ALBERT-LUDWIGS-UNIVERSITÄT
FREIBURG
INSTITUT FÜR INFORMATIK

Lehrstuhl für Mustererkennung und Bildverarbeitung
Prof. Dr.-Ing. Hans Burkhardt

Algorithms for Protein Retrieval and Classification based on Structural Features

Diplomarbeit

von
Maja Temerinac

Betreuer: Dipl.-Inf. Marco Reisert
1. Gutachter: Prof. Dr.-Ing. H. Burkhardt
2. Gutachter: Prof. Dr. R. Backofen

Januar 2006 – Juli 2006

Erklärung

Hiermit erkläre ich, dass ich diese Abschlussarbeit selbständig verfasst habe, keine anderen als die angegebenen Quellen/Hilfsmittel verwendet habe und alle Stellen, die wörtlich oder sinngemäß aus veröffentlichten Schriften entnommen wurden, als solche kenntlich gemacht habe. Darüber hinaus erkläre ich, dass diese Abschlussarbeit nicht, auch nicht auszugsweise, bereits für eine andere Prüfung angefertigt wurde.

Maja Temerinac

Freiburg, den 30. Juni 2006

Danksagung

*Ich möchte mich herzlich bei **Prof. Dr.-Ing. H. Burkhardt** für die Möglichkeit bedanken, diese eher untypische Arbeit am Lehrstuhl für Mustererkennung und Bildverarbeitung anfertigen zu dürfen.*

*Ein großer Dank gebührt meinem Betreuer **Dipl.-Inf. Marco Reisert**, der immer da war, wenn ich fragen hatte, sei es programmiertechnischer, mathematischer oder allgemeiner Natur. Er hat mich nie unter Druck gesetzt und mir Freiräume gelassen, um mit eigenen Ideen zu spielen. Danke auch für das gründliche und kritische lesen dieser Arbeit.*

*Ich danke auch **Prof. Dr. R. Backofen** für die Übernahme des zweiten Gutachters.*

*Danke **Margret**, für das Ausmerzen meiner englischen Sprachfehler und dafür, dass ich wegen Dir gerne in dem Diplomandenpool gearbeitet habe.*

*Bei meiner Schwester **Dunja** möchte ich mich für die Erklärungen der chemischen Struktur und Funktion von Proteinen bedanken. Danke, dass Du an mich glaubst!*

*Meinem Freund **Christian** danke ich dafür, dass er immer da ist und zu mir hält.*

*Last but not least möchte ich noch **meinen Eltern** danken, dass sie mich immer in jeder Hinsicht unterstützt haben.*

ALBERT-LUDWIGS-UNIVERSITÄT FREIBURG
INSTITUT FÜR INFORMATIK
Lehrstuhl für Mustererkennung und Bildverarbeitung
Prof. Dr.-Ing. Hans Burkhardt

Georges-Köhler-Allee
Geb. 052, Zi 01-029
D-79110 Freiburg
Tel. 0761 - 203 - 8260

Aufgabenstellung für die Diplomarbeit von
Frau Maja Temerinac

Algorithmen zur merkmalsbasierten Proteinsuche und -klassifikation

Klassischerweise werden Proteine anhand ihrer Aminosäuresequenz verglichen, jedoch ist die funktionale Einordnung der Proteine stark durch ihre dreidimensionale Struktur gegeben. Es ist bekannt, dass auch bei sehr geringer Sequenzähnlichkeit (unter 25%), Proteine sehr ähnliche räumliche Strukturen aufweisen und somit auch funktionale Zusammenhänge bestehen. Typischerweise werden strukturelle Vergleiche mittels 'Alignment'-Methoden vorgenommen. Es wird versucht die zu vergleichenden Proteine hinsichtlich der euklidischen Bewegung optimal aneinander auszurichten (z.B. DALI, SSM), der resultierende Fehler wird als Ähnlichkeitsmass benutzt. Solche Methoden sind jedoch sehr zeitaufwendig und rechenintensiv. Eine Methode um das 'Alignment' zu umgehen ist die Extraktion von Merkmalen, die invariant gegenüber Lagetransformationen sind.

In dieser Arbeit sollen aufbauend auf [1, 2] Möglichkeiten zur Verbesserung der Proteinsuche und -klassifikation untersucht werden. Dies kann beinhalten, sowohl eine verbesserte Merkmalsextraktion, als auch bessere Distanzmaße sowie Merkmalstransformation und -selektion. Die Ergebnisse sollen zum Beispiel mit DALI oder PRIDE verglichen werden.
Literatur:

[1] Maja Temerinac, *Invariant Feature Extraction for Protein Fold Recognition* Studienarbeit, Albert-Ludwig-Universität Freiburg, 2005, temerina@informatik.uni-freiburg.de

[2] Marco Reisert and Hans Burkhardt, *Second Order 3D-Shape-Features: An Exhaustive Study*, accepted for publication in Special Issue of Computers & Graphics on Shape Reasoning and Shape Understanding, 2005

Referent:	Prof. Dr.-Ing. H. Burkhardt
Betreuer:	Dipl.-Inf. M. Reisert
Ausgabedatum:	02. 01. 2006
Abgabedatum:	02. 07. 2006
Bearbeitungszeit:	6 Monate

Abstract

Most methods for protein functional classification rely on the comparison of their amino acid sequence. However functional classification of proteins is strongly defined by their three dimensional structure. This approach is based on the fact that proteins from one fold keep the same structure even after evolutionary changes of their sequence. The underlying theme of the work is: 'can we find a unique invariant representation for the structure of each protein fold?'.

A new method for protein structure description based on Group Integration (GI) is presented. Since spatial information is generalized using GI, Spherical Harmonics are added to retain rotational information. These structural features are evaluated on well-known testing sets from SCOP and CATH databases and their performance is compared to existing protein classification algorithms like DALI and PRIDE. Compared to DALI the proposed method has significantly lower time consumption while the classification results are slightly worse. The PRIDE algorithm yields worse results then the proposed method.

The results achieved in this work can compete and in some cases outperform the established methods considering the classification accuracy and the time requirements. The method can be scaled according to different applications by adjusting the feature size and implementing different kernels.

Zusammenfassung

Die meisten Methoden zur funktionalen Proteinsuche und -klassifikation betrachten nur die Aminosäurensequenz und nicht die dreidimesionale Struktur eines Proteins. Die Funktion eines Proteins ist allerdings sehr stark durch seine dreidimensionale Struktur festgelegt. Der Grund für diese Sichtweise ist, dass Proteine aus einer Faltungsklasse dieselbe Struktur bewahren, selbst nachdem sich während der Evolution die primäre Sequenz des Proteins verändert hat. Das zentrale Thema dieser Arbeit ist: Können wir eine invariante Repräsentation für jede Proteinfaltungsklasse finden?

Eine neue Methode zur Beschreibung von Proteinstrukturen basierend auf Gruppenintegralen (GI) wird vorgestellt. Da mit Hilfe von GI die räumliche Information verallgemeinert wird, werden Kugelflächenfunktionen (Spherical Harmonics) hinzugenommen um Rotationsinformationen zu behalten. Diese strukturellen Merkmale werden auf wohl bekannten Testdatensätzen von den SCOP und CATH Datenbanken ausgewertet und ihre Leistung mit den bereits existierenden automatischen Klassifikationsalgorithmen DALI und PRIDE verglichen. Im Vergleich zu DALI hat die vorgeschlagene Methode wesentlich geringere Zeitanforderungen, aber die Klassifikationsergebnisse sind etwas schlechter. Der PRIDE Algorithmus liefert schlechtere Ergebnisse als die vorgeschlagene Methode.

Die Ergebnisse dieser Arbeit können sich mit den bewährten Methoden messen und sie in manchen Fällen hinsichtlich der Klassifikationsgenauigkeit und der Zeitanforderungen sogar überbieten. Die Methode kann für verschiedene Anwendungen angepasst werden, indem man die Größe der Merkmale und die Kernelfunktion den Anforderungen anpasst.

Contents

Chapter 1

Introduction

If we could not only describe the [protein] sequences but also pronounce the law by which they assemble, the secret of life would be declared open - the ultima ratio discovered!

Jaques Monod (1970)
from *Chance and Necessity, P.94*

1.1 Proteins

Proteins are the main essential active agents in biochemistry: without them almost none of the metabolic processes that we associate with life would take place. Therefore, most reviews on proteins concentrate on their chemical kinetics, interactions and detailed stereochemical arrangement of the catalytic groups. In this work, however, proteins will be viewed from a different angle - their biology and chemistry will be completely ignored. Instead, their overall structure will form the central topic of investigation. This approach is based on the fact that proteins from one fold keep the same structure even after evolutionary changes of their biochemical relations.

The underlying theme of the work is : 'can we find a unique invariant representation of the structure of each protein fold?'. If we can find such a 'fingerprint' for each protein fold, we could draw a complete map of the protein structure space making it easy to classify new proteins and also contribute to answering the question: 'which structural forms can proteins adopt at all?'

1.1.1 Protein folding

In Figure 1.1 the core machinery of life is depicted which is inherent in every cell. This machinery has two main functions: To transform DNA [1] to RNA [2] and to translate RNA to proteins. The double stranded DNA is transformed by the enzyme polymerase to single stranded RNA which in turn is translated by the ribosome to an amino acid sequence. The

[1] Deoxyribonucleic acid
[2] Ribonucleic acid

main building blocks of proteins are amino acids which are generated from RNA via the genetic code. As shown in Figure 1.1 one amino acid is coded by three nucleotides.

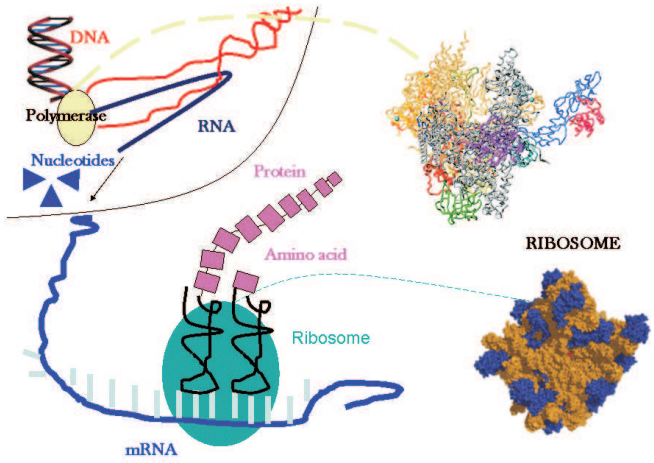

Figure 1.1: **Core Machinery of Life.** DNA is transformed to RNA and RNA is translated to amino acids which are the building blocks of proteins.

Although the genetic code is known since 1961 [3], the prediction of the three dimensional structure a protein will fold into is still a large field of research. The interaction between the atoms of the amino acids and different forces acting on them are so complex that it is impossible to predict their behavior. Only by examining already known protein structures we can guess what is going to happen during the folding process.

1.1.2 Relation between structure and function

Why do we want to know a protein's fold? If we could predict the protein's fold from its amino acid sequence, we would be in a much better position to predict the protein's function as well. The binding site of a protein has a particular shape which enables it to bind to a ligand, a virus or other molecules. Thus, in order to perform a certain function, a protein needs to have an adequate shape. Throughout evolution proteins did mutate in order to achieve the desired shape by changing the amino acid sequence which uniquely defines the protein's structure. The cycle of life at the molecular level as described by Michael Levitt is depicted in Figure 1.2.

1.2 Pattern Recognition

The main topic of pattern recognition is the operation and design of systems that recognize patterns in data. It encloses subdisciplines like discriminant analysis, feature extraction, error estimation, cluster analysis (together sometimes called statistical pattern

[3]The German biochemists H. Matthaei and M. Nirenberg solved the genetic code in Mai 27th 1961 at 3 a.m.

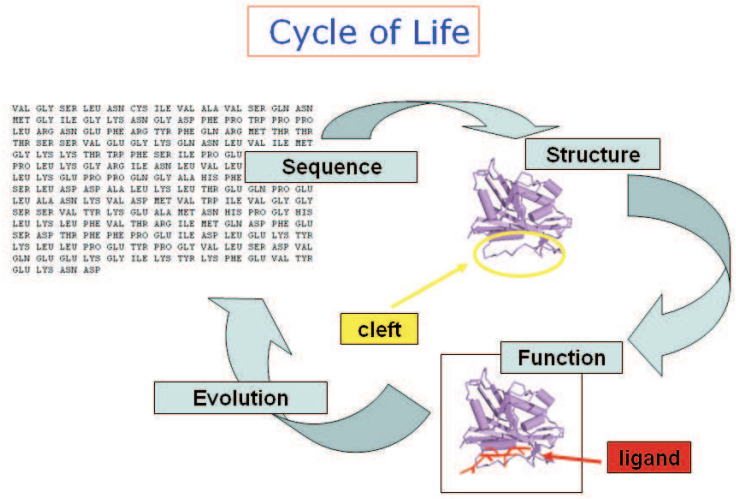

Figure 1.2: **Cycle of Life.** The structure is defined by a unique amino acid sequence and imposes a function on the protein. Through evolution the primary sequence has changed to achieve the necessary function. Here the cleft in the structure of the protein is the perfect binding site for a ligand.

recognition) and grammatical inference and parsing (sometimes called syntactical pattern recognition). Important application areas are image analysis, character recognition, speech analysis, man and machine diagnostics, person identification and industrial inspection.

In the following areas, closely related systems are studied or similar tools are developed:

- Artificial Intelligence (expert systems and machine learning)
- Neural Networks
- Computer Vision
- Cognitive Sciences and Biological Perception
- Mathematical Statistics (hypothesis testing and parameter estimation)
- Nonlinear Optimization
- Exploratory Data Analysis

In this work pattern recognition is used to describe and classify protein structures. The process of description and classification of new protein structures can be enhanced by finding intelligent data structures and fast classification algorithms. The generation of automatic models relies on expert opinion and does not make experts obsolete. Rather it should improve the classification and help on getting more insight into the protein structure universe.

1.3 Protein Shape Recognition

The protein shape recognition approach used in this work could be roughly described as in Figure 1.3. In a preprocessing step, protein structures are solved by crystallographers and the 3D coordinates of the protein structures as well as further information is written in a text file called PDB File. The PDB Files are freely available in the internet and are downloaded for further processing to a protein database. For each structure of the protein database features are extracted and saved to a feature database. When a user poses a query for a protein structure, the classifying server returns a list of similar structures. If the structures have already been classified according to some experts, the quality of the returned results can be evaluated and then further processed to select the features which yielded the best results.

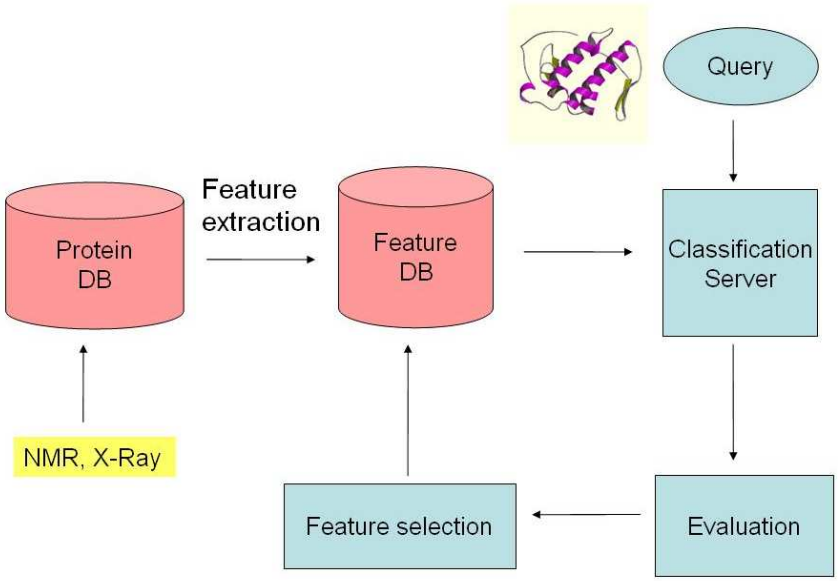

Figure 1.3: **Protein Structure Recognition.** The pattern recognition process used for classification of protein structures. After feature extraction structural information is saved in a protein database. When a user performs a structural query, a classification server outputs the most similar structures to the query structure. The output is evaluated and with this knowledge the features can be reselected.

1.4 Goal of this work

The number of currently known protein structures is 40,000 and still increasing by over 1000 new solved protein structures per year. In order to organize the protein structure space, different methods have been applied and different protein organization platforms already exist in the internet (See Chapters 3 and 4 for examples).

In former applications, protein structures were compared using alignment techniques [13, 18] or by visual inspection of experts [20]. Also techniques from machine learning

and artificial intelligence [14] were applied which use some learning network for classification. Statistical methods were applied in PRIDE [4] by biologists, while the Gauss integral [26] was used by mathematicians in order to describe proteins.

A good overview of protein structure classification is given in [33]. The desired properties of a protein structure comparison algorithm are listed in [6]. These properties are:

1. insertions or deletions to the sequence should not be penalized too heavily
2. reasonably robust algorithm: small pertrubations of the definition should not make too much difference in the measure
3. easy to compute
4. able to discover local and global alignments
5. its success should be validated by empirical studies on the PDB
6. acceptance by the protein scientists

Regarding these aspects, a new method for protein structure comparison was developed.

The approach of this work is completely new since as to the author's knowledge no other method has been introduced which uses invariant theory to describe proteins. We started our work with the idea of shape histograms describing protein structure [34] and extended the idea to group integrals with spherical harmonics [24]. In order to describe protein structure only structural features should be used.

Our goal is to establish a scalable method for protein structure description providing every wished trade-off between quality and complexity.

1.5 Structure of this work

After the introduction and the motivation of this work, the document is organized as follows:

- **Chapter 2:** The basic knowledge about protein's chemistry/biology is presented.
- **Chapter 3:** Protein structures can further be grouped to form protein classification hierarchies. Classification can either be performed by human experts (SCOP and CATH) or by automatic classification (DALI) hierarchies. SCOP and CATH are explained in this chapter.
- **Chapter 4:** Automatic classification methods and the DALI hierarchy are explained and discussed.
- **Chapter 5:** The new method for protein structure classification is presented.
- **Chapter 6:** A benchmark of the existing methods and the new method is computed.
- **Chapter 7:** Conclusions and an outlook is presented.

Chapter 2

Protein Structure

The data used in this thesis consists of coordinates and sequence numbers of atoms building a protein structure. This information is gained from PDB files which are publicly available from the RCSB (Research Collaboratory for Structural Bioinformatics) Protein Database (PDB) homepage [1].

In the first part of this chapter the protein structure hierarchy is explained. In the second part the PDB file format and the parsing of the relevant information will be spotlighted.

2.1 Protein Architecture

The protein structure can be described at four levels of detail starting with the primary structure at the very basic level and leading to the quaternary structure being the most global description level of a protein structure.

2.1.1 Primary Structure

The *primary structure* of a protein is defined by the sequence of amino acids on its chain. An amino acid is made up of a C_α-atom (the central atom), a hydrogen atom (H) attached to the C_α-atom, a carboxyl group (COO^-), an amino group (NH_3^+) and a side chain (R) (See Figure 2.1). The side chain determines the chemical property of the amino acid. Removing side chains leaves us with linked repeating units $NH - C_\alpha H - CO$, which make up what is called the 'main chain' or 'backbone' of the protein.

A variety of side chain conformations can be seen in proteins because rotation can occur around many side chain bonds. However, most amino acids have a restricted set of preferred conformations, called rotamers.

[1] http://www.pdb.org

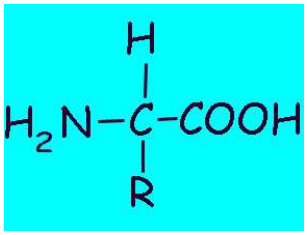

Figure 2.1: **Amino acid structure.** The chemical structure of an amino acid. In the middle the C_α atom, left the amino terminal, right the carboxyl terminal. H denotes the hydrogen atom, R the side chain.

There are 20 amino acids which can occur in proteins. They can be divided into three groups according to their chemical properties: hydrophobic, charged and polar (See Appendix A). Hydrophobic amino acids try to keep away from water and are therefore found in the core of a protein.

The amino acids in a protein are held together by peptide bonds (Figure 2.2). A peptide bond is a chemical bond formed between two molecules when the carboxyl group of one molecule reacts with the amino group of the other molecule, releasing a molecule of water (H_2O). Peptide bonds are very stable: There is no rotation around the C-N bond.

Two or more amino acids linked by a peptide bonds are referred to as a peptide. Once an amino acid is incorporated into a peptide, it is referred to as an amino acid residue.

