

*Ecology*

# **STRUCTURAL INFORMATION OF PROTEIN BASED ON MULTIPLE DATABASES**

*A dissertation submitted to Jawaharlal Nehru University  
in partial fulfillment of the requirements  
for the award of the degree of*

**Master of Technology  
in  
Computer Science & Technology**

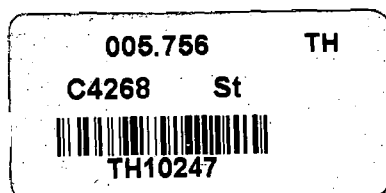
By

**V. MALLIKARJUNA CHETTY**



**SCHOOL OF COMPUTER & SYSTEMS SCIENCES  
JAWAHARLAL NEHRU UNIVERSITY  
NEW DELHI 110067**

**JANUARY 2003**





जवाहरलाल नेहरू विश्वविद्यालय

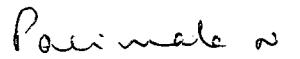
SCHOOL OF COMPUTER & SYSTEMS SCIENCES  
JAWAHARLALNEHRU UNIVERSITY  
NEW DELHI – 110067 (INDIA)

**CERTIFICATE**

This is to certify that the dissertation titled “**Structural Information of Protein Based on Multiple Databases**” which is being submitted by **Mr. V. Mallikarjuna Chetty** to the **School of Computer & Systems Sciences, Jawaharlal Nehru University, New Delhi**, in partial fulfillment of the requirements for the award of **Master of Technology in Computer Science & Technology** is a bonafide work carried out by him under the supervision of **Prof. Parimala .N.** The matter embodied in the dissertation has not been submitted for the award of any other degree or diploma.

  
6-01-03

Prof. K.K. Bharadwaj  
Dean, SC & SS  
Jawaharlal Nehru University  
New Delhi 110067



Prof. Parimala. N  
SC & SS  
Jawaharlal Nehru University  
New Delhi 110067

## ACKNOWLEDGEMENTS

I am thankful to my guide, Prof. Parimala N. for suggesting me to do the work in *Structural Information of Protein Based on Multiple Databases* and being there to direct me all the time.

I would like to thank Prof. K.K. Bharadwaj, Dean, School of Computers and Systems Sciences (SC&SS), JNU for providing excellent lab facilities.

I extend my thanks to Dr. Andrew, Bioinformatics Center (BIC), JNU who helped us in successful completion of the project.

I would also like to thank my classmates Mr. V. Praveen kumar Reddy, Mr. B.Damodar and all my batch mates for their cooperation and advice.

V. Mallikarjuna Chetty

## **ABSTRACT**

In the present emerging biotechnological scenario, there is a need to know about the internals of the existing proteins. So we took up the task to develop a query interface for retrieving data across multiple protein databases. As part of this project, this thesis is concerned with displaying the query results of these multiple protein databases and refining the query results. These results can be displayed in List form, Table form and Graphical structural representation. Each of these is implemented in this thesis. Additionally, we look at refining a query. In this, the attribute list to be retrieved can be modified or further conditions can be added to narrow down the search.

## CONTENTS

<b>CERTIFICATE</b>	I
<b>ACKNOWLEDGEMENTS</b>	II
<b>ABSTRACT</b>	III
<b>1. INTRODUCTION</b>	1
1.1 Heterogeneous databases	2
1.1.1 <i>sequence Databases</i>	2
1.1.2 <i>Structural Databases</i>	3
1.1.3 <i>Literature Databases</i>	3
1.1.4 <i>Genome Databases</i>	4
1.2 Different Formats of Databases	4
1.2.1 <i>Flatfile databases</i>	4
1.2.2 <i>Relational databases</i>	4
1.2.3 <i>Object Oriented databases</i>	5
1.3 Data Retrieval Systems	5
1.3.1 <i>Hyper Text Navigation</i>	5
1.3.2 <i>Data ware Housing</i>	6
1.3.3 <i>Unmediated Multi DB Queries</i>	6
1.3.4 <i>Federated Approach</i>	6
1.4 Problems in Existing Data Retrieval Systems	7
1.5 Proposed Approach	9
1.5.1 <i>Providing User Interface to Build query</i>	9
1.5.2 <i>Accessing Database efficiently</i>	10
1.5.3 <i>Displaying Output in User-friendly Environment</i>	10
<b>2. OVERVIEW OF PROTEIN DATABASES</b>	12
2.1 Protein Databases	12
2.1.1 <i>SWISS-PROT</i>	12
2.1.2 <i>PDB</i>	12
2.1.3 <i>Prosite</i>	13
2.2 Formats	14
2.2.1 <i>Swiss-Prot Format</i>	14
2.2.2 <i>PDB Format</i>	15
2.2.3 <i>Prosite Format</i>	17

3. SYSTEM ARCHITECTURE AND DESIGN	19
3.1 Over View Of the Architecture	19
3.2 Design	21
3.2.1 <i>Structure chart for display of Textual information</i>	22
3.2.1 <i>Structure chart for display of Graphical information</i>	23
3.2.1 <i>Structure chart for Refinement of query</i>	24
4. RESULTS PRESENTATION	26
4.1 List View	26
4.2 Table View	27
4.3 Functions Provided by the Textual View of Data	29
4.4 Structural View	31
4.4.1 <i>Navigation Through 3D world</i>	31
4.5 Refining Query	37
5. IMPLEMENTATION	38
5.1 Display of Textual information	38
5.1.1 <i>Table View</i>	38
5.1.2 <i>List View</i>	41
5.2 Display Of Graphical Information	42
5.3 Refinement of Query	43
6. CONCLUSION	44
REFERENCES	46
APPENDIX A	48

# CHAPTER 1

## INTRODUCTION

In an emerging field of biotechnological scenario, there is a need to know about the internals of existing proteins. The researchers need to analyze the fundamental biomedical problems, which includes gene organization, sequence analysis, and structure prediction. Many of the current research projects also include detection and analysis of genome organization, repeating sequence patterns, protein domains and structural elements.

In this scenario the biological data has gained greater importance. The biological data is scattered around the globe. The different biological databases are maintained at different sites in the World Wide Web. Some well-known examples are SWISS-PROT [3], Prosite [4], PDB [5], GDB [16], and Gen Bank [8]. These databases are such as Nucleic acid databases, sequence databases, protein three-dimensional structural databases, Literature databases, sequence related databases et cetera.

Different institutes maintain databases, which are relevant to their research programs. For example the European Bioinformatics Institute (EBI)[1] is a center for research and services in bioinformatics. This institute manages and analyses databases of biological data including DNA, Protein sequences and molecular structures. Swiss Institute of bioinformatics (SIB)[2] is dedicated to the analysis of protein sequences and structures and it is the responsible to maintain the SWISS-PROT [3] a Protein knowledge base, PROSITE [4] a database of protein families and domains. The PDB [5] is the single international repository for the processing and distribution of experimentally determined three-dimensional macromolecular structure data. Rutgers, the State University of

Newjersy, the San Diego Super Computer Center (SDSC) [6] at the University of California, San Diego, and the National Institute of Standards Technology manage the PDB [5]. The NCBI [21] is responsible for maintaining the GenBank DNA sequence database. In addition to GenBank, NCBI supports and distributes a variety of databases for the medical and scientific communities. These include the Online Mendelian Inheritance in Man (OMIM) [21], the Molecular Modeling Database (MMDB) [21] of 3D protein structures, the Unique Human Gene Sequence Collection (UniGene), a Gene Map of the Human Genome, the Taxonomy Browser, and the Cancer Genome Anatomy Project (CGAP), in collaboration with the NCI (National Cancer Institute) [21].

## **1.1 Heterogeneous Databases**

Molecular Biology databases are very heterogeneous in nature. This is as a result of different aims, and usage's that they have been developed for. The molecular databases contain only data gathered on one specific organism and/or developed and maintained by only one research team, other molecular biological databases aim at collecting all data available on biological interesting concepts and are the result of long listing international co-operations between research laboratories. Further more, different approaches are used for data modeling, for storing and for data analysis and query purposes. The different kinds of information that exist in molecular biological databases are discussed below.

### **1.1.1 Sequence Databases**

The Sequence databases contain the information about the sequence information like the DNA sequence information, Protein sequence information, Nucleic acid sequence information and Peptide sequence information. The most commonly used sequence databases are Nucleic Acid Sequences and Peptide Sequences. The Nucleic Acid Sequences like EMBL (Compiled at the EBI [1], Europe) [7], Genbank (Compiled in the



USA) [8], cDNA (HGMP-RC generated cDNA's) [9], and EPD (Eukaryotic Promoter Database) [10] contain the Nucleic acid sequence information. The Peptide Sequences like SwissProt (Compiled at the EBI & Switzerland)[1], PIR (Protein Identification Resource) [12], TREMBL (Translation of EMBL coding sequences) [7] contain the peptide sequence information. EMBL nucleotide database [7] is a primary repository for genetic sequences. EMBL-DB contains information scanned from the literature and submitted directly. GenBank [8] is similar in principle to the EMBL nucleotide Sequence Database [7] and serves as repository for known genetic sequences. GenBank [7] has close connections to the EMBL database and all information available in GenBank is also contained in the EMBL database.

### 1.1.2 Structural Databases

The Structural databases contain information on the three-dimensional structure of molecules, chiefly proteins. Data primarily based on x-ray crystallography (> 80%) NMR, or theoretical models (< 2%). For example Protein databases [13], Molecular Modeling Databases [14] contains the information on 3-D structural information of molecules. The PDB [5] is the largest repository for 3D protein structures determined by X-ray crystallography or nuclear magnetic resonance (NMR) and contains examples of all known unique protein families.

### 1.1.3 Literature Databases

Most Molecular Biology databases contain literature references. More and more computational biologists consider data documentation by means of references a premier objective. For example PubMed [15] is a well-known Literature database.

#### 1.1.4 Genome Databases

Genome databases contains the information of genomes and chromosome information. GDB [16] holds data on Human gene loci, polymorphism's, mutations, probes, genetic maps, GenBank [8], citations and contacts. For example Human Genome Databases [17], Human Chromosome Specific Databases [17].

### 1.2 Different Formats of Databases

The Biological databases available on the web are not in the same format. Many important molecular Biological databases are of different formats. These databases are a collection of unstructured flat-files to highly structured Object Oriented Databases. Molecular databases can be classified as follows:

#### 1.2.1 Flat-File Databases

In the early days of molecular biological databases, database management systems were rarely used. Instead most molecular biological databases were built up as indeed ASCII text files called "flat-files".

Molecular biological databases [14] implemented as flat-files in general have no explicit data model. Their entries are usually structured either implicitly or explicitly by search indexes. Most flat-file collections are explicitly structured using key words. The term "line type" is often used as for these keywords. Sequence databases are often flat-file collections. For example GenBank [8], SWISS-PROT [3], EMBL [7], PDB [5], Prosite [4] etc are Flat-File databases.

#### 1.2.2 Relational Databases

The Relational database consists of tables with homogeneous content, where each table contains records (items) and records has one or more fields (properties). Key fields relate

records in different tables. Contents from different tables are brought together using these key values. Since structure is very constrained the basic operations can be performed very easily. For example Genome Sequence DB [18], and DDBJ (DNA Data Bank of Japan)[19] and are Relational Databases.

### 1.2.3 Object Oriented Databases

In the Object Oriented Databases data is organized into a hierarchy of concepts or classes. Each concept has a set of attributes, which can have typed values. Concepts can inherit values of attributes from parents in the hierarchy. Object oriented databases can model richer set of relations than the relational model. For example AtDB [20] is an Object Oriented Database.

## 1.3 Data Retrieval Systems

As the Research in the Biological field is growing day to day, the biological data has gained more significance. As discussed above, the Biological databases are very heterogeneous in nature and are distributed with a culture of sharing and rapid circulation of information. Frequently the information sources have different structures, contents and query languages. The goal of DB interoperation research is to allow users to interact with a set of disconnected, heterogeneous DBs as seamlessly as they interact with each individual database. Below we describe the different ways in which access to interoperable databases has been provided.

### 1.3.1 Hypertext Navigation

This approach allows users to interactively navigate from some presentation of an entry in one member DB, to an entry in another member DB, by traversing links between the two. Generally, only two operations are supported: searching within one DB to find a starting entry (such as retrieving a Genbank [8] entry using a protein name), and then

requesting a linked entry from another DB. Given a display of a Genbank entry [8], for example, the user could request to see an associated Medline entry that was explicitly listed in the Genbank entry [8]. Representative systems include Entrez [23] by NCBI, the ExPASy WWW server and Genome Net.

