescore the ten regions with a scoring matrix

After the ten best local regions are found in the first step, they are rescored using a scoring matrix that allows runs of identities shorter than ktup residues and conservative replacements to contribute to the similarity score. The ends of the region are trimmed to include only those residues contributing to the highest score. Each region is a partial alignment without gaps. For each of the best diagonal regions rescanned with the scoring matrix, a subregion with the maximal score is identified. The highest-scoring subregion is reported as the *init1* score.

Step 3. Join initial regions to form an approximate alignment

The third "joining" step increases the sensitivity of the search method because it allows for insertions and deletions as well as conservative replacements. The modification does, however, decrease selectivity. The degradation of selectivity is limited by including in the optimization step only those initial regions whose scores are above a threshold.

The program checks to see whether several initial regions can be joined together in a single alignment to increase the initial score. Given the locations of the initial regions, their respective scores, and a joining penalty (usually 20) that is analogous to a gap penalty, FASTA calculates an optimal alignment of initial regions as a combination of compatible regions with a maximal score. FASTA uses the resulting score, referred to as the *initn* score, to rank the library sequences.

Step 4. Construct an optimal alignment of the best initial region

In this final step of the comparison, an optimal alignment of the query sequence and the library sequence is constructed, considering only those residues that lie in a band 32 residues wide centered on the best initial region found in Step 2. The optimization employs the same scoring matrix used in determining the initial regions. FASTA reports this score as the optimized *opt* score.

Because FASTA calculates an initial similarity score based on an optimization of initial regions during the library search, the initial score is close to the optimized score for many sequences.

6 Running the FASTA Program

The program requires the name of the query sequence file, a library file, and the *ktup* parameter. The program can accept arguments on the command line or it will prompt for the file names and *ktup* value.

6.1 FASTA Options

To start the program, the command line has the form:

```
FASTA options
```

where only the program name FASTA is required. It is possible to specify several options on the command line preceded by a dash; the following options are available:

```
-a        on output, show the complete sequence instead of
          just the overlap of the two aligned sequences.

-b #      number of sequence scores to be shown on output

-c #      threshold to be used for optimization in a band
          around the best initial region.  Normally this OPTCUT
          value is calculated from the length of the sequence and
          the ktup value (for a 200 residue sequence, it is about
          28). Set "-c 1" and "-o" to optimize every sequence in
          a database. This is the most sensitive option, but
          slows the program down about 5-fold.

-d #      number of alignments to be reported by default. (Used
          in conjunction with -Q).

-f        identical match score from scoring matrix in the scan
          for initial regions. (default for protein) (PAMFACT=1)

-g #      Threshold for joining init1 segments to build an initn
          score (GAPCUT).

-k        use constant score in scan for initial regions (like
          old fastp, fastn, default for DNA) (PAMFACT=0)

-l file   location of library menu file (FASTLIBS)

-m #      MARKX = # (0, 1, 2) highlight differences between two
          aligned sequences in one of three ways:
          -m 0 (default)          -m 1               -m 2

            MWRTCGPPYT         MWRTCGPPYT          MWRTCGPPYT
            ::..:: :::           xx  X             ..KS..Y...
            MWKSCGYPYT         MWKSCGYPYT

          : identities      x conservative      only differences
          . conservative      replacements      are shown
            replacements    X nonconservative
                              replacements

-n        Force the query sequence to be treated as a DNA
          sequence.  This is particularly useful for query
          sequences that contain a large number of ambiguous
          residues, e.g. transcription factor binding sites.

-o        optimize all scores greater than OPTCUT.  If '-c' is
          not specified, OPTCUT will be calculated from the
          length of the sequence and the ktup setting.
```

```
-Q        quiet - does not prompt for any input.  Writes scores
          and alignments to the terminal or standard output file.

-r file   save a results summary line for every sequence in the
          sequence library.  The summary line includes the
          sequence identifier, superfamily number (if available)
          position in the library, and the similarity scores
          calculated.  This option can be used to evaluate the
          sensitivity and selectivity of different search
          strategies (see W. R. Pearson (1991) Genomics 11:635-
          650.)

-s file   SMATRIX is read from file.  Several SMATRIX files are
          provided with the standard distribution.  For protein
          sequences: codaa.mat - based on minimum mutation
          matrix; idnaa.mat - identity matrix; idpaa.mat -
          identity matrix for mismatches, but identical matches
          weighted according to the PAM250 matrix; pam250.mat -
          the PAM250 matrix developed by Dayhoff et al (Atlas of
          Protein Sequence and Structure, vol. 5, suppl. 3,
          1978); pam120.mat - a PAM120 matrix.  The SMATRIX also
          specifies the penalties for the first residue in a gap
          and additional residues in a gap; FASTA, the other
          alignment programs, and the SMATRIX files use -12 and
          -4. Currently, to change the -12, -4 gap penalties, the
          SMATRIX file must be edited.

-v        (LINEVAL) values used for line styles in plfasta

-w #      line length (width) = number (<200)

-x        specifies offsets for the beginning of the query and
          library sequence.  For example, if you are comparing
          upstream regions for two genes, and the first sequence
          contains 500 nt of upstream sequence while the second
          contains 300 nt of upstream sequence, you might try:

              fasta -x "-500 -300" seq1.nt seq2.nt

          If the -x option is not used, FASTA assumes numbering
          starts with 1.  This option will not work properly with
          the translated library sequence with tfasta.  (You
          should double check to be certain the negative
          numbering works properly.)

-1        sort output by init1 score.

-3        (TFASTA only) translate only three forward frames
```

For example:

```
fasta -w 80 -a seq1.aa seq2.aa
```

would compare the sequence in seq1.aa to that in seq2.aa and display the results with 80 residues on an output line, showing all of the residues in both sequences. Be sure to enter the options before entering the file names, or just enter the options on the command line, and the program will prompt for the file names.