The peptide bond has partial double-bond character because of resonance between the carbonyl and amide groups. The double-bond character inhibits rotation around the peptide bond. As a consequence, atoms participating in the bond and all atoms within one bond of it are restricted to the same plane.

While peptide bonds resist rotation, most $N-C_\alpha$ and $C_\alpha-C$ bonds are only bounded by steric constraints. Rotation about a $N-C_\alpha$ bond is described by an angle called ϕ, and rotation about a $C_\alpha-C$ bond is described by an angle called ψ. Steric interference between backbone and side chain atoms restricts ϕ and ψ rotation.

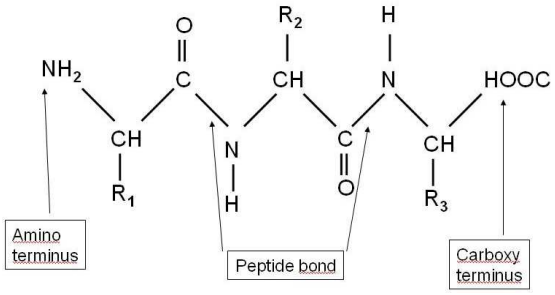

Figure 2.2: **Peptide bond.** The carboxyl group of one molecule reacts with the amino group of the other molecule to form a peptide bond.

One of the most important principles in understanding protein structure is that most combinations of ϕ and ψ are unfavorable for steric reasons. There are, however, two broad sets of permitted combinations: regular *secondary structures* are formed by repetition of these conformations - in one case, an $\alpha-$helix, and in the other , a $\beta-$sheet.

2.1.2 Secondary Structure

Three types of regular arrangements are dominating the protein *secondary structure*: the α*-helix*, the β*-sheet* and the β*-turn*.

α-helices are common structures in proteins partly because the ϕ and ψ angles required to form them are favorable (see Section 2.1.1). In addition, the conformation allows stabilizing hydrogen bonds to form between the amide nitrogens and carbonyl oxygens of residues close together in the sequence. Hydrogen bond acceptors and donors in an α-helix exhibit a characteristic spacing, by convention referred to as i, i+4. The carbonyl oxygen of each residue (i) accepts a hydrogen bond from the amide nitrogen located four residues further along ($i + 4$) in the sequence. Repetition of the i, $i + 4$ pattern covers the length of the α-helix with hydrogen bonds running roughly parallel to the helix axis (See Figure 2.3).

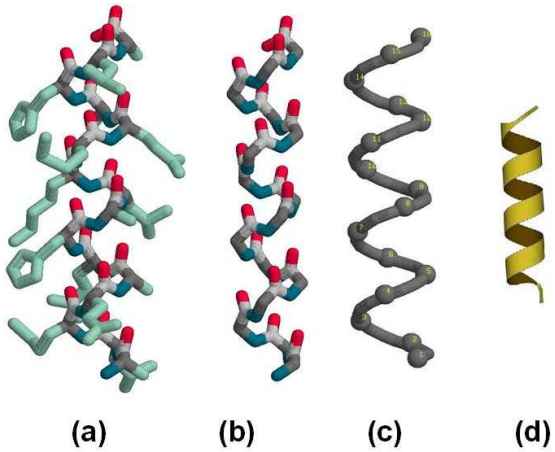

Figure 2.3: **Alpha helix.** Different representations of one α-helix: amino acids are represented by (a) all atoms (b) only their backbone (c) only α-atoms (d) as a cartoon. Images from [1] with Chime.

The other common secondary structure found in proteins - the β-sheet - is made up of smaller structures called β strands. Like α-helices, β-strands are formed by repetition of a favored amino acid conformation. In this case, repetition leads to an extended conformation in which the side chains project out more or less on alternating sides of the backbone. A broad arrow is often used to schematically represent a β-strand. The arrow points in the direction of the carboxyl terminus (See Figure 2.4 (b)). Hydrogen bonding groups of the backbone lie more or less in the plane of the arrow. Turning off side chains and turning on oxygen and nitrogen atoms makes this easier to see (Figure 2.4 (c)). β-strands hydrogen bond with each other to form β-sheets. In an antiparallel sheet arrangement, strands run in opposite directions (Figure 2.4 (d)). However, parallel arrangements are also possible.

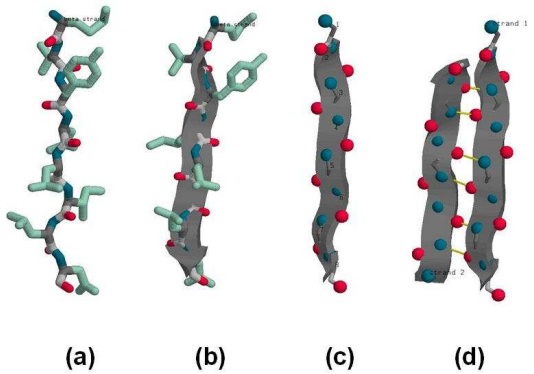

Figure 2.4: **Beta sheet.** Different representations of a β-strand (a)-(c) and the forming of a β-sheet (d). Images from [1] with Chime.

Many antiparallel sheets are connected by sharp turns, called *β-turns* (See Figure 2.5). Several common and well-defined β-turn structures have been identified; most have characteristic sequence features. For example, some types of turns require a glycine at a certain position. Strand pairs that are parallel or that are not contiguous in the polypeptide sequence must be connected by something other than a β-turn, for example, by an α-helix.

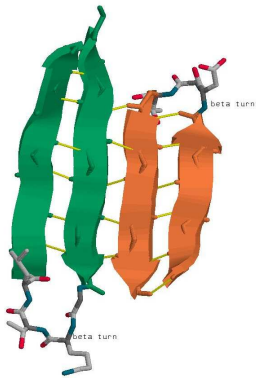

Figure 2.5: **Beta turn.** A β-turn connects two antiparallel β-strands. Image from [1] with Chime.

2.1.3 Tertiary Structure

The global structure of a polypeptide chain is called *tertiary structure* or *fold*. A globular structure buries many of the protein's atoms in the interior, and isolates them from the surrounding medium. Partitioning of charged and partially charged atoms to the surface and uncharged atoms to the core is an important feature of globular proteins, and thought to be the key to the forming of tertiary structure.

Something that is not apparent from the representations of secondary structures that we looked at in the previous section is the tight packing of atoms within a folded protein (See Figure 2.6). An exception to the rule of tightly packed globularity can be seen in fibrous proteins like collagen. (Another exception are amino acid sequences consisting of only one or a small number of the twenty amino acids. These are often found within

otherwise normal-looking protein sequences, particularly in eukaryotes. They are thought to lack well-ordered structure.)

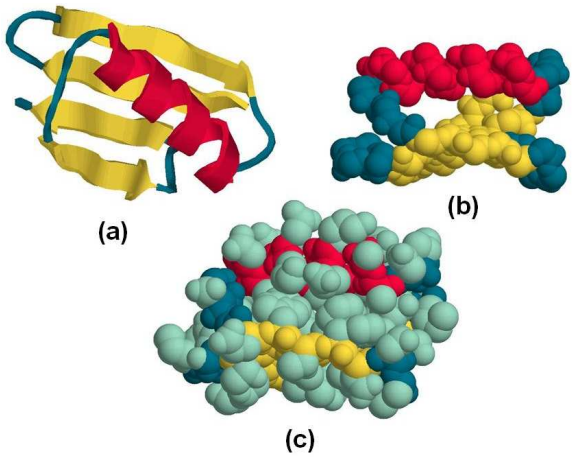

Figure 2.6: **Protein fold.** In (a) the cartoon of a protein structure is shown. The packing of the backbone of the same protein structure is shown in (b). In (c) all the atoms of the protein are pictured revealing a dense packing of the structure. Images from [1] with Chime.

Some polypeptide chains fold into two or more compact globular units which could be imagined as pearls stringed on a chain. These structural units are called *domains*. Domains usually contain between 30 and 400 amino acids. Although the tertiary structure of two proteins differs, they do have similar domains in many cases.

2.1.4 Quaternary Structure

Proteins that have more than one polypeptide chain have a fourth level of structural organization called *quaternary structure*. The polypeptide chain of one protein is called a *subunit*. The quaternary structure describes the spatial organization and the interaction between these subunits. The most basic representative of a quaternary structure is a dimer. A dimer is a protein consisting of two identical subunits. Usually the quaternary structure is more complex since many different subunits can occur.

For example, the human hemoglobin which is the oxygen transporting protein in the blood has two subunits (See Figure 2.7), which are called α-unit and β-unit.

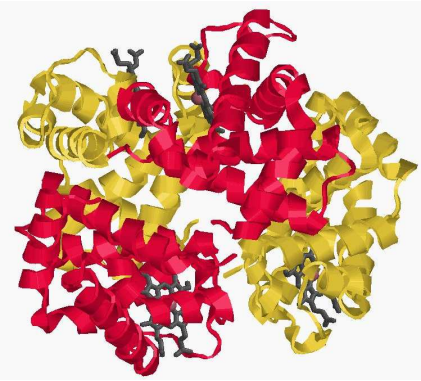

Figure 2.7: **Subunits.** The quaternary structure of hemoglobin consisting of two different types of subunits (red and yellow) is shown. Image from [1] with Chime.

2.2 Protein Representation

One of the most exciting questions for a protein analyst is: How does the three dimensional structure of a protein look like? The structure of a protein determines its function, since the features of the activating center and the binding cites depend on the precise three dimensional conformation of the protein. With the help of NMR[2]-spectroscopy and X-ray crystallography the protein structure can be solved.

NMR spectroscopy is due to a phenomenon based upon the magnetic property of the nucleus of an atom: When the nuclei of certain atoms are immersed in a static magnetic field and exposed to a second oscillating magnetic field, resonance can be observed. The rotation of the nuclei, the spin, of a proton produces a magnetic moment. The magnetic moment can assume two orientations (α and β) if an electro-magnetic radio-frequency (RF) impulse acts on it. We obtain resonance by changing the RF-impulse and can therefore detect different magnetic spectra. NMR can be measured by observing the atomic structure of macro molecules in a very high concentrated solution ($\sim$ 1 mM for a protein with 15kD).

X-ray crystallography is a technique in crystallography in which the pattern produced by the diffraction of X-rays through the closely spaced lattice of atoms in a crystal is recorded and then analyzed to reveal the nature of that lattice. This technique can depict the spatial position of the most atoms in a protein molecule very precisely by applying Bragg's law. In fact the first structure of a protein, namely myoglobin, was solved by X-ray crystallography in 1959.

The Protein Data Bank (PDB) is the single international source for 3D structure files-not only proteins but also nucleic acids and macromolecular complexes. These are experimentally determined structures (solved by X-ray crystallography or NMR spectroscopy) or theoretical models. Structure files are contributed by research labs from around the world and available for viewing or downloading. As to May 2006, there are 36,710 solved structures in the PDB. The PDB structure statistics [3] are pictured in Table 2.1.

[2] nuclear magnetic resonance

[3] http://www.rcsb.org/pdb/holdings.do

Exp. method	
	X-Ray
	NMR
	Electron Microscopy
	Other
	Total

Molecule Type

Proteins	Nucleic Acids	Protein/NA Complexes	Other	**Total**
28,835	899	1,349	28	**31,111**
4,568	699	121	6	**5,394**
88	9	28	0	**125**
73	4	3	0	**80**
33,564	**1611**	**1501**	**34**	**36,710**

Table 2.1: **PDB statistics.** The types of structures in the PDB by May 2006 and the kind of technique used to determine the structures.

In the year 2005, 5431 new structures were added. However, structures were also removed if they had a large similarity to related structures. For the growth statistics of the PDB see Appendix B.

2.2.1 PDB file format

The information stored in a PDB file can be divided into four major sections: title, primary structure, secondary structure and coordinate section. In the *title section* data about the author, the number of chains, functional information as well as the experimental technique with which the structure was obtained is listed. In Figure 2.8 the title section for human hemoglobin is shown.

```
HEADER    OXYGEN TRANSPORT                        07-JUN-02   1GZX
TITLE     OXY T STATE HAEMOGLOBIN: OXYGEN BOUND AT ALL FOUR HAEMS
COMPND    MOL_ID: 1;
COMPND   2 MOLECULE: HEMOGLOBIN ALPHA CHAIN;
COMPND   3 CHAIN: A, C;
COMPND   4 OTHER_DETAILS: LIGANDED T STATE;
COMPND   5 MOL_ID: 2;
COMPND   6 CHAIN: B, D;
COMPND   7 MOLECULE: HEMOGLOBIN BETA CHAIN;
COMPND   8 OTHER_DETAILS: LIGANDED T STATE
SOURCE    MOL_ID: 1;
SOURCE   2 ORGANISM_SCIENTIFIC: HOMO SAPIENS;
SOURCE   3 ORGANISM_COMMON: HUMAN;
SOURCE   4 TISSUE: BLOOD;
SOURCE   5 CELL: RED BLOOD CELLS;
SOURCE   6 MOL_ID: 2;
SOURCE   7 ORGANISM_SCIENTIFIC: HOMO SAPIENS;
SOURCE   8 ORGANISM_COMMON: HUMAN;
SOURCE   9 TISSUE: BLOOD;
SOURCE  10 CELL: RED BLOOD CELLS
KEYWDS    HAEM PROTEIN, OXYGEN BINDING, TRANSPORT, COOPERATIVITY
EXPDTA    X-RAY DIFFRACTION
AUTHOR    M.PAOLI,R.LIDDINGTON,J.TAME,A.WILKINSON,G.DODSON
REVDAT   1   08-JUL-02 1GZX    0
```

Figure 2.8: **Title section.** The title section of the PDB file for hemoglobin is presented.

The *primary structure section* lists the sequence of the amino acids as three letter codes according to the chain they belong to (See Figure 2.9).

The *secondary structure section* reports about the helices, sheets and turns of a protein chain. In every line the start and end amino acid for each secondary structure element of a chain is reported (See Figure 2.10). Hemoglobin consists of 38 α-helices. For each of

```
SEQRES   1 B  146  VAL HIS LEU THR PRO GLU GLU LYS SER ALA VAL THR ALA
SEQRES   2 B  146  LEU TRP GLY LYS VAL ASN VAL ASP GLU VAL GLY GLY GLU
SEQRES   3 B  146  ALA LEU GLY ARG LEU LEU VAL VAL TYR PRO TRP THR GLN
SEQRES   4 B  146  ARG PHE PHE GLU SER PHE GLY ASP LEU SER THR PRO ASP
SEQRES   5 B  146  ALA VAL MET GLY ASN PRO LYS VAL LYS ALA HIS GLY LYS
SEQRES   6 B  146  LYS VAL LEU GLY ALA PHE SER ASP GLY LEU ALA HIS LEU
SEQRES   7 B  146  ASP ASN LEU LYS GLY THR PHE ALA THR LEU SER GLU LEU
SEQRES   8 B  146  HIS CYS ASP LYS LEU HIS VAL ASP PRO GLU ASN PHE ARG
SEQRES   9 B  146  LEU LEU GLY ASN VAL LEU VAL CYS VAL LEU ALA HIS HIS
SEQRES  10 B  146  PHE GLY LYS GLU PHE THR PRO PRO VAL GLN ALA ALA TYR
SEQRES  11 B  146  GLN LYS VAL VAL ALA GLY VAL ALA ASN ALA LEU ALA HIS
SEQRES  12 B  146  LYS TYR HIS
```

Figure 2.9: **Primary structure section.** The primary section of the PDB file for chain B of hemoglobin is presented.

the four chains (A-D) the beginning and the end of each α-helix is listed. The number on the far right denotes the number of residues involved in the particular α-helix.

```
HELIX    1   1 SER A    3  GLY A   18  1                                  16
HELIX    2   2 HIS A   20  PHE A   36  1                                  17
HELIX    3   3 PRO A   37  PHE A   43  5                                   7
HELIX    4   4 SER A   52  HIS A   72  1                                  21
HELIX    5   5 ASP A   75  LEU A   80  1                                   6
HELIX    6   6 LEU A   80  LYS A   90  1                                  11
HELIX    7   7 PRO A   95  LEU A  113  1                                  19
HELIX    8   8 THR A  118  THR A  137  1                                  20
HELIX    9   9 THR B  147  LYS B  160  1                                  14
HELIX   10  10 ASN B  162  TYR B  178  1                                  17
HELIX   11  11 PRO B  179  PHE B  185  5                                   7
HELIX   12  12 PHE B  185  GLY B  189  5                                   5
HELIX   13  13 THR B  193  ASN B  200  1                                   8
HELIX   14  14 ASN B  200  LEU B  218  1                                  19
HELIX   15  15 ASN B  223  PHE B  228  1                                   6
HELIX   16  16 PHE B  228  LYS B  238  1                                  11
HELIX   17  17 PRO B  243  GLY B  262  1                                  20
HELIX   18  18 LYS B  263  PHE B  265  5                                   3
HELIX   19  19 THR B  266  HIS B  286  1                                  21
HELIX   20  20 SER C  403  GLY C  418  1                                  16
HELIX   21  21 HIS C  420  PHE C  436  1                                  17
HELIX   22  22 PRO C  437  PHE C  443  5                                   7
HELIX   23  23 SER C  452  ALA C  471  1                                  20
HELIX   24  24 ASP C  475  LEU C  480  1                                   6
HELIX   25  25 LEU C  480  LYS C  490  1                                  11
HELIX   26  26 ASP C  494  VAL C  496  5                                   3
HELIX   27  27 ASN C  497  LEU C  513  1                                  17
HELIX   28  28 THR C  518  THR C  537  1                                  20
HELIX   29  29 THR D  547  GLY D  559  1                                  13
HELIX   30  30 ASN D  562  TYR D  578  1                                  17
HELIX   31  31 PRO D  579  PHE D  585  5                                   7
HELIX   32  32 PHE D  585  GLY D  589  5                                   5
HELIX   33  33 THR D  593  ASN D  600  1                                   8
HELIX   34  34 ASN D  600  ALA D  619  1                                  20
HELIX   35  35 ASN D  623  LYS D  638  1                                  16
HELIX   36  36 PRO D  643  GLY D  662  1                                  20
HELIX   37  37 LYS D  663  PHE D  665  5                                   3
HELIX   38  38 THR D  666  HIS D  686  1                                  21
```

Figure 2.10: **Secondary structure section.** The secondary section of the PDB file for chain B of hemoglobin is presented.

The *coordinate section* is the most important part for our experiments. Here the three dimensional coordinates for each atom of the protein structure are listed, as well as the sequence numbers for every chain (See Figure 2.11). Notice that the sequence number does not necessarily start by one and can contain characters as well, e.g. 1A, 1B.

```
ATOM   1071  N   VAL B 144     -15.723  11.024  11.494  1.00 49.45           N
ATOM   1072  CA  VAL B 144     -14.698  11.472  12.455  1.00 50.32           C
ATOM   1073  C   VAL B 144     -14.849  12.971  12.844  1.00 50.87           C
ATOM   1074  O   VAL B 144     -15.837  13.708  12.440  1.00 51.58           O
ATOM   1075  CB  VAL B 144     -13.286  11.049  11.975  1.00 50.45           C
ATOM   1076  CG1 VAL B 144     -12.938  11.593  10.592  1.00 51.41           C
ATOM   1077  CG2 VAL B 144     -12.220  11.387  13.036  1.00 50.85           C
ATOM   1078  N   HIS B 145     -13.817  13.367  13.663  1.00 48.24           N
ATOM   1079  CA  HIS B 145     -14.021  14.764  14.064  1.00 47.65           C
ATOM   1080  C   HIS B 145     -13.197  15.725  13.272  1.00 46.40           C
ATOM   1081  O   HIS B 145     -12.114  16.232  13.570  1.00 45.43           O
ATOM   1082  CB  HIS B 145     -14.027  14.959  15.607  1.00 54.64           C
ATOM   1083  CG  HIS B 145     -15.414  14.466  16.058  1.00 59.18           C
ATOM   1084  ND1 HIS B 145     -16.609  14.815  15.395  1.00 60.59           N
ATOM   1085  CD2 HIS B 145     -15.748  13.630  17.097  1.00 60.06           C
ATOM   1086  CE1 HIS B 145     -17.619  14.231  16.044  1.00 61.31           C
ATOM   1087  NE2 HIS B 145     -17.130  13.502  17.055  1.00 61.18           N
```

Figure 2.11: **Coordinate section.** A snapshot of the coordinate section of the PDB file for chain B of hemoglobin is presented. The C_α coordinates of Valine (Val) and Histidine (His) are: (-14.698,11.472,12.455) and (-14.021,14.764,14.064). They have the sequence numbers 144 and 145 on chain B of hemoglobin.