### 1.3.2 Data Warehousing

In this approach, a set of heterogeneous DBs are translated and physically loaded into a single, homogeneous DBMS. For each DB to be integrated, we must define a translator from the format and conceptualization of that DB, into the format and conceptualization of the warehouse (central) DB. The conceptualization of the warehouse DB must span the conceptualizations of all member DBs that are to be included in the warehouse. As an example, this approach might be applied to load Swiss-Prot [3], PDB [5], and PIR [12] into one large Oracle DB. Once all DBs are present in the Oracle warehouse, arbitrary DBMS queries can be applied to the data.

### 1.3.3 Unmediated MultiDB Queries

This approach allows users to construct complex queries that are evaluated against multiple heterogeneous DBs. A query explicitly identifies both the member DBs that it applies to, and the tables and attributes that are to be queried within each DB.

### 1.3.4 The Federated Approach

The Federated Approach combines aspects of approaches Data Warehousing and Unmediated MultiDB Queries. Like Unmediated MultiDB Queries, it does not force the member DBs to be physically integrated within one DBMS. Like Data Warehousing, this approach defines translations between a single global data model and conceptualization (the *federated schema*), and the data model and conceptualization of the member DBs.

User queries are dynamically translated from the environment of the federated schema into the environment of the member DBs.

## **1.4 Problems in Existing Data Retrieval Systems**

The available analysis tools have no common user interface and often work only on limited subset of data. When Biologists need to retrieve the information, he/she should perform the following tasks during query information execution.

The users must identify source and their locations. They must know content/function of sources. They have to recognize components of query and target them to appropriate sources in the optimal order. They have to communicate with sources, need to transform the data between source formats, express syntactically complex queries and manage the results from different sources. For example, The Sequence Retrieval System (SRS) [22], A form-based interface allows the user to ask complex, although restricted, queries over multiple sources that are executed simultaneously. Queries composed of sub-queries, which have to be executed in a given order, must be issued separately by the user, and the results of one sub-query piped into another by hand. And even it does not provide guidance as to which source is most appropriate for a given query.

The foregoing implies that all of these tasks are burden on biologists, most of whom are not experts in Bioinformatics, and limits the use that can be made of available information.

We now, consider the problems in particular that exist in protein data retrieval systems. In an emerging field of Biological databases, the data about of proteins is available in three different databases. The sequence database SWIS\_PROT, the sequence related database PROSITE [4] and protein 3D structure in PDB (Protein Database) [5] are three such databases. Each of these protein databases represents different information of the protein.

In all these databases attributes used in the databases are not unique and the same information is represented with different names in different databases. It is expected that the user remember all the attribute names and their meaning while fetching data from these databases.

Systems like SRS (Sequence Retrieval System) [22] and ENTREZ [23] provides a graphical user interface but they are database specific user interfaces. User has to know from which database he has to extract data.

As these databases are not conventional relational databases and information is not centralized in one database, information retrieval system for these databases is an ongoing research problem. If the details of a particular protein are to be fetched all the three databases are to be queried. The cross-references from one database to the other that are present are to be utilized in order to extract particular information for a given protein.

Systems like SRS (Sequence Retrieval System) [22] provide query interface for these databases as well as other biological databases. But in these systems the user must explicitly select the database from which information is to be fetched. This places a burden on the user.

The databases SWISS-PROT [3], PROSITE [4] and PDB [5] contains the view of textual data. User may be interested to see this textual data either in List form or Table forms, He/she may also like to see the structural information in graphical views.

Systems that exist to view textual data in List form or Table form is SRS (Sequence Retrieval System) [22], and to view Graphical representation is the PDB Searchlite [5]. Unless one knows the PDB Id or keyword, they cannot see the structural information from the PDB Searchlite [5]. In SRS in order to have a view of structural information of protein, user has to search SWISS-PROT [3] and PROSITE [4] to get PDB entry [5] and

then he/she can view structure with help of PDB searchlite [5]. To get detailed information one has to navigate through hyperlinks provided by the search results of existing system.

In addition there is no option for refining the query. Users invariably like to narrow down the search. Refinement of query in terms of adding/removing an attribute and adding/removing/changing the condition is enquired from the user point of view.

## **1.5 Proposed Approach**

In this project we are developing query interface and data retrieval system for three protein databases SWISS-PROT, Prosite and PDB. Each of these three databases contains the different information. SWISS-PROT contains the sequence information, Prosite contains the sequence-related information and PDB contains the structural information.

### **1.5.1 Providing User Interface to Build Query**

In our system of approach the user will be provided with graphical user interface for building the query for the databases PDB [5], Prosite [4] and SWISS-PROT [3]. There are two aspects in querying databases, Query formation and condition evaluation. For query formation user will be provided with the user terms with which he/she is familiar than the actual database terms. Based on the condition built by the user the information will be retrieved.

In this approach the user need not be aware of, from which databases he/she is extracting the data. But user must be aware of what is the data present in the existing databases. A data dictionary is provided for this with all users terms available, so that user can select what he wants. After developing the query and condition, a graphical display of query is presented to user from which databases the information is to be retrieved. User also

provided with an option of saving the query and printing the results. In addition Help is also provided to know about databases and query development.

### 1.5.2 Accessing Databases Efficiently

As we discussed above, the users need not to select the databases explicitly. To make the accessing of databases efficiently, first we decompose the query into multiple sub queries, where each sub query may accesses information from particular database. This is done with the help of built in thesaurus. We create index structure for each of these databases, which makes search faster. In next step, the condition evaluation, in which condition may be specified on different databases. We have to navigate through these databases based on cross-references present in the databases in order to evaluate the conditions and retrieve the information. After collecting the data from all databases the result is collated and displayed.

### 1.5.3 Displaying Output in User Friendly Environment

The results of the query from three protein databases can be viewed in both the Textual format and Graphical format. In textual format data can be viewed either in List format or Table format. And graphical information can be displayed with a simple mouse click.

In List form the data can be displayed in file like format. The results will be displayed for the requested fields in the query. And for more analysis the user can also view the structural information for any displayed record in the results. This can be done with a simple mouse click on the required record without any extra navigation.

In the Table form the retrieved results can be displayed in the Table format. And if user wish to see the detailed view of structure information for any record, this can be accomplished with a simple button click in each record. The detailed view can be provided for each accession entry in protein database without any extra traversal through



hyperlinks. User is also provided with an option of changing from current List view to the Table view.

In addition to these options, the user is also provided with an extra option of refining query. Modifying the attribute list or changing the condition can refine queries. With this option he/she can narrow down the search to get their exact results.

The layout of this thesis is as follows. In chapter 2 we describe the protein databases and their structure. Here three protein databases i.e. SWISS-PROT, Prosite and PDB are explained. In chapter 3 we discuss the architecture of the system, the design of presentation of output of the system and what are the facilities provided to the user. In chapter 4 we talk about the manner in which the output is presented to the user and what are the facilities and functionalities that are provided and how to use them. In chapter 5 we explain the implementation details of the presentation of the output, facilities and functionalities provided to the user.

## CHAPTER 2

### OVERVIEW OF PROTEIN DATABASES

#### 2.1 Protein DataBases

Databases on proteins have grown rapidly because of research on proteins. PDB [5], SWISS-PROT [3], PROSITE [4] are three protein databases, which contain different information of proteins.

##### 2.1.1 SWISS-PROT

SWISS-PROT [3] is a protein knowledgebase established in 1986 and maintained collaboratively, since 1987, by the Department of Medical Biochemistry of the University of Geneva. The SWISS-PROT [3] protein knowledgebase consists of sequence entries. SWISS-PROT [3] is a curated protein sequence database which strives to provide a high level of annotations (such as the description of the function of a protein, its domains structure, post-transnational modifications, variants, etc.), a minimal level of redundancy and high level of integration with other databases.

Sequence entries are composed of different line-types, each with their own format. For standardization purposes the format of SWISS-PROT [3] follows as closely as possible that of the EMBL Nucleotide Sequence Database [7].

##### 2.1.2 PDB

The PDB [5], single international repository for the processing and distribution of experimentally determined three-dimensional macromolecular structure data. The PDB [5] is managed by Rutgers, The State University of New Jersey, the San Diego

Supercomputer Center at the University of California, San Diego, and the National Institute of Standards and Technology. International participants in data deposition and processing include the European Bioinformatics Institute Macromolecular Structure Database group (UK) and the Institute for Protein Research at Osaka University (Japan). There is also a worldwide network of distribution sites as listed at <http://www.rcsb.org/pdb/mirrors.html>.

### 2.1.3 PROSITE

PROSITE [4] is a database of protein families and domains. It is based on the observation that, while there is a huge number of different proteins, most of them can be grouped, on the basis of similarities in their sequences, into a limited number of families. Proteins or protein domains belonging to a particular family generally share functional attributes and are derived from a common ancestor.

It is apparent, when studying protein sequence families, that some regions have been better conserved than others during evolution. These regions are generally important for the function of a protein and/or for the maintenance of its three-dimensional structure. By analyzing the constant and variable properties of such groups of similar sequences, it is possible to derive a signature for a protein family or domain, which distinguishes its members from all other unrelated proteins. A pertinent analogy is the use of fingerprints by the police for identification purposes. A fingerprint is generally sufficient to identify a given individual. Similarly, a protein signature can be used to assign a newly sequenced protein to a specific family of proteins and thus to formulate hypotheses about its function.

PROSITE [4] currently contains patterns and profiles specific for more than a thousand protein families or domains. Each of these signatures comes with documentation providing background information on the structure and function of these proteins.

## 2.2 Formats

The different formats of Protein databases i.e., SWISS-PROT, PDB, and Prosite are discussed below.

### 2.2.1 SWISS-PROT Format

The entries in the SWISS-PROT [3] database are structured so as to be usable by human readers as well as by computer programs. The explanations, descriptions, classifications and other comments are in ordinary English. Wherever possible, symbols familiar to biochemists, proteinchemists, and molecular biologists are used. Each sequence entry is composed of lines. Different types of lines, each with their own format, are used to record the various data that make up the entry. A sample sequence entry of this swiss-prot is shown in Appendix A.

In a Swiss-Prot database entry each line begins with two-character code, which indicates the type of data contained in the line. The current line type and codes and the order, in which they appear in an entry, are shown below.

ID Identification once; Occurrence in entry: starts the entry  
AC Accession number(s) ; Occurrence in entry: Once or more  
DT Date; Occurrence in entry: Three times  
DE Description ; Occurrence in entry: Once or more  
GN Gene name(s) ; Occurrence in entry: Optional  
OS Organism species; Occurrence in entry: Once or more  
OG Organelle ; Occurrence in entry: Optional  
OC Organism classification; Occurrence in entry: Once or more  
OX Taxonomy cross-reference(s) ; Occurrence in entry: Once or more  
RN Reference number; Occurrence in entry: Once or more  
RP Reference position; Occurrence in entry: Once or more  
RC Reference comment(s) ; Occurrence in entry: Optional  
RX Reference cross-reference(s) ; Occurrence in entry: Optional

RA Reference authors; Occurrence in entry: Once or more  
 RT Reference title ; Occurrence in entry: Optional  
 RL Reference location ; Occurrence in entry: Once or more  
 CC Comments or notes; Occurrence in entry: Optional  
 DR Database cross-references; Occurrence in entry: Optional  
 KW Keywords ; Occurrence in entry: Optional  
 FT Feature table data ; Occurrence in entry: Optional  
 SQ Sequence header Once ; Occurrence in entry:  
     (blanks) sequence data ; Occurrence in entry: Once or more  
 // Termination line ; Occurrence in entry: Once; ends the entry

### 2.2.2 PDB Format

Every PDB [5] file may be broken into a number of lines terminated by an end-of-line indicator. Each line in the PDB [5] entry file consists of 80 columns. The last character in each PDB [5] entry should be an end-of-line indicator. Each line in the PDB [5] file is self-identifying. The first six columns of every line contain a record name, left justified and blank-filled. This must be an exact match to one of the stated record names.