The *file-name* parameter is the name of the file containing the sequence to be used in the search. If the sequence of interest is in a database, use the COPY command of the ATLAS program to copy it into an external file prior to running the FASTA program.

The *library file* parameter is the name of the database file to be searched or the appropriate letter from the displayed list of databases. Multiple databases may be searched by typing a percent sign followed by the letters of the desired databases.

The *ktup* parameter determines how many consecutive identities in a match. For proteins a ktup of 1 or 2 is used; ktup=1 is more sensitive but is about 3 times slower than ktup=2. DNA sequences can be searched with a ktup value of 1 to 6.

The title and the number of residues in the query sequence as well as the data library selected to be searched are displayed and the program begins the search procedure. When the search has been completed, a histogram of similarity scores will be displayed. Immediately after the histogram, the following statistics are displayed:

- The number of residues and the number of sequences with which the test sequence was compared.
- The mean and standard deviation (in parentheses) of the *initn* and *init1* similarity scores obtained in the search.
- The number of highest-scoring sequences saved as a result of the search.

The program optionally writes output to a file created in your directory. You will be prompted by:

Enter filename for results:

Press the RETURN key for no output file. The output contains the histogram and statistics in addition to the information generated at the request of the user in response to the next few prompts.

You are next prompted by:

How many scores would you like to see? [20]

Respond with the number of top-scoring sequences you would like to see. If you press the return key, the value of 20 is assumed. If the number entered is larger than the number of sequences saved, only the sequences saved will be shown.

Note: If no sequences are saved, there will be no output.

The program lists the titles of the top-scoring sequences, the *initn*, *init1*, and *opt* scores. These results are also copied to the output file.

You are then prompted by:

More scores? [0]

Respond as above. If the return key is pressed, no more sequences are shown and you will be prompted by:

Display alignments also?

If you respond with yes (Y), the program continues and you are prompted by:

Number of alignments [20]?

Respond with the number of top-scoring alignments you wish to have displayed. The alignments are not displayed on the terminal, but are written to the output file, and then the program terminates.

6.2 The Output

In the output, the region of alignment that was first selected and on which the *inti1* similarity score is based is enclosed between the X characters. Identities are marked by double dots and conservative replacements are marked by single dots within the region corresponding to the optimized alignment.