Each atom has a number. The C_α-atom is denoted by CA. Notice, that the hydrogen atom is not listed, since it is too small to be measured. Protein coordinates are measured in Angstroms (A°). Ten billion Angstroms equal one meter:

$$1A^\circ = 10^{-10}\text{m}.$$

Chapter 3

Protein Classification

The basic approach of biologists in describing the nature is to use some kind of classification hierarchy. Similar objects are grouped at different level of detail. For example, human beings, are bipedal primates belonging to the mammalian species Homo sapiens under the family Hominidae (the great apes). A similar classification hierarchy is applied on protein structures.

For protein classification the most popular three hierarchies are: SCOP [20], CATH [21] and DALI/FSSP [12]. SCOP is a protein database labeled by human experts, while CATH uses automatic tools before processing the classification task to experts. The classification is performed completely automatically by DALI/FSSP.

In the first section the hierarchies defined as by SCOP and by CATH are presented. In the second section the principle of automatic classification is explained. In the third section, the complexity of the different approaches is discussed.

3.1 Manual Classification

Presently there are three main protein databases for protein structure classification: Those are SCOP, CATH and FSSP/DALI. While SCOP and CATH rely on experts opinion, the FFSP/DALI classification was generated automatically with the distance matrix alignment (DALI) algorithm. The methods used by DALI will be introduced in the next chapter, while we will have a closer look on the classifying schemes of SCOP and CATH in this section.

3.1.1 SCOP

In the SCOP[1] (Structural Classification of Proteins) database published in 1995, all proteins of known structure are ordered according to their evolutionary and structural relationship. The protein domains are hierarchically grouped into families, superfamilies, folds and classes (See Figure 3.1). The last update of the hierarchy dates from October 2004.

[1] http://scop.mrc-lmb.cam.ac.uk/scop/

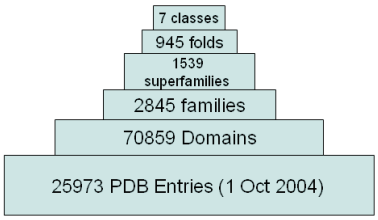

Figure 3.1: **SCOP classification hierarchy.** Protein domains are grouped into families, superfamilies, folds and classes.

The basic unit in SCOP is a protein domain. The domain is either a monomer or a part of a protein and it should reflect a structure that did not change throughout evolution. Since this definition is very hard to measure by an algorithm, SCOP solely relies on visual inspection by experts.

Each domain can be addressed either by a unique integer (sunid) or by a concise classification string (sccs). For example, the protein with the PDB identity 1dlr has the sunid 34906 and the sccs 'c.71.1.1', where 'c' stands for the class, '71' the fold, '1' the superfamily and the last '1' for the family. In the 'dir.des.scop.txt' file the sunid and the sccs for each domain and English names for proteins, families, superfamilies, folds and classes are listed. Also the sequence number where the domain in the chain starts and ends is contained in this file.

A *family* consists of proteins which either have residue identities over 30% or have similar structure or function. Globins and Triosephosphate isomerase (TIM) are examples of protein families.

A *superfamily* consists of proteins with lower than 30% sequential identity and a probable common evolutionary origin. Examples for superfamilies are Actin-crosslinking proteins.

A *fold* contains proteins having the same major secondary structures in the same arrangement with the same topological connections. The most interesting members of a fold are those with low sequential similarity where there exists an evolutionary link to the other proteins of the fold.

A *class* contains folds with similar secondary structure and is the most general way of defining a protein structure. Table 3.1 shows the distribution of folds over the seven classes.

SCOP Class	number of folds
All alpha-helices	218
All beta-strands	144
Alpha-helices and beta-strands dispersed (a/b)	136
Alpha-helices and beta-strands segregated (a+b)	279
Multi-domain proteins	46
Membrane and cell surface proteins	47
Small proteins	75

Table 3.1: The SCOP distribution of folds into classes as in October 2004.

3.1.2 CATH

The CATH (Class Architecture Topology Homology) database published in 1997 classifies protein domains semi-automatically into class (C-level), architecture (A-level), topology (T-level) and homologous superfamily (H-level). The distribution of CATH domains into the four levels of hierarchy is presented in Figure 3.2 .The last update from the database was in May 2006.

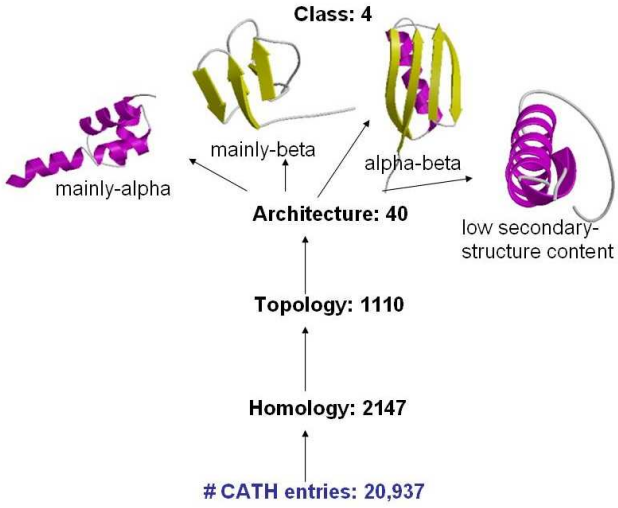

Figure 3.2: **CATH classification hierarchy.** The protein domains are grouped into classes, architectures, topologies and homologies.

All the classification is performed on individual protein domains: Note that these domains differ from domains as defined by SCOP. The domains are defined using a combination of automatic and manual techniques.

The *C-level* consists of four protein classes determined according to their secondary structure composition. The 'all-alpha' and the 'all-beta' class are defined as in SCOP, while the classes 'alpha+beta' and 'alpha/beta' form one class called 'alpha-beta'. The fourth class consists of proteins with low secondary structure content.

The *A-level* contains protein domains with overall shape similarity ignoring however the connectivity between the secondary structures. This level is assigned manually.

At the *T-Level* the connectivity of the secondary structures is considered as well. The connectivity is computed with the help of structure comparison algorithms.

The *H-level* groups protein domains that are supposed to have an evolutionary connection. This is determined either using the SSAP algorithm (Taylor and Orengo, 1989) or if the domains share high sequential similarity.

3.2 Automatic Classification

The goal of this approach is to perform the protein classification automatically and to compare these classification results with the results obtained by experts. The state-of-the-art automatic classification methods are introduced in the next chapter.

Automatic classification methods try to find a concise representation of the protein structure, e.g. by a matrix or a feature vector. In order to compare these representations, a distance measure has to be introduced. Based on the distance measure the classification is performed using a classifier.

The simplest classifier in pattern recognition is the Nearest Neighbor classifier where no learning is performed and no parameters have to be adjusted. More complex classifiers are Neural Networks and Support Vector Machine.

3.2.1 Nearest Neighbor Classifier

The representations are compared using a distance measure. One could think of different distance measures (See Chapter 5), but for the beginning the L1-norm or *Manhattan Distance* is used:

$$\|\mathbf{x} - \mathbf{y}\|_1 = d(\mathbf{x}, \mathbf{y}) = \sum_{i=0}^{n} |\mathbf{x}_i - \mathbf{y}_i| \,,$$

where $\mathbf{x}, \mathbf{y} \in \mathbb{R}^n$.

The distance between the representation of one protein structure to every other structure is computed. These distances are sorted in ascending order. The structure with the smallest distance to the query structure has the highest similarity to the query structure. The query structure is thus assigned to the class of its 'nearest neighbor'.

3.3 Complexity Considerations

The complexity of protein classification depends on the size of the samples to be classified and the number of these samples. For example, a protein data set of N proteins requires $N \cdot (N-1)/2$ protein-protein comparisons. Human experts are not enough if one wants to classify a database consisting of 40,000 protein structures. It needs years of experience in order to classify protein structures just by looking at them and the memory data space is enormous. An objective and reproducible classification measure is not guaranteed this way.

The computational complexity of the automatic methods strongly depends on the size of the protein structure representation and the comparison method. If the representation for each structure has the same size M, than the comparison can be performed in $O(N^2 \cdot M)$ by using a distance measure which can be computed in linear time. This methods are called *Structural Fingerprint* methods. If the representation of a structure depends on the number of C_α atoms, the comparison can be performed only after solving an optimization problem between two structures of different size. In the naive approach the complexity of this optimization problem is NP-hard. However, special algorithms exist which can solve the problem in polynomial time. This is the truth for *Alignment* methods.

Chapter 4

State of the Art

Usually, protein structures are compared using *Alignment*, which is very expensive for large protein databases. The most cited three *Alignment Methods* for protein structures are RMSD, DALI [11] and the Contact Map Overlap [18] approach. In the first section, the three approaches are explained in detail.

As opposed to alignment techniques there are methods using a *Structural Fingerprint* to describe the protein structure. These methods still lack the experts' approval since their quality has not yet reached the quality of the alignment approach. However, their computational time is very low and the classification results are still improving. In the second section, these methods are explained in detail.

4.1 Alignment

Alignment methods try to find a pairing of amino acids between two protein structures. One of the first methods for protein structure comparison was to fix one of the protein structures and to rotate and translate the second structure as a rigid body to minimize its Root Mean Square Distance (RMSD) from the first structure. The most used automatic classification server DALI/FSSP computes a similarity score for two protein structures with the distance matrix alignment algorithm introduced by Holm and Sander [11]. The Contact Map Overlap (CMO) approach uses contact maps to represent the distances between protein structures. The alignment of contact maps is an NP-hard problem.

4.1.1 RMSD

A protein conformation is a set of n vectors (the 3D coordinates of its atoms), x_n, where each x_n has three components. The difference between two protein conformations can be computed through the difference of the two sets of vectors x_n and y_n. Note that the two vector sets must have the same size.

Before the difference is computed, the center-of-mass of y_n and x_n is shifted to the origin of the coordinates, so that translation can be neglected. Then the difference can be captured by the square distance:

$$E(U) = \sum_n |x_n - Uy_n|^2 ,$$

where U is an orthogonal rotation matrix.

The RMSD is then defined by:

$$RMSD = \sqrt{\frac{E(U_{min})}{n}},$$

where U_{min} is the rotation matrix which induces the smallest distance between the two structures.

The classic paper of Wolfgang Kabsch [15] showed how to minimize E to get the RMSD using Lagrange multipliers and Singular Value Decomposition.

However, the RMSD only works well as an indicator of structural similarity if the structures are closely related and structures with same size are compared. Also, RMSD can not manage 'outliers', single atoms which have a large distance to the rest of the structure.

4.1.2 DALI/FSSP

The DALI algorithm tries to align distance matrices. A distance matrix is defined by:

$$D_{i,j} = d(i, j),$$

where $d(i, j)$ is the Euclidean distance between the C_α atoms of the ith and the jth amino acid on the protein chain. Usually, the distance matrices are depicted using gray scale images, where black indicates the distance zero and is only present at the diagonal. Figure 4.1 shows the distance matrix of 1ash.

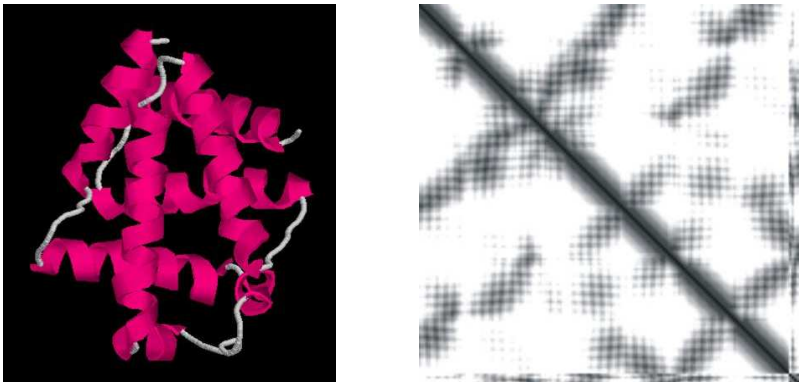

Figure 4.1: **Distance matrix.** The cartoon and the distance matrix of the protein structure 1ash. In the distance matrix the C_α-C_α distances are represented as gray values.

How should one compare two distance matrices? The simple idea is to slide one (transparent) matrix over the other and detect similar submatrices. This idea implies a combinatorial optimization problem of merging corresponding similar submatrices to larger blocks

of agreement by removing redundant rows and columns. The solution of this optimization problem is computed with the Monte Carlo method. In the trial-and-error method the structurally similar regions are found by defining a cutoff function on the intramolecular distances between two detected submatrices. The result of the alignment is typically reported as an equivalent set of amino acids and visualized as a 3D super-imposition.

Since algorithms of the alignment of two structures have been known for a long time, the main contribution of DALI was to apply alignment on large data sets in order to compute automatically a complete map of the protein universe. Hence the alignment algorithm should not only compare two structures but induce a global similarity measure between the two. This similarity measure is defined by:

$$S(A,B) = \sum_i \sum_j \left(0.2 - \frac{\left| D_{ij}^A - D_{ij}^B \right|}{D_{ij}^*} \right) e^{-(D_{ij}^*/20A^\circ)^2}, \tag{4.1}$$

where the summation is over all amino acids of the common core, d_{ij}^* denotes the arithmetic mean of the $C_\alpha - C_\alpha$ distances d_{ij}^A and d_{ij}^B of the proteins A and B, a relative deviation of 0.2 is the threshold of similarity and the exponential factor downweights contribution from parts at longer distances. The optimal structural alignment is that set of equivalences (i^A, i^B) that maximizes S.

The DALI algorithm performs two steps for searching in large databases. In the first step a fast algorithm is used to compute a group of potential similarity candidates. In the second step, a refinement is performed on the set of the previous step using slow but more sophisticated algorithms.

In the first step, the proteins are represented by the means of their secondary structure elements (SSE). Thus, every SSE is represented as a 3D vector at a spatial position. A lookup table is then used to represent the protein's structure. When a structural query is performed on a protein database, the lookup table is used to quickly determine the candidates for similarity. The secondary structures of each protein of this set of candidates are then further aligned with the query structure by dynamic programming.

In the second step, a branch-and-bound algorithm is used to compare the C_α coordinates rather than only the secondary structures thus exploiting the whole space of possible structure-structure alignments. Since the search space is very large, this step requires much more computation time. In order to reduce the computation time of the branch-and-bound algorithm, the protein structures are decomposed first into smaller compact units and then the correspondence problem for the smaller matrices is solved.

Since the protein structures are too large, they are cut into domains. Protein chains are decomposed into domains using the criteria of recurrence and compactness [12]. In the "Dali domain dictionary" each domain is assigned a domain classification number DC_l_m_n_p representing:

1. a fold space attractor region (l),
2. a globular folding topology (m),
3. a functional family (n) and

4. a sequence family (p).

The finest level of classification is *level p* and the highest level of the fold classification corresponds to *level l*. The most evolutionary interesting part of the DALI classification hierarchy is *level m*. The globular folding topology defines the fold type. Fold types are defined as clusters of structural neighbors in fold space with average pairwise Z-scores above 2. The mean and standard deviations of similarity scores S were calibrated against pairwise all-on-all comparisons in a database of 220 proteins, as a function of protein size. Shape similarity quantified with the distance matrix comparison scores can then be expressed in terms of normalized Z scores that is, standard deviations above the mean.

4.1.3 Contact Map Overlap

A contact map is a concise representation of a protein's native three-dimensional structure. It is specified by a matrix C, with entries indexed by pairs of protein residues:

$$C_{i,j} = \begin{cases} 1 & \text{if residue } i \text{ and } j \text{ are in contact,} \\ 0 & \text{otherwise.} \end{cases}$$

Residues i and j are said to be in contact if they lie within R Angstroms from each other in the protein's native fold. R is called the threshold of the contact map. In the Figures 4.2 and 4.3, two protein structures and their contact maps are depicted. The contact between two amino acids corresponds either to a non-zero value of the matrix or to an edge in the graph representation.

The contact maps are a simplification of the distance matrices introduced in the previous section. In fact, a contact map is a distance matrix which contains only the values 0 or 1 depending on the threshold R.

Rather than representing contact maps as matrices, these are usually represented as undirected graphs $G = (V, E)$. The vertices of the graph represent the amino acids, while the edges are the contacts between two amino acids.

The goal of the alignment is to find correspondences between the endpoints of contacts in the first map and the amino acids that are also in contact in the second map. The number of these correspondences is called the overlap. In Figure 4.4, the alignment of the two contact maps from the Figures 4.2 and 4.3 is presented. The alignment of the two contact maps matches the vertices of the first graph to the vertices of the second graph. The mappings (in blue) do not cross, since the order of the vertices needs to be preserved.

In graph theoretic language: Two undirected graphs $G_1 = (V_1, E_1)$ and $G_2 = (V_2, E_2)$ are given, where $n_i = |V_i|$ is the number of vertices and $m_i = |E_i|$ is the number of edges for $i = 1, 2$. A total order is defined on $V_1 = \{a_1 < ... < a_{n_1}\}$ and $V_2 = \{b_1 < ... < b_{n_2}\}$. A *non-crossing map* of V_1 in V_2 is defined by any two subsets of the same size k and $a_{i_1} < ... < a_{i_k} \subseteq V_1$ and $b_{u_1} < ... < b_{u_k} \subseteq V_2$. The indices i_h and u_h are ordered according to their value and u_h is the image of i_h. Two edges $(a_i, a_j) \in E_1$ and $(b_u, b_v) \in E_2$ are *shared* by the map if there are $l, t < k$ such that $a_i = a_{i_l}$, $a_j = a_{j_t}$, $b_u = b_{u_l}$ and $b_v = b_{v_t}$.

The goal of the alignment is to find the maximal number of non-crossing sharings.

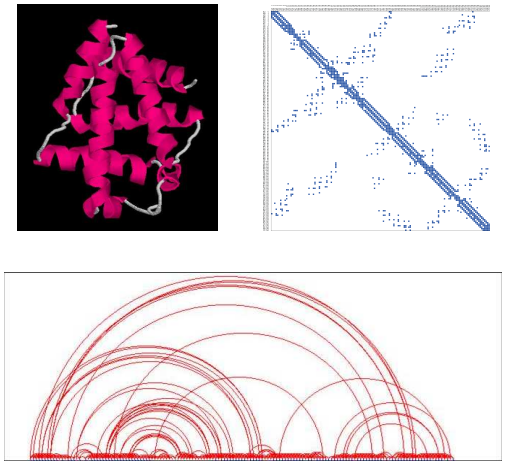

Figure 4.2: The protein structure of 1ash and its contact map represented as a matrix and as a graph.

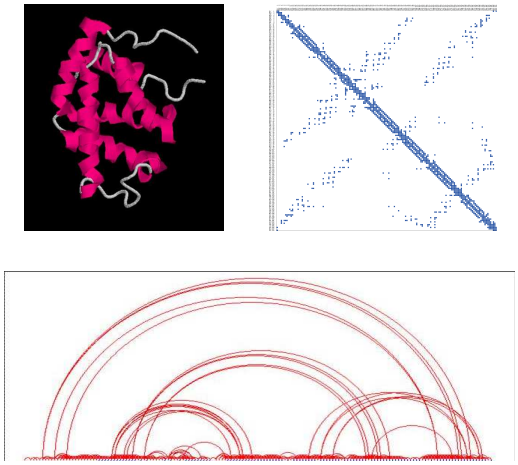

Figure 4.3: The protein structure of 1hlm and its contact map represented as a matrix and as a graph.

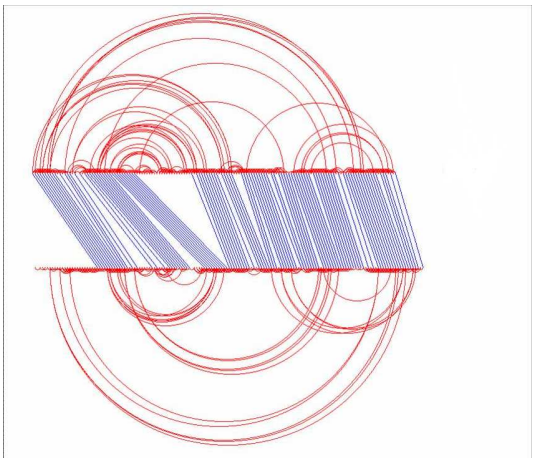

Figure 4.4: **Contact map alignment.** The contact maps of the protein structures 1ash and 1hlm are aligned with the CMO algorithm by Lancia et. al. Picture from [18].

