Preprint MCS-P417-0294, Mathematics and Computer Science Division, Argonne National Laboratory, Argonne, Ill., March 1994.

# Automating the Determination of 3D Protein Structure*

Karen D. Rayl†,
Terry Gaasterland, and Ross Overbeek
Mathematics and Computer Science Division
Argonne National Laboratory

**Abstract.** The creation of an automated method for determining 3D protein structure would be invaluable to the field of biology and presents an interesting challenge to computer science. Unfortunately, given the current level of protein knowledge, a completely automated solution method is not yet feasible; therefore, our group has decided to integrate existing databases and theories to create a software system that assists X-ray crystallographers in specifying a particular protein structure. By breaking the problem of determining overall protein structure into small subproblems, we hope to come closer to solving a novel structure by solving each component. By generating necessary information for structure determination, this method provides the first step toward designing a program to determine protein conformation automatically.

*Work supported in part by the Office of Scientific Computing, Department of Energy, under contract W-31-109-Eng-38.

†Participant in the Summer 1993 Student Research Participation Program. This program is coordinated by the Division of Educational Programs. Home institution is Austin College, Sherman, TX 75091.

# 1 Protein Structure and Computer Science

*The properties of a protein are largely determined by its three-dimensional structure* [Voet and Voet 1990]. This statement would seem to simplify the process of understanding proteins and their functions. Actually, it exposes the major reason that so little is really understood about specific proteins or even proteins in general. For many proteins, certain properties of their function can be described, but no explanation of *how* they function can be given. The result is a gaping hole in our understanding of the fundamental processes of life. Nothing in the cell remains unaffected or unprocessed by proteins in some way. Many of the structural components of the cell are proteins; and, perhaps more important, proteins lie at the heart of the biological reactions that occur within the cell. Most objects known to have catalytic activity contain proteins, whether the protein is an enzyme and responsible itself for the catalysis or is the structural component of ribozymes. An understanding of proteins would unravel most, if not all, the processes necessary to regulate and sustain life. Such an understanding would open new doors in medicine, industry, agriculture, pharmaceuticals, and many other areas.

The determination of protein structure is a nontrivial problem. Much of the necessary information about which structure a protein will fold into is contained in the linear sequence of amino acids; therefore, it may be possible to determine the 3D structure of a protein from its amino acid sequence. Many scientists are currently working toward such an understanding, but to date, the mechanisms and rules by which a protein folds remain elusive. Key to current research are crystallographic techniques, which provide electron density maps that provide incomplete, but highly informative, data about the conformation of a protein molecule in its crystalline form. The density map is generated from 2D diffraction patterns representing *electron* densities (the nuclei are not visible) in a series of cross sections of the protein crystal. Unfortunately, for many proteins, X-ray crystallographic data is not easy to obtain, and the environment required to produce a clear diffraction pattern may actually deform the native conformation of the protein [Voet and Voet 1990]. In addition, given a 2D pattern there are still many possible 3D structures, because of the nondeterministic nature of the 2D to 3D mapping. Any aid that can be given to a crystallographer trying to determine a novel protein's structure thus increases the probability and speed of determining the proper structure.

The massive amount of information that needs manipulation when determining protein structure naturally leads to an alliance with computer science. It is assumed that there are specific rules by which proteins organize themselves and by which other biological processes occur. The mathematical nature of the underlying chemical reactions provides hope that the process may be simulated and structure determination automated. Computer visualization tools are already an indispensable part of a crystallographer's repertoire. Computer science has also assisted biology as databases of biological information have been created and maintained. These databases support the storage, sorting, and retrieving of massive amounts of biological data. In short, complex structural problems in biology require large-scale numerical analysis, 3D graphical representation, and parallel processing, thus presenting an interesting challenge to the computer scientist.

Our efforts have been focused on creating an assistant for determining protein structure. This paper presents our ideas for applying existing database information and biological theories to the problems

of protein structure. After establishing the basis for our theoretical approach in Section 2, we discuss the tools we plan to use to implement our ideas in Section 3. In Section 4, we outline the completed preliminary steps of integrating DBEMP, a database of functional information, and GenoBase, which already provides an interface for many structural databases. We highlight the intergration of Bowie et al.'s environmental classification method in Section 5. In Section 6, we present the current model for how we would obtain, and apply, for a particular peptide sequence, a specialized database of knowledge pertinent to protein structure determination. In Section 7, we summarize out efforts. We emphasize that our focus on integrating and utilizing *existing* knowledge (as opposed to the generation of a new theory) and on analyzing the problem from a computer science perspective makes our efforts unique.

# 2 Approach

Although the ultimate goal is to create a fully automated procedure to determine protein structure, our current goal is to create an "assistor" that could be used as a tool by a human trying to predict the structure of an unknown protein. The problem of fully determining any protein structure is very difficult and, given the current level of knowledge about proteins, not feasible. For a protein of 150 residues (a small to moderate-sized protein), $100^{200}$ possible constructions exit, of which $10^{38}$ can be viewed as nonrelated and independent, although it is estimated that only $10^{36}$ of these structures would be stable. Also, for a particular structure, an estimated $10^{33}$ different side chain arrangements could create the structure [Branden and Tooze 1991]. The search area for determining a protein structure is thus *huge*, especially if performing a "dumb" search. Note, however, that estimations predict fewer than 1000 topologically different domain structures [Branden and Tooze 1991]. Thus, by looking for domains within the 150 amino acid chain, instead of all possible conformations, we reduce the search area by many orders of magnitude. Each time more information on a novel protein is available, this search area is further decreased. Possibly, as more is discovered about the rules by which proteins fold into their proper conformations, the search area will be reduced to one structure.