A Commands and Command Modifiers of ATLAS

```
ACCESSION accession-num   - search accession index for accession number
  /BRIEF                    - display brief output
  /COUNTS                   - display number of entries containing search term
  /SHOW                     - display how search string is parsed
  /PIR                      - restrict search to PIR1, PIR2, and PIR3 databases
  /CURRENT                  - restrict search to current list
  /ADD                      - add entries found to current list
  /SUBTRACT                 - subtract entries found from current list
  /KEEP                     - do not modify current list
  /PRINTER(=file-spec)      - direct screen output to printer (or disk file)
  /ANCHOR (default)         - match terms beginning with character string
  /NOANCHOR                 - match character string anywhere in term
  /ENTRY                    - match each character string to separate terms

AUTHOR author-name        - search author name index
  /BRIEF                    - display brief output
  /COUNTS                   - display number of entries containing search term
  /SHOW                     - display how search string is parsed
  /PIR                      - restrict search to PIR1, PIR2, and PIR3 databases
  /CURRENT                  - restrict search to current list
  /ADD                      - add entries found to current list
  /SUBTRACT                 - subtract entries found from current list
  /KEEP                     - do not modify current list
  /PRINTER(=file-spec)      - direct screen output to printer (or disk file)
  /ANCHOR (default)         - match terms beginning with character string
  /NOANCHOR                 - match character string anywhere in term
  /ENTRY                    - match each character string to separate terms

BASES database-list       - define list of active databases
  /BRIEF                    - display list without active database indicators
  /PRINTER(=file-spec)      - direct screen output to printer (or disk file)

COPY code                 - copy entry to an external file
  /CURRENT                  - process current list entries
  /TEXT                     - copy only text of entry
  /SEQUENCE                 - copy sequence only
  /OUTPUT=file-spec         - specify output file name

CROSS cross_reference     - search the cross_reference index
  /BRIEF                    - display brief output
  /COUNTS                   - display number of entries containing search term
  /SHOW                     - display how search string is parsed
  /CURRENT                  - restrict search to current list
  /ADD                      - add entries found to current list
  /SUBTRACT                 - subtract entries found from current list
  /KEEP                     - do not modify current list
  /PRINTER(=file-spec)      - direct screen output to printer (or disk file)
  /ANCHOR                   - match terms beginning with character string
  /NOANCHOR (default)       - match character string anywhere in term
  /ENTRY                    - match each character string to separate terms

DEFINE symbol-definition  - define database-list or command abbreviations
  /BASES (default)          - define abbreviations for database-list
  /COMMAND                  - define abbreviations for commands
  /DISPLAY                  - define symbol for display layout
```

```
EXTRACT                   - construct and display a modified sequence
  /FULL_NUMBERING           - display numbering for each line of sequence
  /ONE_LETTER (default)     - display one-letter amino acid abbreviations
  /THREE_LETTER             - display three-letter amino acid abbreviations
  /CURRENT                  - process only entries on current list
  /TABLE                    - the Specification is read from the entry
  /OUTPUT                   - output constructed sequence to a file
  /OUTPUT=file-spec         - output to the indicated file
  /EXTRACT                  - output without verification
  /PRINTER(=file-spec)      - direct screen output to printer (or disk file)

FEATURE feature-name      - search the feature table index file
  /BRIEF                    - display brief output
  /COUNTS                   - display number of entries containing search term
  /SHOW                     - display how search string is parsed
  /PIR                      - restrict search to PIR1, PIR2, and PIR3 databases
  /CURRENT                  - restrict search to current list
  /ADD                      - add entries found to current list
  /SUBTRACT                 - subtract entries found from current list
  /KEEP                     - do not modify current list
  /PRINTER(=file-spec)      - direct screen output to printer (or disk file)
  /ANCHOR                   - match terms beginning with character string
  /NOANCHOR (default)       - match character string anywhere in term
  /ENTRY                    - match each character string to separate terms

FIND string-expression    - search title index (also includes CONTAINS: field
                              and ALTERNATE NAMES: field for PIR entries)
  /BRIEF                    - display brief output
  /COUNTS                   - display number of entries containing search term
  /SHOW                     - display how search string is parsed
  /PIR                      - restrict search to PIR1, PIR2, and PIR3 databases
  /CURRENT                  - restrict search to current list
  /ADD                      - add entries found to current list
  /SUBTRACT                 - subtract entries found from current list
  /KEEP                     - do not modify current list
  /PRINTER(=file-spec)      - direct screen output to printer (or disk file)
  /ANCHOR                   - match terms beginning with character string
  /NOANCHOR (default)       - match character string anywhere in term
  /ENTRY                    - match each character string to separate terms
  /MAIN                     - search titles only NOT alternate names or contains
                              fields
  /TEXT                     - perform string search on titles

GENE gene-name            - search gene_name index
  /BRIEF                    - display brief output
  /COUNTS                   - display number of entries containing search term
  /SHOW                     - display how search string is parsed
  /CURRENT                  - restrict search to current list
  /ADD                      - add entries found to current list
  /SUBTRACT                 - subtract entries found from current list
  /KEEP                     - do not modify current list
  /PRINTER(=file-spec)      - direct screen output to printer (or disk file)
  /ANCHOR                   - match terms beginning with character string
  /NOANCHOR (default)       - match character string anywhere in term
  /ENTRY                    - match each character string to separate terms

GET file-name             - get current list from an external file
  /CURRENT                  - restrict search to current list
  /ADD                      - add entries found to current list
  /SUBTRACT                 - subtract entries found from current list
  /USER                     - entry-codes are specified by user

HELP                      - obtain help from program
  /BRIEF                    - list commands and modifiers
  /PRINTER(=file-spec)      - direct screen output to printer (or disk file)
```

```
JOURNAL                   - list literature and the number of citations
  /BRIEF                    - display brief output
  /COUNTS                   - display number of entries containing search term
  /SHOW                     - display how search string is parsed
  /PIR                      - restrict search to PIR1, PIR2, and PIR3 databases
  /CURRENT                  - restrict search to current list
  /ADD                      - add entries found to current list
  /SUBTRACT                 - subtract entries found from current list
  /KEEP                     - do not modify current list
  /PRINTER(=file-spec)      - direct screen output to printer (or disk file)
  /ANCHOR                   - match terms beginning with character string
  /NOANCHOR (default)       - match character string anywhere in term
  /ENTRY                    - match each character string to separate terms

KEYWORD keyword           - search the keyword index file
  /BRIEF                    - display brief output
  /COUNTS                   - display number of entries containing search term
  /SHOW                     - display how search string is parsed
  /PIR                      - restrict search to PIR1, PIR2, and PIR3 databases
  /CURRENT                  - restrict search to current list
  /ADD                      - add entries found to current list
  /SUBTRACT                 - subtract entries found from current list
  /KEEP                     - do not modify current list
  /PRINTER(=file-spec)      - direct screen output to printer (or disk file)
  /ANCHOR                   - match terms beginning with character string
  /NOANCHOR (default)       - match character string anywhere in term
  /ENTRY                    - match each character string to separate terms

LIST                      - list current list titles
  /ALL                      - list titles of all entries in active databases
  /CURRENT (default)        - process current list entries
  /OUTPUT=file-spec         - copy current list codes to an external file
  /RESTORE                  - restore previous current list
  /SET                      - active databases become current list
  /PRINTER(=file-spec)      - direct screen output to printer (or disk file)

MATCH                     - search for protein segments allowing mismatches
  /CURRENT                  - restrict search to current list