The PDB [5] file may also be viewed as a collection of record types. Each record type consists of one or more lines. Each record type is further divided into fields. Each record type is detailed in this document. The description of each record type includes the sections Over View, Record Format, Details, Verification/Validation/Value Authority Control, Relationship to other Record Types, Example, and Known Problems.

For records that are fully described in fixed column format, columns not assigned to fields *must be left blank*. An example entry of this type is shown in Appendix A. The currently used line types, along with their respective line codes, are listed below:

CRYST1	Unit cell parameters, space group, and Z.
END	Last record in the file.

HEADER	First line of the entry, contains
PDB ID	code, classification, and date of deposition.
MASTER	Control record for bookkeeping.
ORIGXn	Transformation from orthogonal coordinates to the submitted coordinates (n = 1, 2, or 3).
SCALEn	Transformation from orthogonal coordinates to fractional crystallographic coordinates (n = 1, 2, or 3).
AUTHOR	List of contributors.
CAVEAT	Severe error indicator. Entries with this record must be used with care.
COMPND	Description of macromolecular contents of the entry.
EXPDTA	Experimental technique used for the structure determination.
KEYWDS	List of keywords describing the macromolecule.
OBSLTE	Statement that the entry has been removed from distribution and list of the ID code(s) which replaced it.
SOURCE	Biological source of macromolecules in the entry.
SPRSDE	List of entries withdrawn from release and replaced by current entry.
TITLE	Description of the experiment represented in the entry.
ANISOU	Anisotropic temperature factors.
ATOM	Atomic coordinate records for standard groups.
CISPEP	Identification of peptide residues in cis conformation.
CONNECT	Connectivity records.
DBREF	Reference to the entry in the sequence database(s).
HELIX	Identification of helical substructures.
HET	Identification of non-standard groups or residues (heterogens)
HETSYN	Synonymous compound names for heterogens.
HYDBND	Identification of hydrogen bonds.
LINK	Identification of inter-residue bonds.
MODRES	Identification of modifications to standard residues.
MTRIXn	Transformations expressing non-crystallographic symmetry (n = 1, 2, or 3). There may be multiple sets of these records.

REVDAT	Revision date and related information.
SEQADV	Identification of conflicts between PDB and the named sequence database.
SEQRES	Primary sequence of backbone residues.
SHEET	Identification of sheet substructures.
SIGATM	Standard deviations of atomic parameters.
SIGUIJ	Standard deviations of anisotropic temperature factors.
SITE	Identification of groups comprising important sites.
SLTBRG	Identification of salt bridges SSBOND Identification of disulfide bonds.
TURN	Identification of turns.
TVECT	Translation vector for infinite covalently connected structures.
FORMUL	Chemical formula of non-standard groups.
HETATM	Atomic coordinate records for heterogens.
HETNAM	Compound name of the heterogens.
ENDMDL	End-of-model record for multiple structures in a single coordinate entry.
MODEL	Specification of model number for multiple structures in a single coordinate entry.
TER	Chain terminator.
JRNL	Literature citation that defines the coordinate set.
REMARK	General remarks, some are structured and some are free form.

### 2.2.3 Prosite Format

The PROSITE [4] database is composed of two ASCII (text) files. The first file (PROSITE.DAT) [4] is a computer readable file that contains all the information necessary to programs that will scan sequence(s) with patterns and/or matrices. The second file (PROSITE.DOC) [4] contains textual information that fully documents each pattern and profile.

The entries in the database data file (PROSITE.DAT) [4] are structured so as to be usable by human readers as well as by computer programs. Each entry in the database is composed of lines. Different types of lines, each with its own format, are used to record

the various types of data, which make up the entry. The general structure of a line is the following:

Characters	Content
-----	-----
1 to 2	Two-character line code. Indicates the type of information Contained in the line.
3 to 5	Blank
6 up to 128	Data

An Example of a pattern entry is shown in Appendix A. The currently used line types, along with their respective line codes, are listed below:

ID	Identification	(Begins each entry; 1 per entry)
AC	Accession number	(1 per entry)
DT	Date	(1 per entry)
DE	Short description	(1 per entry)
PA	Pattern	(>=0 per entry)
MA	Matrix/profile	(>=0 per entry)
RU	Rule	(>=0 per entry)
NR	Numerical results	(>=0 per entry)
CC	Comments	(>=0 per entry)
DR	Cross-references to SWISS-PROT	(>=0 per entry)
3D	Cross-references to PDB	(>=0 per entry)
DO	Pointer to the documentation file	(1 per entry)
//	Termination line	(Ends each entry; 1 per entry)



## CHAPTER 3

### SYSTEM ARCHITECTURE AND DESIGN

#### 3.1 Over View of the Architecture

The system is based on two-tier architecture. It is shown in Fig 3.1 given below.

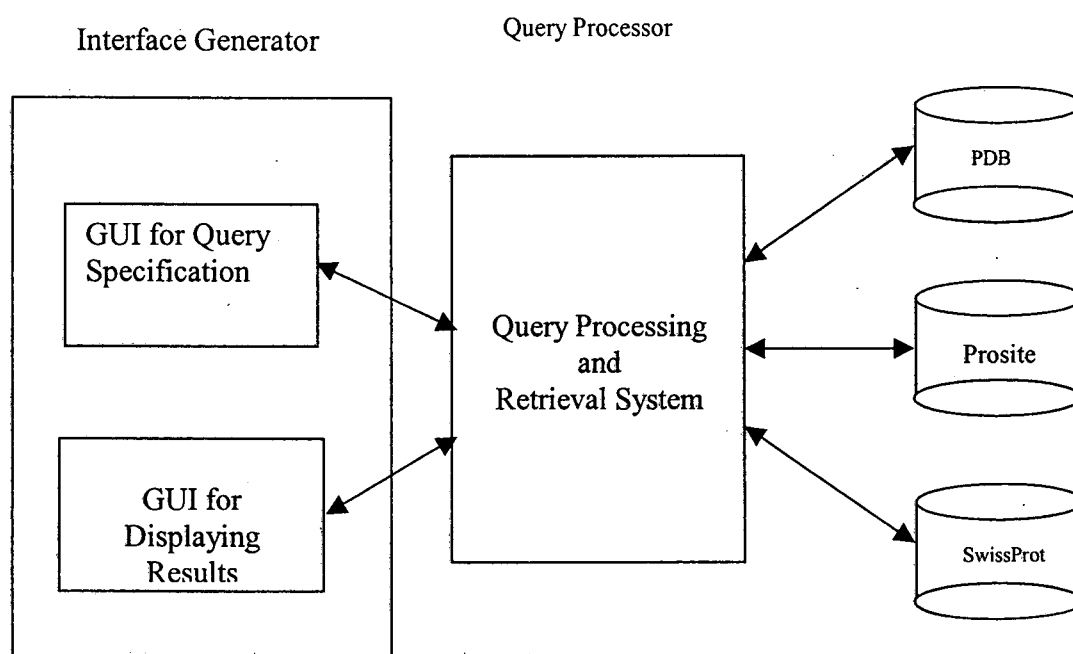


Fig: 3.1 Two-tier Architecture

The modules in the Fig: 3.1 above consist of Interface generator and query processing retrieval systems. The interface generator consists of GUI for query specification and GUI for Displaying results, which are discussed below.

*GUI for query specification* module provides GUI interface for the user to submit the query to retrieve the required data from protein databases. This module provides user-friendly environment to build the query. Here the user needs not to remember all database terminology, rather he can use simply user terminology to build the query. To build the query it provides the logical operators like less than (<), greater than (>), less than or equal to (<=), greater than or equal to (>=), AND, and OR operators. The submitted query in user terminology is mapped into actual database terminology by this module. In addition to facility of building the query it also provides the option of saving the user submitted query. The user submitted query would be given to the query processing and retrieval system module to get the required result.

*Query processing and retrieval system* module takes the user-submitted query and it will be divided into subtasks. And this module identifies, to which databases each sub query belongs. Based on the user request it will fetch the data from the corresponding databases through net connectivity. The databases SWISS-PROT, PDB, and Prosite updated day to day. These updating will be reflected in the retrieved results. These collected results from different databases SWISS-PROT, PDB, and Prosite will be collated and given to the Displaying results module to present output to the user.

*GUI for Displaying results* module presents the output in users requested form. The results can be presented to the user can be either in Textual format or Graphical format. The textual information can be displayed either in the List form or in the Table format. The structural information can be displayed in a 3D window. It also provides the facility of switching from List view to Table view and vice-versa. In addition to the display of results it also provides facility for Refinement of query. This module is discussed in detail in following sections. We are concerned with the design of the module GUI for displaying results.

### 3.2 Design

Fig 3.2 below shows the structure chart for the Displaying results, which consists of three sub modules in it. These are discussed in detail below. In the structure chart below 'T' stands for Textual information, 'PI' stands for PDB identification number, and 'RQ' stands for query for refinement.

*Display of Textual information* module takes the textual information from the retrieved results to display either in List view or in Table view. In the List view the data is displayed in File like view and in Table view the displayed in a Table form for the user convenience. This also provides facility of switching to the graphical view and to see the structural information. This sub module is discussed in detail in following sections.

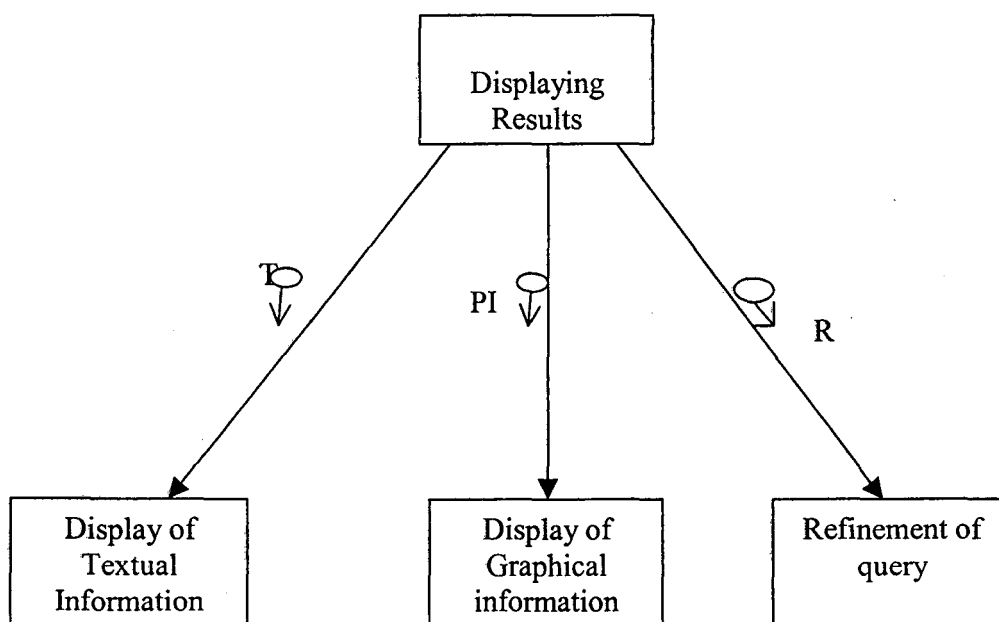


Fig: 3.2 Structure chart for Displaying results

*Display of Graphical information* module provides the facility to explore the structural information in Cortona window. This window contains 3D window and toolbars to navigate through the 3D window. Here we support the PDB searchlite to display the structural information. This sub module is discussed in detail in following sections.



TH-10247

*Refinement of query* module provides the facility of refinement of query, which is not provided in many biological data retrieval systems. By adding or removing the attribute and/or by modifying the condition the search can be enhanced or narrow down the search to retrieve the required results. This module is discussed in detail in following sections.

### 3.2.1 Structure chart for Display of Textual information

Fig: 3.3 below represent the structure chart for Display of structural information. In this diagram 'T' represents the textual information. The textual information can be viewed either in the Table view or List view. In either of these views the user is provided with the many user-friendly options. These are discussed below in detail.