Much effort currently focuses on creating individual computational methods that can either determine protein structure automatically or provide assistance to humans trying to solve novel structures. An automated system that integrates these methods has not been created. Therefore, we have decided to tackle the problem by exploring ways to fully utilize data and methods already collected by other groups. Connection of disparate bodies of existing information is sorely lacking in biology and may provide insights that are not readily apparent when each body of information is manipulated independently.

Current protein prediction methods rely on using X-ray crystallographic information complemented by homology[1] matches against proteins of known structure. It is much more difficult to determine the structure of a protein that is not homologous to any known protein, even given excellent crystallographic data. An alternative approach is to break the protein into parts, analyze each part

[1] *Homologous proteins* are evolutionarily related proteins which therefore share, to variable degrees, structure and function. Lower homology implies more evolutionary distance and less similarity.

independently, and then establish connections between the separate domains. Instead of relying on full-sequence homology matches, the more common local-sequence homologies can be utilized. By saying

partA looks (90% homology) like domain_type_1, partB looks (75%) like domain_type_5, partC resembles (30%) domain_type_1, ...;

the problem is reduced to determining the connecting sequences and adjusting each part for its unique characteristics. With computer technology, many small domains can be manipulated and shown at once. The biologist thus has a better feeling for the overall protein structure, and the raw jumble of electron densities is simplified. Each part becomes a "box" that can be either looked into or looked at. Breaking the protein into parts also increases the likelihood that helpful information will exist about what is "inside" the box. Fully utilizing collected biological data to decode information in each box might thus conquer a divided protein.

Many protein databases are really just data banks of flat-formatted files containing data; therefore, it is appropriate to collectively refer to databases and data banks as "data repositories." By translating these bodies of data into Prolog, as is done with GenoBase, and by providing access to alignment tools through Prolog commands, we gain not only to use the data repositories in a common framework but also the ability to reason about the proteins. The GenoBase paradigm provides a framework for accessing multiple databases through a single query language and using alignment tools as "operations" over the data.

In this paper, we describe

- the integration of the Databse of Enzymes and Metabolic Pathways into GenoBase,
- plans to integrate the Bowie, Luthy, and Eisenberg paradigm of aligning by "profiling," and
- a method for using this environment to gather data about a new peptide sequence.

Integrating protein data repositories and sequence alignment tools into a single environment is the first step to providing an automated assistant to the crystallographer.

# 3 Tools

GenoBase [Overbeek and Price 1993], an ongoing project at Argonne, is a methodology for integrating biological databases. In this section, we briefly describe GenoBase and data repositories such as PDB, PIR, PROSITE, SWISS-PROT, and DBEMP, all of which may be accessed through GenoBase, and which capture a large quantity of carefully maintained specialized information. Then we examine a new method for determining sequence homologies based on local amino acid environmental classes, as opposed to direct sequence alignment. This methodology can find structure homologies where sequence homologies have degenerated and thus is a powerful addition to our arsenal of tools for discovering new relationships between proteins.

## 3.1 GenoBase

**GenoBase** provides coherent access to many different biological data repositories, including Brookhaven PDB (Protein Data Bank), PIR (Protein Identification Resource), PROSITE, and SWISS-PROT. These repositories are primary sources of the current knowledge on protein structure. DBEMP (DataBase of Enzymes and Metabolic Pathways) is currently being added to GenoBase. GenoBase provides the unique service of allowing one to access many different repositories at once and relate data from one source to data in another source. Biology has many areas of inquiry, including structure determination, for which simultaneous access of more than one database is needed [Sillince and Sillince 1991]. In addition to the features obtainable from the individual databases, the connections between the data sets provide knowledge that can be employed in the search for information relating to protein structure. GenoBase views each data source as a collection of data objects. A complementary set of connections between data objects relates data within and across sources. Users query GenoBase by asking about attributes of data objects and about the connections between objects. Interaction with data repositories may occur by using each independent database directly, through a subset of GenoBase, or through a modified version of GenoBase, to retrieve information pertinent to analyzing protein structures [Overbeek and Price 1993]. GenoBase was designed by Ross Overbeek.

## 3.2 Databases

**PDB** is the largest repository for 3D protein structures determined by X-ray crystallography or nuclear magnetic resonance and contains examples of all known unique protein families [Hobohm et al. 1992]. The proteins of known 3D structure, provided by the PDB, are commonly used to check current attempts to automate the process of protein structure determination. A homology match to a protein in the PDB provides invaluable clues to the structure of an unknown protein. As more protein structures are determined, it becomes increasingly probable that such a homology match will be found within the PDB. The April 1993 release of the PDB contains 1110 fully annotated atomic coordinates entries, and the number of entries in the PDB increases dramatically with each new release. PDB is the product of researchers at Brookhaven National Laboratory [PDB staff 1993].

**PIR** is a collection of sequences originally designed to study the evolutionary relationships between proteins. PIR is therefore organized on the idea of protein superfamilies. Superfamilies consist of homologous proteins that appear evolutionarily related on the basis of amino acid sequence. Such superfamilies are often the products of gene duplication. However, the superfamily design was initiated before it became apparent that large proteins are often composed of domains from different evolutionary origins, obtained through fusion, gain and loss of exons, shifted reading frames, incorporation of foreign DNA, or rearrangement of native DNA. To accommodate such "unevolutionary" homologies, PIR assigns sequences that are mostly unrelated evolutionarily to separate superfamilies, even though they may contain related domains. Margaret Dayhoff initiated the PIR database [Barker et al. 1991].

**PROSITE** contains biologically significant patterns and sites that can be easily used to classify an unknown protein into a known family of proteins. Instead of relying on overall sequence alignment to

identify structure, PROSITE allows for local sequence alignment matches. The patterns for matches are short, allowing for specificity. The goal is to obtain a core pattern that will detect all the proteins in a certain family, or subfamily, without matching proteins outside of this group. It is worth noting that more than one motif may be indicative of one protein family, allowing multiple checks of family assignment. PROSITE was created by Amos Bairoch [Bairoch 1993b].