  /DEFINE                   - affect the resultant current list
  /DEFINE/ADD               - add entries found to current list
  /DEFINE/SUBTRACT          - remove entries found from the current list
  /FULL                     - report "No matches found" message
  /PEPTIDE=peptide          - specify the peptide
  /MISMATCHES=number        - specify maximum number of mismatches allowed
  /SHOW                     - show positions in peptide and matching residues
  /PRINTER(=file-spec)      - direct screen output to printer (or disk file)

MEMBERS entry-code        - display list of alignments containing code
  /BRIEF                    - display brief output
  /COUNTS                   - display number of entries containing search term
  /SHOW                     - display how search string is parsed
  /CURRENT                  - restrict search to current list
  /ADD                      - add entries found to current list
  /SUBTRACT                 - subtract entries found from current list
  /KEEP                     - do not modify current list
  /PRINTER(=file-spec)      - direct screen output to printer (or disk file)
  /ANCHOR                   - match terms beginning with character string
  /NOANCHOR (default)       - match character string anywhere in term
  /ENTRY                    - match each character string to separate terms

PRINT file-spec           - display contents of an external file

QUIT                      - quit the ATLAS program
```

```
REFERENCE reference       - search reference index
  /BRIEF                    - display brief output
  /COUNTS                   - display number of entries containing search term
  /SHOW                     - display how search string is parsed
  /CURRENT                  - restrict search to current list
  /ADD                      - add entries found to current list
  /SUBTRACT                 - subtract entries found from current list
  /KEEP                     - do not modify current list
  /PRINTER(=file-spec)      - direct screen output to printer (or disk file)
  /ANCHOR (default)         - match terms beginning with character string
  /NOANCHOR                 - match character string anywhere in term
  /ENTRY                    - match each character string to separate terms

Report                    - tabulate ancillary data
  /CURRENT                  - restrict search to current list
  /ADD                      - add entries found to current list
  /SUBTRACT                 - subtract entries found from current list
  /KEEP                     - do not modify current list
  /PRINTER(=file-spec)      - direct screen output to printer (or disk file)

SCAN peptide              - rapid sequence search for identically
                              matching segments
  /CURRENT                  - restrict search to current list
  /ADD                      - add entries found to current list
  /SUBTRACT                 - subtract entries found from current list
  /KEEP                     - do not modify current list
  /TITLE                    - display only titles of entries found
  /SEGMENT                  - display only segments of entries found
  /PRINTER(=file-spec)      - direct screen output to printer (or disk file)

SEARCH/FIELD=field-name   - search specified index(es)
  /BRIEF                    - display brief output
  /COUNTS                   - display number of entries containing search term
  /SHOW                     - display how search string is parsed
  /PIR                      - restrict search to PIR1, PIR2, and PIR3 databases
  /CURRENT                  - restrict search to current list
  /ADD                      - add entries found to current list
  /SUBTRACT                 - subtract entries found from current list
  /KEEP                     - do not modify current list
  /PRINTER(=file-spec)      - direct screen output to printer (or disk file)
  /ANCHOR                   - match terms beginning with character string
  /NOANCHOR (default)       - match character string anywhere in term
  /ENTRY                    - match each character string to separate terms
  /MAIN (for /FIELD=title)  - search titles only NOT alternate names or contains
                              fields
  /TEXT (for /FIELD=title)  - perform string search on titles

SELECT                    - select entries on basis of ancillary information
  /BRIEF                    - suppresses warning messages
  /CURRENT                  - restrict search to current list
  /ADD                      - add entries found to current list
  /SUBTRACT                 - subtract entries found from current list
  /KEEP                     - do not modify current list
  /EQUALS=entry-identifier  - searches for value equal to specified sequence
  /PRINTER(=file-spec)      - direct screen output to printer (or disk file)

SET                       - set program operation parameters
  /CLOCK                    - set internal timer
  /MENU                     - return from command mode to menu mode (PC only)
  /SWITCH=switch character  - change modifier character / to specified character
  /NOWRAP                   - turns line wrap off (affects /printer=file-spec output)
  /WRAP                     - turns line wrap on (affects /printer=file-spec output)
```

```
SHOW                       - show information about current program operation
  /COMMANDS                  - show command definitions for each field
  /DATABASE                  - show header information for active databases
  /DISPLAYS                  - show symbols defined for display layouts
  /FIELDS (default)          - show indexed fields for each database
  /INFORMATION               - show ATLAS program header
  /PRINTER(=file-spec)       - direct screen output to printer (or disk file)
  /TIME                      - show time
  /TOTALS                    - display composition table generated by TYPE command

SPECIES species-name       - search the species index file
  /BRIEF                     - display brief output
  /COUNTS                    - display number of entries containing search term
  /SHOW                      - display how search string is parsed
  /CURRENT                   - restrict search to current list
  /ADD                       - add entries found to current list
  /SUBTRACT                  - subtract entries found from current list
  /KEEP                      - do not modify current list
  /PRINTER(=file-spec)       - direct screen output to printer (or disk file)
  /ANCHOR                    - match terms beginning with character string
  /NOANCHOR (default)        - match character string anywhere in term
  /ENTRY                     - match each character string to separate terms

SUPERFAMILY superfamily    - search superfamily classifications
  /BRIEF                     - display brief output
  /COUNTS                    - display number of entries containing search term
  /SHOW                      - display how search string is parsed
  /CURRENT                   - restrict search to current list
  /ADD                       - add entries found to current list
  /SUBTRACT                  - subtract entries found from current list
  /KEEP                      - do not modify current list
  /PRINTER(=file-spec)       - direct screen output to printer (or disk file)
  /ANCHOR                    - match terms beginning with character string
  /NOANCHOR (default)        - match character string anywhere in term
  /ENTRY                     - match each character string to separate terms

SFNUM superfamily number   - search superfamily numbers
  /BRIEF                     - display brief output
  /COUNTS                    - display number of entries containing search term
  /SHOW                      - display how search string is parsed
  /CURRENT                   - restrict search to current list
  /ADD                       - add entries found to current list
  /SUBTRACT                  - subtract entries found from current list
  /KEEP                      - do not modify current list
  /PRINTER(=file-spec)       - direct screen output to printer (or disk file)
  /ANCHOR                    - match terms beginning with character string
  /NOANCHOR (default)        - match character string anywhere in term
  /ENTRY                     - match each character string to separate terms

TAXONOMY                   - select entries on basis of taxonomic classification
  /CURRENT                   - restrict search to current list
  /ADD                       - add entries found to current list
  /SUBTRACT                  - subtract entries found from current list
  /KEEP                      - do not modify current list
  /EQUALS=entry-identifier   - searches for value equal to specified sequence
  /MAIN                      - searches only species of sequence shown in entry
  /PRINTER(=file-spec)       - direct screen output to printer (or disk file)

TYPE code                  - display sequence entry
  /COMPOSITION               - display composition only
  /COMPOSITION=quiet         - compute composition but suppress its display
  /CURRENT                   - process current list entries
  /EXCHANGE                  - display entry in CODATA format
  /FULL_NUMBERING            - display numbering for each line of sequence
  /OLD                       - display an entry in old order
  /ONE_LETTER (default)      - display sequence in one-letter code
  /SEQUENCE                  - display sequence only
  /TEXT                      - display only text of entry
  /TEXT=display-layout       - display only selected text lines of entry
  /THREE_LETTER              - display sequence in three-letter code
  /PRINTER(=file-spec)       - direct screen output to printer (or disk file)
```