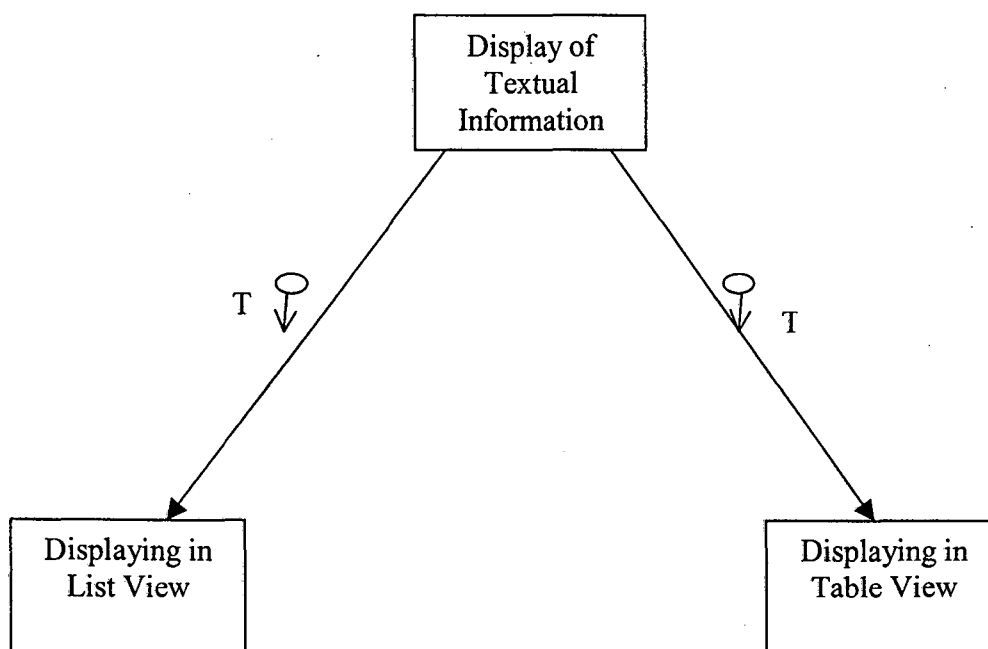


Fig: 3.3 Structures chart for Display of Textual information

*Displaying in List View* displays the textual information (T) in file like view. In this view it displays the fetched information in record wise. In each record the user requested values would be shown against the requested fields. In addition to display of results, it also provides facilities for switching from list view to the Table view, to switch to refining query option, and also to switch to view the structure information. It also provides the support for saving of submitted query and support for printing the results of

this view. The saving and printing options are discussed in GUI interface for Biological databases [23].

*Displaying in Table View* displays the textual information (T) in Table format. In this view it presents the user requested fields in column headers and the values of those fields in subsequent rows, one record per row. In addition to display of results it provides an option for changing from the List view to Table view. It also provides options to switch to refinement of query and to switch to view structural information. It also supports the facilities of saving the submitted query and printing the results of this view.

Both the above textual views provides options in menus and Toolbars for user friendly saving, printing, refinement, go home, exit, and change view options.

### 3.2.2 Structure chart for Display of graphical information

Fig: 3.4 below represents structure chart for Display of graphical information. In this diagram below 'PI' stands for PDB identification number and 'G' stands for structural information.

From the retrieved results, this module gets a PDB id information for the user selected record. As user can select with a simple click on the hypertext to view the structure information for the requested record from List view, or, he/she can click a dark bulb icon against each record in a Table View.

*Retrieving from PDB searchlite* module collects PDB id information and is submitted to PDB searchlite by connecting to PDB site to retrieve this information. This module supports to retrieve this information and to display in Cortona window.

*Displaying Structure Information in 3D-window* module, which supports 3D window, and Toolbars provided by the PDB searchlite. The retrieved structural information can be

displayed in 3D world to navigate through it. This module supports all navigation tools to explore 3D world.

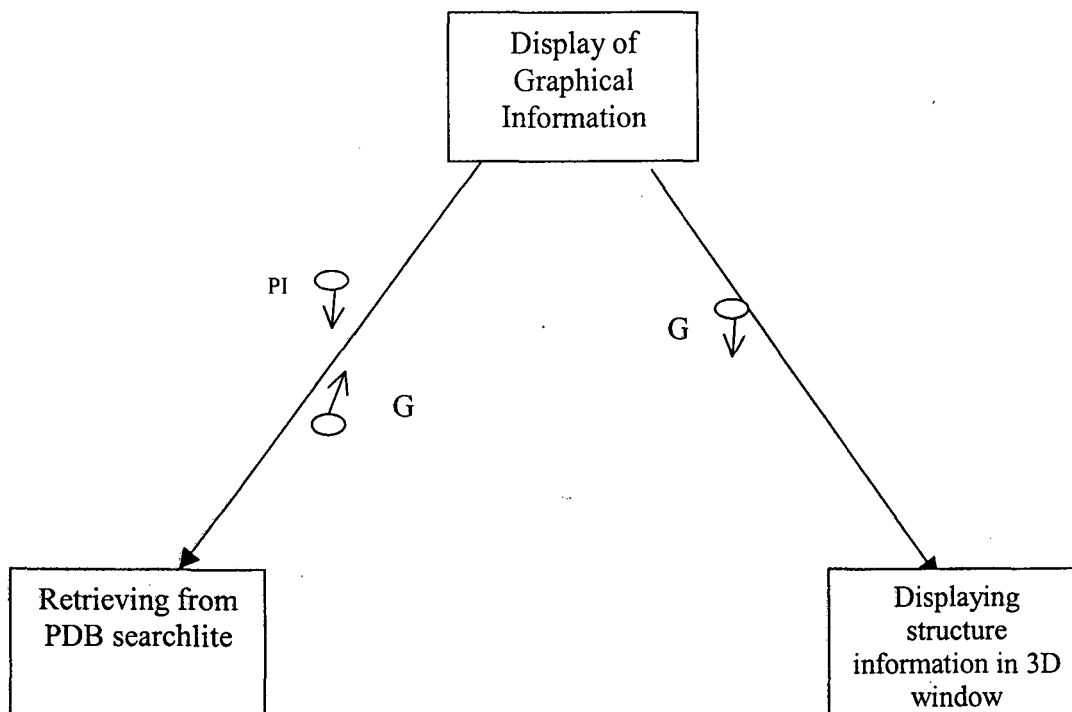


Fig: 3.4 Structures chart for Display of Graphical information

### 3.2.3 Structure chart for Refinement of query

Fig: 3.5 represent the structure chart for refinement of query. In this figure the 'RQ' stands for query for refinement, and 'MQ' represents the Modified query. The refinement of query provides the facility to refine the submitted query to retrieve the refined data. This module consists of two sub modules, which are discussed below.

*Query modification* module provides the facility to modify the all ready submitted query. The user can save the submitted query with a save option in List view and Table view. If user wishes to modify already submitted query by adding/removing the attributes and/or by changing the conditions to enhance or narrow down the search to get his/her exact results, this can be provided by this module. This modified query can be submitted to the

*Query processing and retrieval system* to retrieve the results. This is discussed in detail in *Accessing Multiple Biological Databases* [24].

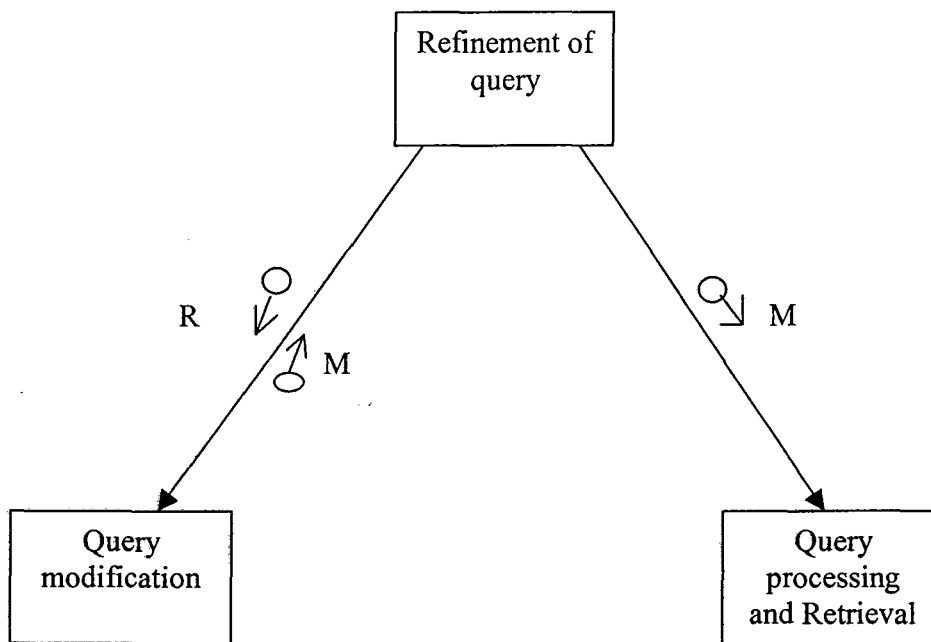


Fig: 3.5 Structures chart for Refinement of query

## CHAPTER 4

### RESULTS PRESENTATION

The retrieved results are displayed either in the Textual format or in Graphical view with user friendly in nature. The Textual view can be either in Table format or in List format. In addition to the displaying of results these views also provides many user-friendly options.

These Textual views provide a Menu bar, which contains the menus, Query Output, View, Go, Refine, Tools, and Help. Each of these menus provides many options as their Menu Items. User is provided with a facility of selecting the menu item using the keyboard Keys in addition to mouse click. And when he/she is using the mouse, the mouse pointer is also set with Tool Tip Text, which is very user friendly. When he/she using the keyboard short cuts to select menu item, he/she can also use up/down keys to switch within menu.

#### 4.1 List View

This view provides file like view. In this view the results are displayed of query specific. All these records can be viewed by scrolling the screen. The results for each of these records will be displayed opposite to their corresponding requested attributes. The very first attribute in each record is the Structure. Opposite to this Structure attribute a text containing the data "click to view the structure" is written, with a mouse click on this text it displays the Structural information for the requested record.



For example for the following query:

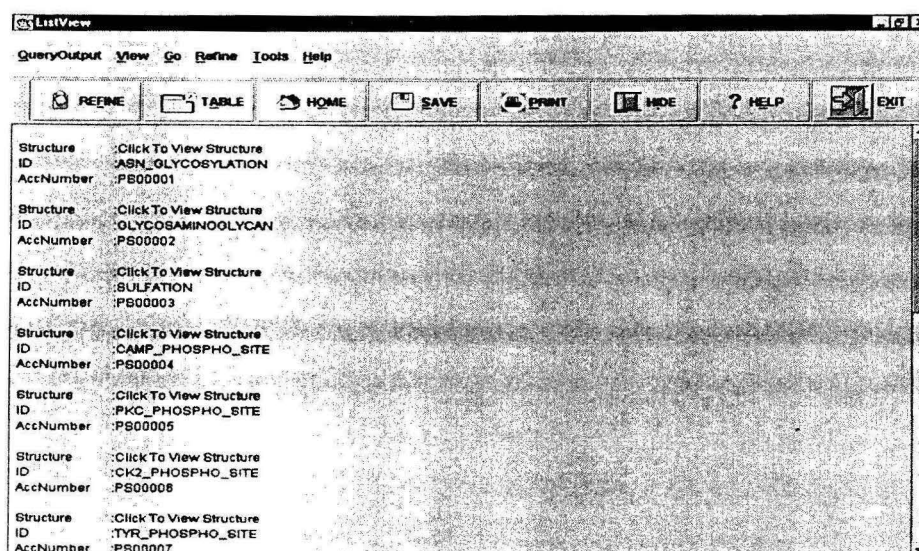
Select

Prosite\_identification, pattern, Genome, Organism Species, Function\_classification,  
entered\_date

Where

Accession Number = ps00001 and swissprot Accession = p01375 and entry authors  
conatins "C.C.F.BLAKE"

The result displays in the following format for the List View.



In addition to the display of results it also includes some additional functions such as Refining of query, switching to Table view. These functions are discussed below in detail.

## 4.2 Table View

In this view it displays the results in user friendly table view. The user requested field names in the query are displayed in the column header and all the query-matched records are placed in the subsequent rows. Only one record per row is displayed. And always the content of first column header displayed is Structure, and subsequent headers are

containing the user-requested fields. Under structure column in each row of results, it's provided with a bulb icon, by pressing with the simple mouse click on bulb icon the user can view the structural information of that record in a separate window.

For example for the following submitted query

Select

**Prosite\_identification, pattern, Genome, Organism Species, Function\_classification, entered\_date**

Where

**Accession Number = ps00001 and swissprot Accession = p01375 and entry authors conatins "C.C.F.BLAKE"**

The result displays in the following format for the Table View.