Lancia et al. [18] found an optimal way to align contact maps of protein structures using Integer Programming (IP). IP is a technique to solve optimization problems defined as linear functions with integer variables. First the contact map overlap problem is reduced to the Maximum Independent Set [1] (MIS) problem. This problem can be defined as an IP problem with the binary variable x_v:

$$\max \sum_{v \in V_{optimal}} x_v$$

with

$$x_u + x_v \leq 1 \text{ for all edges } \{u, v\} \in E$$

[1] independent set = a set of vertices such that there is no edge between any two of them.

We define x_u for an edge $E_i = \{a, b\} \in E$ and $a, b \in V$ as:

$$x_u = \begin{cases} 1 & \text{if } u \in E_i, \\ 0 & \text{otherwise.} \end{cases}$$

The search space of the problem is exponential, since for a Graph with n vertices the search space is $O(2^n)$

Then IP is applied on this optimization problem using a Branch-and-Cut algorithm to solve the set of equations. The lower bounds are defined by several heuristics, while the upper bounds are found by linear programming relaxation, in which the variables are not restricted to integers and thus the computation time is polynomial. The cuts are mainly clique-inequalities.

An example of a clique-inequality is:

$$\max \sum_{v \in Q} x_v \leq 1,$$

which says that any clique Q can have at most one vertex in common with any independent set. Finding more clique-inequalities can further reduce the complexity of the problem.

For reducing the CMO problem to the MIS problem and its overall IP formulation, the reader is referred to [18].

4.2 Structural Fingerprints

These methods do not compute an alignment between two structures. Rather they try to find a structural fingerprint of the proteins. However, this problem is not the same like describing and finding human fingerprints in a large database, where only an exact match is of interest. The protein fingerprint should on the one hand describe the protein structure as good as possible. On the other hand, it should leave space for slight modifications of the basic structure. Thus, similar proteins can be retrieved from a large database.

Since no alignment is performed, these methods are very fast. The Probability of Identity (PRIDE)[4] method uses a set of histograms to describe each protein. The Gauss Integral approach used by Rogen and Fain [26] describes a protein backbone (= the polygonal curve connecting its C_α atoms) with the help of knot theory.

4.2.1 PRIDE

In this approach by Carugo and Pongor the distribution of the $D_{i,i+n}$ distances in the range of $n = 3, ..., 30$ chain distance is used to describe the protein structure. For each protein the distance distribution which is a histogram is computed for $C_\alpha - C_\alpha$ pairs with a distance of n on the chain. In Figure 4.5 the histograms of $D_{i,i+n}$ distances for $n = 8, 15, 22, 29$ are shown.

Hence we get 28 distance histograms associated with one protein structure. These distance histograms are then compared pairwise for two protein structures using contingency table analysis.

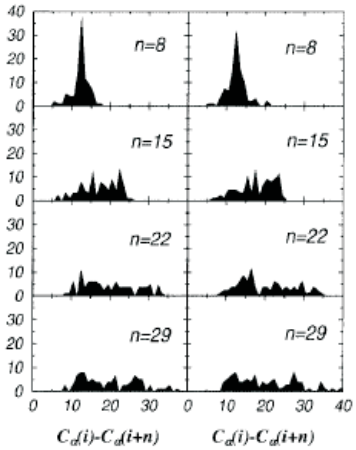

Figure 4.5: **PRIDE distance histograms.** The $D_{i,i+n}$ distance histograms for n = 8, 15, 22, 29 as computed by PRIDE. Rather than computing one histogram of the distance matrix $D_{i,j}$, several distance matrices of C_α atoms with the distance n are computed. The x-axis scales the distances in A° and the y-axis denotes the percentage of observations found in one bin. Image from [4].

This analysis answers the question: Is their a dependency between structure 1 and structure 2? The contingency table for two protein structure histograms is shown in Table 4.1.

	Bin 1	Bin 2	...	Bin m	Total
Structure 1	O_{11}	O_{12}	...	O_{1m}	R_1
Structure 2	O_{21}	O_{22}	...	O_{2m}	R_2
Total	C_1	C_2	...	C_m	n

Table 4.1: **Contingency table.** The contingency table constructed of two histograms with m bins.

The expected value for the observation is:

$$e_{ij} = \frac{R_i \cdot C_j}{n},$$

where R_i is the total sum of the ith row, C_j is the total sum of the jth column and n is the total number of samples. The chi-square test statistic, χ^2, used by PRIDE is then computed by:

$$\chi^2 = \sum_{i=1}^{r} \sum_{j=1}^{c} \frac{(O_{ij} - e_{ij})^2}{e_{ij}},$$

where r and c are the number of rows and columns of the contingency table respectively. The observed number of responses in the cell in row i and column j is O_{ij}.

The PRIDE score is the arithmetic average of the 28 χ^2 values. It ranges between 0 and 1, where 1 implies the maximum dependency between the two structures.

In order to perform the contingency table analysis, the bins of the histograms of the two structures are combined so that at least 5% of the observations are included in each bin.

4.2.2 Gauss Integrals

Rogen et al. compute a set of 29 writhe-based features associated with the protein backbone structure. The backbone can be parametrized by a polygonal curve μ (See Figure 4.6).

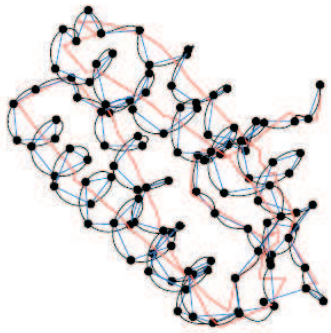

Figure 4.6: **Polygonal representation of the C_α trace.** The structure of the protein is described as a connected linear trace of its C_α atoms.

The writhe of a closed space curve γ can be computed by using the Gauss Integral:

$$Wr(\gamma) = \frac{1}{4\pi} \int \int_{\gamma \times \gamma \setminus D} \omega(t_1, t_2) dt_1 dt_2,$$

where

$$\omega(t_1, t_2) = \frac{[\gamma'(t_1), \gamma(t_1) - \gamma(t_2), \gamma'(t_2)]}{|\gamma(t_1) - \gamma(t_2)|^3} dt_1 dt_2.$$

D is the diagonal of $\gamma \times \gamma$ and $[\gamma'(t_1), \gamma(t_1) - \gamma(t_2), \gamma'(t_2)]$ is the triple scalar product. The *triple scalar product* for three vectors $\mathbf{a}, \mathbf{b}, \mathbf{c}$ is defined by:

$$[\mathbf{a}, \mathbf{b}, \mathbf{c}] = \mathbf{a} \cdot (\mathbf{b} \times \mathbf{c})$$

The writhe can be described as the average number of crossings seen when looking at the structure from all directions of the 3D-space.

For a polygonal curve μ the integral is reduced to a sum:

$$Wr(\mu) = I_{(1,2)}(\mu) = \sum_{1 < i_1 < i_2 < N} W(i_1, i_2),$$

where $W(i_1, i_2)$ is the contribution to the writhe coming from the i_1th and i_2th line segment. The concrete computation of W is explained in [25].

28 more writhe-based descriptors are constructed by taking absolute values and looking only at certain interesting configurations such as in:

$$I_{|1,2|}(\mu) = \sum_{1<i_1<i_2<N} |W(i_1, i_2)|$$

$$I_{|1,3|(2,4)}(\mu) = \sum_{1<i_1<i_2<i_3<i_4<N} |W(i_1, i_3)|\, W(i_2, i_4)$$

and

$$I_{(1,5)(2,4)(3,6)}(\mu) = \sum_{1<i_1<i_2<i_3<i_4<i_5<i_6<N} W(i_1, i_5)W(i_2, i_4)W(i_3, i_6).$$

The number of structural descriptors is 30 since the number of C_α atoms is also one of the descriptors.

4.3 Discussion

The alignment methods have a very high time complexity for DALI and the CMO. If the size of the protein database is N and the average size of a protein is M, than the complexity of the alignment algorithms is $O(N^2 \cdot M^2)$. Further there are limitations to the size of the data set considered for CMO: Not more than 300 structures can be compared.

The structural finger print methods are very fast, however they lack to describe the protein structure in detail.

This motivates the new approach: We want to find a precise and effective representation of the protein's structure. It should have the precision of the alignment methods and the time requirements of the structural fingerprint methods.

Chapter 5

Structural Fingerprints by the use of GI

Here, a new method for protein structure classification based on structural features is presented. The idea of the structural fingerprint methods is followed, so the central topic of this thesis is to find a concise representation of the protein structure. This representation is obtained by *Group Integrals* (GI).

In the first section, the feature extraction for protein structure representation by GI is introduced. GI are extended by *Spherical Harmonics* (SH) and *D-Wigner Matrix* expansion to keep more structural information back.

In the second section, the feature selection algorithms RELIEF and SIMBA are introduced. They should help to extract the relevant features.

In the third section different distance measures for feature vectors are considered. Also, a simple domain partitioning algorithm is introduced dividing a protein into several domains. Finally, several distance measures for vector sets are presented.

In the fourth section algorithms for computing the new method are presented.

5.1 Feature Extraction

In pattern recognition features play an important role. If the extracted features are not relevant for describing the object, then no other technique will improve the classification results. Therefore, we are interested in thoroughly describing the structural features of proteins. For this reason we examined the protein structure in detail in the previous chapter.

We want to find a translation and rotation invariant representation of protein tertiary structures in order to compare them. In [34], histograms were used to describe the distribution of distances and angles occurring in the structure of a protein (= its 3D coordinates). These histograms are invariant to rotation and translation since only the frequency of the occurrence of inter-atomic distances and angles is considered and not the actual atom coordinates. Since histograms could be described as GI, a more general invariant can be defined.

The 3D structure of the protein in Figure 5.1 is described by a set of invariant features. Invariance for the Euclidean group of motion is considered. In order to obtain translational and rotational invariance, GI are computed. Later, GI are extended by SH since more

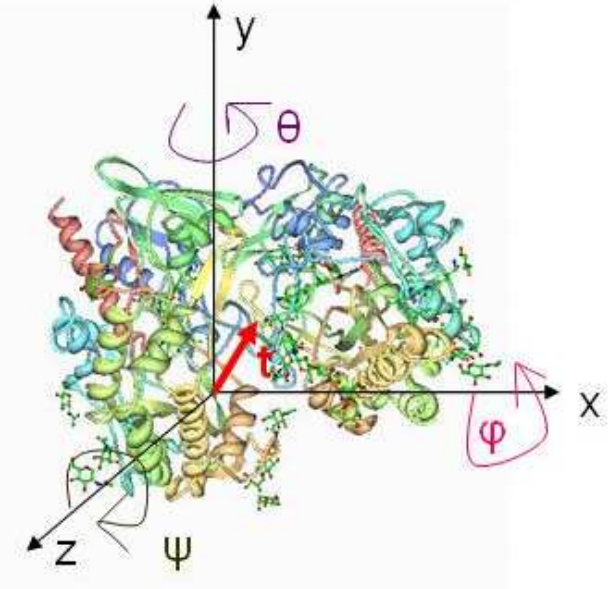

Figure 5.1: **Euclidean motion of protein structure in 3D.** The protein structure with the PDB identity 1GPE is depicted as a cartoon in 3D space. The relative position of the structure to the origin of the coordinate axis is defined by the translational vector $\mathbf{t}$ and the rotational angles ϕ, θ and ψ around the x, y and z axis respectively.

rotational information can be kept in this way. Even more detailed shape description can be achieved by *D-Wigner matrices*. Here, a trade-off between computational cost and the level of detail has to be made depending on the application's requirements.

5.1.1 Invariant Features

In mathematics, given a set $\mathbf{X}$ and an equivalence relation $\sim$ on $\mathbf{X}$, the equivalence class of an element $\mathbf{x}$ in $\mathbf{X}$ is the subset of all elements in $\mathbf{X}$ which are equivalent to $\mathbf{a}$:

$$[\mathbf{a}] = \{\mathbf{x} \in \mathbf{X} | \mathbf{x} \sim \mathbf{a}\}$$

The equivalence relation $\sim$ is reflexive ($\mathbf{a} \sim \mathbf{a}$), symmetric ($\mathbf{a} \sim \mathbf{b} \Rightarrow \mathbf{b} \sim \mathbf{a}$) and transitive ($\mathbf{a} \sim \mathbf{b} \wedge \mathbf{b} \sim \mathbf{c} \Rightarrow \mathbf{a} \sim \mathbf{c}$).

An equivalence relation on a group $\overset{G}{\sim}$ can be defined by:

$$\mathbf{x_1} \overset{G}{\sim} \mathbf{x_2} \Leftrightarrow \exists g \in G \; g\mathbf{x_1} = \mathbf{x_2}$$

In order to obtain invariant features, we need to find a mapping $\mathbf{I}$ such that $\mathbf{I}$ is able to extract the intrinsic features of an object. All the representatives of one equivalence class should be mapped into one point of the feature space by $\mathbf{I}$. This could be expressed by the following formula:

$$\mathbf{x_1} \overset{G}{\sim} \mathbf{x_2} \Rightarrow \mathbf{I}(\mathbf{x_1}) = \mathbf{I}(\mathbf{x_2}).$$

There are three canonical possibilities to obtain invariants: *Normalization*, the *Differential Approach* and *Integration*. Normalization techniques utilize extreme points on orbits and normalize the object with respect to these. Fourier descriptors, for example, normalize the objects with respect to the main axes of the ellipses. The Differential approach considers invariant features obtained by solving partial differential equations (Lie theory). This approach is suitable only if the parameter set is small and the solution for the resulting differential equations is easy to find. The Integral Approach assumes that the equivalence class of an object $\mathbf{x}$ forms an orbit in the object space parametrized by λ. The idea is to average arbitrary functions evaluated on the orbit (Haar Integrals). A good overview of invariant theory and especially of the Integration approach used in this work is presented in [3].

Group Integration is the name for a class of methods which compute invariant features for a transformation group. Invariance is achieved by summing up all possible positions a signal can take under a certain transformation group. Additionally, a non linear kernel function k should be applied to every transformation of the signal in order to obtain separable features.

In general, an invariant feature obtained by Group Integration is defined by:

$$I_k(\mathbf{x}) = \int_G k(g\mathbf{x})\, dg \tag{5.1}$$

Here, g stands for a group element of the transformation group G and acts on the signal $\mathbf{x}$. The kernel function k is evaluated for each group transformation of the signal $\mathbf{x}$. The quality of the invariant depends on the kernel function k.

Usually, the kernel functions are monoms which take into account a certain neighborhood of the signal in order to describe local features. A special feature of the Haar Integrals is their robustness towards small perturbations of the signal. The disadvantage of Haar-Integrals is their high computational cost: For each point of the signal and each possible transform, the kernel function needs to be evaluated.

GI stands in contrast to Normalization techniques, which obtain invariance by computing features relative to a global reference frame. The determination of the reference frame makes Normalization techniques extremely sensitive to noise, whereas GI is known to be very robust to many kinds of noise.

At the Chair for Pattern Recognition and Image Processing, University of Freiburg, GI techniques were already used for different applications: Haasdonk [8, 9] applied GI to character recognition and joined the GI-framework with Kernel-techniques. Ronneberger et al [27, 28] used GI for the classification of Pollen grains and segmentation of cell nuclei. In [29, 32, 30] GI was successfully applied to texture-classification and image retrieval.

Group Integrals for 3D structures

In this section, we introduce GI for 3D objects. Later, we will extend the model to describe proteins. In general, a 3D object $\mathbf{x}$ could be defined using the following function:

$$\mathbf{x} : \mathbb{R}^3 \mapsto \mathbb{R}$$

The domain of the function is the volume of the 3D object $\mathbf{x}$. For intensity objects $\mathbf{x}$ could be defined as the coloring function for the object, where the value of $\mathbf{x}$ is the gray value.

Consider eq. (5.1): If G is the Euclidean group $\mathcal{E}$ and $\mathbf{x}$ is an object in 3D space, then $g \in \mathcal{E}$ acts on $\mathbf{x}$ by:

$$(g\mathbf{x})\,(\mathbf{n}) := \mathbf{x}(\mathbf{R}^T\mathbf{n} - \mathbf{t})$$

The action of the group on a point $\mathbf{n}$ of the object $\mathbf{x}$ could be parameterized by the translational vector $\mathbf{t}$ and the rotational angles ϕ, ψ and θ. For example, if we have a point $\mathbf{n}$ in three dimensional space, than its Euclidean motion to point $\mathbf{n}'$ could be described by the formula:

$$\mathbf{n}' = \mathbf{R}\mathbf{n} + \mathbf{t}$$

If we denote $\cos a$ by c_a and $\sin a$ by s_a the formula above could be rewritten to:

$$\mathbf{n}' = \begin{pmatrix} 1 & 0 & 0 \\ 0 & c_\phi & -s_\phi \\ 0 & s_\phi & c_\phi \end{pmatrix} \begin{pmatrix} c_\theta & 0 & s_\theta \\ 0 & 1 & 0 \\ -s_\theta & 0 & c_\theta \end{pmatrix} \begin{pmatrix} c_\psi & -s_\psi & 0 \\ s_\psi & c_\psi & 0 \\ 0 & 0 & 1 \end{pmatrix} \mathbf{n} + \begin{pmatrix} t_0 \\ t_1 \\ t_2 \end{pmatrix}$$

The angles ϕ, θ and ψ describe the rotation around the x, y and z axis respectively (see Figure 5.1).

Typical choices for the kernel function k in eq. (5.1) are $k(\mathbf{x}) = \mathbf{x}(0) \cdot \mathbf{x}(d)$. Here the value of a reference point $\mathbf{x}(0)$ and another point of the object $\mathbf{x}(d)$ at distance d is considered. Further kernels could be constructed by $k(\mathbf{x}) = h_1(\mathbf{x}(0)) \cdot h_1(\mathbf{x}(d))$, where h_1 and h_2 are some arbitrary nonlinear functions.

Choosing the kernel

In order to compute the GI in eq. (5.1), a suitable kernel function should be found. On the one hand, the kernel function should have a high discrimination power for the different object classes. On the other hand it should be flexible enough to describe all objects that belong to one class.

In this application, the protein structure is viewed as an atom cloud (See Figure 5.2). The interesting parts of this structure are regions of high variety. These regions can be described by using the gradient ∇x. This leads to the following kernel choice:

$$k_d(x, \nabla x) = h_1(\nabla x(0)) \cdot h_2(\nabla x(d)),$$

where d is the width parameter of the kernel function. The functions h_1 and h_2 should be direction specific to keep the relative directions of the gradients. If the gradient is large, a strong feedback is desired. The simplest idea is to choose:

$$h_n(v) = \left|v^T n\right|,$$

where n is some fixed unit vector. The function above is not able to decide whether it has to deal with a large disoriented or a small oriented gradient v. Thus, a more rational choice for h_n is:

$$h_n(v) = |v| \cdot \delta_1(\left(\frac{|v^T n|}{|v|}\right), \tag{5.2}$$

where $\delta_x(y) = \delta(y-x)$ and δ is Dirac's delta function. It gives contribution to the integral only if $x - y = 0$.

Thus $h_n(v)$ is unequal to zero whenever n and v are parallel or antiparallel. The kernel function suggested is therefore:

$$k_{d,n,n'}(x) = h_n(\nabla x(0)) \cdot h_{n'}(\nabla x(d)), \tag{5.3}$$

Spherical Harmonics

Spherical Harmonics are a powerful tool to describe the way rotations act on a spherical function. Spherical harmonics, $Y_l^m(\phi, \psi)$, are single-valued, smooth (infinitely differentiable), complex functions of two variables, ϕ and ψ, indexed by two integers, l and m. In quantum physics terminology, l is the angular quantum number and m the azimuthal quantum number. Roughly speaking, l gives the number of local minima of the function and therefore represents a spatial frequency.

Spherical harmonics form an orthonormal basis of a vector space analogue to Fourier coefficients. In the same way that Fourier coefficients could be used to describe the spectrum of an image, expansion coefficients based on Spherical Harmonics can be used to describe functions defined on a sphere. Any square-integrable function of ϕ and ψ on the two-sphere [1] can be expanded as follows:

$$\mathbf{f}(\phi, \psi) = \sum_{l=0}^{\infty} \sum_{m=-l}^{l} a_m^l Y_m^l(\phi, \psi). \tag{5.4}$$

Depending on the cutoff parameter l, errors are introduced to the representation of $\mathbf{f}$.

In order to compute the spherical harmonics, $Y_l^m(\phi, \psi)$, associated *Legendre polynomials* P_l^m and P_l^{-m} are necessary with $m = 0, 1, ..., l$. The associated Legendre polynomials are solutions to the associated *Legendre differential* equation for $l \in \mathbb{N}$ and $x \in \mathbb{R}$:

$$(1 - x^2)y'' - 2xy' + l(l+1)y = 0$$

The P_l^m are given by:

$$P_l^m(x) = \frac{(-1)^m}{2^l l!}(1 - x^2)^{m/2} \frac{d^{l+m}}{dx^{l+m}}(x^2 - 1)^l.$$

For negative associated Legendre polynomials P_l^{-m} are defined by:

$$P_l^{-m}(x) = (-1)^m \frac{(l-m)!}{(l+m)!} P_l^m(x).$$

[1] All the points (x, y, z) such that: $(x - x_0)^2 + (y - y_0)^2 + (z - z_0)^2 = r^2$, where (x_0, y_0, z_0) is the center of the sphere and r its radius.