B One- and Three-letter Amino Acid Abbreviations

Table B–1 One- and Three-letter Amino Acid Abbreviations

A	Ala	Alanine
C	Cys	Cysteine
D	Asp	Aspartic acid
E	Glu	Glutamic acid
F	Phe	Phenylalanine
G	Gly	Glycine
H	His	Histidine
I	Ile	Isoleucine
K	Lys	Lysine
L	Leu	Leucine
M	Met	Methionine
N	Asn	Asparagine
P	Pro	Proline
Q	Gln	Glutamine
R	Arg	Arginine
S	Ser	Serine
T	Thr	Threonine
V	Val	Valine
W	Trp	Tryptophan
Y	Tyr	Tyrosine
B	Asx	Asp or Asn, not distinguished
Z	Glx	Glu or Gln, not distinguished
X	X	Undetermined or atypical amino acid

These abbreviations conform to those suggested by the IUPAC-IUB Commission on Biochemical Nomenclature, J. Biol. Chem. 243, 3557-3559, 1968.

C Punctuation in Protein Sequences

	No punctuation between two adjacent amino acids indicates that they are connected as determined experimentally
( )	Encloses a region, the composition but not the complete sequence of which has been determined experimentally, or encloses a single residue that has been tentatively identified
=	Indicates)(, the juxtaposition of two regions of indeterminate sequence, while preserving proper spacing between amino acids
/	Indicates that the adjacent amino acids are from different peptides, not necessarily connected. When the amino end of a protein has not been determined, / precedes the first residue. When the carboxyl end has not been determined, / follows the last residue. When)/, /(, or)/(are needed, only / is used
.	Outside of parentheses, indicates the ends of sequenced fragments. The relative order of these fragments was not determined experimentally but is clear from homology or other indirect evidence
.	Within parentheses, indicates that the amino acid to its left has been placed with at least 90% confidence by homology with known sequences
,	Indicates that the amino acid to its left could not be positioned with confidence by homology

D PIR-International Protein Sequence Database Entry Format

Each entry consists of a variable number of consecutive records. The primary information contained in these lines is divided into four sections. The sections are listed below in the order in which they occur in the entry.

1 **HEADER** (exactly 1 record) Information that marks the line as the first line of an entry and that identifies the sequence contained in the entry. It contains the entry type specifier and the entry-code.

2 **TITLE** (exactly 1 record) The protein name and one or more species names.

3 **SEQUENCE** (variable number of records) The amino acid sequence ending with an asterisk.

4 **TEXT** (variable number of records) Reference citations and text information.

To be accessed by the NBRF programs, external sequence files must conform to the same format as the database entries. Only the HEADER, TITLE, and SEQUENCE lines are necessary. Any number of entries may be contained in a single external file provided they all have different identification codes.

D.1 HEADER

The first line of each entry is known as the HEADER line. The first character on this line is a right arrowhead (>) which indicates the beginning of an entry; it is followed by a two-character entry type, 'P1' for a complete protein or 'F1' for a fragment. The fourth character on the line is a semicolon; this character separates the protein type from the entry-code that immediately follows. An entry-code can be any combination of four to six alphanumeric characters containing no internal blanks.

```
Header Line Format
---------------------
Field    Length   Contents of field
-----    ------   -----------------
 '>'        1     marks the line as an entry header
TYPE        2     type of sequence in the entry:
                        'P1'  protein, complete
                        'F1'  protein, fragment
 ';'        1     field separator
CODE     4 to 6   unique retrieval key we have assigned to
                    the entry
```

The CODE contains four to six alphanumeric characters.

```
Header line examples:
--------------------
   >P1;AZBR
   >P1;DENCEN
   >F1;XNECV
   >P1;LQBP37
```

D.2 TITLE

The second line of the entry contains the entry title, which consists of a protein name and a species name separated by a dash flanked by two blank spaces (' - '). The species name, which lists the biological source from which the protein was obtained, is always on the right of the dash.

```
Title Line Format
--------------------
Field    Length   Contents of field
-----    ------   -----------------
SEQNAM  variable  name of the protein
 ' - '      3     field separator
ORGNAM  variable  name of biological source of the protein
```

If the organism or organelle translates nucleic acid to protein by a special genetic code, this fact is noted in the ORGNAM field by the symbol '(SGCn)', where n denotes the special genetic code.

```
Title line examples:
--------------------
azurin precursor - Pseudomonas aeruginosa
DNA ligase (ATP) (EC 6.5.1.1) - phage T7
cytochrome c - castor bean
thyroliberin - pig
hemoglobin alpha chain - eastern gray kangaroo
epidermal growth factor precursor - human
```

D.3 SEQUENCE

The sequence section of an entry consists of a variable number of records. Each sequence record may be up to 500 characters long. The characters represent the amino acid sequence stored from the amino end to the carboxyl end. The symbols used for the amino acids are the one-letter abbreviations shown in Appendix B. The amino acid sequence is terminated by an asterisk, which is the last character on the last line of the sequence section. In addition, the sequence may contain punctuation symbols to indicate various degrees of reliability of the data (see Appendix C). One punctuation symbol may precede any amino acid symbol or the terminating asterisk. The sequence lines contain no blank characters (if space characters are used to separate amino acid symbols, they are ignored by the NBRF programs when they occur within sequences).

```
Sequence line examples:
----------------------
GDVE(G.K.G.I.F=T,M,C.S.Q,C.H.V,E.K.G.G.K.H)
FTGPNLHGLFGRK.TGQAVGYSYTAANK.NK.GIIWGDDTLM
EYLENPK.RYIPGTK.MVFTGLSK.YRE
RTNLIAYLK.EK.TAA*
```

D.4 TEXT

The TEXT section contains annotation information such as species, reference citations, genetic information, superfamily classification and search keywords.

The format of the TEXT section consists of a variable number of records. Each TEXT record may be up to 500 characters long. Each TEXT record, with the exception of the citation record, begins with a Tag that indicates the type of information contained on that record. Certain Tags mark the beginning of a block of data. The types of Tags or data items are listed below with examples. Required or optional notations with regard to tags refer to entry standards for PIR-International. A more compact format specification is given in Example D–1.