The screenshot shows a window titled "TableView" with a menu bar (QueryOutput, View, Go, Refine, Tools, Help) and a toolbar with buttons for REFINE, List, HOME, SAVE, PRINT, Hide, Help, and EXIT. Below the toolbar is a table with the following data:

Structure	ID	AccNumber	Description
⚡	ASN_GLYCOSYLATION	PS00001	N-glycosylation site
⚡	GLYCOSAMINOGLYCAN	PS00002	Glycosaminoglycan attachment site
⚡	SULFATION	PS00003	Tyrosine sulfation site
⚡	CAMP_PHOSPHO_SITE	PS00004	cAMP- and cGMP-dependent protein kinase phosphorylation s...
⚡	PKC_PHOSPHO_SITE	PS00005	Protein kinase C phosphorylation site
⚡	CK2_PHOSPHO_SITE	PS00006	Casein kinase II phosphorylation site
⚡	TYR_PHOSPHO_SITE	PS00007	Tyrosine kinase phosphorylation site
⚡	MYRISTYL	PS00008	N-myristoylation site
⚡	AMIDATION	PS00009	Amidation site
⚡	ASX_HYDROXYL	PS00010	Aspartic acid and asparagine hydroxylation site
⚡	GLU_CARBOXYLATION	PS00011	Vitamin K-dependent carboxylation domain
⚡	PHOSPHOPANTETHEINE	PS00012	Phosphopantetheine attachment site

In addition to the display of results it also includes some additional functions such as Refining of query, switching to List view. These functions are discussed below in detail.

### 4.3 Functions provided by the Textual view of data

In Textual view of both List View and Table View Inaddition to the views, they are provided with some other functions. Both of these views conatins menu bar and toolbars, which are assigned with some functionality's. Those functions are discussed below.

**QueryOutput** menu contains two options, **Save** and **Print**. The Save option saves the submitted query and print option prints the results of this view. These Save and Print aspects are discussed in detail in GUI Interface to Biological Databases [23].

The menu **View** contains two options, **ListForm** and **TableForm**. If user selects the ListForm option from List View it simply displays a dialogue box containing the message "Current View is List View of Data". Similarly if user selects a Table form option from the Table form then it displays a dialogue box containing the message "Current View is List View of Data". Instead, if he/she selects TableForm from List form he can switch to Table form and vice versa. This View option provides the facility of switching from List View to Table view and Table view to List view.

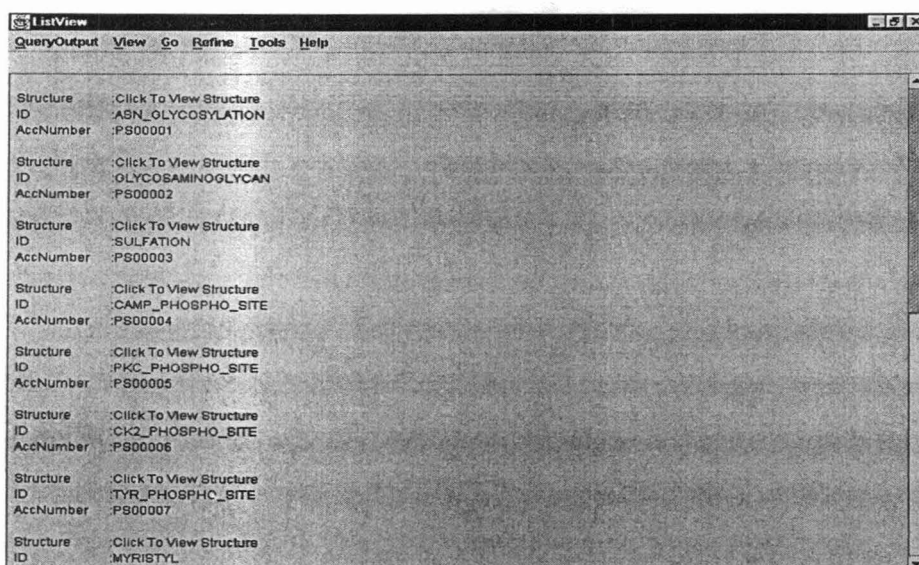
The menu **Go** is provided with options of **Home** and **Exit**. By selecting the Home option the user can switch to the Query screen. The aspect of Query screen is discussed in GUI Interface to Biological Databases [23]. If user selects an option of Exit then he/she can exit the system. This option is provided in both the Table view and List view.

The menu **Refine** is provided with menu item **RefineQuery**. If user selects this option then he is provided with a facility of refining query. This refining can be done by adding/removing attributes or changing the condition. This options makes the narrow down of the search and it may matches the exact results of the user requirements.

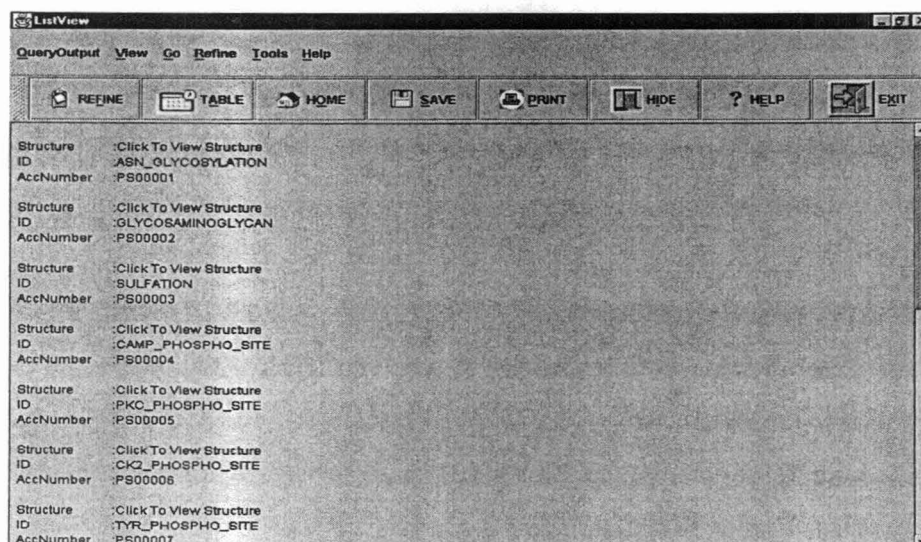
The menu **Tools** contains two options **HideTools** and **ViewTools**. With the selection of ViewTools option, it provides a facility of movable toolbar, which contains the Refine

button, Table, Home, Save, Print, Hide, Help and Exit buttons. This movable toolbar can be placed anywhere on the screen according to user's convenience. This movable toolbar can be hidden with the selection of HideTools option or by clicking the Hide button on toolbar. This toolbar option is provided both in Table View and List View.

The List View without toolbar is shown below.



The List view with tool bar is shown below.



The **Help** menu contains **Help** and **About** menu items. By clicking the Help menu item the user is displayed with Help contents and with selection of About, it displays about this querying and search retrieval system. These two options are discussed in the GUI interface to Biological databases [23].

#### 4.4 Structural View

The Structural view can be displayed by clicking the hyper text in List view or image icon displayed in Table view against the specified record. In this we support the view provided by the PDB Searchlite. The structure information can be displayed in Cortona window, which is used to explore 3D worlds. Cortona window consists of two parts, Toolbars and a 3D window.

Toolbars contain Vertical toolbar and Horizontal toolbar. The *vertical toolbar*, which contains buttons used to specify navigation type in a world. The *horizontal toolbar*, which contains buttons with predefined actions to change the position in a world. The *3D window*, which shows VRML world. There is also *pop-up menu*, which can be accessed by pressing the right mouse button while the pointer is over toolbars or 3D window.

##### 4.4.1 Navigation through 3D world

Moving through a 3D space is similar to moving a video camera. The movements in the world continually position and orient that camera. Using the camera controls on the vertical toolbar to move the camera through the 3D Space. This concept assumes that there is a real person viewing and interacting with the VRML world. This section describes the mechanisms that Cortona provides for navigating in a three-dimensional space.

##### *Moving around: Walk, Fly, and Study*

There are three main navigation modes that Cortona offers: WALK, FLY, and STUDY. One can switch the navigation mode by clicking buttons on the vertical toolbar. Each

navigation mode may have several options: PLAN, PAN, TURN, and ROLL. The combination of navigation mode and its option determines the possible camera motion and its orientation. Some worlds don't allow the user to use navigation controls, but they may provide on-screen cues to navigation. One can navigate with the mouse, the keyboard, or both mouse and keyboard.

### *Navigation around 3D world*

The *distance* that the user drags the mouse determines the *speed* with which the camera moves. If he/she stop moving the mouse, the camera will continue moving until they release the mouse button. To accelerate the camera's movement or rotation, one has to press SHIFT, CTRL, or SHIFT + CTRL.



By using WALK and PLAN to the camera can move in a Horizontal plane of 3D world. By holding the left mouse click pressed and by dragging the mouse on the 3D world the distance can be moved. If mouse is dragged FORWARD then the camera moves closer, if it dragged BACKWARD then the camera moves further, if it dragged RIGHT then the camera turn to the right, if it dragged LEFT then camera turns to the left.



By using WALK and PAN to the camera can move left or right in a Horizontal plane of 3D world. By holding the left mouse click pressed and by dragging the mouse on the 3D world the distance can be moved. If mouse is dragged FORWARD then the camera moves closer, if it dragged BACKWARD then the camera moves further, if it dragged RIGHT then the camera moves to the right, if it dragged LEFT then camera moves to the left.



By using WALK and TURN to change the angle of the camera in the 3D world. By holding the left mouse click pressed and by dragging the mouse on the 3D world the distance can be moved. If mouse is dragged FORWARD then the image turns upward, if it dragged BACKWARD then the image turns downward, if it dragged RIGHT then the image turn to the right, if it dragged LEFT then image turn to the left.



By using FLY and PLAN to move the camera, left or right in the 3D world. By holding the left mouse click pressed and by dragging the mouse on the 3D world the distance can be moved. If mouse is dragged FORWARD then it moves the camera forward towards its longitudinal access, if it dragged BACKWARD then moves the camera backward, if it dragged RIGHT then turn the camera to the right around it's vertical access, if it dragged LEFT then turn the camera to the left around it's vertical access.



By using FLY and PAN to move the camera within a vertical plane. By holding the left mouse click pressed and by dragging the mouse on the 3D world the distance can be moved. If mouse is dragged FORWARD then it moves the camera up, if it dragged BACKWARD then moves the camera down, if it dragged RIGHT then moves the camera to the right, if it dragged LEFT then moves the camera to the left.



By using FLY and TURN to turn the camera in the 3D world. By holding the left mouse click pressed and by dragging the mouse on the 3D world the distance can be moved. If mouse is dragged FORWARD then it turns the camera upward around its horizontal access, if it dragged BACKWARD then it turns the camera downward around its horizontal access, if it dragged RIGHT then turn the camera to the right around it's vertical access, if it dragged LEFT then turn the camera to the left around it's vertical access.



By using FLY and ROLL to incline the camera in the 3D world. By holding the left mouse click pressed and by dragging the mouse on the 3D world the distance can be moved. If mouse is dragged RIGHT then incline to the left, if it dragged LEFT then incline to the right.



By using STUDY and PLAN to examine the object from various angles in the 3D world. By holding the left mouse click pressed and by dragging the mouse on the 3D world the distance can be moved. If mouse is dragged FORWARD then the camera moves forward, if it dragged BACKWARD then the camera moves backward, if it dragged LEFT or RIGHT then the camera moves around the central point in the 3D scene.





By using STUDY+TURN to examine an object from various angles. Forward, Backward, Right, and Left - move the camera around the central point which is defined by the center of bounding box of the geometry in the 3D scene.



By using STUDY+ROLL to incline the camera around the central point, which is defined, by the center of bounding box of the geometry in the 3D scene. If mouse is dragged RIGHT then camera incline to the left, if it dragged LEFT then camera incline to the right.



With use of GOTO to move close to object in a world. If user Selects GOTO in the toolbar and then click on an object in the world, Then he/she will move directly to it.

### *Restore, Fit, and Align*

Cortona provides three mechanisms that can help to re-orient a camera if user have lost his/her way in a world. Unlike the navigation tools, these buttons invoke predefined actions that take place as user clicks on them.



By clicking the RESTORE, it automatically returns to the loaded world's original active viewpoint.



By clicking the FIT, user can see the view fully visible in the Cortona 3D window.