**SWISS-PROT** houses translations of sequences contained in the EMBL (European Molecular Biology Laboratory) Nucleotide Sequence Database and all PIR annotated sequence data, as well as original data. Sequence data corresponds to protein form before posttranslation processing and is closely linked and cross referenced to other EMBL databases, such as PDB and PROSITE. A strong effort was made to include annotations of the extent of different sequence domains present in each entry, and all information relating sequence features is stored in computer-readable tables. The ultimate goal of SWISS-PROT "is to provide a complete annotated protein sequence data bank where all the data is easily retrievable by computer programs and is stored in a format similar to that of the EMBL Nucleotide Sequence Database" [Bairoch 1993a]. SWISS-PROT is the work of Amos Bairoch.

**DBEMP** currently includes approximately 10,000 articles published in a variety of biological journals encoded into a computer-accessible format. This has created a vast store of data relating to protein function which is not available elsewhere because of the difficulty of creating and organizing such a database. The goal in the creation of DBEMP was to obtain a warehouse of information upon which to create models and simulations of living systems; this database thus contains information on many aspects of biology. DBEMP contains not only logical descriptions of how the reactions of metabolic pathways interconnect but also raw and evaluated numeric data. The numeric data provides a basis for simulation; the logical, symbolic data provides connections to other databases. (For information on the DBEMP integration, see below.) Since DBEMP's central objects are metabolic pathways, the connected system as a whole is a rich source of interconnected functional data that can be tied into structural data. Evgeni Selkov designed DBEMP.

## 3.3 New Method of Determining Structural Similarity

Protein structure is determined more by the interaction of properties of amino acid side chains than by the particular individual amino acids in the primary sequence. It is often possible for two dissimilar sequences to form the same 3D structure because of the following:

1. amino acids of similar type are located in key positions even though specific residues are not conserved,
2. replacements or movements may occur in neighboring side chains,
3. shifts in the backbone may occur, and
4. compensating changes in sequences at neighboring or distant sites may occur.

Although specificity[2] is a prime characteristic of protein interactions, and protein function is intimately tied to protein structure, proteins are actually very tolerant of residue substitutions [Bowie et al. 1990]. The detection of structural similarities must therefore look beyond the level of simple sequence homology.

In an attempt to find a new way to look at protein structure, Bowie, Luthy, and Eisenberg [Bowie et al. 1991] classified residue microenvironments into 18 environmental classes. 3D protein structure can be converted into a 1D string or sequence of environmental classes. Environmental class sequences can then be aligned to measure the compatibility of a new sequence with the given 3D structure. This 3D structure profile is, in general, less sensitive to the specific sequence relations detectable with standard primary structure alignments but more sensitive to general structural similarity. Using this method, we may be able to obtain information about structural homologies, a key to determining unknown structures where sequence homologies do not exist because of degeneration of the primary sequence[3] or convergent evolution of similar protein sequences.[4]

## 3.4 Additional Databases and Techniques

Biological databases are dynamic, being constantly replaced, corrected, and updated (generally manually). This fact complicates the problem of determining the maximum amount of information that can be extracted from a data repository. Not only does the informational content change, but the manner in which the information is stored often changes. Original database schemes frequently must be revamped as the sheer amount of data overwhelms the old system. Maintenance problems occasionally cause fields to be deleted, or at least not to be kept current, while other fields, cross references, and cross links are newly formed as natural relatedness becomes more apparent. Many new databases try to create formats consistent with present databases, especially those housed by EMBL. Other data banks, however, originate with one scientist creating a resource intended only for personal use. Determining what information can, and cannot, be utilized within each of the data sets is thus an ongoing challenge.

In addition, a large number of biological databases have been created containing other subject-specific information that may also be of use to protein structure determination. We are still exploring many of these possibilities, including BLOCKS, 3D-ALI, DSSP, HSSP, and PKCDD.

Many groups worldwide are trying to determine the rules for protein folding. The ultimate goal of many of these groups is a method for determining tertiary structure of any protein from the primary sequence (which can often be determined given the gene sequence). The pattern recognition and matching patterns of computers may provide insight for scientists trying to determine the general

[2] Because proteins interact with their substrates largely on the basis of topology and geometry, they express a great deal of *specificity* for substrates. Substrate charge, polarity, and other factors that influence the overall shape of the enzyme-substrate complex are just as important as the substrate's individual 3D shape.

[3] It is sometimes possible to effect great changes in amino acid sequence without greatly affecting function, as indicated earlier. Primary sequence homology may have diverged to such an extent that homology is not detectable by normal sequence alignments.

[4] Since protein structure is closely related to protein function, two proteins of similar function may show convergence to similar amino acid sequences without being evolutionary descendants of the same DNA archetypal sequence.

rules of protein structure; additionally, computers many assist those scientists trying to determine a specific protein structure. Meanwhile, groups such as Bowie, Luthy, and Eisenberg are discovering many general rules and constraints that we can utilize. Other methods for "profiling" a protein by assigning properties of its structure to its sequence exist[Tcheng and Subramaniam 1993]. Once the Bowie et al. profiling methods are integrated, we plan to move on to the use of other profiling methods as well. Coevolution between these groups and ours will lead to the availability of an automated system of determining protein structure given only the DNA sequence.