D.4.1 Alternate names specification

(optional) An alternate name specification consists of a single text record that is identified by having 'N;Alternate names:' Tag as the first 18 characters. The remainder of the line contains a list of other common names for the protein. Alternate names are separated by semicolons.

```
Alternate name line examples:
----------------------
N;Alternate names: soluble cytochrome f; cytochrome c553
N;Alternate names: cusacyanin; plantacyanin
N;Alternate names: component II; nitrogenase reductase
```

D.4.2 Contains line

(optional) A contains line consists of a single text line that is identified by having 'N;Contains:' as the first 11 characters. This record specifies other activities or functions that are included in the sequence shown in the entry. Multiple "contains" titles are separated by semicolons.

```
Contains line examples:
----------------------
N;Contains: cytochrome b2 core
N;Contains: ribonuclease (EC 3.1.-.-) activity
N;Contains: Arg-vasopressin; neurophysin 2
N;Contains: intestinal peptide PHM-27
```

D.4.3 Species line

(required) The Species line is a single record identified by 'C;Species:' as the first 10 characters. This record describes the source or organism from which the sequence is derived. Each entry contains this record and it may mark the beginning of a Species data block.

D.4.4 Species Note record

(optional; contained in Species block) The Species Note line is a single record identified by 'A;Note:' as the first 7 characters. This record describes special Species information and is contained in the Species block. All information in the 'C;Host:' records from previous versions of the database has been transferred to this record.

```
Examples of maximal Species block:
---------------------------------
  C;Species: vaccinia virus
  A;Note: host Homo sapiens (man)
  C;Species: equine herpesvirus 1, equine abortion virus
  A;Note: host Equus caballus (domestic horse)
```

D.4.5 Date record

(required) The Date line is a single record identified by 'C;Date:' as the first 7 characters. This record indicates the date the entry was added to the data set, the date the SEQUENCE was modified and the date the TEXT was modified. Each of the dates (add, seq, text) is optional but at least one must appear.

```
Examples:
  C;Date: 31-Jul-1979  sequence_revision 30-Sep-1992
     text_change 14-Oct-1993
  C;Date: 31-May-1979  sequence_revision 25-Feb-1985
     text_change 14-Oct-1993
  C;Date:  text_change 30-Jun-1993
  C;Date:  sequence_revision 30-Sep-1991  text_change 02-Dec-1993
```

D.4.6 Entry-specific Accession record

(required) The Entry-specific Accession line is a single record identified by 'C;Accession:' as the first 12 characters. This record indicates a list of Accession numbers associated with the entry.

```
Examples:
  C;Accession: A92196; A92218; A00169
  C;Accession: A90383; A92053; B93774; A92231; A00170
  C;Accession: JS0745; A00172; S07960
```

D.4.7 Author/citation records

(required; start of Reference block) The Author line is a single record identified by 'R;' as the first 2 characters. This record indicates a list of authors, separated by semicolons, associated with the Reference block. Immediately following is the citation record that specifies the source of the Reference. This pair of records is required and begins the Reference block. The Reference block may be repeated in the TEXT section.

```
Examples:
  R;Aigle, M.; Biteau, N.; Crouzet, M.
  submitted to the Protein Sequence Database, March 1992
  R;Cossart, P.; Katinka, M.; Yaniv, M.
  Nucleic Acids Res. 9, 339-347, 1981
  R;Skala, J.; Purnelle, B.; Goffeau, A.
  Yeast 8, 409-417, 1992
```

D.4.8 Authors record

(optional; repeating; contained in Reference block) The Authors line is an optionally repeating record identified by 'A;Authors:' as the first 10 characters. This record is used as a supplement to the Author list in the

'R;' record. If the 'R;' record exceeds the maximum record length of 500 characters then additional authors are listed on the Authors line.

D.4.9 Reference Title record

(optional; contained in Reference block) The Reference Title line is a single record identified by 'A;Title:' as the first 8 characters or 'A;Description:' as the first 14 characters. This record is the publication title (A;Title) or description of a sequence submission (A;Description:). Either Tag may be present but not both.

```
Examples:
  A;Title: Amino acid sequence of ragweed allergen Ra3.
  A;Description: The amino acid sequence of a type I copper protein
      with an unusual serine- and hydroxyproline-rich C-terminal
      domain isolated from cucumber peelings.
```

D.4.10 Reference number record

(required; contained in Reference block) The Reference number line is a single record identified by 'A;Reference number:' as the first 19 characters. This record contains standardized information relating a reference with a six character alphanumeric string (ref_num). Optionally a Medline reference number may appear in the record identified by the 'MUID:' Tag and separated form the ref_num by a semicolon.

```
Examples:
  A;Reference number: A94561
  A;Reference number: A00100; MUID:82075747
```

D.4.11 Contents record

(optional; contained in Reference block) The Contents line is a single record identified by 'A;Contents:' as the first 11 characters. This record specifies the source of the protein (species and/or strain), the portion of the sequence reported, the method of sequence determination, or the extent of experimental detail reported. The record may indicate that the reference is included as a source of ancillary information such as X-ray crystallography or active site identification.

```
Examples:
  A;Contents: ATCC 16455
  A;Contents: annotation; methylation
  A;Contents: X-ray crystallography, 2.8 angstroms
  A;Contents: Strain BALB/c
```

D.4.12 Reference Note record

(optional; repeating; contained in Reference block) The Reference Note line is a repeating record identified by 'A;Note:' as the first 7 characters beneath the start of a Reference block. This record describes reference specific comments.

```
Examples:
  A;Note: This is the final paper in a series.
  A;Note: The nucleotide sequence is not given in this paper.
```