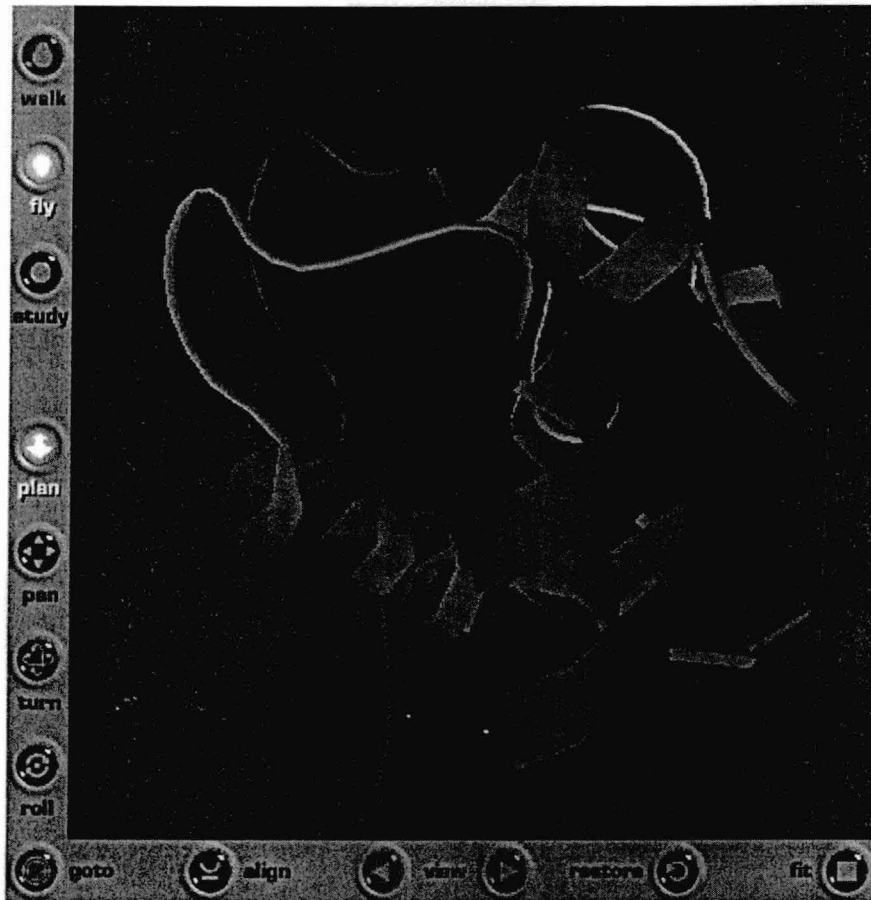


The ALIGN can be used to position the camera's horizontal and longitudinal axes parallel to the scene horizontal plane.



View is used to switch between predefined viewpoints in 3D world. But here we are displaying by default overall viewpoints. So, here view option is disabled and only empty message will be visible.

For example the structure information of protein structure 8LYZ is shown below.



#### 4.5 Refining Query

Users may interest to widen or narrow down the search. These widen or narrow down of search is to retrieve and to match the exact requirement of the user specifications. Refining of query in terms of adding or removing an attribute and modifying the condition is enquired from the user point of view. Modifying the attributes of saved query can do this refining of query. This facility of refining query can be provided from both the List view and Table view.

For example in the above given query, we can add some condition or we can add/remove some attributes to result in widen the search.

Select

**Prosite\_identification, pattern, Genome, Organism Species, Function\_classification,  
entered\_date, journal**

**Where**

**Accession Number = ps00001 and swissprot Accession = p01375 and entry authors  
contains "C.C.F.BLAKE"**

In our example here to widen the search we added **journal** attribute in addition to the  
existing attributes in the query to widen the search through refining.

# CHAPTER 5

## IMPLEMENTATION

This chapter explains the implementation details of the output presentation to the user, and implementation of the facilities provided for refining of query. In our implementation we used Java to provide a user-friendly environment, which also having many features. Java is created by Sun Micro Systems by adapting features of C++ and removing complexities like pointers in it.

The main reason in choosing Java is built-in support for GUI. Java consists of advanced swing package which provides very complex structures like table structure views in very easy way to use. It provides net connectivity to access and retrieve the data, also provides support for the images to retrieve very easily form net access. It provides iconified buttons and many widgets for user-friendly to use. This also provides good look and feel of Java components. Apart from these the execution time of the programs is very less on this platform.

### 5.1 Display of Textual Information

The presentation of output can be viewed either in textual format or graphical format, as discussed earlier. The textual data can be displayed either in List form or Table form. The implementations of these are discussed in detail in following sections.

#### 5.1.1 Table View

As discussed in Structure chart for Display of Textual information [section 3.2.1] in Table view, this aspect of the module retrieves data for requested query would be collected from the query processing and retrieving module. And this data will be collected into the data structures Column\_names (array) and names (2D array). This data will be displayed in the Table form. To represent the data in Table form it makes use of

the classes provided in java.swing.table package. In addition to provide the user-friendly tools such as menus, menu items, buttons, scrollbar, JFrame options it makes use of the widgets provided by the java.swing package.

Swing tables display rows and columns of data and are Swing's most complex component. In fact, Swing provides separate package--swing.table--that contains table support interfaces and classes. Swing tables composed of a table header that displays column headings, table columns, and cell values. Tables also contain rows and table cells, although rows and cells are not objects as are headers and columns. Tables are capable of supporting a number of selection modes including row, column, and cell selection. An object that implements the *TableCellRenderer* interface and are edited by an object that implements the *CellEditor* interface renders table cell values.

To implement the Table view, we used JTable class provided by the java.swing.table package. The different JTable constructors that are provided by this package are

- public JTable(Object[][] rowData, Object[] columnNames)
- public JTable(Vector rowData, Vector columnNames)
- public JTable(TableModel, TableColumnModel)
- public JTable(TableModel)

Cell values can be specified at construction time as data--in the form of either object arrays or vectors--or as table and/or table column models.

In this implementation we used the constructor

```
public JTable(Object[][] rowData, Object[] columnNames)
```

If user is interested to see the structural information from the table view, he/she can do it by simple click on the dark bulb displayed below the Structure column header against each record. When he clicks the dark bulb, it changes the icon to bright bulb in user selected row, and it displays the structure information. Extending *DefaultTableCellRenderer* and *AbstractCellEditor* can do the change of icon in the selected cell. These classes are provided by the package java.swing.table.

The bulbs in the left column are rendered by an instance of `BulbRenderer`. If a row is selected, the Boolean value representing the bulb is set to false, and the bulb is rendered with brighter icon. Conversely, if a row is not selected, the value is set to false, and the bulb is rendered with a darker icon. `BulbRenderer` extends `DefaultTableCellRenderer`. The `getTableCellRendererComponent` method sets the label's icon depending upon whether the cell's value is true or false.

The prototype of the method `getTableCellRendererComponent` is as shown here.

```
public Component getTableCellRendererComponent(JTable, Object value, boolean
isSelected, boolean hasFocus, int row, int column)
```

The class `DefaultTableCellRenderer` provides only constructor as shown below.

```
Public DefaultTableCellRenderer( )
```

Dark bulbs in the left column are switched to bright bulbs when the mouse is pressed in a cell. The switching is performed by an instance of `BulbEditor` in our implementation. `BulbEditors` always return a renderer that displays a bright version of the bulb by indicating too the renderer's `getTableCellRendererComponent` method that the cell is selected. As a result, anytime a bulb editor is activated, it immediately displays a bright bulb regardless of the previous state of the bulb. The implementation of `BulbEditor` class, which returns an instance of `BulbRenderer` from its `getTableCellEditorComponent` method. The prototype of `getTableCellEditorComponent` is as shown below.

```
public Component getTableCellEditorComponent(JTable, Object, boolean, int, int)
```

The implementation of the structural view is discussed in detail below.

### 5.1.2 List View

As discussed in Structure chart for Display of Textual information [section 3.2.1] in List view, the data can be displayed in File like view. In this aspect the retrieved data for requested query would be collected from the query processing and retrieving module. And this data will be collected into the data structures `Column_names` (array) and `names` (2D array). This collected data would be displayed in the `JTextArea` provided by the `javax.swing` package. The prototype in our implementation that we used is `JTextArea ()`.

When user wishes to switch from List view to Structural view to see a particular structure, he/she can do this with a simple mouse click the "click to structure view", which is highlighted in each record. To highlight the text written against the structure attribute, we use implement the interface *Highlighter.HighlightPainter* defined in `javax.swing.text` package, and we use the function *paint* defined in the interface, whose prototype is shown below.

```
void paint(Graphics g, int p0, int p1, Shape bounds, JTextComponent c)
```

And in both the above views we used the `javax.swing` widgets to provide user-friendly options to user. These Textual views provide a Menu bar, which contains the menus, Query Output, View, Go, Refine, Tools, and Help. Each of these menus provides many options as their Menu Items and also provided iconified buttons in toolbar to be an user friendly. To implement these it uses the widgets like buttons, Menu, MenuItem, JPanel, and Jframes derived in `javax.swing` package to give good look and feel. And to provide the actions to these, when these widgets generates events these are made get register with `ActionListeners`. The prototypes for these widgets are shown below.

We used `JButton(String text, Icon icon)` for iconified buttons in our implementation.

The `JMenu` is available in `javax.swing` package and we used `JMenu(String s)` in our implementation to facilitate different menus as discussed above.



The prototype for JMenuBar available in javax.swing package is JMenuBar() And we used the same in our implementation. The JMenuItem available in javax.swing package, and we used JMenuItem(String text) in our implementation.

The JScrollPane available in javax.swing package that is used to scroll down the message in displaying window. We used JScrollPane(Component view) in our implementation.

And We used JPanel(LayoutManager layout) in our implementation.

And to provide the functionality we made registered these widgets to the corresponding widget listeners. For Buttons made registered with the ActionListener(new ActionEvent).

## **5.2 Display of Graphical Information**

As discussed in Structure chart for Display of Graphical information [section 3.2.2], this aspect of the module presents the structural information. In this implementation we support for the structural information and Cortona window retrieved from PDB searchlite. The structural information can be displayed in VRML (Virtual Reality Modeling Language) 3D world provided by the PDB searchlite.

To retrieve this information from PDB site we use java.net package. Here we use the URL class defined in java.net package. We use URL(String spec) in our implementation. And we use the function getContent( ) defined in the URL class to get the content from the URL specified. The prototype for this method is Object getContent()

To display the content on to the browser we use the calss JApplet and methods getAppletContext( ) and getImage( ) methods. We also use BufferedReader calss and the methods InputStreamReader (url.openStream( )).

We also use the package java.net.MalformedURLException to handle the exceptions while retrieving the information from the URL.

### **5.3 Refinement of Query**

As we discussed in the module Structure chart for Refinement of query [section 3.2.3], this module provides a facility of refining the query. It allows the user to resubmit the query with change in condition and/or add or remove the attributes to narrow down the search. To refine the query, it provides the GUI interface with saved query retrieval, where user can make the changes. The GUI interface is discussed in detail in The GUI interface for Biological databases [23]. The modified query can be submitted to the query and processing retrieval system, where this query is parsed and retrieves the data of user requirement. The Query processing and Retrieval system is discussed in detail in Accessing Multiple Biological Protein Databases [24].

## CONCLUSION

In this thesis we developed a user-friendly query interface for protein biological databases. The databases that we have considered for this system is SWISS-PROT, which contains the protein sequence information, Prosite which contains sequence related information, and PDB contains the structural information. Here the user can build the query in user terminology rather than actual database terms. In this system the user need not be aware of what information that exists in each database. Even user need not to specify explicitly the databases while submitting query. After submission of query, it is divided into many sub queries and is submitted to the corresponding web databases for required information. And these retrieved results are collated and presented to the user. This information can be displayed in user specified view. If user wishes to enhance or narrow down the search he/she can also do that by refining the query.

The resultant information consists of both the textual information and graphical structure information. The textual information can be displayed either in List view or Table view. And the Graphical structural information can be explored in VRML 3D world. No existing data retrieval system provides this facility to view both the textual view and structural view. With the existing systems for example SRS is a sequence retrieval system to view the textual information, and if user wish to see the structural view that he/she has to pipe these results to the PDB searchlite, which is a burden for the user.