# 4 Integrating a Database: DBEMP

As mentioned above, DBEMP contains functional data not available in other databases, which could be applied to the problem of protein structure determination. Establishing interconnections between GenoBase and DBEMP would unite the functional data of DBEMP with the structural data available in current large databases. This union is of extreme importance in biology, because structure and function are such intertwined concepts as to be ultimately inseparable, thus giving another path of attack when hypothesizing the structure of an unknown protein. Given a structureless protein, say, proteinA, the current first step would be a search against the complete PDB for a global sequence homology match. Suppose, however, a quick look at DBEMP shows that another author has determined that proteinA is a DNA binding protein. This would suggest looking for a specific DNA binding motif or local sequence homology contained in PROSITE, which would then suggest a specific protein family within the implied protein superfamily of DNA binding proteins, thus greatly reducing the search universe in the PDB. With the organization of PIR this knowledge becomes an asset that can be fully utilized. Suppose further information emerges, perhaps from a second author within DBEMP, that proteinA has a much stronger binding constant to ssDNA than to dsDNA. One could then search PROSITE for a subfamily[5] within the family implied by the DBEMP, that is, the family of ssDNA binding proteins. The search could then focus on PDB and PIR entries of proteins within this subfamily. As observed by Branden and Tooze, reducing the number of possible structures is the key to determining a novel protein's 3D shape:

> *Why is this prediction of protein structure so difficult? The answer is usually formulated in terms of the complexity of the task of searching through all the possible conformations of a polypeptide chain to find those with low energy. It requires enormous amounts of computing time.* [Branden and Tooze 1991]

Therefore, much effort has been directed toward putting DBEMP into a form that is easily accessible by the proposed protein structure assistant. Specifically, this task involves encoding DBEMP's data as Prolog objects and then integrating the database with GenoBase. The following discussion presents the work accomplished as part of this effort.[6]

[5] In this example, the term subfamily would be used to describe an ssDNA binding protein that uses a particular motif as its binding domain.

[6] The team members involved in the DBEMP integration were Terry Gaasterland, Natalia Maltsev, Jennifer Mankoff, Ross Overbeek, Murali Raju, Karen Rayl, and Rahul Singhal.

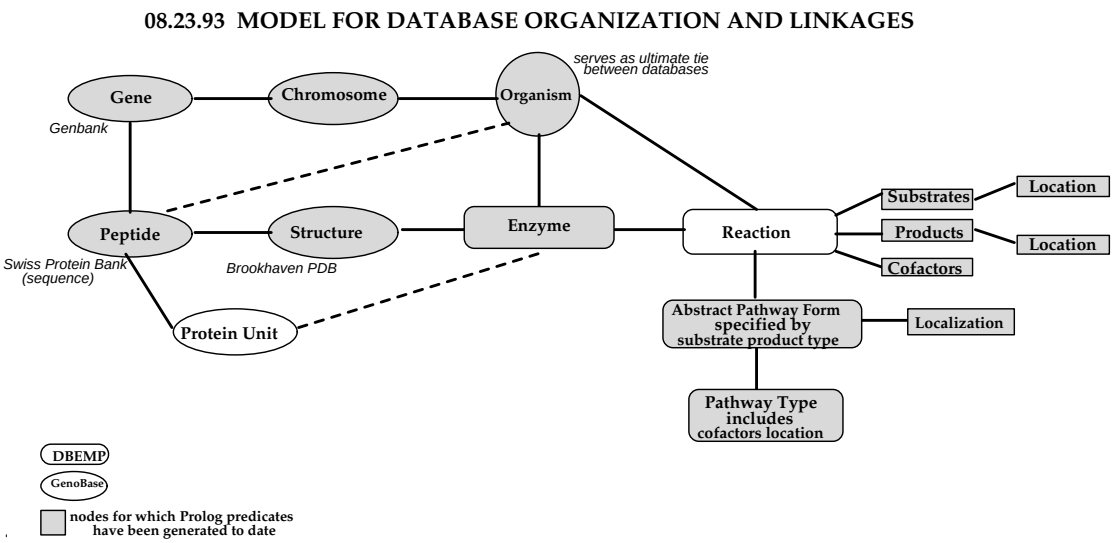


Figure 1: Current Model for DBEMP Integration

## 4.1 Organizing DBEMP Information

Like most biological databases, DBEMP was created with a specific purpose in mind that controlled the format of data storage. This individuality of data sets makes the process of integrating multiple databases very challenging, especially when the normal problems of maintaining each individual database are also considered. The problem is compounded in integrating DBEMP into the GenoBase framework, since DBEMP contains information of a much different type from that stored in most data banks, that is, large numbers of detailed facts, extracted and analyzed by hand from a broad range of scientific articles. The integration of DBEMP thus relies on being able to take the flat DBEMP files and create Prolog objects that can be related to other fields within DBEMP, to other records within DBEMP, and, most excitingly, to existing structural databases already incorporated into GenoBase.

Figure 1 shows the present model for the organization of DBEMP information into structures that would be integrable with GenoBase. The figure shows nodes representing information obtained from DBEMP, as well as the primary connections to data types already contained within GenoBase. GenoBase was created with Prolog, and thus all data is structured such that it can be easily related to other objects in a variety of ways. DBEMP, on the other hand, contains flat data, where natural connections between each entry, and each field within an entry, are not easily retrieved. Structure is first imposed by delineating which fields within DBEMP contains what kind of information. After the data stored in these files has been organized, it is possible to form the interconnections between different "nodes" of information.

## 4.2 Exploiting the Concept of Metabolic Pathways

The design of DBEMP involved thinking carefully about how to represent sufficient information to uniquely identify enzymes. Enzyme form, as presented in Figure 1, involves several distinct properties that differentiate instances within an enzyme code[7] and organism. For example, DBEMP can discriminate between enzymes based on cellular location, presence or absence of prosthetic group, posttranslational modifications, or amino acid sequence differences. Since all enzymes are linked to metabolic pathways, these pathways are the key to connecting the DBEMP records and enzyme information to other databases. Our first step toward unifying the data repositories was thus the creation of Prolog atoms encoding the metabolic pathway name for all applicable records. A description of the creation of Prolog objects containing pathway name data will serve as a paradigm to illustrate how DBEMP data is converted to a form integrable with GenoBase.

