Symbolic Protein Data Base

Igor Mozetič
Center for Applied Mathematics and Theoretical Physics
University of Maribor
Krekova 2, 2000 Maribor, Slovenia
and
Jožef Stefan Institute, Department of Intelligent Systems
Jamova 39, 1000 Ljubljana, Slovenia
igor.mozetic@ijs.si

Milan Hodoščček
National Institute of Chemistry,
Laboratory of Molecular Modeling and NMR Spectroscopy
Hajdrihova 19, 1000 Ljubljana, Slovenia
milan@kihp6.ki.si

January 1997

Abstract

The report describes the process of selecting and transforming protein data with known 3D structures into Symbolic Protein Data Base (SPDB). SPDB is a relational database, used to analyze properties of protein structures and as a source of examples to be used by machine learning. The report gives a detailed structure of SPDB, and some preliminary analysis results.

1 Introduction

Determination of 3D protein structure is of great importance for the prediction of their function in pharmacology and in medicine. So far, there are over 50.000 known proteins, but the structure is known for around 500 only. Experimental methods for protein structure determination (X-ray diffraction, NMR) are expensive and time consuming.

A long-term goal of our project is the prediction of 3D protein structure for a given amino-acid sequence, on the basis of a database of known protein structures. Protein molecules are organized in the following structural hierarchy:

primary — a linear sequence of amino-acids which form a protein.

secondary — hydrogen-bonded and geometrical features of parts of amino-acid sequence, such as an α-helix, a β-strand, a turn, a bend, ...

tertiary — positions of elements of the secondary structure, i.e., coordinates of each protein atom.

quaternary — spatial relations between several amino-acid chains, for multiple-chain proteins.

A short-term goal is the prediction of a secondary structure for a given primary structure. The paper describes the first phase, the selection of an appropriate set of proteins with known structure, and the transformation of their 3D coordinates into Symbolic Protein Data Base (SPDB). SPDB consists of symbolic protein descriptions in terms of their relevant properties and in the form of a relational database, implemented as a Prolog program [5]. Prolog enables flexible analysis of SPDB, and its use as a source of learning examples for different learning systems.

2 Generation of SPDB

Figure 1 gives an overview of the selection and transformation of the protein data.

2.1 Brookhaven Protein Data Bank (PDB)

PDB [4] consists of detailed 3D coordinates of all proteins with known structure. Coordinates are given, in principle, for all atoms of amino-acid residues which constitute proteins. However, often some atoms are missing, their positions may be determined at low resolution, or even entire residues in a chain may be missing (broken chains). Currently, PDB consists of around 5000 proteins, but majority of them are only small variations in residues. There are about 500 considerably different proteins with structure determined at relatively high resolution.

PDB files can be fetched from *http://www.pdb.bnl.gov*.