The textual information that is displayed in List view is like a File like view. In this view the user is provided with the resultant values against requested attributes. In addition he/she also provided with an option of switching from List view to Table view. If the user is interested to view the structural information for any requested entry, that can also be done by clicking the structure field in each record.

The textual information that can be displayed in Table view with requested fields as column headers and the resultant values in subsequent rows. Each resultant record will be displayed in a row. Here it is provided with an option of changing the view from Table view to List view. If user wishes to see the structural information for any particular record, that can also be done with a simple mouse click. In both the List and Table views the user is provided with many user-friendly options like to save the query, print the results, exit the system, refinement of query et cetera.

The structural information can be displayed in VRML 3D world, in which the user can interact with the structure information. He is provided with many tools to navigate through the 3D world to explore it. Here we retrieve the user required structure information from PDB searchlite and we support the navigation environment that is provided by this site.

In addition to the above views we provide the user with a facility of refining the query. With refinement of query he/she can enhance or narrow down the search. In refining of query the user can change the attributes and/or he can add or remove or modify the condition in already submitted query.

In this thesis we have considered only 3 protein biological databases Swiss-Prot, Prosite and PDB, but it has lot of scope to enhance it. We can extend this for Nucleotide sequences, genome databases. This also has scope for extending it, for animal and plants biological databases.

## REFERENCES

1. European Bioinformatics Institute(EBI ) <http://www.ebi.ac.uk/Information/index.html>
2. Swiss Institute of Bioinformatics <http://www.isb-sib.ch/>
3. Swiss-prot [http:// www. expasy. ch/ sprot/](http://www.expasy.ch/sprot/)
4. Prosite [http:// www. expasy. ch/ prosite/](http://www.expasy.ch/prosite/)
5. PDB [http:// www. rcsb. org/ pdb/](http://www.rcsb.org/pdb/)
6. San Diego Super Computer Centre <http://www.sdsc.edu/>
7. EMBL [http:// www. ebi. ac. uk/ embl/](http://www.ebi.ac.uk/embl/)
8. Genbank [http:// www. ncbi. nlm. nhi. gov/ Genbank](http://www.ncbi.nlm.nih.gov/Genbank)
9. cDNA <http://www.cbc.umn.edu/ResearchProjects/Arabidopsis/>
10. EPD (Eukaryotic Promoter Database) [http:// www. epd. isb- sib. ch/](http://www.epd.isb-sib.ch/)
11. PIR [http:// pir. georgetown. Edu](http://pir.georgetown.edu)
12. Rebase <http://rebase.neb.com/rebase/rebase.html>
13. HSC-2DPAGE 2-DE Gel Protein Databases at Harefield  
<http://www.harefield.nthames.nhs.uk/nhli/protein/>
14. Molecular Modelling databases  
<http://www.ncbi.nlm.nih.gov/Structure/MMDB/mmdb.shtml>
15. Pubmed [http:// www. ncbi. nlm. nih. gov/ PubMed/](http://www.ncbi.nlm.nih.gov/PubMed/)

16. GDB [http:// www. gdb. Org](http://www.gdb.Org)
17. Human genome database <http://www.ncbi.nlm.nih.gov/genome/guide/human/>
18. Genome Sequence databases  
<http://inn.weizmann.ac.il/hg3m/databases/sequence.html>
19. DDBJ <http://www.ddbj.nig.ac.jp/E-mail/homology.html>
20. AtDB <http://www.arabidopsis.org/>
21. NCBI <http://www.ncbi.nlm.nih.gov/>
22. SRS <http://srs.embl-heidelberg.de:8000/srs5/>
23. GUI Interface for Biological Databases
24. Accessing Multiple Biological Protein Databases
25. A Molecular Biology Database Digest, Francois Bry and Peer Kröger
26. TAMBIS - Transparent Access to Multiple Bioinformatics Information Sources, Patricia G. Baker a, Andy Brass a, Sean Bechhofer b, Carole Goble b, Norman Paton b, Robert Stevens b.
27. Heterogeneous Data and Algorithm Integration in Bioinformatics, Barbara Eckman, Julia Rice, William Swope
28. Overview Of Selected Molecular Biological Databases ,Karen D.Rayal and Terry Gaasterland
29. QUICK:Graphical User Interface to Multiple Databases,Wang Chiew Tan, Ke Wang, Limsoon Wong
30. A Strategy for Database Interoperation, Peter D.Carp

# APPENDIX A

The sample entries for SWISS\_PROT, PDB, and Prosite are shown below.

## 1. A Sample entry for SWISS-PROT

A sample sequence entry is shown below:

ID GRAA\_HUMAN STANDARD; PRT; 262 AA.  
AC P12544;  
DT 01-OCT-1989 (Rel. 12, Created)  
DT 01-OCT-1989 (Rel. 12, Last sequence update)  
DT 16-OCT-2001 (Rel. 40, Last annotation update)  
DE Granzyme A precursor (EC 3.4.21.78) (Cytotoxic T-lymphocyte proteinase  
DE 1) (Hanukkah factor) (H factor) (HF) (Granzyme 1) (CTL tryptase)  
DE (Fragmentin 1).  
GN GZMA OR CTLA3 OR HFSP.  
OS Homo sapiens (Human).  
OC Eukaryota; Metazoa; Chordata; Craniata; Vertebrata; Euteleostomi;  
OC Mammalia; Eutheria; Primates; Catarrhini; Hominidae; Homo.  
OX NCBI\_TaxID=9606;  
RN [1]  
RP SEQUENCE FROM N.A.  
RC TISSUE=T-cell;  
RX MEDLINE=88125000; PubMed=3257574;  
RA Gershenfeld H.K., Hershberger R.J., Shows T.B., Weissman I.L.;  
RT "Cloning and chromosomal assignment of a human cDNA encoding a T  
RT cell- and natural killer cell-specific trypsin-like serine  
RT protease.";  
RL Proc. Natl. Acad. Sci. U.S.A. 85:1184-1188(1988).  
RN [2]  
RP SEQUENCE OF 29-53.  
RX MEDLINE=88330824; PubMed=3047119;  
RA Poe M., Bennett C.D., Biddison W.E., Blake J.T., Norton G.P.,  
RA Rodkey J.A., Sigal N.H., Turner R.V., Wu J.K., Zweerink H.J.;  
RT "Human cytotoxic lymphocyte tryptase. Its purification from granules

RT and the characterization of inhibitor and substrate specificity.";

RL J. Biol. Chem. 263:13215-13222(1988).

RN [3]

RP SEQUENCE OF 29-40, AND CHARACTERIZATION.

RX MEDLINE=89009866; PubMed=3262682;

RA Hameed A., Lowrey D.M., Lichtenheld M., Podack E.R.;

RT "Characterization of three serine esterases isolated from human IL-2  
RT activated killer cells.";

RL J. Immunol. 141:3142-3147(1988).

RN [4]

RP SEQUENCE OF 29-39, AND CHARACTERIZATION.

RX MEDLINE=89035468; PubMed=3263427;

RA Kraehenbuhl O., Rey C., Jenne D.E., Lanzavecchia A., Groscurth P.,  
RA Carrel S., Tschopp J.;

RT "Characterization of granzymes A and B isolated from granules of  
RT cloned human cytotoxic T lymphocytes.";

RL J. Immunol. 141:3471-3477(1988).

RN [5]

RP 3D-STRUCTURE MODELING.

RX MEDLINE=89184501; PubMed=3237717;

RA Murphy M.E.P., Moulton J., Bleackley R.C., Gershenfeld H.,  
RA Weissman I.L., James M.N.G.;

RT "Comparative molecular model building of two serine proteinases from  
RT cytotoxic T lymphocytes.";

RL Proteins 4:190-204(1988).

CC -!- FUNCTION: THIS ENZYME IS NECESSARY FOR TARGET CELL LYSIS IN CELL-  
CC MEDIATED IMMUNE RESPONSES. IT CLEAVES AFTER LYS OR ARG. MAY BE  
CC INVOLVED IN APOPTOSIS.

CC -!- CATALYTIC ACTIVITY: HYDROLYSIS OF PROTEINS, INCLUDING FIBRONECTIN,  
CC TYPE IV COLLAGEN AND NUCLEOLIN. PREFERENTIAL CLEAVAGE: ARG-|-XAA,  
CC LYS-|-XAA >> PHE-|-XAA IN SMALL MOLECULE SUBSTRATES.

CC -!- SUBUNIT: HOMODIMER; DISULFIDE-LINKED.

CC -!- SUBCELLULAR LOCATION: CYTOPLASMIC GRANULES.

CC -!- SIMILARITY: BELONGS TO PEPTIDASE FAMILY S1; ALSO KNOWN AS THE  
CC TRYPSIN FAMILY. STRONGEST TO OTHER GRANZYMES AND TO MAST CELL  
CC PROTEASES.

CC -----



CC This SWISS-PROT entry is copyright. It is produced through a collaboration  
CC between the Swiss Institute of Bioinformatics and the EMBL outstation -  
CC the European Bioinformatics Institute. There are no restrictions on its  
CC use by non-profit institutions as long as its content is in no way  
CC modified and this statement is not removed. Usage by and for commercial  
CC entities requires a license agreement (See <http://www.isb-sib.ch/announce/>  
CC or send an email to [license@isb-sib.ch](mailto:license@isb-sib.ch)).

CC -----

DR EMBL; M18737; AAA52647.1; -.

DR PIR; A28943; A28943.

DR PIR; A30525; A30525.

DR PIR; A30526; A30526.

DR PIR; A31372; A31372.

DR PDB; 1HF1; 15-OCT-94.

DR MEROPS; S01.135; -.

DR MIM; 140050; -.

DR InterPro; IPR001254; Trypsin.

DR Pfam; PF00089; trypsin; 1.

DR SMART; SM00020; Tryp\_SpC; 1.

DR PROSITE; PS50240; TRYPSIN\_DOM; 1.

DR PROSITE; PS00134; TRYPSIN\_HIS; 1.

DR PROSITE; PS00135; TRYPSIN\_SER; 1.

KW Hydrolase; Serine protease; Zymogen; Signal; T-cell; Cytolysis;

KW Apoptosis; 3D-structure.

FT SIGNAL 1 26

FT PROPEP 27 28 ACTIVATION PEPTIDE.

FT CHAIN 29 262 GRANZYME A.

FT ACT\_SITE 69 69 CHARGE RELAY SYSTEM (BY SIMILARITY).

FT ACT\_SITE 114 114 CHARGE RELAY SYSTEM (BY SIMILARITY).

FT ACT\_SITE 212 212 CHARGE RELAY SYSTEM (BY SIMILARITY).

FT DISULFID 54 70 BY SIMILARITY.

FT DISULFID 148 218 BY SIMILARITY.

FT DISULFID 179 197 BY SIMILARITY.

FT DISULFID 208 234 BY SIMILARITY.

FT CARBOHYD 170 170 N-LINKED (GLCNAC...) (POTENTIAL).

SQ SEQUENCE 262 AA; 28968 MW; DA87363A0D92BAF4 CRC64;

MRNSYRFLAS SLSVVVSLLL IPEDVCEKII GGNEVTPHSR PYMVLLSLDR KTICAGALIA

KDWVLTAABC NLNKRQVIL GAHSITREEP TKQIMLVKKE FPYPCYDPAT REGDLKLLQL  
 TEKAKINKYV TILHLPKKGD DVKPGTMCQV AGWGRTHNSA SWSDTLREVN ITIHDRKVCN  
 DRNHYNFNPV IGMNMVCAGS LRGGRDSCNG DSGSPLLCEG VFRGVTSFGL ENKCGDPRGP  
 GVIYLLSKKH LNWIIMTIKG AV

//

## 2. A Sample entry for PDB

A sample entry for PDB is shown below, in which the records are fully described in fixed column format, columns not assigned to fields *must be left blank*.