We decided upon two primary ways to look at the concept of a metabolic pathway. The *Abstract Pathway Form* is much more general and is delineated only by a pathway's substrates, products, and type. The *Pathway Type* is further specified by necessary cofactors and the location of the pathway, initial substrates, and final products. Records where one or more elements of the *Pathway Type* are not included were defined as undesignated, as explained below. Information to create these Prolog objects about pathways was stored primarily in two fields: MPW (Metabolic PathWay) and SPN (Specific Pathway Name). Roughly, the MPW field corresponds to *Abstract Pathway Form*, while the SPN field corresponds to *Pathway Type*.

To obtain a complete list of SPN and MPW entries, we first transferred DBEMP into a Unix format and created a *Perl* script capable of extracting particular fields within each record. Because some fields, including the MPW field, contain information stored in tables, we also decomposed wanted, or removed unwanted, tables and subfields. Two complete lists were extracted from DBEMP, a "rawspn list" including record ID and the value of the SPN field, and a "rawmpw list" including record ID and the MPW field value. These raw data files were used to generate files of Prolog objects, respectively *rawspn.pl* and *rawmpw.pl*, using an Emacs macro. Our goal was to use these very rough atoms to generate a single predicate, designated "spn," for all records possible. The spn predicate records would then represent the *Pathway Type* node and could be used to back-generate the *Abstract Pathway Form* data items.

The MPW field, as defined by Selkov, has the following form:

substrates--products_type

In contrast, the SPN field, as defined by Selkov, has the form

substrates--products_type_(cofactors)_(pathway location)[8]

[7] Enzyme code numbers are not assigned one number to one physical object. Instead, enzyme code numbers are assigned to reactions (taking a simplistic view). Therefore, one physical object called an enzyme may have many code numbers.

[8] Note: Some records contained extra information in the form of initial substrate and/or final product location. See later in this section for treatment.

The spn predicate would contain the SPN value, where it existed, or the MPW value, if it existed and the SPN value did not exist. "Undesignated" was used for any information not contained in the record, above the minimal required of substrate, product, and type.

Because of the large volume of records, there were understandably many typographical errors and some records where the SPN and/or MPW fields were not in the proper format. To create a useful spn predicate, first the SPN and MPW values had to be "cleaned" of such errors. Common typographical errors included spacing, location of commas or other punctuation, and spelling. Records that were not in the proper format or were missing minimal information were designated as problems. They were removed from the *rawspn.pl* and *rawmpw.pl* files and held until they could be fixed by Selkov's team.

Typographical errors were transferred to a *sed* file. As the *sed* file was too large to be used directly, it was converted into a *Perl* script that could then be run against the *rawspn.pl* and *rawmpw.pl* files. This generated *spn1.pl* and *mpw1.pl* files "without" problems or typographical errors. As the files were used, especially as attempts were made to parse the spn atoms, more corrections were made (see below).

The *spn1.pl* and *mpw1.pl* files were then merged to generate the spn predicate. Where the predicate of *spn1.pl* existed for the ID, it became the predicate in *spn2.pl*. Where the *spn1.pl* atom for that ID no longer existed, was removed as a problem, or never existed in DBEMP, the predicate in *mpw1.pl* became the predicate in *spn2.pl*. The *spn2.pl* atoms were of the form

spn('ID','value').

where

value = substrates--products_type_(cofactors)_(location)

and

location = pathway location + substrate location (optional) +
product location (optional)

This generated the first "complete" list of pathway names and associated records for the unification of DBEMP and GenoBase. The next phase was to parse the spn predicate into its separate informational parts: *substrates*, *products*, *type*, *pathway location*, *substrate location*, and *product location*.

## 4.3 Parsing the *spn* Predicate

Parsing the spn predicate was complicated by the fact that the SPN and MPW strings had not originally been designed to be divisible and there was no real separator between most of the subfields. The *substrate* substring was separated from the rest of the string by "--", but the rest of the string was not naturally divisible. Dividing *products* from *type* at "_" was not possible because "_" was also used to separate each word in multiword products and to separate different products from

each other. It was not practical to divide *type* from *cofactors*, or *cofactors* from *location*, at "(" or ")" because there exist cofactors with "( )" within their names, e.g., NADP(+). We thereofre chose to divide *substrates* from the rest of the string at "--", search for a match of *type* against a list of possibilities, and separate *products* as being "everything before *type*." The next step was to determine whether the *cofactor/location* unit existed and, if so, whether *cofactors*, *location*, or both *cofactors* and *location* values were present. This was accomplished by matching the contents of "( )" against lists of possible cofactors and then of possible locations. The final parser generated also created individual units of each cofactor and each location listed within the subfields.

In the attempt to parse the spn predicate, many more errors (typographical and formatting) were discovered. Instead of adding the corrections to the original problems file and *sed* file of typographical errors, we made the corrections directly to *spn2.pl*. The file was declared "clean" when no more errors were detected during parsing. The problems and *sed* aliases files, in addition to aliases files in other formats, were then back-generated by running the totally corrected *spn2.pl* against the original *spn.pl*, using a Prolog program. (Note: this method does not take into account the merging of *mpw.pl* files, as explained earlier. This "flaw" should be remembered when trying to use the aliases and problem files to clean the original database.)

We then created a "crude" parser (using `append`) and a DCG parser to completely divide the spn predicate. This process allows for the creation of the *Substrate, Substrate Location, Product, Product Location, Cofactors,* and *Localization* nodes represented in Figure 1. The parsed spn were then used to generate a predicate of the form

spn_parsed('ID', spn(mpw), cofactors, pathway_location, substrate_location, product_location).

where

mpw = mpw(substrate, products, type).