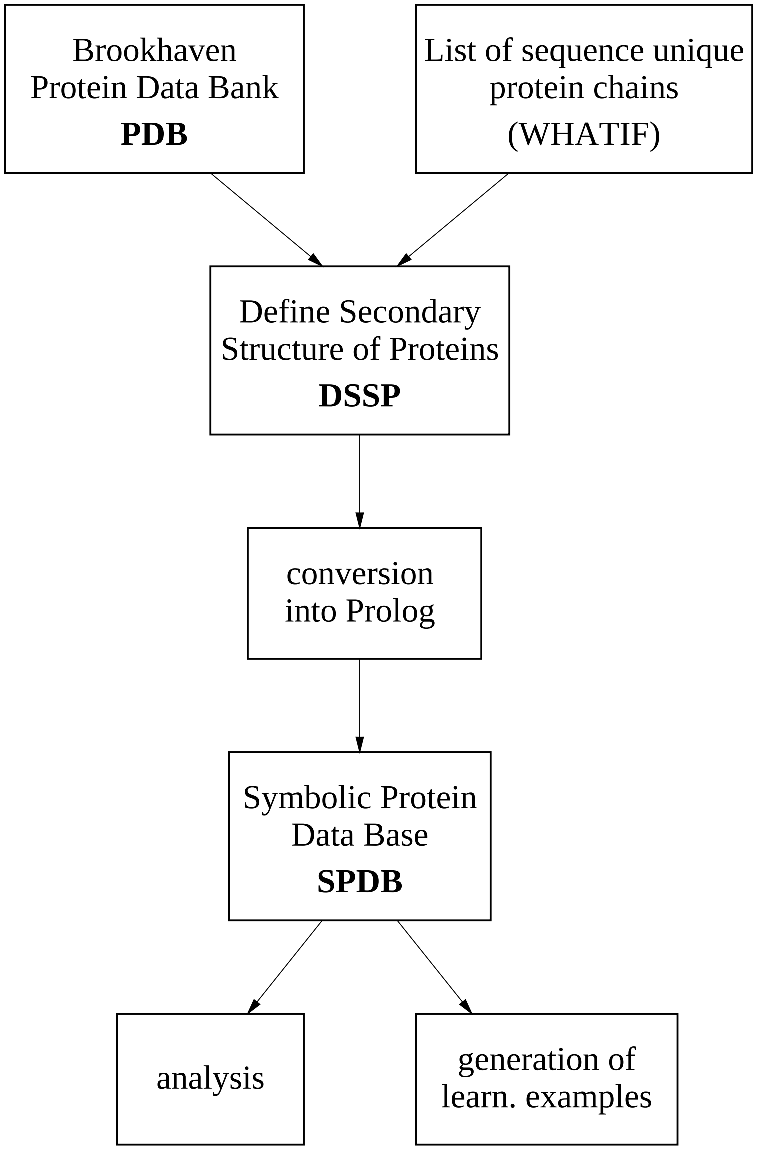

Figure 1: An overview of the procedure for protein structure analysis.

2.2 List of sequence unique protein chains

A subset of PDB protein chains is selected by the authors of WHATIF [6] and used in the WHATIF relational database. The selection is a representative set of sequence-unique chains generated from the X-ray protein PDB files available at a certain moment.

The procedure used to generate this database is similar to the *PDB select* algorithm, but rather than focusing on maximum size of the subset, this algorithm focuses on getting representative structures of the highest available quality. For the selection an empirical quality value is defined: a composite score depending on the Resolution and the R-factor (to be published). The criteria for selecting protein chains is made more strict whenever the number of selected chains would otherwise be higher than 320.

In the list, each protein chain is identified by the 4-letter PDB identifier, plus (if applicable) a one-letter chain name. Protein chains are ordered by decreasing quality value.

The list is updated approximately once per month, and can be fetched from *http://www.embl-heidelberg.de/whatif/select.*

2.3 Define Secondary Structure of Proteins (DSSP)

DSSP [2] is a program which computes secondary structure and solvent exposure (accessible surface) of proteins from atomic coordinates as given in the PDB format. Level of detail is reduced from individual atoms to residues. Each residue is assigned a position identifier in the sequence. *Note:* this might be different from the PDB position and there might be breaks in chains. There is also a number of other residue features, like coordinates of Cα atoms (to which residue side chains are attached). For this work, of interest is only the secondary structure assignment, Φ and Ψ angles, and accessible surface of individual residues (given in Å^2).

2.4 Conversion into Prolog

From the DSSP files, we extracted and converted the above mentioned relevant features into the Prolog format. In addition, for each residue, a set of residues within the radius of 7 Å and at least two positions away in the sequence, is computed. More precisely, computed are distances between corresponding Cα atoms. The disulfide bridges between pairs of cysteine residues are also extracted from the DSSP files.

The Prolog files are at the National Institute of Chemistry (NIC) and can be ftp-ed from: *~milan/proj/fold/pdb/*.pl.*

2.5 Symbolic Protein Data Base (SPDB)

SPDB consists of three files: *pc_select.pl, spdb.pl,* and *aa_back.pl.*

2.5.1 *pc_select.pl* is a list of sequence unique protein chains in the Prolog form. It is converted from the WHATIF list (section 2.2) by the *conv_pc.exe* program. It defines a binary predicate which enumerates all selected protein chains:

```
prot_chain( Pr, Ch )
```

2.5.2 *spdb.pl* is simply a concatenation of all Prolog files from NIC. Alternatively, all the files can be consulted by the *spdb_consult.pl* program. In *spdb.pl* there are three predicates defined:

```
struct( Pr, Ch, N, AA, SS, Phi, Psi, Acc )
dist( Pr, N, [N1:D1, N2:D2 | Ds] )
ssbond( Pr, N, N1:D1 )
```

Predicate arguments have the following interpretation:

`Pr`	— PDB 4-letter protein name with prefix 'p'
`Ch`	— PDB one-letter chain name, or '0' for single chain proteins
`N,Ni`	— DSSP assigned residue position in a protein
`Di`	— distance (in Å) between Cα atoms at positions N and Ni
`AA`	— one-letter amino-acid name
`SS`	— DSSP secondary structure assignment
`Phi`	— DSSP Φ angle in degrees
`Psi`	— DSSP Ψ angle in degrees
`Acc`	— DSSP computed accessible surface in Å^2

`struct/8` defines properties of an amino-acid `AA` at position `N` in a `Protein Chain`: `SS, Phi, Psi` and `Acc`.