The first four associated Legendre polynomials are:

$$P_0^0(x) = 1,\ P_1^0(x) = x,\ P_1^1(x) = -1(1-x^2)^{1/2},\ P_2^0(x) = \frac{1}{2}(3x^2-1).$$

For a definition of the associated Legendre polynomials and further properties see [2]. The $Y_l^m(\phi,\psi)$ are computed by:

$$Y_{lm}^*(\phi,\psi) = \sqrt{\frac{2l+1}{4\pi}\frac{(l-m)!}{(l+m)!}} \cdot P_l^m(\cos\phi) \cdot e^{im\psi}. \tag{5.5}$$

The expansion coefficients, a_m^l, can be obtained by multiplying eq. (5.4) by the complex spherical harmonics and integrating over the solid angle Ω,

$$a_m^l = \int_{S^2} f(\phi,\psi) Y_{lm}^*(\phi,\psi) d\Omega.$$

The energy $\sum_{m=-l}^{l} \left|a_m^l\right|^2$ of the coefficients a_m^l is invariant to rotations of the underlying object. The key idea is that the amount of energy $\sum_{m=-l}^{l} \left|a_m^l\right|^2$ a function $\mathbf{f}$ contains at different frequencies does not change.

D-Wigner Expansion

A real function $\mathbf{f}(g) : SO_3 \mapsto \mathbb{R}$ defined on the rotation group can be orthogonally expanded in terms of D-Wigner matrices:

$$\mathbf{f}(g) = \sum_{l=0}^{\infty} \sum_{m=-l}^{l} \sum_{m'=-l}^{l} b_{m,m'}^l (D_{m,m'}^l(g)),$$

where the b^l are expansion matrices obtained by the projections on the basis functions:

$$b_{m,m'}^l = \frac{2l+1}{8\pi^2} \left\langle D_{m,m'}^l, \mathbf{f} \right\rangle$$

The scalar product $\langle \mathbf{x}, \mathbf{x}' \rangle$ is defined by:

$$\langle \mathbf{x}, \mathbf{x}' \rangle = \int_{SO(3)} \mathbf{x}^*(g)\mathbf{x}'(g) dg,$$

where $\mathbf{x}^*$ denotes the conjugate transpose of $\mathbf{x}$.

Hence, we are able to use the projections to keep more information back in our group integration framework. Instead of a simple integration over the rotation group we compute projections on the D-Wigner matrices.

Since the D-Wigner matrices are unitary representations of the rotation group one can show that the norms of the columns of the b^l are invariant to right multiplications $\mathbf{f}(R) \mapsto \mathbf{f}(RR')$ and similar the norms of the rows of the b^l are invariant to left multiplications. Hence we can obtain invariance by taking the norms of the columns or rows of the expansion matrices, respectively.

Further, the Spherical Harmonics coefficients a_m^l show a nice transformation behavior: If $\mathbf{f}(\phi, \psi)$ is rotated by some matrix R, then the a_m^l are transformed by a D-Wigner matrix $D_l(R)$ such that:

$$a_m^l \mapsto \sum_{m=-l}^{l} D_m^l(R) a_m^l$$

An introduction to D-Wigner matrices is given in [19].

5.1.2 Application for Proteins

A 3D model of the protein shape is constructed by considering only the 3D coordinates of the atoms forming a protein structure. In Figure 5.2, a typical protein structure is shown.

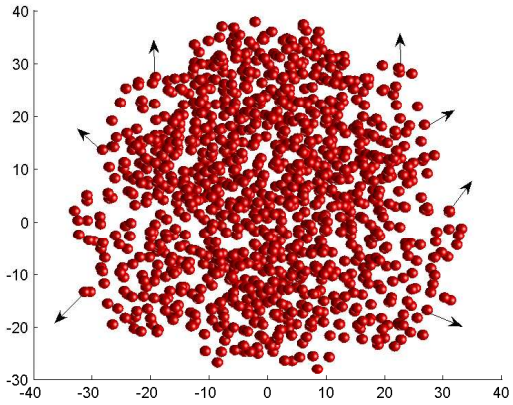

Figure 5.2: **Atom cloud.** Only the C_α atoms of the protein with the PDB identity 1GPE are plotted. The arrows indicate the gradient at each C_α atom. In Figure 5.1, the same protein is plotted as a cartoon.

In order to include directional information for each atom, the gradient of x, $\nabla x(r)$, is defined. By using the δ function only the gradient for C_α locations u_i is defined.

$$\nabla \mathbf{x}(r) = \sum_i \delta_{u_i}(r) \underbrace{\frac{2}{\sigma^2} \sum_j (u_i - u_j) \cdot e^{-(\frac{\|u_i - u_j\|}{\sigma})^2}}_{\nabla x(u_i)} \tag{5.6}$$

$$= \sum_i \delta_{u_i}(r) \nabla \mathbf{x}(u_i), \tag{5.7}$$

where u_i are the point coordinates and the indices i and j range over the whole point set. They are chosen according to the sequence numbers of the amino acids. Thus, the gradient in point r is influenced by the distance to all other points of the protein structure. In other words, the gradient is the orientation of one atom with respect to all the other atoms of the structure.

Group Integrals for Proteins

The kernel introduced in eq. (5.3) contains three vector valued parameters n, d and n'. Their configuration is shown in Figure 5.3. Two atoms of a protein structure are considered having the orientation n in $A1$, n' in $A2$ and the Euclidean distance d.

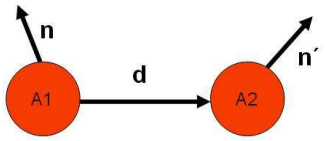

Figure 5.3: **Configuration of n, n′ and d.** Two atoms $A1$ and $A2$ of the protein are considered. Their configuration could be uniquely defined by the parameter set $\Pi = \{\alpha, \beta, \gamma, \Delta\}$ constructed of the vectors n, d and n'.

Using three pair-wise dot products and the absolute value of the distance d, we define the parameter set $\Pi = \{\alpha, \beta, \gamma, \Delta\}$:

$$\alpha = \frac{n^T \cdot d}{|d|} \tag{5.8}$$

$$\beta = \frac{n'^T \cdot d}{|d|} \tag{5.9}$$

$$\gamma = n^T \cdot n' \tag{5.10}$$

$$\Delta = |d| \,. \tag{5.11}$$

Now we want to solve the group integral:

$$I_\Pi(\mathbf{x}) = \int_{\mathcal{E}} k_{d,n,n'}(g\mathbf{x})\, dg \tag{5.12}$$

$$= \int_{\mathcal{E}} h_n(g\nabla\mathbf{x}(0)) \cdot h_{n'}(g\nabla\mathbf{x}(d))\, dg \tag{5.13}$$

$$= \int_{\mathbb{R}^3} \int_{O_3} h_n(R\nabla\mathbf{x}(u)) \cdot h_{n'}(R\nabla\mathbf{x}(u + R^T d))\, dR\, du \tag{5.14}$$

In eq.(5.14) the Euclidean group is represented by the translation u and a rotation matrix R. The points are transformed by an arbitrary orthogonal matrix $R \in \mathbb{R}^{3\times3}$. The rotation matrix has three degrees of freedom (rotation around x, y and z axes). In $h_{n'}$, the distance d is rotated by R^T before $\nabla x(u + R^T d)$ is transformed by R. This is due to the fact that we want to conserve the relative direction of d to the point u.

The second kernel factor $h_{n'}(R\nabla x(u + R^T d))$ is substituted with the help of the relation:

$$h(u) = \int_{\mathbb{R}^3} h(u')\delta_{u'}(u)\, du'.$$

So, we obtain:

$$h_{n'}(R\nabla x(u + R^T d)) = \int_{\mathbb{R}^3} h_{n'}(R\nabla x(u')) \cdot \delta_{u'}(u + R^T d)\, du'$$

and substitute the result to eq. (5.14):

$$I_\Pi(x) \quad = \quad \int_{\mathbb{R}^6} \int_{O_3} h_n(R\nabla x(u)) \cdot h_{n'}(R\nabla x(u')) \cdot \delta_{u'}(u + R^T d)., \tag{5.15}$$

Now we have to consider the functions h_n more closely. According to eq.(5.2), h_n has a non-zero value only if n and v are parallel or anti-parallel. From this fact, it can be concluded, that the integrand of $I_\Pi(x)$ differs from zero only if the following conditions hold:

$$\begin{aligned} n &\parallel R\nabla x(u) && (5.16)\\ n' &\parallel R\nabla x(u') && (5.17)\\ d &\parallel (u - u') && (5.18)\\ |d| &= |u - u'| && (5.19) \end{aligned}$$

The first three conditions are fulfilled if n, n' and d have the same configuration up to rotation as $\nabla x(u)$, $\nabla x(u')$ and $u - u'$.

We can now rewrite the eq. (5.15) to:

$$I_\Pi = \int_{\mathbb{R}^6} \theta_{n,n',d} \cdot \delta_\Delta(|u - u'|) \cdot |\nabla x(u)| \cdot |\nabla x(u')|\, du\, du'\, dR, \tag{5.20}$$

Here $\theta_{n,n',d}$ is a configuration specific function. In eq.(5.9-5.11), we have uniquely defined the configuration of n, n' and d by α, β, γ and δ.

Now, the definition of the gradient $\nabla x(r)$ of eq. (5.7) is inserted into the integral of eq.(5.20). The integral reduces to a sum:

$$I_\Pi = \sum_{i,k} \theta_{n,n',d} \cdot \delta_\Delta(|u_i - u'_k|) \cdot |\nabla x(u_i)| \cdot |\nabla x(u'_k)|, \tag{5.21}$$

The result of the computation is nothing else but a special histogram: The frequency of occurrence of two gradients in a specific distance with a particular relative configuration is computed. Each bin of the three dimensional histogram represents a particular configuration. For each pairwise configuration of atoms of the protein, the appropriate bin is updated by the product $|\nabla x(u_i)| \cdot |\nabla x(u'_k)|$. In fact, the GI is closely related to the Shape distribution computed by Osada et al. [22], where a histogram of the pairwise distances of points on a 3D object surface is computed.

Extending GI with Spherical Harmonics

The SH transform is now applied to the GI in eq.(5.20). The Delta function $\delta_\Delta(|u-u'|)$ is evaluated. The Delta function has a none-zero value only if $u' = u + \Delta \cdot s$, such that $|s| = 1$ and $s \in S^2$.

So we obtain the sphere integral, where s ranges over the unit-sphere S^2:

$$I_\Pi = \int_{S^2} \int_{\mathbb{R}^3} \theta_\Pi \cdot |\nabla x(u)| \cdot |\nabla x(u + \Delta \cdot s)| \, du \, ds \tag{5.22}$$

Instead of integrating the function, we compute its projection onto the spherical harmonics $Y_l^m(s)$:

$$I_\Pi^{lm} = \int_{S^2} \int_{\mathbb{R}^3} \theta_\Pi \cdot |\nabla x(u)| \cdot |\nabla x(u + \Delta s)| \cdot Y_{lm}^*(s) \, du \, ds. \tag{5.23}$$

And by substituting the definition of $\nabla x(u)$:

$$I_\Pi^{lm} = \sum_{i,k} \theta_\Pi \cdot |\nabla x(u_i)| \cdot |\nabla x(u_k)| \cdot Y_{lm}^* \left(\frac{u_i - u_k}{|u_i - u_k|} \right). \tag{5.24}$$

For $l = 0$, the integral has the same value as the integral in eq.(5.20). For $l > 0$ the computation is very similar to eq.(5.21). Instead of adding $|\nabla x(u_i)| \cdot |\nabla x(u_k)|$ to the histogram bin related to the configuration Π, the added value is multiplied by the complex factor $Y_{lm}(\frac{u_i - u_k}{|u_i - u_k|})$.

Each bin has now two more indices, namely l and m. After computation, the results are made invariant by computing the bandwise energy:

$$\sum_{m=-l}^{l} |I_\Pi^{lm}|^2.$$

So, for example if $l = 1$ (See Figure 5.4), we obtain two values for each multidimensional histogram bin, after computing the bandwise energy. In general, $l + 1$ invariant features are computed for each n, n', d confirmation.

Extending GI with D-Wigner Matrices

An even more accurate way of describing the rotation group $SO(3)$ is using D-Wigner matrices. The group integral I_Π can be rewritten to:

$$I_\Pi(g) = \int_G k(g) D^l(g) dg$$

The irreducible representatives of $SO(3)$ are called D-Wigner matrices and are denoted $D^{(l)}$, where $\dim D^l = 2l + 1$. In particular, $D^0(R) = 1$ and $D^1(R) = U^+ R U$.

In order to compute $D^1(R)$, we need to find the rotation R which transforms the configuration of the gradients $\nabla x(u)$ and $\nabla x(u')$ and the difference vector $u - u'$ into

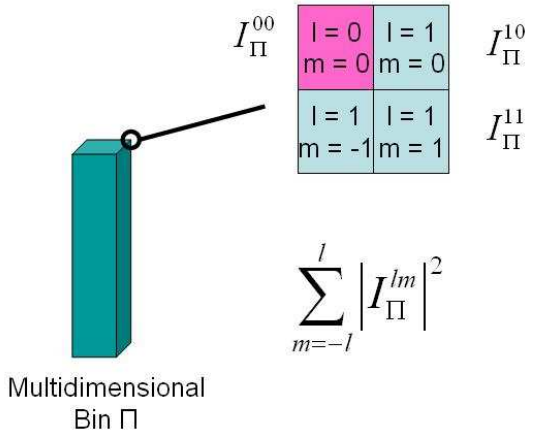

Figure 5.4: **The bandwise energy of the SH.** For the bin $\Pi = \{\alpha, \beta, \gamma, \Delta\}$ the bandwise energy of the SH is computed. For $l = 1$, two invariant features are obtained by taking the bandwise energy.

the standard configuration of the parameters n, n' and d. Rather than taking the actual parameters n, n' and d, the angles $\overline{\alpha}, \overline{\beta}$ and $\overline{\gamma}$ will be considered. So we have to solve the equation:

$$RU = Z, \tag{5.25}$$

where U is a 3×3 matrix with columns consisting of the vectors $d = (1, 0, 0)^T$, $n = (\cos\overline{\alpha}, \sin\overline{\alpha}, 0)$ and $n' = (\cos\overline{\beta}, \sin\overline{\beta}\cos\overline{\gamma}, \sin\overline{\beta}\sin\overline{\gamma})$, and:

$$\cos\overline{\alpha} = \frac{n^T \cdot d}{|d|}$$

$$\cos\overline{\beta} = \frac{n'^T \cdot d}{|d|}$$

$$\cos\overline{\gamma} = \left(\frac{n - \cos\overline{\alpha}\frac{d}{|d|}}{\left| n - \cos\overline{\alpha}\frac{d}{|d|} \right|} \right)^T \cdot \frac{n' - \cos\overline{\beta}\frac{d}{|d|}}{\left| n' - \cos\overline{\beta}\frac{d}{|d|} \right|}$$

The standard confirmation is depicted in Fig.5.5. The angle $\overline{\gamma}$ is the angle between the projection of n on d and the projection of n' on d.

We can compute $D^1(R)$ by:

$$D^1(R) = R = [d, n, n'] \cdot \begin{pmatrix} 1 & \cos\overline{\alpha} & \cos\overline{\beta} \\ 0 & \sin\overline{\alpha} & \sin\overline{\beta}\cos\overline{\gamma} \\ 0 & 0 & \sin\overline{\beta}\sin\overline{\gamma} \end{pmatrix}^{-1}$$

Since the values of D^l for $l > 0$ are matrices of size $(2 \cdot l + 1) \times (2 \cdot l + 1)$, we have to compute the norm of each column of the matrix to obtain invariant features. Thus each

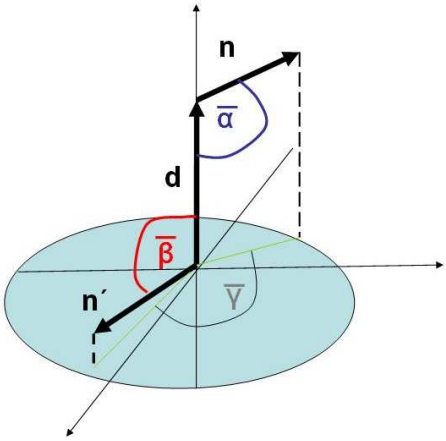

Figure 5.5: **Configuration** $(\overline{\alpha}, \overline{\theta}, \overline{\gamma})$**.** The standard configuration can be described by three angles $(\overline{\alpha}, \overline{\theta}, \overline{\gamma})$ which are formed by the three unit vectors d, n and n'. The three angles describe uniquely the configuration of the three vectors.

configuration D^l produces $l \cdot (l+2) + 1$ invariant features, e.g for $l = 1$ we have four invariant features.

In [23], an explicit formula for D-Wigner matrix computation is given. The computation of $D^l(R)$ only depends on the the values of $D^1(R)$ and $D^{l-1}(R)$ and thus can be computed iteratively.

Extending the kernel function

Since the protein hierarchy starts with a linear sequence, every amino acid is associated with a sequence number. This information should be incorporated into the kernel in eq. (5.3). Since every C_α atom belongs to one amino acid and is hence associated with a sequence number we can define an index mapping $I : \mathbb{R}^3 \mapsto \mathbb{R}$. The new kernel is:

$$k^P_{\Pi,i}(x) = k_{d,n,n'}(x) \cdot \delta_i(|I(0) - I(d)|).$$

This formula guarantees that the coordinate-distance and the sequence-distance between two C_α-atoms are closely related. Depending on the bin size for i, we can enhance or diminish the importance of the sequential distance.

Local Feature Extraction

In the previous section, the feature extraction for features describing the global shape of proteins was introduced. Another idea would be to consider features describing the local surroundings of each atom and include this information in the global description. For each atom, only a restricted neighborhood is considered. One GI is computed per atom j, where j indicates the sequence number on the chain :

$$I^{lm}_{\Pi,n}(j) = \sum_{i=-n/2}^{n/2} \theta_\Pi \cdot |\nabla x(u_i)| \cdot |\nabla x(u_j)| \cdot Y^*_{lm}\left(\frac{u_i - u_j}{|u_i - u_j|}\right), \tag{5.26}$$

where n is the cutoff and determines the size of the considered neighborhood. The pairwise comparison of the local features of two proteins is too expensive. Therefore, a histogram of the group integrals is computed:

$$I^{lm}_{\Pi,n}(q) = \sum_{j=1}^{N} \delta(\left|I^{l}_{\Pi,n}(j)\right| - q), \tag{5.27}$$

where N is the number of C_α-atoms of the protein and:

$$\left|I^{l}_{\Pi,n}(j)\right| = \sqrt{\sum_{m=-l}^{l} \left|I^{lm}_{\Pi,n}(j)\right|^2}.$$

5.2 Feature Selection

Since large feature vectors can be computed using SH expansion or D-Wigner matrices, the question arises: Which of the features are relevant at all for the classification process? The goal is to eliminate the misleading features and keep the features with good discrimination properties.

Since the labels on the training set are known, a supervised classification problem needs to be solved. Two conceptual frameworks for feature selection exist: the *filter model* and the *wrapper model*. In the filter model the selection is done in a preprocessing step with the help of an evaluation function. In the wrapper model the performance of a specific algorithm is optimized directly. However, the wrapper model is very high time consuming. A good overview of feature selection techniques is presented in [7].

The RELIEF [16] and SIMBA [5] algorithms are filter model based approaches. They were chosen because they are easy to implement and fast to compute. For a set S of $T = |S|$ labeled feature vectors, a weighting vector w is computed indicating which features are relevant. Each feature vector $x \in S$ has the size N. The vector w contains the relevance of each feature. Features with a weight w_i below a certain threshold τ can be discarded.

For the computation of the algorithms, the *nearest hit* and the *nearest miss* for a feature vector x needs to be defined. It is assumed that the vectors are labeled, so we define:

$$nearhit(x) := \underset{\substack{y \in S \\ label(y)=label(x)}}{argmin} \; d(x,y)$$

$$nearmiss(x) := \underset{\substack{y \in S \\ label(y)\neq label(x)}}{argmin} \; d(x,y).$$

So the nearest neighbor to sample x from the same class is called $nearhit$ while the nearest neighbor from a different class is called $nearmiss$.