Similar Prolog predicates were created for

- *Organism Name* or(ID, Genus, species).
- *Enzyme* ec(ID, enzyme_code_number).
  en(ID, enzyme_common_name).
- *Reaction* reactions(ID, reaction), where reaction =
  reaction(complete substrates, complete products,reversible/irreversible, direction).
- *Compound* compound_id(abbrev_for_compound, compound_name).

objects, as shown in Figure 1.

The next step, which is under way, is to create the ties between the nodes within DBEMP and then with GenoBase.

# 5 Integrating a Tool: The Environmental Classification Method

Most theories and methods for detecting protein homology detect amino acid sequence similarity. As indicated earlier, Bowie et al. [1991] developed a system, and computer application, that detects structure homology without relying on primary sequence. Specifically, they create a 1D representation of the protein as a series of environmental classes. These linear sequences are then aligned and compared to detect structural based homology between proteins. The program, as written by Bowie et al., searches only for global homology, using a matrix with parameters of amino acid and environmental class, with values representing the probability of each amino acids being found in that environmental class. Our idea is to extend this method to detect *local* homologies.

Normal sequence alignment programs compare two linear sequences of amino acid and employ a probability matrix called a PAM (Percentage of Accepted point Mutations) matrix. The PAM values, representing the probability that one amino acid is substituted for another, are the accepted values that have been observed and estimated. They are used as a measure of the rate of evolution, by expressing the number of substitutions (point mutations) per codon per $10^{10}$ years [Schulz and Schirmer 1990]. An apparent similarity exists between the design of the homology *search* itself, whether using the Bowie et al. environmental classes or traditional sequence alignments. The primary difference between our application of Bowie et al.'s method and their original intent is that we include procedures to detect local sequence alignments, as well as global alignments of protein structures.

Our approach involves using Miller's sequence alignment program [Miller 1991], which is a "traditional" alignment program based on the PAM matrix paradigm, to detect environmental class homology. Our method is to assign each environmental class, 18 in all, a one-letter code, similar to the idea behind the one-letter amino acid code. We then modify the probability matrix of Bowie et al. to "look like" a PAM matrix. By presenting Miller's algorithm with

1. an amino acid sequence we wish to test for homology, to
2. a second sequence with known structure represented by the 1D environmental class, in the one letter code,
3. and the pseudo-PAM matrix, which is actually a matrix of the probability that the second sequence will assume the shape (hence, environmental classes) presented in 2,

we can then employ Miller's algorithm to calculate the homology between the questioned amino acid sequence and the amino acid structure represented in the sequence of environmental classes.

As stressed earlier, it is the interaction of amino acids and not the strict primary sequence that determines protein shape. The primary sequence and specific combination of amino acids associated to give the structure are of importance only in that they aid in determining the tertiary structure. By using Bowie et al.'s method, we can remove part of our dependence on strict sequence homology to get an initial set of possible structures. To fully integrate this tool with the database, we next create a GenoBase operator that calls the environmental aligner and retains potential domains for parts of an input amino acid sequence.

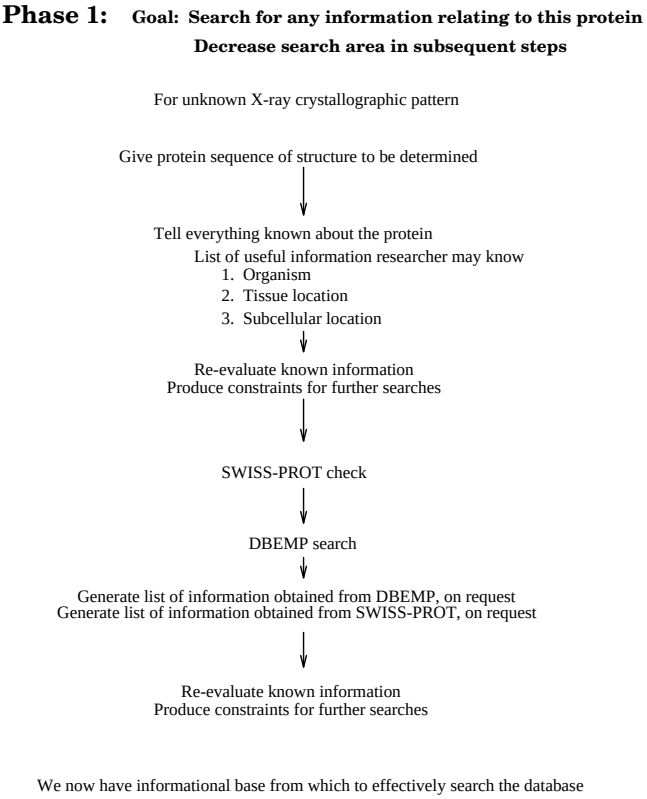


Figure 2: Restriction of Search Area

# 6 Using the Integrated System: A Sketch

We can now begin to create an automated assistant for determining protein structure that applies information using Bowie et al.'s classification system and the data sets introduced above. A model for the creation of such an assistant, divided into three phases, is represented in Figures 2, 3, and 4. Each of the smaller or more easily accessed databases is used to restrict the query before entering the larger or more cumbersome data banks. This model will undoubtedly be changed greatly as an actual program is implemented and as more is discovered about other data repositories. We repeat that the information presented here is part of a larger project utilizing X-ray crystallographic data. The program modeled here is meant solely to complement the larger program by providing additional information for creating a hypothetical structure.