`dist/3` defines a list of residues which are within the 7 Å radius of the residue `N` in a `Protein`, but at least two positions away in the sequence. With each residue `Ni` its distance `Di` to `N` is associated. The relation is commutative.

`ssbond/3` defines a disulfide bridge in a `Protein` between cysteines at positions `N` and `N1`, at a distance `D1`. The relation is not commutative.

The predicates `struct/8, dist/3, ssbond/3` are defined for entire `Protein`s (for all chains in the case of multi-chain proteins) from the `prot_chain( Pr, Ch )` relation, and not just for the selected `Chain`s!

2.5.3 *aa_back.pl* defines some background knowledge about amino-acids and secondary structures. First, all twenty amino-acid names (one-letter, 3-letter and normal) are defined:

```
amino_acid( a, ala, alanine ).
amino_acid( c, cys, cysteine ).
amino_acid( d, asp, aspartic_acid ).
amino_acid( e, glu, glutamic_acid ).
amino_acid( f, phe, phenylalanine ).
amino_acid( g, gly, glycine ).
amino_acid( h, his, histidine ).
amino_acid( i, ile, isoleucine ).
amino_acid( k, lys, lysine ).
amino_acid( l, leu, leucine ).
amino_acid( m, met, methionine ).
amino_acid( n, asn, asparagine ).
amino_acid( p, pro, proline ).
amino_acid( q, gln, glutamine ).
amino_acid( r, arg, arginine ).
amino_acid( s, ser, serine ).
amino_acid( t, thr, threonine ).
amino_acid( v, val, valine ).
amino_acid( w, trp, tryptophan ).
amino_acid( y, tyr, tyrosine ).
```

Secondary structure assignments, as used by DSSP, are defined:

```
sec_struct( h, helix_4 ).     % ’H’ alpha-helix
sec_struct( b, beta_1 ).      % ’B’ isolated beta-bridge
sec_struct( e, beta_strand ). % ’E’ extended strand, in beta-ladder
sec_struct( g, helix_3 ).     % ’G’ 3-helix
sec_struct( i, helix_5 ).     % ’I’ pi-helix
sec_struct( t, turn ).        % ’T’ H-bonded turn
sec_struct( s, bend ).        % ’S’ bend
sec_struct( r, random ).      % ’ ’ no assignment
```

Finally, there is a number of binary predicates which define some chemical, physical and geometrical properties of amino-acids. A predicate name corresponds to an attribute (as used in learning), the first argument is a 3-letter amino-acid name, and the second is the attribute value, i.e., `Attribute( AAA, Value )`. Note that these properties were not yet verified by any protein folding expert, and should be used with caution! In the following, the background knowledge predicates are illustrated on the *asparagine* example.

```
access( asn, 16 ).
```

defines accessible surface as percentage of the average (through selected protein chains) to maximum (for an isolated amino-acid) and intervalized into intervals of width 8. I.e., $Xint = Val \implies Val \leq X < Val + 8$.

```
    hydro( asn, philic ).
```

defines hydrophobicity or hydrophylicity, as used by GOLEM [3].

```
    size( asn, small ).
```

defines qualitative size of an amino-acid, as used by GOLEM.

```
    sidech( asn, 4 ).
```

defines the number of side-chain atoms, without hydrogens.

```
    charge( asn, no ).
```

defines the amino-acid charge, as used by CHARMM [1].

```
    polar( asn, neg ).
```

defines the polarity. All charged amino-acids are also polar, and some neutral may also be polar.

```
    shape( asn, normal ).
```

defines the shape (aromatic, aliphatic or normal), as used by GOLEM.

```
    hbond_don_acc( asn, 2, 1 ).

    hdon( AAA, N ) :- hbond_don_acc( AAA, N, _ ).
    hacc( AAA, N ) :- hbond_don_acc( AAA, _, N ).
```

define the number of hydrogen-bond donors and acceptors in the side-chain, as used by CHARMM.