5.2.1 RELIEF

The RELIEF algorithm holds a weight vector over all features and updates this vector according to the sample points presented. The algorithm selects features which can separate neighboring samples well. The weighting vector w is computed by considering the difference of a single feature from the sample x and its $nearhit$ and $nearmiss$. If these differences are far apart from each other, than the single feature has a good separation ability and thus should be selected. The difference with the $nearmiss$ value should be as large as possible, while the difference to the $nearhit$ should be as small as possible.

The threshold τ determines which features are chosen.

Algorithm 1 RELIEF

1: Initialize the weighting vector w with a zero vector: $w = (0, ..., 0)^T$.
2:
3: **for** $t = 1$ to T **do**
4: Pick randomly an instance x of S.
5:
6: **for** $i = 1$ to N **do**
7: $w_i = w_i + (x_i - \text{nearmiss}(x)_i)^2 - (x_i - \text{nearhit}(x)_i)^2$
8: **end for**
9: **end for**
10: Choose the feature set $\{i | w_i > \tau\}$, where τ is the threshold.

RELIEF does not re-evaluate the distances according to the weight vector w. In particular, RELIEF has no mechanism for eliminating redundant features. Thus, the SIMBA algorithm is also considered since it incorporates the already computed weights, while computing new weights.

5.2.2 SIMBA

The SIMBA algorithm uses a margin based criteria to measure the quality of each feature. The margin of x is defined by

$$\theta_P^w = \frac{1}{2}(\|x - nearmiss(x)\|_w - \|x - nearhit(x)\|_w),$$

where P is a set of points and w is the weight vector. The w-Norm $\|z\|_w$ and $w, z \in \mathbb{R}^N$ is defined by:

$$\|z\|_w = \sqrt{\sum_{i=1}^{N} w_i^2 z_i^2}.$$

The margin measures the classifiers confidence when making its decision. If many sample points have a large margin, a good generalization is guaranteed. An evaluation function is defined to measure the margin induced by a set of features.

Given a training set S and a weight vector w, the evaluation function is:

$$e(w) = \sum_{x \in S} \theta^w_{S \setminus x}(x)$$

SIMBA first finds the weight vector w that maximizes $e(w)$ and then uses a threshold in order to get a feature set. A stochastic gradient ascent is used to maximize $e(w)$.

Algorithm 2 SIMBA

Initialize the weighting vector w by: $w = (1, ..., 1)^T$.

for $t = 1$ to T **do**
  Pick randomly an instance x of S.
  Calculate $nearmiss(x)$ and $nearhit(x)$ with respect to $S \setminus x$ and the weight vector w.

  for $i = 1$ to N **do**
    $\Delta_i = \frac{1}{2} \left(\frac{(x_i - \text{nearmiss}(x)_i)^2}{\|x - \text{nearmiss}(x)\|_w} - \frac{(x_i - \text{nearhit}(x)_i)^2}{\|x - \text{nearhit}(x)\|_w} \right) \cdot w_i$
    $w = w + \Delta$
  end for
  $w \leftarrow w^2 / \|w^2\|_\infty$ where $(w^2)_i := (w_i)^2$
end for

The great advantage of SIMBA is that it can even choose correlated features if this contributes to the overall performance.

In terms of computational complexity, RELIEF and SIMBA are equivalent. They can be computed in $O(T \cdot N \cdot m)$ where T is the number of iterations, N is the number of features and m is the size of the sample S.

5.3 Distance Measures

After the features for the domains are computed different distance measures can be used to determine their similarity score. In section 3.2.1, for example, the L1-norm is used to compute the distance. In the first section different distance measures for feature vectors are introduced.

In the second section the idea of partitioning the domain into smaller units is presented. For each partitioned domain a set of feature vectors is computed. Thus, different similarity measures for vector sets are defined.

5.3.1 Simple Distance Measures

It is assumed that feature vectors $\mathbf{x} = (x_0, x_1, ..., x_n)^T$ and $\mathbf{y} = (y_0, y_1, ..., y_n)^T$ of the same length n are compared. Different L-distance measures $d(\mathbf{x}, \mathbf{y})$ as described in Table 5.1 were tested on the four data sets.

Two statistical χ^2-distances were also used (See Table 5.2).

Distance measure	abbreviation	Formula
Manhattan Distance	L1	$d_{L1}(\mathbf{x},\mathbf{y}) = \sum_{i=0}^{n} \lvert x_i - y_i \rvert$
Euclidean Distance	L2	$d_{L2}(\mathbf{x},\mathbf{y}) = \sqrt{\sum_{i=0}^{n}(x_i - y_i)^2}$
Maximum Distance	L_∞	$d_\infty(\mathbf{x},\mathbf{y}) = \max_i \lvert x_i - y_i \rvert$

Table 5.1: The L-distance measures used for feature vector comparison.

Distance measure	abbreviation	Formula
χ_1^2- Distance	χ_1^2	$d_{\chi_1^2}(x,y) = \sum_{i=0}^{n} \frac{(x_i-y_i)^2}{x_i+y_i}$
χ_2^2- Distance	χ_2^2	$d_{\chi_2^2}(x,y) = \sum_{i=0}^{n} \frac{(x_i-y_i)^2}{x_i}$

Table 5.2: The χ^2-distance measures used for feature vector comparison.

5.3.2 Domain Partitioning

We could use domains as defined by SCOP or CATH for classification. But is there a simple algorithm to define the domain automatically?

An easy way to define a domain would be to consider a chain and perform a cut if two conditions are fulfilled:

1. The distance between the coordinates of two C_α-atoms on the chain is very large.
2. The resulting domain contains at least K amino acids.

The first condition implies that different domains on a chain are separated in space by large distances. The second condition ensures that the chosen domains have a minimal size.

In order to check condition one, a distance map is computed with two free parameters a and b:

$$c_{ij} = e^{-(a|i-j|+b\|u_i-u_j\|_1^2)} \tag{5.28}$$

The minimum of the distance map is computed by:

$$m^* = \underset{1\le m\le n}{\operatorname{argmin}} \sum_{i<m<j} c_{ij}, \tag{5.29}$$

where n is the number of amino acids on the chain. The visualization of the sum in eq.5.29 is presented in Figure 5.6.

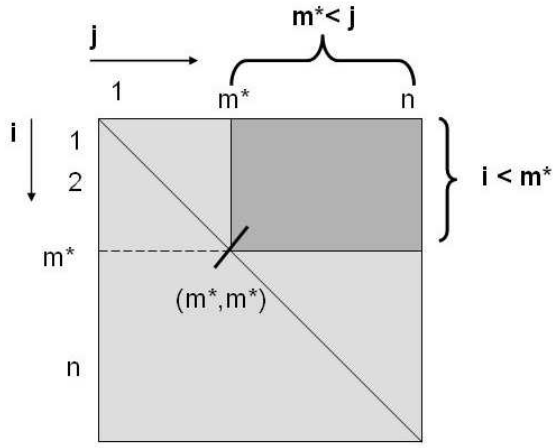

Figure 5.6: **Domain partitioning** The distance map for domain partitioning is presented. The cut is made at the point (m^*, m^*). The dark gray rectangle marks the sum indicated in eq. 5.30.

The cut is made if:

$$\sum_{i<m^*<j} c_{ij} \leq \alpha(n - m^*)m^* \tag{5.30}$$

The second condition is fulfilled if:

$$m^* > t \quad \text{and} \quad (n - m^*) > t.$$

Here the parameters α and t have to be chosen.

5.3.3 Measures on Vector Sets

After the domain partitioning, a chain is associated with a set of domains. For each of these domains a feature vector is computed. Thus a distance measure for vector set comparison has to be found.

The general problem is stated as follows: Find the distance between two vector sets $\mathbf{V} = \{\mathbf{V}_1, \mathbf{V}_2, ..., \mathbf{V}_n\}$ and $\mathbf{W} = \{\mathbf{W}_1, \mathbf{W}_2, ..., \mathbf{W}_m\}$ where the two vector sets $\mathbf{V}$ and $\mathbf{W}$ contain vectors of same size. Manhattan or χ^2 distance are used to compute the distance d between two vectors.

Next Neighbor Distance

The *Next Neighbor Distance* computes the minimal distance between two vectors from the two vector sets.

$$D1(\mathbf{V}, \mathbf{W}) = \min_{i,j} d(\mathbf{V}_i, \mathbf{W}_j)$$

Sum of Next Neighbor Distance

All vectors in the set could be considered by finding for each vector in one set the corresponding vector in the other set. These distances are then summed up to compute the final distance. Since the matching is not injective, the problem can be solved in $O(n)$. However this method does not induce a metric.

$$D2(\mathbf{V}, \mathbf{W}) = \sum_i \min_{i,j} d(\mathbf{V}_i, \mathbf{W}_j)$$

Minimum Matching Distance

The *Minimum Matching Distance* (MMD) is computed using the Hungarian algorithm [17]. The Hungarian algorithm solves the problem of assigning to each vector in one set one and only one corresponding vector in the other set so that the sum of the pairwise distances is minimal. We assume that $|\mathbf{V}| < |\mathbf{W}|$ and let π be an injective function.

$$D3(\mathbf{V}, \mathbf{W}) = \min_{\pi} \sum_i d(\mathbf{V}_i, \mathbf{W}_{\pi(i)})$$

For the computation of the MMD the C-implementation libhungarian-v0.1.2.tgz library from Cyrill Stachniss was used (http://www.informatik.uni-freiburg.de/ stachnis/misc.html). The computation complexity is $O(n^3)$.

5.4 New Algorithm

The algorithm of the new method has 3 steps:

1. Extract the features.
2. (Select the relevant features.) *optional*
3. Compute the distance matrix.

Extract the features.

The feature extraction for GI features can be described by Algorithm 3, where N is the number of protein structures in the data base.

Algorithm 3 GI Algorithm

Initialize $I_\Pi = 0$
for $i = 1$ to N **do**
  for $j = 1$ to N **do**
    Compute $\Pi = \{\alpha, \beta, \gamma, \Delta\}$
    $\alpha = \frac{\nabla x(u_i)}{|\nabla x(u_i)|}^T \frac{(u_i - u_j)}{|(u_i - u_j)|}$
    $\beta = \frac{\nabla x(u_j)}{|\nabla x(u_j)|}^T \frac{(u_i - u_j)}{|(u_i - u_j)|}$
    $\gamma = \frac{\nabla x(u_i)}{|\nabla x(u_i)|}^T \frac{\nabla x(u_j)}{|\nabla x(u_j)|}$
    $\Delta = |u_i - u_j|$
    Update $I_\Pi \rightarrow I_\Pi + |\nabla x(u_i)| \cdot |\nabla x(u_j)|$
  end for
end for

The algorithm can be further extended by SH and D-Wigner matrices. For the SH only the last line of the algorithm needs to be changed (See Algorithm 4). The Legendre polynomials necessary for the Spherical Harmonics computation were computed with the MATPACK 1.8.1 library [2] by Berndt M. Gammel. The resulting array gets two additional indices l and m. After computation, the results are made invariant by computing the bandwise energy $\sum_{m=-l}^{l} |I_\Pi^{lm}|^2$.

For the D-Wigner matrices, Algorithm 5 is applied. Now, the resulting array I_Π^l is matrix valued, where each $I_\Pi^l \in \mathbb{C}^{(2l+1)\times(2l+1)}$ is a complex-valued matrix. The result array is made invariant by taking norms columnwise. Hence, for each l we obtain $2l + 1$ invariant features, instead of one feature as for the SH-based algorithm.

In all three cases, I_Π is a four dimensional histogram. The expansion parameter l adds to I_Π after normalization a further dimension for the SH and the D-Wigner methods. This histogram is transformed to a one dimensional vector by Algorithm 6.

[2] http://www.matpack.de

Algorithm 4 SH Algorithm

Initialize $I_{\Pi}^{l,m} = 0$
for $i = 1$ to N **do**
  for $j = 1$ to N **do**
    Compute $\Pi = \{\alpha, \beta, \gamma, \Delta\}$
    Update $I_{\Pi}^{l,m} \rightarrow I_{\Pi}^{l,m} + Y_m^l(\frac{u_i - u_j}{|u_i - u_j|}) \cdot |\nabla x(u_i)| \cdot |\nabla x(u_j)|$
  end for
end for

Algorithm 5 D-Wigner Algorithm

Initialize $I_{\Pi}^{l} = 0$
for $i = 1$ to N **do**
  for $j = 1$ to N **do**
    Compute $\Pi = \{\alpha, \beta, \gamma', \Delta\}$
    Determine $R = MV_{\alpha,\beta,\gamma'}^{-1}$
    where $M = [\frac{\nabla x(u_i)}{|\nabla x(u_i)|}, \frac{\nabla x(u_j)}{|\nabla x(u_j)|}, \frac{(u_j - u_i)}{|u_j - u_i)|}]$
    Update $I_{\Pi}^{l} \rightarrow I_{\Pi}^{l} + |\nabla x(u_i)| \cdot |\nabla x(u_j)| \ \cdot D^l(R)$
  end for
end for

Algorithm 6 I_{Π} to Feature vector

In the four dimensions of I_{Π} the number of bins is n_1, n_2, n_3 and n_4.
count = 0;
for $i = 1$ to n_1 **do**
  for $j = 1$ to n_2 **do**
    for $k = 1$ to n_3 **do**
      for $l = 1$ to n_4 **do**
        feat_vector[count] = I_{n_1,n_2,n_3,n_4}
        count = count + 1
      end for
    end for
  end for
end for

Compute the distance matrix.

The distance matrix ($dmat$) is computed by computing the pairwise distance d between two feature vectors (*feat_vector*). If the data set contains N protein structures, than the distance matrix is computed by:

Algorithm 7 dmat computation

for $i = 1$ to N **do**
  for $j = 1$ to N **do**
    $dmat(i, j) = d(\text{feat\_vector}_i, \text{feat\_vector}_j)$
  end for
end for

Chapter 6

Benchmark of Methods

For the evaluation of the new method and the comparison with existing methods, the Princeton Shape Benchmark (PSB)[31] is used. The PSB is presented in the first section.

In the second section, the experimental framework is introduced and the new method is evaluated for different parameters. In the third section, different distance measures and domain partitioning was tested. The results of feature selection on the experimental datasets i presented in section four. In section five the results of the new method are compared to the results of state-of-the-art automatic classification methods. Finally, the time requirements for the different datasets and methods are presented in section 6.

6.1 Evaluation Tools

The PSB provides a suite of tools for comparing shape matching and classification algorithms. The evaluation is based on five statistical measures: Nearest Neighbor (NN), First-Tier, Second-Tier, E-Measure and the Discounted Cumulative Gain (DCG). The same procedure starts the computation for all five measures: Each object of the database is taken as a query object and the distances to all other query objects are computed and stored in a distance matrix. The five statistical measures are computed based on the distance and the class label.

The *Nearest Neighbor* measures the percentage of the closest matches that belong to the same class as the query. This provides an intuition on how well a nearest neighbor classifier would perform. The desired value for this measure is of course 100%.

The *First-* and the *Second-Tier* measure the percentage of models in the query's class that appear within the top K matches, where K depends on the size of the query's class. Specifically, for a class with $|C|$ members, $K = |C| - 1$ for the first tier, and $K = 2 \cdot (|C| - 1)$ for the second tier. The optimal result has the value 100%.

The *E-Measure* is a composite measure of the precision P and recall R for a fixed number of retrieved results, where P and R are defined by:

$$P = \frac{|\{relevant\ structures\}| \cap |\{found\ structures\}|}{|\{found\ structures\}|} \tag{6.1}$$

$$R = \frac{|\{relevant\ structures\}| \cap |\{found\ structures\}|}{|\{relevant\ structures\}|} \tag{6.2}$$

Since the user is more interested in the query results with a high similarity to the input query, only the first 32 most similar retrieved results are considered. After computing the precision and recall for those results, the E-Measure is obtained by:

$$E = \frac{2}{\frac{1}{P} + \frac{1}{R}}.$$

The higher the E-Measure value the better the result, with the perfect score being 100%.

The DCG weighs the results near the front of the list more than correct results later in the ranked list. For details on the computation of DCG see [31]. The best value for the DCG is 1.0.

6.2 Classification Results

In this section the results for the Group Integral features are presented and evaluated with the PSB on experimental data sets.

6.2.1 Experimental data sets

The features were evaluated on four representative data sets called: 'all-classes', 'all-alpha', '27fold' and 'cath'. The first three data sets are labeled according to the SCOP classification. The fourth data set consists of all CATH 2.4 entries.

The 'all-classes' data set consists of all SCOP entries with the sccs ?.1.1.1. The 'all-alpha' data set contains the entries with the sccs a.?.?.?. The '27fold' data set was proposed for the testing of fold classification by [14]. The data set was selected by their characteristics so that all proteins in the data set have less than 40% of the sequence identity for the aligned subsequences longer than 80 residues.

The number of entries and the classification used for the four data sets is represented in Table 6.1.

dataset	**# of domains**	**classification level**	# of classification classes
all-classes	2,650	SCOP-class	7
all-alpha	3,680	SCOP-fold	172
27fold	685	SCOP-fold	27
cath	20,937	CATH-homology	2147

Table 6.1: **Testing datasets.** Number of domains for the classification of the four testing sets: 'all-classes', 'all-alpha', '27fold' and 'cath'.

6.2.2 Implementation details

The feature extraction was implemented as described in the previous chapter. Experiments were conducted for the Group Integrals without Spherical Harmonic Expansion (noSH),

with Spherical Harmonic Expansion (SH) and with D-Wigner Expansion (D-Wigner).

According to experimental results, the parameters were chosen as in Table 6.2. The parameter σ is the variance used in eq. 5.7. Since the histograms only consider values in the range between 0 and 1, the distance Δ and the sequential distance between proteins is downscaled to this range. Here it is assumed that atoms further apart than $20A^\circ$ in coordinate distance and 40 apart according to their sequence distance can be discarded. The four dimensional histogram consists of bins as indicated in the table. The coordinate distance can be assigned to 16 bins, the angles α and γ to two bins and the sequential distance to 8 bins. For the Spherical Harmonics and the D-Wigner matrices l_{sharm} and l_{dwig} are the cutoff for the expansion.

If not stated otherwise these parameters were used for computation.

Gradient computation	σ	400
Coordinate Distance Scaling	DScale	0.02
Sequence Distance Scaling	SeqDScale	40
Histogram Bin Dimension	hist Π	[16,2,2,8]
Spherical Harmonics Coefficient	l_{sharm}	1
D-Wigner Matrix Coefficient	l_{dwig}	1

Table 6.2: **Parameter set for the new method.**If not stated otherwise, this parameter set was used for the computation of the GI, SH and D-Wigner features.

Further, the performance of different expansion coefficients l was examined for the SH and the D-Wigner features.

Finally, the Local Features based on Spherical Harmonics (LF-SH) were examined. Here the histogram bin size per atom (perAtom) and per protein (perProtein) could be variated.

The implemented C++ classes and methods are briefly presented in Appendix C.

6.2.3 Experimental results

In Table 6.3 the results for the 'all-classes' dataset are presented. As expected, the classification rate is very high since division into SCOP classes is quite an easy task. The SH did not improve the already very good classification results. The D-Wigner results are worse than SH, although we would expect them to give better results. The D-Wigner matrix features will be evaluated in more detail later in this section.

Feature	**1NN**	**1T**	**2T**	**EM**	**DCG**
noSH	99.8	86.8	91.4	13.4	96.7
SH	99.8	87.6	92.5	13.4	97.2
D-Wigner	99.5	86.1	89.9	13.3	96.3

Table 6.3: **Results 'all-classes'.** Results on the 'all-classes'-dataset with GI, SH and D-Wigner features.

In Table 6.4, the search is performed on the 'all-alpha' dataset. The results are not as good as on the 'all-classes'-dataset since the classification into folds is a more difficult task. However, 97.8% is a quite high classification rate and the SH improve the GI features.

Feature	1NN	1T	2T	EM	DCG
noSH	97.4	84.8	88.6	35.6	94.4
SH	97.8	89.3	92.2	37.4	96.0
D-Wigner	97.4	87.5	90.4	36.8	95.2

Table 6.4: **Results 'all-alpha'.** Results on the 'all-alpha'-dataset with GI, SH and D-Wigner features.

In Table 6.5, the results for the '27-folds'-dataset are presented. They are much worse than the previous two testing sets since the domains of the '27folds' have less than 40% sequential similarity and are thus hard to classify. Also, the number of samples per class is far less than in the previous two sets. The number of samples per class is important, since it increases the probability to find a similar structure in one class.