Phase 1 (Figure 2) is directed toward gathering as much information as possible to make further searches more effective and to reduce the search tree. As the researcher presents the amino acid sequence, with the accompanying X-ray crystallographic structure, queries can be initiated to obtain any additional information the researcher may have. Since the SWISS-PROT database has been created by inferring amino acid sequences directly from nucleotide sequences, a SWISS-PROT search

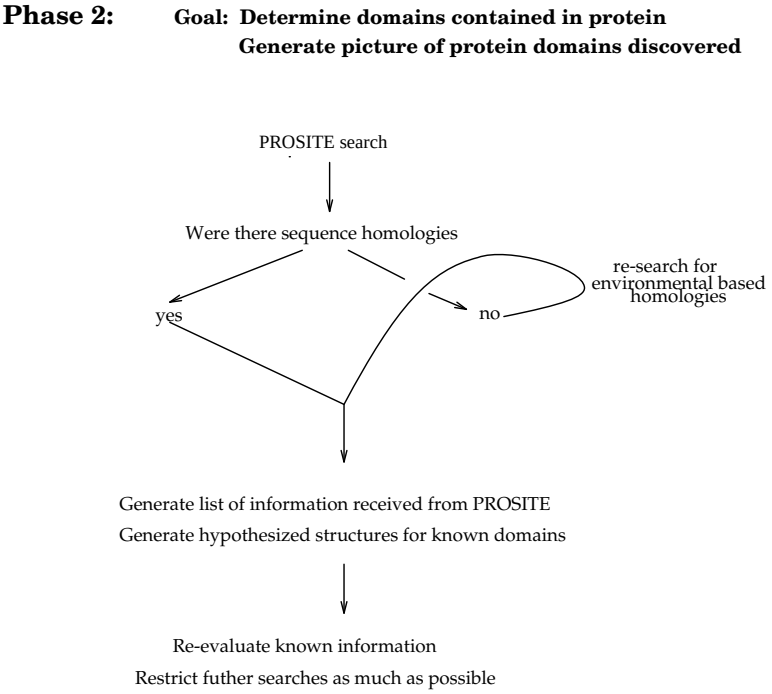


Figure 3: Establish Domains and Local Homology

may be performed on a given amino acid sequence to determine the gene from which the protein was transcribed and translated. If information exists regarding this gene (for example, regulation and activation data), this may shed light on possible protein function or pattern of production. Next, DBEMP can be searched for information relating to the novel protein. This information can then be filtered and sorted for information that can be used to restrict further searches or that may indicate aspects of protein function/structure not previously realized by the researcher.

Phase 2 (Figure 3) involves a search of PROSITE. The data contained in PROSITE allows the protein to be broken into the boxes suggested earlier and may also highlight features inside each box. Where PROSITE contains structural data, this can be directly utilized. For domains without specific structural information, example proteins may be stored for future cross checking with PDB. Thus, if the novel protein contains, say, domainA, which is also contained in, say, proteinX, proteinY, and proteinZ, it is possible that structural information will exist for at least one of protein[X-Z] within the PDB that can then be applied to the unknown protein structure. PROSITE matches are based on amino acid sequence homology, but it may be possible to create a subprocedure that can search for homologies with local domains contained in PROSITE via the classification system of Bowie et al.. The final step includes the specification of domains discovered during the PROSITE search and, where possible, generation of hypothetical structures. Further searches can again be constrained using information gathered at this step.

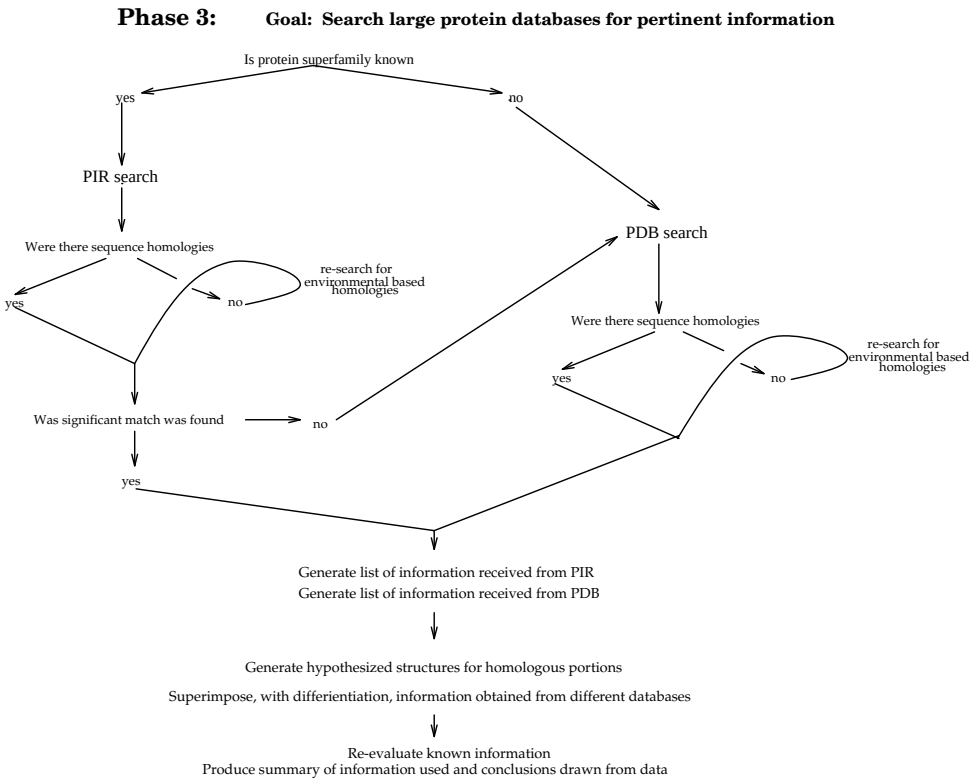


Figure 4: Search Protein Structure Database – Global Homology

Phase 3 (Figure 4) commences after as much restrictive data as possible has been gathered. In this phase, PDB is searched for information. This restriction of the search area specifically within the PDB will become more important as more protein structures have been generated. In cases where a protein superfamily can be hypothesized, it becomes beneficial to first access PIR to see whether homologous structures can be determined. Note that searches in both PIR and PDB could also utilize a subprocedure involving Bowie, Luthy, and Eisenberg's environmental classifications.