3 Analysis of SPDB

The SPDB analysis programs are in two files: *analys.pl* and *gplot.pl.*

3.0.1 *analys.pl* consists of a number of procedures, which write the analysis results into the following (recommended) files:

```
    counts.pl    — basic counts
```

`freq_aa.pl`	— frequencies of AA
`freq_ss.pl`	— frequencies of SS
`pc_gaps.pl`	— duplicates, gaps in protein chains
`access.pl`	— accessible surface for AA
`ramach.pl`	— 3D Ramachandran Φ, Ψ distributions for SS
`phi_psi.pl`	— 2D Φ, Ψ distributions for SS
`aa_dist.pl`	— AAs within distance of AA
`prot_ssX.pl`	— compact representation of SS for protein chains
`ss_aa.pl`	— distributions of AAs for SS

3.0.2 *gplot.pl* reads the analysis results from the above files and transforms them into the GNUPLOT data. The data files are for individual amino-acids or secondary structures, and are in different directories:

`/surf`	— histograms of accessible surfaces for AA
`/angles`	— 2D and 3D Φ, Ψ plots for SS
`/dist`	— histograms of AAs close to AA
`/ssaa`	— distributions of AAs for SS

Have a look at the above files and README for more detailed information. Try GNUPLOT in the above mentioned directories.

4 Generation of learning examples

Program which generates learning examples from SPDB is in the *gen_sds.pl* file. At this stage, learning examples are generated for the SDS [7] attribute-value learning program. This is due to the fact that we have to pre-process a large number of learning examples (over 50.000) with an efficient system, before we can proceed to an ILP system.

There are two user definable predicates which affect the generation of learning examples.

```
seq_window( -4, 5 ).
```

defines the window size of a sequence of residues. With the above definition, 4 residues before, and 5 residues after the selected residue, i.e. 10 in total, form an individual example.

```
attributes( Class, Attributes )
```

defines the `Class` (decision, dependent) attribute and a list of independent `Attributes` with which examples are described. For example:

```
attributes( ss8,
  [access, hydro, size, sidech, charge, polar, shape, hdon, hacc] ).
```

defines a binary predicate `ss8/2` as a class, and all the background knowledge predicates from *aa_back.pl* as independent attributes. `ss8/2` is defined by:

```
ss8( SS, SS ) :- sec_struct( SS, _ ).
```

Alternatively, the user can specify the following:

```
attributes( ss3, [aa] ).
```

where `aa/2` corresponds to a 3-letter amino-acid name:

```
aa( AAA, AAA ) :- amino_acid( _, AAA, _ ).
```

and `ss3/2` maps 8 different secondary structures into 3:

```
ss3( h, turn ).
ss3( b, beta ).
ss3( e, beta ).
ss3( g, turn ).
ss3( i, turn ).
ss3( t, turn ).
ss3( s, rand ).
ss3( r, rand ).
```

For efficiency reasons, first all the possible attribute values are encoded into the *codes.pl* file. The file has to be regenerated whenever attributes or their values are changed.

5 Summary

The report represents the state-of-the-art of SPDB in January 1997. To generate the SPDB, we used a list of 253 PDB chains, created by the WHATIF authors from the PDB on November 12, 1996. There are over 51.000 amino acids in the selected chains. The SPDB, consulted as a Prolog program, requires 30 Mb of storage.

Acknowledgment

This work was supported by the Department of Intelligent Systems, IJS, and Laboratory of Molecular Modeling and NMR Spectroscopy, NIC, as part of the ESPRIT IV project 20237 ILP2.

References

[1] Brooks, B.R., Bruccoleri, R.E., Olafson, B.D., States, D.J., Swaminathan, S., Karplus., M. CHARMM: A program for macromolecular energy, minimization, and dynamics calculations, *J. Comp. Chem. 4(2)*, 187–217, 1983.

[2] Kabsch, W., Sander, C. Dictionary of protein secondary structure: Pattern recognition of hydrogen-bonded and geometrical features. *Biopolymers 22*, 2577–2637, 1983.

[3] Muggleton, S., King, R.D., Sternberg, M.J.E. Protein secondary structure prediction using logic. *Prot. Eng. 5(7)*, 647–657, 1992.

[4] Protein Data Bank, Brookhaven National Lab, USA, http://www.pdb.bnl.gov.

[5] SICStus Prolog user's manual, Release 3.0. Swedish Institute of Computer Science, Kista, Sweden, 1995.

[6] Vried, G. WHATIF, http://www.sander.embl-heidelberg.de/whatif.

[7] Zupan, B., Bohanec, M. Learning concept hierarchies from examples by function decomposition. IJS-DP–7455, Jozef Stefan Institute, Slovenia, 1996.

A Distribution of accessible surfaces per amino-acid

B Distribution of Φ angles per secondary structure

C Distribution of Ψ angles per secondary structure

D Number of close neighbors (within 7 Å radius, but at least 5 positions away in the sequence) per amino-acid

E Normalized number of amino-acids (within 7 Å radius, but at least 5 positions away in the sequence) per amino-acid

F Relative distribution of amino-acids per secondary structure