HEADER	HYDROLASE (O-GLYCOSYL) 16-SEP-77	8LYZ	8LYZ 3
COMPND	LYSOZYME (E.C.3.2.1.17) IODINE-INACTIVATED		8LYZ 4
SOURCE	HEN (GALLUS GALLUS) EGG WHITE		8LYZ 5
AUTHOR	C.R.BEDDELL,C.C.F.BLAKE,S.J.OATLEY		8LYZ 6
REVDAT 9	14-JUL-86 8LYZH 3	SEQRES TURN ATOM	8LYZH 1
REVDAT 8	22-OCT-84 8LYZG 1	SHEET	8LYZG 1
REVDAT 7	27-JAN-84 8LYZF 1	REMARK	8LYZF 1
REVDAT 6	30-SEP-83 8LYZE 1	REVDAT	8LYZE 1
REVDAT 5	01-MAR-82 8LYZD 1	REMARK	8LYZE 2
REVDAT 4	21-MAY-81 8LYZC 3	ATOM	8LYZE 3
REVDAT 3	25-MAY-78 8LYZB 1	SEQRES	8LYZE 4
REVDAT 2	01-NOV-77 8LYZA 1	SSBOND	8LYZE 5
REVDAT 1	24-OCT-77 8LYZ 0		8LYZE 6
JRNL	AUTH C.R.BEDDELL,C.C.F.BLAKE,S.J.OATLEY		8LYZ 7
JRNL	TITL AN X-RAY STUDY OF THE STRUCTURE AND BINDING		8LYZ 8
JRNL	TITL 2 PROPERTIES OF IODINE-INACTIVATED LYSOZYME		8LYZ 9
JRNL	REF J.MOL.BIOL. V. 97 643 1975		8LYZ 10
JRNL	REFN ASTM JMOBAC UK ISSN 0022-2836 070		8LYZ 11
REMARK 1			8LYZ 12
REMARK 1	REFERENCE 1		8LYZ 13
REMARK 1	AUTH R.DIAMOND		8LYZ 14
REMARK 1	TITL REAL-SPACE REFINEMENT OF THE STRUCTURE OF HEN		8LYZ 15
REMARK 1	TITL 2 EGG-WHITE LYSOZYME		8LYZ 16
REMARK 1	REF J.MOL.BIOL. V. 82 371 1974		8LYZ 17

REMARK 1	REFN ASTM JMOBAK UK ISSN 0022-2836 070	8LYZ 18
REMARK 1	REFERENCE 2	8LYZ 19
REMARK 1	AUTH D.C.PHILLIPS	8LYZ 20
REMARK 1	TITL CRYSTALLOGRAPHIC STUDIES OF LYSOZYME AND ITS	8LYZ 21
REMARK 1	TITL 2 INTERACTIONS WITH INHIBITORS AND SUBSTRATES	8LYZ 22
REMARK 1	EDIT E.F.OSSERMAN,R.F.CANFIELD,S.BEYCHOK	8LYZ 23
REMARK 1	REF LYSOZYME 9 1974	8LYZ 24
REMARK 1	PUBL ACADEMIC PRESS,NEW YORK	8LYZ 25
REMARK 1	REFN ISBN 0-12-528950-2 977	8LYZD 1
	[REF 3-12 deleted]	
REMARK 2		8LYZ 95
REMARK 2	RESOLUTION. 2.5 ANGSTROMS.	8LYZ 96
REMARK 3		8LYZ 97
REMARK 3	REFINEMENT. BY THE MODEL-BUILDING AND REAL-SPACE	8LYZ 98
REMARK 3	REFINEMENT PROCEDURES OF R. DIAMOND. REFER TO REFERENCE 1	8LYZ 99
REMARK 3	ABOVE AND REMARK 4 BELOW.	8LYZ 100
REMARK 4		8LYZ 101
REMARK 4	THE ONLY SIGNIFICANT FEATURES ON THE DIFFERENCE MAP ARE IN	8LYZ 102
REMARK 4	THE REGION OF GLU 35 AND TRP 108 SIDE CHAINS - THE OE2 ATOM	8LYZ 103
REMARK 4	OF GLU 35 FORMS A COVALENT BOND WITH THE CD1 ATOM OF TRP	8LYZ 104
REMARK 4	108. AN INTERACTIVE COMPUTER GRAPHICS SYSTEM WAS USED TO	8LYZ 105
REMARK 4	MANIPULATE THESE SIDE CHAINS IN THE RSSD COORDINATE SET OF	8LYZ 106
REMARK 4	R. DIAMOND (1974), ENTRY 2LYZ IN THE PROTEIN DATA BANK, SO	8LYZ 107
REMARK 4	THAT A FIT TO THE ELECTRON DENSITY MAP WAS OBTAINED.	8LYZ 108
REMARK 4	THESE COORDINATES, THEREFORE, ARE IDENTICAL TO THE RSSD	8LYZ 109
REMARK 4	ENTRY APART FROM PORTIONS OF THESE TWO SIDE CHAINS.	8LYZ 110
REMARK 5		8LYZA 1
REMARK 5	CORRECTION.	8LYZA 2
REMARK 5	ADD SSBOND RECORDS.	8LYZA 3
REMARK 5	01-NOV-77.	8LYZA 4
	[REMARKS 6-12 deleted]	
SEQRES 1	129 LYS VAL PHE GLY ARG CYS GLU LEU ALA ALA ALA MET LYS	8LYZ 111
SEQRES 2	129 ARG HIS GLY LEU ASP ASN TYR ARG GLY TYR SER LEU GLY	8LYZ 112
SEQRES 3	129 ASN TRP VAL CYS ALA ALA LYS PHE GLU SER ASN PHE ASN	8LYZ 113
SEQRES 4	129 THR GLN ALA THR ASN ARG ASN THR ASP GLY SER THR ASP	8LYZB 3
SEQRES 5	129 TYR GLY ILE LEU GLN ILE ASN SER ARG TRP TRP CYS ASN	8LYZB 4
SEQRES 6	129 ASP GLY ARG THR PRO GLY SER ARG ASN LEU CYS ASN ILE	8LYZB 5

SEQRES 7	129 PRO CYS SER ALA LEU LEU SER SER ASP ILE THR ALA SER	8LYZ 117
SEQRES 8	129 VAL ASN CYS ALA LYS LYS ILE VAL SER ASP GLY ASN GLY	8LYZH 7
SEQRES 9	129 MET ASN ALA TRP VAL ALA TRP ARG ASN ARG CYS LYS GLY	8LYZ 119
SEQRES 10	129 THR ASP VAL GLN ALA TRP ILE ARG GLY CYS ARG LEU	8LYZ 120
HELIX 1	A ARG 5 HIS 15 1	8LYZ 121
HELIX 2	B LEU 25 GLU 35 1	8LYZ 122
HELIX 3	C CYS 80 LEU 84 5	8LYZ 123
HELIX 4	D THR 89 LYS 96 1	8LYZ 124
SHEET 1	S1 2 LYS 1 PHE 3 0	8LYZ 125
SHEET 2	S1 2 PHE 38 THR 40 -1 N THR 40 0 LYS 1	8LYZG 5
SHEET 1	S2 3 ALA 42 ASN 46 0	8LYZ 127
SHEET 2	S2 3 SER 50 GLY 54 -1 N ASN 46 O SER 50	8LYZ 128
SHEET 3	S2 3 GLN 57 SER 60 -1 N TYR 53 O ILE 58	8LYZ 129
TURN 1	T1 LYS 13 GLY 16 TYPE I.	8LYZ 130
TURN 2	T2 LEU 17 TYR 20 NEARLY TYPE II CONFORMATION.	8LYZ 131
TURN 3	T3 ASN 19 GLY 22 NEARLY TYPE II CONFORMATION.	8LYZ 132
TURN 4	T4 TYR 20 TYR 23 NEARLY TYPE II CONFORMATION.	8LYZ 133
TURN 5	T5 GLY 54 GLN 57 TYPE I,BETW STRNDS 2,3 SHT S2.	8LYZ 134
TURN 6	T6 ASN 59 TRP 62 NEARLY TYPE I CONFORMATION.	8LYZ 135
TURN 7	T7 THR 69 SER 72 NEARLY TYPE I CONFORMATION.	8LYZ 136
TURN 8	T8 ASN 74 ASN 77 TYPE I.	8LYZ 137
TURN 9	T9 ASN 103 ASN 106 TYPE I.	8LYZH 8
TURN 10	T10 CYS 115 THR 118 TYPE II (IMPERFECT).	8LYZ 139
TURN 11	T11 ILE 124 CYS 127 TYPE II (IMPERFECT).	8LYZ 140
SSBOND	1 CYS 6 CYS 127	8LYZA 5
SSBOND	2 CYS 30 CYS 115	8LYZA 6
SSBOND	3 CYS 64 CYS 80	8LYZA 7
SSBOND	4 CYS 76 CYS 94	8LYZA 8
CRYST1	79.100 79.100 37.900 90.00 90.00 90.00 P 43 21 2 8	8LYZ 141
ORIGX1	1.000000 0.000000 0.000000 0.000000	8LYZ 142
ORIGX2	0.000000 1.000000 0.000000 0.000000	8LYZ 143
ORIGX3	0.000000 0.000000 1.000000 0.000000	8LYZ 144
SCALE1	.012642 0.000000 0.000000 0.000000	8LYZ 145
SCALE2	0.000000 0.012642 0.000000 0.000000	8LYZ 146
SCALE3	0.000000 0.000000 .026385 0.000000	8LYZ 147
ATOM	1 N LYS 1 3.240 10.040 10.380 1.00 0.00	8LYZ 148
ATOM	2 CA LYS 1 2.390 10.410 9.250 1.00 0.00	8LYZ 149

ATOM	3 C LYS	1	2.460	11.920	9.100	1.00	0.00	8LYZ 150
ATOM	4 O LYS	1	2.580	12.670	10.100	1.00	0.00	8LYZ 151
ATOM	5 C B LYS	1	.950	9.960	9.490	1.00	0.00	8LYZ 152
ATOM	6 C G LYS	1	-.050	10.450	8.450	1.00	0.00	8LYZ 153
ATOM	7 C D LYS	1	-1.470	10.060	8.820	1.00	0.00	8LYZ 154
ATOM	8 C E LYS	1	-2.350	9.920	7.590	1.00	0.00	8LYZ 155
ATOM	9 N Z LYS	1	-3.680	9.380	7.960	1.00	0.00	8LYZ 156
ATOM	10 N VAL	2	2.390	12.350	7.850	1.00	0.00	8LYZ 157
[ATOM 11-998 deleted]								
ATOM 999	CD1	LEU 129	-12.970	22.550	8.090	1.00	0.00	8LYZ1146
ATOM 1000	CD2	LEU 129	-13.000	20.080	8.010	1.00	0.00	8LYZ1147
TER 1002		LEU 129						8LYZ1148
CONNECT	48	47	981					8LYZ1149
CONNECT	238	237	889					8LYZ1150
CONNECT	277	275	820					8LYZ1151
CONNECT	513	512	630					8LYZ1152
CONNECT	601	600	724					8LYZ1153
CONNECT	630	513	629					8LYZ1154
CONNECT	724	601	723					8LYZ1155
CONNECT	820	277	819	822				8LYZ1156
CONNECT	889	238	888					8LYZ1157
CONNECT	981	48	980					8LYZ1158
MASTER	124	0 0 4 5	11 0 6	1000	1	10	10	8LYZH 17
END								8LYZ1160

### 3. A Sample entry for Prosite

Example of a pattern entry of Prosite is shown below.

```

ID HOMEBOX; PATTERN.
AC PS00027;
DT APR-1990 (CREATED); JUN-1992 (DATA UPDATE); DEC-1992 (INFO UPDATE).
DE 'Homeobox' domain signature.
PA [LIVMFY]-x(5)-[LIVM]-x(4)-[IV]-[RKQ]-x-W-x(8)-[RK].
NR /RELEASE=24,28154;
NR /TOTAL=187(187); /POSITIVE=175(175); /UNKNOWN=0(0); /FALSE—POS=12(12);
NR /FALSE—NEG=9(9);
CC /TAXO-RANGE=??E??; /MAX-REPEAT=1;

```

DR P02833, HMAN—DROME, T; P07548, HMDF—DROME, T; P20009, HMDL—DROME, T;  
DR P18488, HMES—DROME, T; P10035, HMD2—DROME, T; P28468, HOX1—HALRO, T;  
DR P17208, BRN3—MOUSE, P; P20266, BRN3—RAT , P; P20912, HM16—XENLA, P;  
DR P20823, HNFA—HUMAN, N; P22361, HNFA—MOUSE, N; P15257, HNFA—RAT , N;  
DR P22197, ALF—ARATH , F; P08704, CDGT—KLEPN, F; P80064, HPPD—PSESP, F;

[70+ DR lines deleted (most were T)]

3D 1HDD;

DO PDOC00027;

//