D.4.13 Reference-specific Accession records

(optional; start of Accession block; contained in Reference block) The Reference-specific Accession line is a single record identified by 'A;Accession:' as the first 12 characters. This record indicates a single unique six character alphanumeric string (acc_num) associated with the shown sequence according to the sequence specification in the Residues record described below. These unique numbers specifiy a unique sequence. The presence of this Accession record implies the start on an Accession block that may be repeated beneath the Reference block; the Accession block(s) is contained in the Reference block.

```
Examples:
  A;Accession: A00086
  A;Accession: JT0008
  A;Accession: S13939
```

D.4.14 Accession Status record

(optional; contained in Accession block) The Accession-specific Status line is a single record identified by 'A;Status:' as the first 9 characters. This record indicates the review status of the sequence refered to by the acc_ num. Currently "preliminary" is the only value for this information.

```
Example:
  A;Status: preliminary
```

D.4.15 Molecule type record

(required if Accession present; contained in Accession block) The molecule type line is a single record identified by 'A;Molecule type:' as the first 16 characters. This record indicates the type of molecule from which the sequence was determined. Valid values for this data item are: "protein", "DNA", "mRNA", "nucleic acid" and "genomic RNA."

```
Examples:
  A;Molecule type: protein
  A;Molecule type: DNA; mRNA
```

D.4.16 Residues record

(required if Accession present; contained in Accession block) The Residues line is a single record identified by 'A;Residues:' as the first 11 characters. This record specifies a reported sequence according to the amino acid sequence as depicted in PIRn.SEQ.

```
Examples:
  A;Residues: 1-85,'SK',88-92,'N',94-100,'K',102-103,'A' <BAH>
  A;Residues: 2-57,'IV',60-61,'ZZ',64-66,'Z',68-69,'ZB',72-105 <HEN>
  A;Residues: 1-107 <CHA>
```

D.4.17 Cross-references record

(optional; contained in Accession block) The Cross-references line is a single record identified by 'A;Cross-references:' as the first 19 characters. This record specifies a list of database/identifier pairs that indicate related sequence information in another database. Current cross referenced databases are: "GB", "EMBL", "PDB", "DDBJ", "CAS", "NCBIP" and "NCBIN."

```
Examples:
  A;Cross-references: GB:J04618; GB:J04619
  A;Cross-references: CAS:124041-95-8
```

D.4.18 Accession Genetics record

(optional; contained in Accession block) The Accession Genetics line is a single record identified by 'A;Genetics:' as the first 11 characters. This record contains a Tag that specifies which 'C;Genetics:' block (defined below) describes the sequence report depicted by the Accession block. This record is present only when more than one 'C;Genetics:' block is present in the entry.

```
Examples:
  A;Genetics: ST1
  A;Genetics: ST2
  A;Genetics: HBA1
```

D.4.19 Accession Note record

(optional; repeating; conntained in Reference block) The Accession Note line is a repeating record identified by 'A;Note:' as the first 7 characters beneath the start of an Accession block. This record describes sequence specific comments.

```
Examples:
  A;Note: The authors translated the codon CTG for residue 169 as Ile.
  A;Note: 175-Ala was also found.
  A;Note: The difference at the carboxyl end is due to a frameshift.
```

D.4.20 Comment records

(optional; repeating) The Comment line is a repeating record identified by 'C;Comment:' as the first 10 characters. This record contains general information in a free format, natural language form about the protein sequence entry. Some Comment records can be decomposed and the information move to more appropriate records; this is an ongoing standardization project.

```
Examples:
  C;Comment: Met preceding 1-Gly is removed after translation.
  C;Comment: The sequence shown is iso-1-cytochrome c.
  C;Comment: Euglena is a genus of green algae.
```

D.4.21 Genetics record

(optional; start of Genetics block) The Genetics line is a single record identified by 'C;Genetics:' as the first 11 characters. This record contains no information except if more than one Genetics block exists in the entry. In the case of multiple gene information this record will contain a unique Tag that points to an Accession block; this indicates which sequence report is related to the genetic data. Presence of this record implies the start of a Genetics block and is required if other genetic information exists such as that defined below. The Genetics block may be repeated within an entry.

```
Examples:
  C;Genetics:
  C;Genetics: <ST1>
  C;Genetics: <ST2>
  C;Genetics: <HBA1>
```

D.4.22 Gene record

(optional; contained in Genetics block) The Gene line is a single record identified by 'A;Gene:' as the first 7 characters. This record specifies the gene symbol used to denote the gene. Some symbols may contain "GDB" as a Tag; this indicates a cross reference to the Genome Database.

```
Examples:
  A;Gene: psbE
  A;Gene: GDB:CYP2D6
  A;Gene: CYP2B1
```

D.4.23 Map position record

(optional; contained in Genetics block) The Map position line is a single record identified by 'A;Map position:' as the first 15 characters. This record specifies a map position on which the gene is located. For viruses this may indicate a segment number.

```
Examples:
  A;Map position: 71.6-76.2
  A;Map position: 85 min
  A;Map position: 4q21-q23
```

D.4.24 Genome record

(optional; contained in the Genetics block) The Genome line is a single record identified by 'A;Genome:' as the first 9 characters. This record specifies which type of genome is described. Current values for this record are: "mitochondrion", "chloroplast", "cyanellle" and "plasmid."

```
Examples:
  A;Genome: chloroplast
  A;Genome: cyanelle
  A;Genome: plasmid
  A;Genome: mitochondrion
```

D.4.25 Genetic code record

(optional; contained in Genetics block) The Genetic code line is a single record identified by 'A;Genetic code:' as the first 15 characters. This record indicates which Special Genetic Code table is used by the specific organism for nucleic acid translation. Current values for this record are "SGC1" - "SGC9."

```
Examples:
  A;Genetic code: SGC1
```

D.4.26 Start codon record

(optional; contained in Genetics block) The Start codon line is a single record identified by 'A;Start codon:' as the first 14 characters. This record indicates the codon in the nucleic acid sequence where translation is initiated.

```
Examples:
  A;Start codon: ATT
  A;Start codon: ATC
  A;Start codon: GTG
```