Feature	1NN	1T	2T	EM	DCG
noSH	77.3	31.0	41.2	27.2	67.9
SH	78.8	32.4	44.7	28.7	69.3
Dwigner	77.8	29.5	39.1	26.2	66.8

Table 6.5: **Results '27fold'.** Results on the 'all-classes'-dataset with GI, SH and D-Wigner features.

In Table 6.6, the results for the 'cath' - dataset are presented. They are very good since the homologous are very well populated and therefore one similar structure to the query structure could be always retrieved.

Feature	1NN	1T	2T	EM	DCG
SH	98.9	72.6	77.7	41.2	91.1
Dwigner	98.8	71.0	75.2	41.0	89.9

Table 6.6: **Results 'cath'.** Results on the 'cath'-dataset with GI, SH and D-Wigner features.

In Figure 6.1, the Precision-Recall graph for the four datasets is plotted. The optimal graph for this plot is when the line is perfectly horizontal and has the value one. The 'all-alpha' and 'all-classes' datasets have a very good retrieval rate, while the '27folds' dataset performs worse.

In Table 6.7 the results for the SH and the D-Wigner features with different expansion coefficients l on the '27folds' dataset are presented. The higher the expansion coefficient

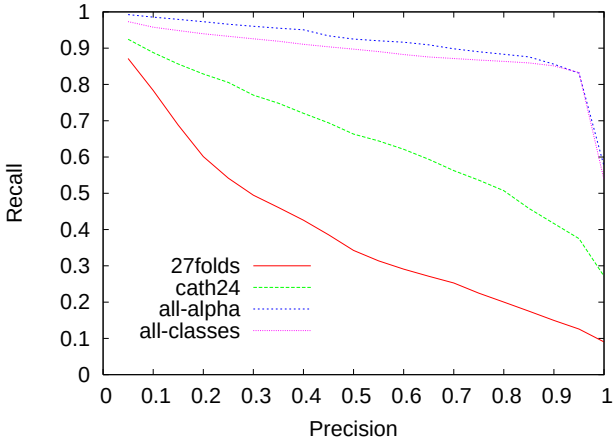

Figure 6.1: **PR-graph for the four datasets.** The PR-graph for the four datasets 'all-classes', 'all-alpha', '27folds' and 'cath'. The features were computed with SH.

l gets, the worse is the classification rate. This is true for both the SH and the D-Wigner expansion coefficients. An explanation for this result is that the level of detail for protein retrieval should not be to high, since we are not looking for the exact same copies of a structure but for similar relatives. Hence given less coefficients, the description of the structure stays more general.

Feature	l	**1NN**	**1T**	**2T**	**EM**	**DCG**
SH	1	78.8	32.4	44.7	28.7	69.3
SH	2	78.4	31.8	43.6	28.1	68.6
SH	3	77.5	31.2	42.8	27.5	68.1
SH	4	76.0	29.7	41.0	26.3	66.9
Dwigner	1	77.8	29.5	39.1	26.2	66.8
Dwigner	2	77.2	29.6	39.2	26.1	66.8
Dwigner	3	76.9	29.4	38.7	25.7	66.5

Table 6.7: **Results for different expansion coefficients l.** Results for the '27folds'-dataset for the SH and the D-Wigner based on different expansion coefficients l.

In Table 6.8 the results for the LF-SH were examined. By varying the bin size per atom and per protein, different results were achieved. The LF-SH features perform by 8.1% worse than the global SH features.

Feature	{DScale, SeqDscale}	perAtom	perProtein	1NN	1T	2T	EM	DCG
LF-SH	{0.02, 24}	{10,2,2,6}	10	68.0	22.5	31.8	19.9	60.3
LF-SH	{0.02, 40}	{10,2,2,6}	10	70.7	23.0	31.0	20.6	60.8
LF-SH	{0.02, 50}	{10,2,2,6}	10	68.6	22.2	32.8	21.3	61.4
LF-SH	{0.01, 24}	{10,2,2,6}	10	60.1	18.6	28.0	16.3	57.1
LF-SH	{0.03, 24}	{10,2,2,6}	10	70.5	23.6	32.9	21.1	61.1
LF-SH	{0.04, 24}	{10,2,2,6}	10	63.9	21.9	30.8	19.4	59.3
LF-SH	{0.03, 40}	{10,2,2,6}	10	69.1	24.0	32.4	21.5	61.3
LF-SH	{0.02, 40}	{5,2,2,6}	10	67.0	20.6	28.2	18.7	58.5
LF-SH	{0.02, 40}	{15,2,2,6}	10	68.1	22.1	30.5	19.8	59.9
LF-SH	{0.02, 40}	{10,2,2,6}	5	67.2	23.1	32.1	20.6	60.6
LF-SH	{0.02, 40}	{10,2,2,6}	15	67.2	20.9	28.6	18.8	58.9
LF-SH	{0.03, 24}	{10,2,2,3}	7	69.2	23.9	33.4	21.4	61.0
LF-SH	{0.03, 24}	{6,2,2,3}	6	67.6	24.8	34.2	22.6	61.3

Table 6.8: **Results for LF-SH features.** Results for the '27folds'-dataset with LF-SH features.

6.3 Results for different Distance Measures

The results obtained by different distance measures are presented on three of the four datasets. In the first experiment, the results without domain partitioning are considered. The distance measures as described in Table 5.1 and Table 5.2 were used. In the second experiment, domain partitioning is performed automatically and the distance is computed by the distance measures $D1 - D3$.

6.3.1 Results without Domain Partitioning

Using domain partitioning according to SCOP, the performance of different L-norms and statistical based χ^2 distance measures is evaluated in Table 6.9. The features were computed with the same parameters as in Table 6.2. The $L1$ and the $L2$ distance measures perform best, while the χ^2_2 distance measure performs worse.

6.3.2 Results with Domain Partitioning

In Table 6.10 the distance measures $D1 - D3$ were tested with the parameters $a = 0.01, b = 0.001, \alpha = 5, t = 50$ (See eq.5.28 and eq.5.30). Since $L1$ performed best in the previous section for the L-norm based distance measures and χ^2_1 performed best for the χ^2 distance measures, they were chosen for the computation of $D1 - D3$. The $D2$ distance measure performs best. However, the results with domain partitioning are by 4.3% worse on the '27folds' dataset than results without domain partioning.

Although the proposed domain partitioning algorithm did not improve the results, the author believes that a good domain partitioning algorithm can be found. For this case, different distance measures on vector sets are defined.

Class	d	1NN	1T	2T	EM	DCG
all-classes	L1	99.8	87.6	92.5	13.4	97.2
all-classes	L2	99.8	86.8	90.7	13.1	96.8
all-classes	L_∞	99.5	84.6	89.3	13.1	96.2
all-classes	χ^2_1	99.8	86.1	91.9	13.4	97.0
all-classes	χ^2_2	93.8	61.1	74.4	9.3	89.3
all-alpha	L1	97.8	89.4	92.2	37.1	96.0
all-alpha	L2	97.8	89.2	92.0	37.0	96.0
all-alpha	L_∞	97.8	88.5	91.9	37.0	95.8
all-alpha	χ^2_1	97.8	88.5	91.9	37.0	95.8
all-alpha	χ^2_2	93.4	52.3	58.5	29.5	83.4
27folds	L1	78.8	32.4	44.7	28.7	69.3
27folds	L2	78.7	32.5	44.4	28.9	68.9
27folds	L_∞	71.7	28.2	40.1	25.1	65.1
27folds	χ^2_1	77.8	31.0	42.8	27.5	68.0
27folds	χ^2_2	54.9	15.2	21.6	13.6	53.4

Table 6.9: **Results without domain partitioning.** Results without domain partitioning using L-norm and the χ distance measures.

Class	D	d	1NN	1T	2T	EM	DCG
all-classes	D1	L1	99.8	87.4	92.5	13.2	97.1
all-classes	D2	L1	99.8	87.9	91.7	13.3	96.9
all-classes	D3	L1	97.3	78.5	88.8	11.6	94.7
all-classes	D1	χ^2_1	99.7	86.6	91.9	13.1	96.9
all-classes	D2	χ^2_1	99.7	87.2	91.2	13.2	96.7
all-classes	D3	χ^2_1	98.6	81.3	88.5	11.6	95.1
all-alpha	D1	L1	97.6	86.0	88.8	35.9	94.5
all-alpha	D2	L1	97.7	87.0	90.2	36.7	95.2
all-alpha	D3	L1	97.7	87.1	90.2	36.7	95.2
all-alpha	D1	χ^2_1	97.4	85.2	88.2	35.7	94.2
all-alpha	D2	χ^2_1	97.8	89.3	92.2	37.4	96.0
all-alpha	D3	χ^2_1	97.4	85.7	88.8	36.2	94.5
27folds	D1	L1	71.0	30.6	41.7	26.7	67.0
27folds	D2	L1	74.1	31.7	42.9	27.5	68.1
27folds	D3	L1	69.4	29.4	39.8	25.5	65.6
27folds	D1	χ^2_1	70.0	29.0	39.9	25.5	65.6
27folds	D2	χ^2_1	73.2	29.0	39.7	25.4	65.8
27folds	D3	χ^2_1	68.4	27.7	38.3	24.5	64.5

Table 6.10: **Results with domain partitioning.** Results with domain partitioning using D1, D2 and D3 distance measures on vector sets.

6.4 Results obtained by Feature Selection

The training and testing set were chosen as in [14] from the '27folds' dataset and have the sizes as in Table 6.11. The training data set has less than 35% of the sequence identity for the aligned subsequences longer than 80 residues, while the testing set has less than 40% of sequence identity per protein pair.

Training Set	27train	303
Testing Set	27test	382

Table 6.11: **Training and testing set for feature selection.** Training and testing set for feature selection are made up of the '27folds' dataset.

After applying SIMBA and RELIEF to the training dataset, the weights in Figure 6.2 are obtained. The greater the value of the weight, the higher the importance of this weight is.

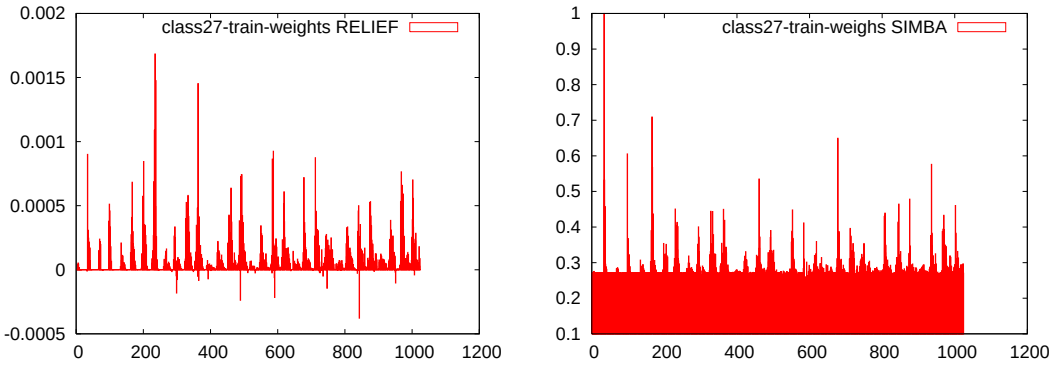

Figure 6.2: **Weight vectors computed by RELIEF and SIMBA.** The weights computed with RELIEF (left) and SIMBA (right) for the dataset '27train'. The x-axis scales the features, the y-axis stands for the computed weight for the feature. Only the features with a weight greater than a certain threshold are selected.

After the weights were computed on the training set, the features computed on the testing set were selected using these weights. For the standard parameter set in Table 6.2 1025 features are computed per protein domain. For the evaluation on the testing dataset, only a certain percentage (10%, 20%, 40%, 60%, 80%) of these 1025 features was used according to the weight vector. The RELIEF algortihm performed worse than SIMBA (see Table 6.12). With the SIMBA algorithm better results by 2.1% could be reached than without using any feature selection. The feature size can also be reduced by using a feature selection algorithm, thus saving memory space.

Feat. Selection	1NN	1T	2T	EM	DCG
without	65.2	30.1	42.2	24.6	63.7
RELIEF (5%)	62.6	29.9	41.8	23.9	62.5
RELIEF (10%)	64.9	30.4	42.8	24.3	63.4
RELIEF (20%)	64.4	31.4	43.4	25.3	64.2
RELIEF (40%)	63.6	31.5	43.6	25.4	64.2
RELIEF (60%)	63.9	30.8	42.8	24.9	63.9
RELIEF (80%)	64.7	30.2	42.7	24.6	63.8
SIMBA (10%)	65.5	33.1	46.3	26.8	66.0
SIMBA (20%)	67.3	33.5	46.7	27.4	66.8
SIMBA (40%)	66.8	32.9	46.0	26.9	66.1
SIMBA (60%)	66.8	32.8	45.7	26.8	66.0
SIMBA (80%)	66.5	32.0	44.7	26.2	65.4

Table 6.12: **Results with feature selection.** Results for feature selection by RELIEF and by SIMBA on the '27test'-dataset.

6.5 Comparison to state of the art methods

In this section, the performance of the alignment methods and methods based on structural fingerprints is evaluated using the PSB. Afterwards, the results are compared with each other and with the results obtained with the proposed method.

6.5.1 Comparison to Alignment Methods

The DALI server [1] provides a standalone application called DaliLite [10]. This program was used to compute the alignments and the pairwise Z-scores of the '27fold' resulting in 235,641 alignments. The results of the evaluation are presented in Table 6.13. DALI performs better by 6.3% than the proposed method. However, the classification time is one week by DALI as opposed to 2 minutes by the proposed method.

Feature	1NN	1T	2T	EM	DCG
SH	78.8	32.4	44.7	28.7	69.3
DALI	85.1	59.1	67.8	45.0	82.8

Table 6.13: **Comparison of results with DALI.** Comparison of the results on the '27folds'-dataset computed by DALI and by the new method.

The dataset used in [18] is the Skolnick clustering set consisting of 33 proteins classified into four families. The validation of the clustering by the CMO method was 98.7% accuracy (1.3% false negatives and 0% false positives). The computation required 528 contact map alignments. The computation time for one alignment ranged between 1 minute and 2 hours.

[1] http://www.ebi.ac.uk/dali/

For comparison, SH features were computed with group integrals on the Skolnick dataset. After the feature extraction, k-means clustering was performed on the dataset and evaluated with the 'silhouette' function provided by Matlab 7.0 Statistics Toolbox. The function *silhouette(X,clust)* plots cluster silhouettes for the n-by-p data matrix X, with clusters defined by $clust$. Rows of X correspond to points, columns correspond to coordinates.

The silhouette vlaue is defined by:

$$\frac{\min(\text{AvgdBetween}(i,k)) - \text{AvgdWithin}(i)}{\max(\text{AvgdWithin}(i), \min(\text{AvgdBetween}(i,k)))},$$

where *AvgdWithin*(i) is the average distance from the i-th point to the other points in its own cluster, and *AvgdBetween*(i,k) is the average distance from the i-th point to points in another cluster k.

The silhouette value for each point is a measure of how similar that point is to points in its own cluster compared to points in other clusters, and ranges from -1 to +1. The optimal value for the silhouette function is 1.

In Figure 6.3 the result of the k-means clustering and the distance matrix based on the SH features is presented. Besides one data sample, all proteins are clearly separated into four clusters. The original functional classes are the same as the classes obtained by k-means clustering, where $k = 4$ is the number of protein families. In the distance matrix, the samples with a small distance have values near zero (which is black in color) and values near one for great distances (which is white in color). The four clusters are clearly separated in the distance matrix as well.

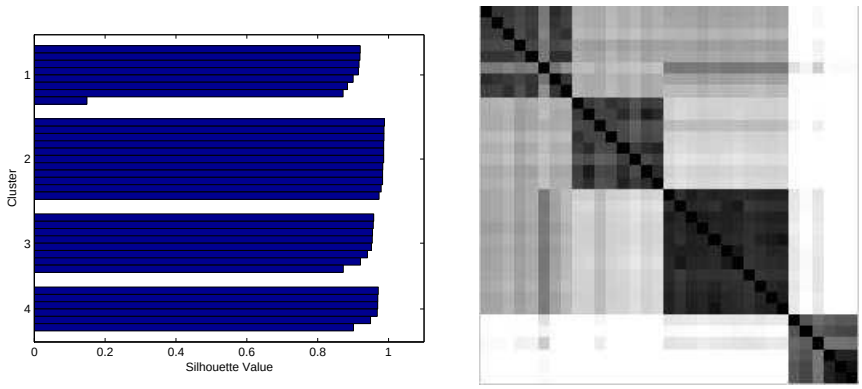

Figure 6.3: **Comparison of CMO with GI.** The results of k-means clustering on the Skolnick dataset. Left the silhouette values as computed by Matlab are presented. The silhouette values can range between -1 and 1, where 1 is the optimal value. They measure the quality of the k-mean clustering. A silhouette value near one indicates, that the samples in one cluster are very close in the feature space and at the same have a great distance to samples in other clusters. Right the distance matrix is visualized. The proteins are listed according to their classes. The black squares along the diagonal indicate, that the features in one cluster have the smallest distance.

The proposed features perform very well on the Skolnick dataset and the classification can be computed in several seconds as opposed to the high time consuming CMO com-

putation. Further, with the CMO approach not more than 300 proteins can be compared, which makes the method not computable on our four data sets.

6.5.2 Comparison to Methods Using Structural Fingerprints

In Table 6.14, the results of the proposed method are compared to the results obtained by PRIDE. The implementation of the PRIDE features was performed as in [4]. However, the bins of the histogram were not combined to contain a certain number of samples. Thus, the PRIDE score was not evaluated by contingency table analysis, but just simply using the $L1$ norm. The results obtained by the new method are better, especially for the '27folds' dataset.

dataset	Feature	**1NN**	**1T**	**2T**	**EM**	**DCG**
all-classes	SH	99.8	87.6	92.5	13.4	97.2
all-classes	PRIDE	99,7	84.8	88.2	13.3	96
all-alpha	SH	97.8	89.3	92.2	37.4	96.0
all-alpha	PRIDE	96.8	80.7	85	34.3	92.7
27folds	SH	78.8	32.4	44.7	28.7	69.3
27folds	PRIDE	70.7	29.4	38.9	25.9	65.1
cath	SH	98.9	72.6	77.7	41.2	91.1
cath	PRIDE	98.8	66.8	73.2	39.1	88.8

Table 6.14: **Comparison with PRIDE features.** Comparison of the results on the '27folds'-dataset computed by PRIDE and by the new method.

In Table 6.15, the results of the proposed method are compared to the results obtained by the Gauss Integrals (Gauss). For the feature computation the program [2] provided by the authors of [26] was used. SH features perform better than Gauss integrals. For the difficult '27folds' dataset, SH outperform Gauss features by 8.1%.

dataset	Feature	**1NN**	**1T**	**2T**	**EM**	**DCG**
all-classes	SH	99.8	87.6	92.5	13.4	97.2
all-classes	Gauss	99.2	73.3	81.2	12.1	93.6
all-alpha	SH	97.8	89.3	92.2	37.4	96.0
all-alpha	Gauss	94.2	63.8	72.9	29.5	87.0
27folds	SH	78.8	32.4	44.7	28.7	69.3
27folds	Gauss	67.6	26.1	35.5	23.2	63.3
cath	SH	98.9	72.6	77.7	41.2	91.1
cath	Gauss	98.4	69.8	76.4	40.2	90.0

Table 6.15: **Comparison with Gauss Integrals.** Comparison of the results on the '27folds'-dataset computed by Gauss Integrals and by the new method.

[2] http://www2.mat.dtu.dk/people/Peter.Roegen/Gauss_Integrals.html

The clustering properties of the Gauss integral features and the SH features are presented in Figure 6.4. There are much more values of the silhouette function below zero for the Gauss integral features (left) than for the features obtained with Spherical Harmonics (right). Thus, the clustering properties of the SH features are better than the clustering properties of the Gauss features on the ’27fold’-dataset.

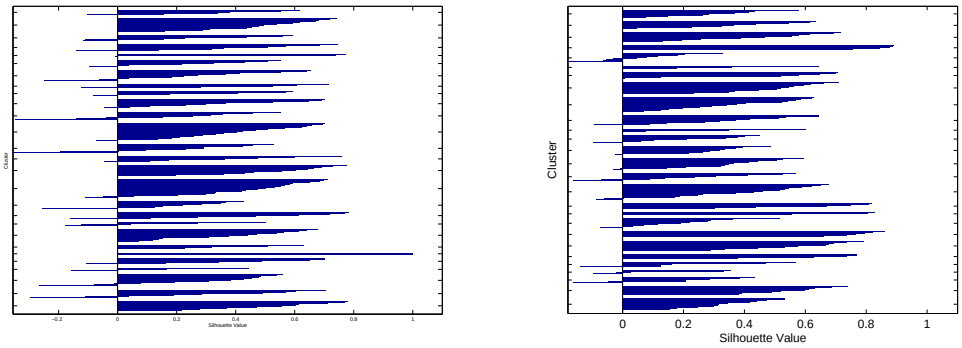

Figure 6.4: **Comparison of clustering properties with Gauss features.** The Matlab function ’silhouette’ applied on the Gauss (left) and the SH-features (right).