At each step in the process, the output of information extracted from each database, as well as a summary of conclusions based on such information, will be accessible to the researcher. Each phase is also independent in that a researcher may terminate the process between phases when it appears that enough information has been withdrawn for a preliminary attempt at creating a protein structure. Because of the intrinsic design of Prolog, it may also be possible to easily modify the process in order to check hypothesized structures and suggest areas that may need more attention. Information would then be gathered and compared with the hypothesized structure in much the same manner as originally proposed.

# 7 Conclusion

We currently have integrated access to many protein structural data sets through GenoBase and will soon be able to simultaneously access the functional data of DBEMP. The next step is to create a system and user interface that is customized to a crystallographer's needs using the data repositories described here (PROSITE, PIR, SWISS-PROT, and DBEMP) and that incorporates new approaches to structure determination, such as environmental classes of sequence alignments. By leaving the system with the utmost flexibility, we hope to create an automated assistant that can grow to include developing databases and theories. When combined with tools being developed to help interpret electron densities, our specialized tool for crystallography will bring scientists a step closer to the ultimate goal of an automated method for prediciting protein structure.

When the Advanced Photon Source is fully operational at Argonne National Laboratory, X-ray crystallographic data will be produced in large quantities, creating a strong local need for tools that assist crystallographers with protein structure determination. With the many diverse groups working world-wide to decipher the mechanisms that determine protein structure, the need for systems that integrate these groups' discoveries is also increasing. Compilation of existing knowledge and theories aids not only the crystallographers who seek a specific structure, but also those who seek the general rules of protein conformation, by highlighting areas in which current models are deficient. By manipulation, integrating, and organizing the massive quantities of biological data being produced on all fronts, computer science can provide an invaluable service to the biologist and can contribute to the solution of problems that are intrinsically challenging.

## Acknowledgments

We acknowledge Marianne Schiffer for her careful and practical guidance; Rick Stevens for all his support and guidance, and Fred Stevens and Priscilla Wilkins for their insight. Thanks to all the member of the "Monday Biology Meeting" for helping to create a friendly environment for interdisciplinary work.

I extend a personal thanks to Terry Gaasterland and Ross Overbeek for creating such a wonderful environment in which to explore and research. Without their superb attitudes, support, and guidance, this summer would have been impossible.

Thanks to Jennifer Mankoff for all the computer glitches she pulled us out of, and for being herself. Thanks to all the other members of our team — Natalia Maltsev, Murali Raju, and Rahul Singhal — for all the help, and a great summer.

This work was completed at Argonne National Laboratory during the Summer 1993 Student Research Participation Program, funded through the Division of Educational Programs. This work was also funded in part by the Office of Scientific Computing, U.S. Department of Energy, under Contract W-31-109-Eng-38.

## References

[Bairoch 1993a] Bairoch, A. 1993. The SWISS-PROT Protein Sequence Data Bank, User Manual. Release 25, April 1993. See also *Nucleic Acid Research* **20**, 2019–2002 (1992).

[Bairoch 1993b] Bairoch, A. 1993. PROSITE: A Dictionary of Protein Sites and Patterns, User Manual. Release 10.2, July 1993. See also *Nucleic Acid Research* **21**, 3097–3103 (1993).

[Barker et al. 1991] Barker, W., D. George, L. Hunt, and J. Garavelli. 1991. The PIR protein sequence database. *Nucleic Acids Research* **19**, 2231–2236.

[Bowie et al. 1990] Bowie, J., J. Reidhaar-Olson, W. Lim, and R. Sauer. 1990. Deciphering the message in protein sequences: Tolerance to Amino Acid Substitutions. *Science* **247**, 1306–1310.

[Bowie et al. 1991] Bowie, J., R. Luthy, and D. Eisenberg. 1991. A method to identify protein sequences that fold into a known three-dimensional structure. *Science* **253**, 164–170.

[Branden and Tooze 1991] Branden, C., and J. Tooze. 1991. *Introduction to Protein Structure*. New York: Garland Publishing.

[Hobohm et al. 1992] Hobohm, U., M. Scharf, R. Schneider, and C. Sander. 1992. Selection of representative protein data sets. *Protein Science* **1**, 409–417.

[Miller 1991] Miller. 1991. A time-efficient, linear-space local similarity algorithm. *Advances in Applied Mathematics* **12**, 337–357.

[Overbeek and Price 1993] Overbeek, R. and M. Price. 1993. *Accessing Integrated Genomic Data Using GenoBase: A Tutorial. Part 1.* ANL/MCS-TM 173, Argonne National Laboratory, Argonne, IL.

[PDB staff 1993] 1993. Protein Data Bank Quarterly Newsletter, Number 64, Brookhaven National Laboratory, April.

[Sillince and Sillince 1991] Sillince, J., and M. Sillince. 1991. *Molecular Databases for Protein Sequences and Structure Studies*. Berlin: Springer-Verlag.

[Schulz and Schirmer 1990] Schulz, G., and R. Schirmer. 1990. *Principles of Protein Structure*. New York: Springer-Verlag, p. 168.

[Tcheng and Subramaniam 1993] Tcheng, D. and S. Subramaniam. 1993. Machine learning approaches to protein feature prediction. *International Journal of Neural Systems*. Suppl: 183-194.

[Voet and Voet 1990] Voet, D., and J. Voet. 1990. *Biochemistry*. New York: John Wiley & Sons, pp. 144–210.