D.4.27 Introns record

(optional; contained in Genetics block) The Introns line is a single record identified by 'A;Introns:' as the first 10 characters. This record specifies the intron segments needed to code for the gene product. Segments are separated by semicolons.

```
Examples:
  A;Introns: 253/3; 270/3
  A;Introns: 139/1; 143/2; 169/2; 253/3; 270/3
```

D.4.28 Genetics Note record

(optional; repeating; contained in Genetics block) The Genetics Note line is a repeating record identified by 'A;Note:' as the first 7 characters beneath the start of a Genetics block. This record describes genetics specific comments.

```
Examples:
  A;Note: strain D273-10B/A21
  A;Note: strain 777-3A
```

D.4.29 Function record

(optional; start of Function block) The Function line is a single record identified by 'C;Function:' as the first 11 characters. This record contains no information except if more than one Function block exists in the entry. In the case of multiple function information this record will contain a unique Tag to distinguish each block. Presence of this record implies the start of a Function block and is required if other function information exists such as that defined below. The Function block may be repeated within an entry.

D.4.30 Function Description record

(optional; contained in Function block) The Function Description line is a single record identified by 'A;Description:' as the first 14 characters and immediately following the Function record. This record indicates the type of function the specified protein may have.

```
Example:
  C;Function:
  A;Description: This protein functions as a molecular chaperone in the
      endosymbiont.
```

D.4.31 Superfamily record

(optional) The Superfamily line is a single record identified by 'C;Superfamily:' as the first 14 characters. This record indicates which Superfamily(s) has the protein as a member. Individual Superfamily names are separated by semicolons.

```
Examples:
  C;Superfamily: phosphorylase
  C;Superfamily: sucrose synthase; sucrose/sucrose-phosphate synthase homology
```

D.4.32 Keywords record

(optional) The Keywords line is a single record identified by 'C;Keywords:' as the first 11 characters. This record indicates which keyword(s) are associated with the entry. Terms in the list are separated by semicolons and are used as a retreival key.

```
Example:
  C;Keywords: homodimer; NAD; oxidoreductase
  C;Keywords: oxidoreductase; pentose phosphate pathway
```

D.4.33 Feature record

(optional; repeating) The Feature line is a repeating record identified by 'F;' as the first 2 characters. This record may be one of many comprising a Feature table. Each line of the Feature table has the following format. Positions 3 to the occurence of a '/' character is a range or site specification. Multiple segments corresponding to a single feature are separated by commas. The feature title appear after the '/' and consists of a feature descriptor, a title, and an optional code extension enclosed by the symbols

< and >. A feature descriptor is a word or short phrase followed by a colon that defines the type of feature (refer to Table IV for a complete listing and explanation of feature descriptors). The code extension consists of a short character string (alphanumerics only) that when associated with the entry identification code defines a logical address for the subsequence that is unique throughout the database. For example, the path DEECK->DKI uniquely defines the feature with code extension DKI in entry DEECK.

```
Examples:
  F;316/Active site: Asp
  F;483/Binding site: carbohydrate
  F;19-69,28-52,44-65/Disulfide bonds:
  F;1-249/Domain: aspartokinase I <DKI>
  F;1-24/Domain: signal sequence <SIG>
  F;50-54/Peptide: Met-enkephalin 1 <ME1>
  F;15-72/Protein: basic protease inhibitor <MAT>
```

Feature Table Descriptors

The following descriptors denote single residue sites. These sites may be represented using the full feature residue specification conventions; however, the specification should be interpreted to specify a collection of single sites.

```
Active site:
Binding site:
Cleavage site:
Inhibitory site:
Modified site:
```

The following descriptors denote residues connected by covalent bonds. The pairs of residues linked by hyphens should be intepreted as being connected by bonds. Single residues may be specified if the bond is between an amino acid in the sequence and another protein chain or molecule. The Cross-links: descriptor specifically denotes bonds linking the sequence shown to an adjacent protein chain.

```
Disulfide bonds:
Cross-links:
```

The following descriptors denoting sequence regions. Pairs of residues linked by hyphens denote a region extending from the first to the second position inclusive. The Domain: descriptor indicates distinct functional regions that may be of separate evolutionary origin. The Duplication: descriptor indicates regions that have evolved by sequence duplication. The Peptide: and Protein: descriptors indicate regions corresponding to the mature protein or peptides derived from the sequence shown. The Region: descriptor is used to indicate all other regions.

```
Domain:
Peptide:
Protein:
Region:
```

Example D–1 Format Specification of PIR-International Entry:

```
>..;code                          HEADER

title - trivial name              TITLE
X X X X X X X X X X *             SEQUENCE
N;Alternate names:                TEXT
N;Contains:
C;Species:                          Species block
  A;Variety:
  A;Note:
C;Date:
C;Accession:
R;                                  Reference block (may repeat)
  citation
  A;Authors:
  A;Title:
  A;Description:
  A;Reference number:
  A;Contents:
  A;Note:
  A;Accession:                      Accession block (may repeat)
    A;Status:
    A;Molecule type:
    A;Residues:
    A;Cross-references:
    A;Experimental source:
    A;Genetics:
    A;Note:
C;Comment:                          Comments (may repeat)
C;Genetics:                         Genetics block (may repeat)
  A;Gene:
  A;Map position:
  A;Genome:
  A;Gene origin:
  A;Genetic code:
  A;Start codon:
  A;Introns:
  A;Other products:
  A;Note:
C;Complex:                          Complex
C;Function:                         Function block
  A;Description:
  A;Pathway:
  A;Note:
C;Superfamily:
C;Keywords:
F;                                  Features block (may repeat)
```

Index

F

G

H

J

K

L

M

N

P

Q

R

S

T