6.6 Time requirements

The computations were performed on a Pentium IV Processor with 2,8 MHz and 1024 MB RAM. The time requirements for the new method on the four data sets are presented in Table 6.16. The time requirements increase polynomially depending on the size of the dataset.

dataset	size	Time
27folds	685	2min
all-classes	2,650	40 min
all-alpha	3,680	1h
cath	20,937	2h

Table 6.16: **Time requirements new method.** Time requirements of the new method on different datasets.

The time requirements for the distance matrix with DALI, PRIDE, Gauss and the new method on the ’27folds’ dataset were considered in Table 6.17. Clearly, the new method is 5040 times faster than DALI. The same computation time is needed by PRIDE and by Gauss as by the new method.

Method	Time
New Method	2 min
PRIDE	2min
Gauss	2min
DALI	1 week

Table 6.17: **Comparison of time requirements.** Comparison of the time requirements on the ’27folds’ dataset with different methods.

Chapter 7

Conclusions

In this work, a new method for protein retrieval and classification based on structural features is introduced. This method is evaluated on existing protein classification databases and compared to other methods. When compared to the alignment based methods, the classification performance is worse, however the computation time is lower. The other methods based on structural fingerprints perform worse than the proposed method. Thus, the Group Integrals (GI) should be used for computing the structural fingerprint.

7.1 Summary

GI are a fast and easy to implement method for protein structure retrieval and comparison. The SH expansion gave a 1.5% improvement to the results computed by GI. The D-Wigner expansion did not improve the results, since the description of the structure is too terse.

The GI rely on a kernel function. The δ-based kernel function proposed in this method can be further extended. So, different angles or other information (maybe sequence based) could be incorporated into the kernel.

The local features did not perform as well as global features (8.1% worse), thus domain partitioning rather than local feature description is desired.

For protein classification, different protein databases exist: SCOP, CATH, DALI, but also many others. All the databases have different input and output formats. A major effort was invested in understanding and parsing these file formats. A standardization of the input output results, e.g. XML files could help making the formats more understandable and easier to compare among each other.

Domain partitioning is a difficult task and there is no unique algorithm for computing the domains, since the description depends strongly on the task. Not only domains, but also motives in one domain can be considered, so that sets rather than whole compact units can be matched. We have not yet found a good domain partitioning algorithm. Ideas on how to compare feature vector sets are presented in this work.

For the evaluation of the classification and the clustering, other tools than PSB could be considered. For example, the evaluation could be performed with a Support Vector Machine or statistic based methods. For the clustering, Principle Component Analysis or

simply the k-means algorithm can be performed to better visualize the data.

Feature selection should be trained in a learning environment: For example, Neural Networks could be used to learn the relevant features.

7.2 Outlook

For future work the following topics are of major interest:

- *Kernel functions* should be further explored. This could be done by using a different description of the atom configuration or incorporating e.g. chemical properties into the kernel. The parameter set associated with the existing kernel function was determined in experiments. However, an optimal parameter set could be found by using some learning framework.
- *Domain and motif partitioning* for proteins still lacks easy to compute algorithms. A mathematically sound way to define a protein's domain should be found. This model should incorporate existing knowledge and also the chemical properties relevant for a domain. Further, the decomposition of domains into motives should be studied.
- *Unknown structures* should be classified using the GI in order to test its classification properties. How is GI going to perform on a dataset which is not labeled?
- Rather than only extracting the features and computing the distance, a *decision framework* should be constructed, so that the statistical relevance of structural similarity computed by GI is defined. In this way, when the classification is performed a third level of classification besides accept and reject, namely unsure can be defined.
- The feature extraction as defined by GI can not replace alignment. However, it can be incorporated as a *preprocessing step to alignment*. The low time requirement and the good results obtained on the data sets approve the use of GI for protein structure classification.

Appendix A

Amino Acids

Amino acids are the basic structural units of proteins. They are commonly classified into the following groups based on the chemical and structural properties of their side chains :

- polar
- charged
- hydrophobic

In Figure A.1 - A.3 the amino acids according to their group are represented as a Rasmol image, using structural formulas and with their three and one letter code. The oxygen atoms from the carboxy-terminal are marked red, the nitrogen from the aminoterminal is blue, while sulfide atoms are marked yellow in the Rasmol image.

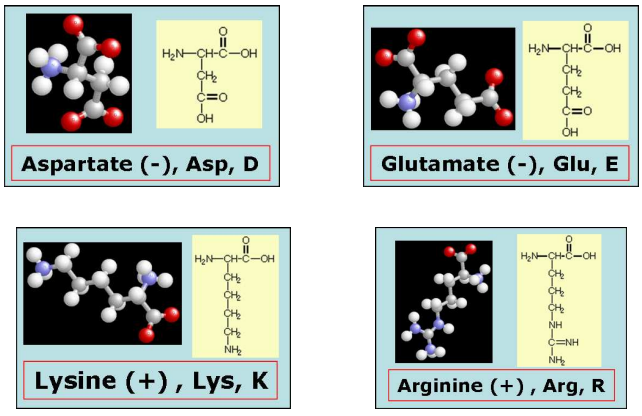

Figure A.1: The charged group.

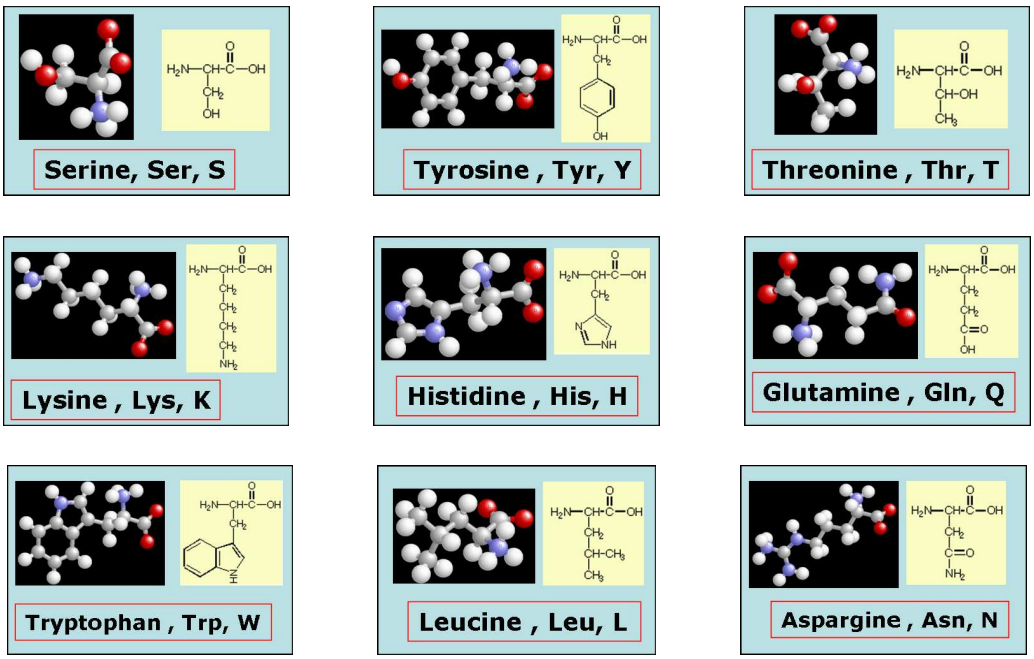

Figure A.2: The polar group.

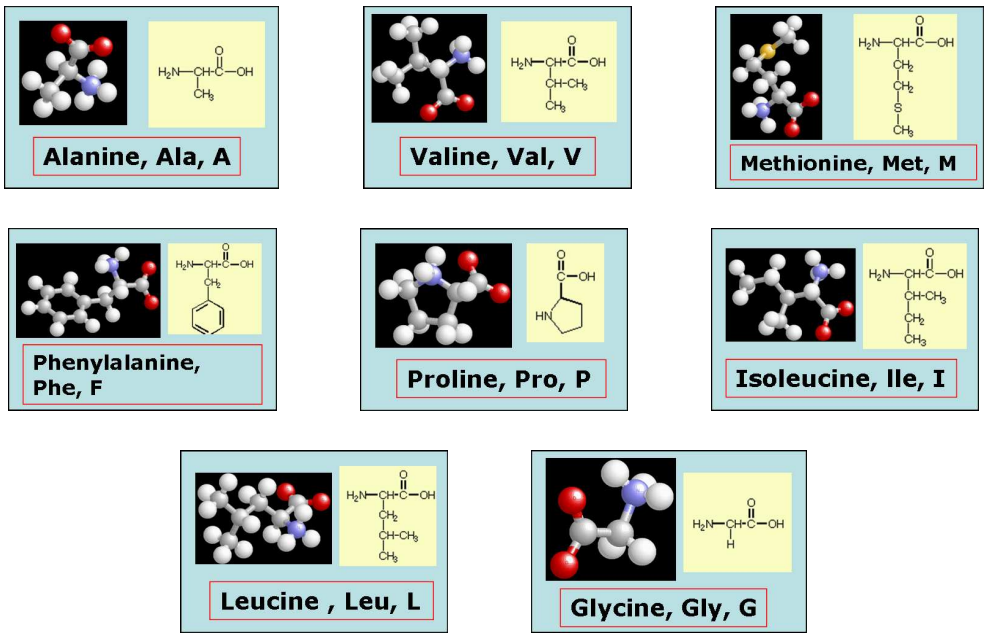

Figure A.3: The hydrophobic group.

Appendix B

PDB statistics

Although the number of structures in the PDB databank is still growing by over 1000 new structures per year (See Figure B.1), the number of folds as defined by SCOP (Figure B.2) and the number of topologies as defined by CATH (Figure B.3) has not changed any more since 2004. Thus it is the basic task of classification to find a fast algorithm that uniquely describes and classifies new folds.

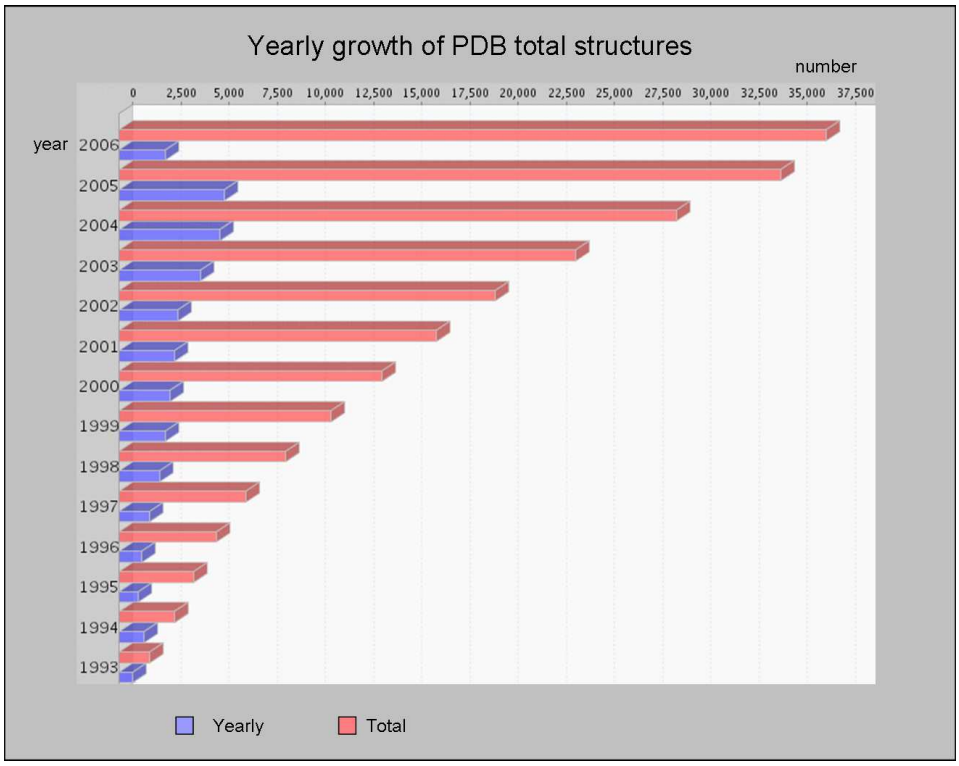

Figure B.1: The growth of molecular structures from 1993-2006.

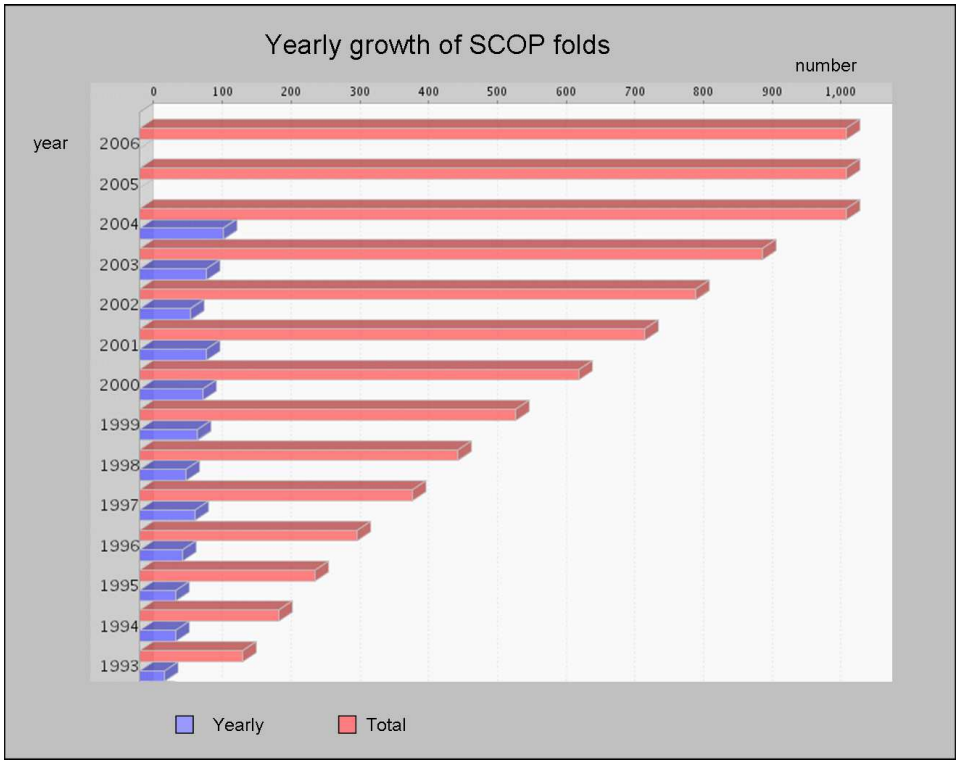

Figure B.2: The growth of unique folds as defined by SCOP from 1993-2006.

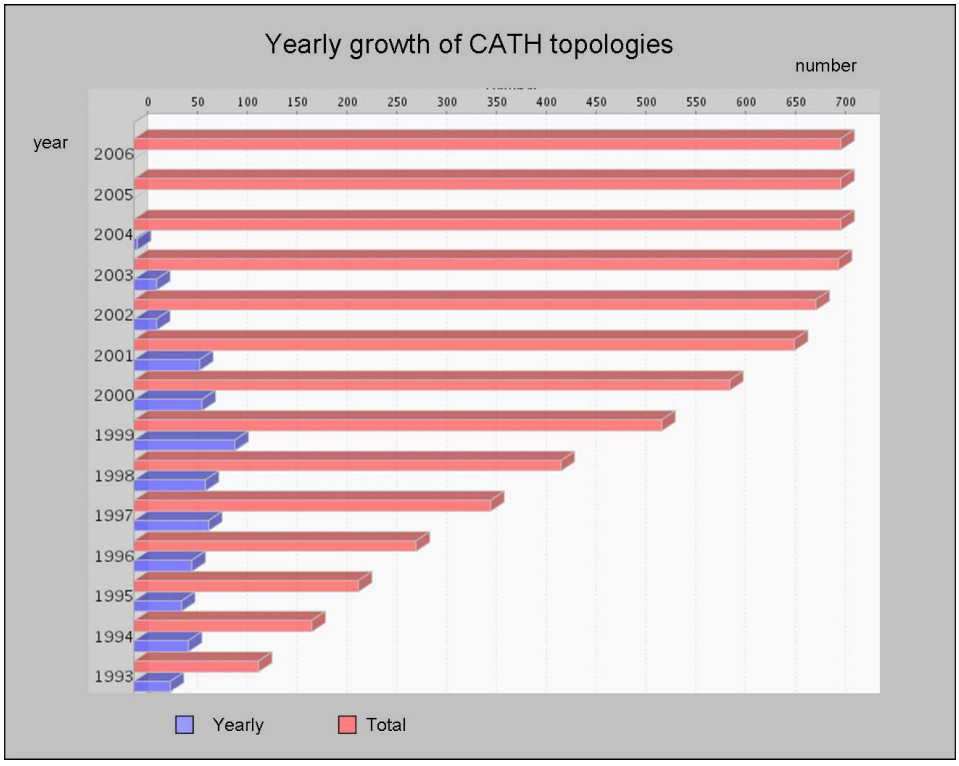

Figure B.3: The growth of the topologies as defined by CATH from 1993-2006.

Appendix C

C++ Classes and Methods

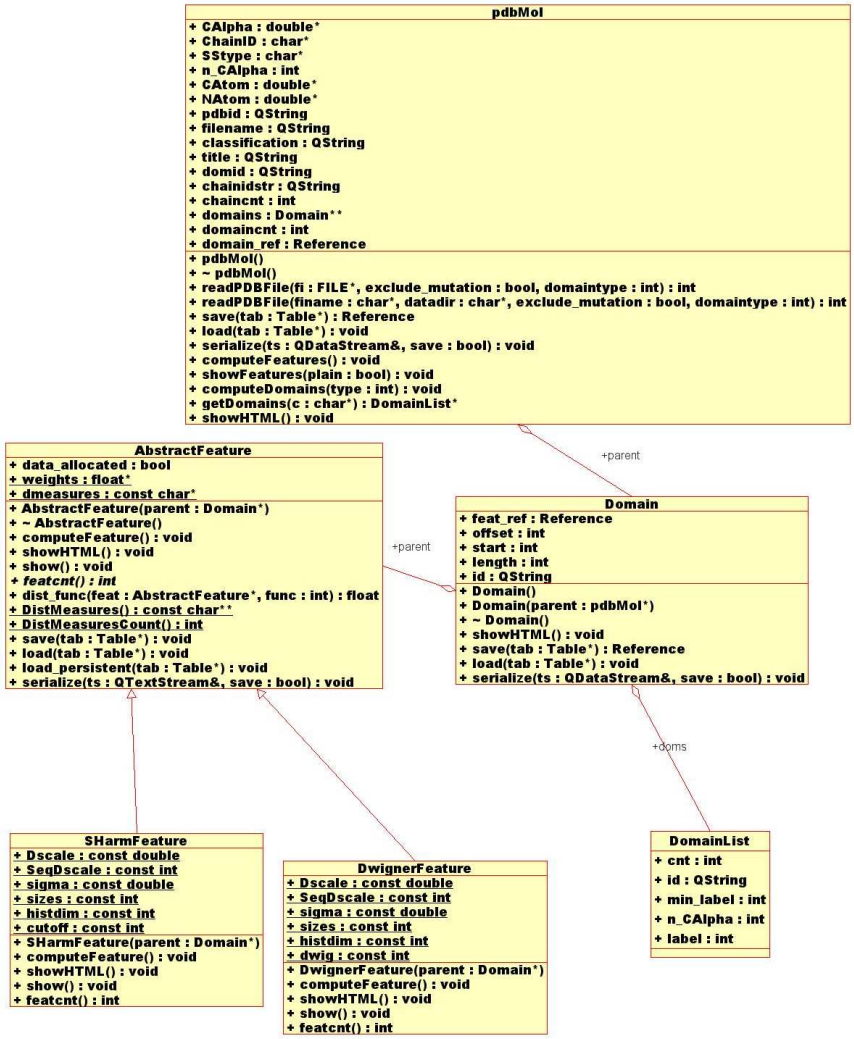

Figure C.1: The Protein Similarity Search by Structural Features (PSF) classes used in the programming framework are depicted.

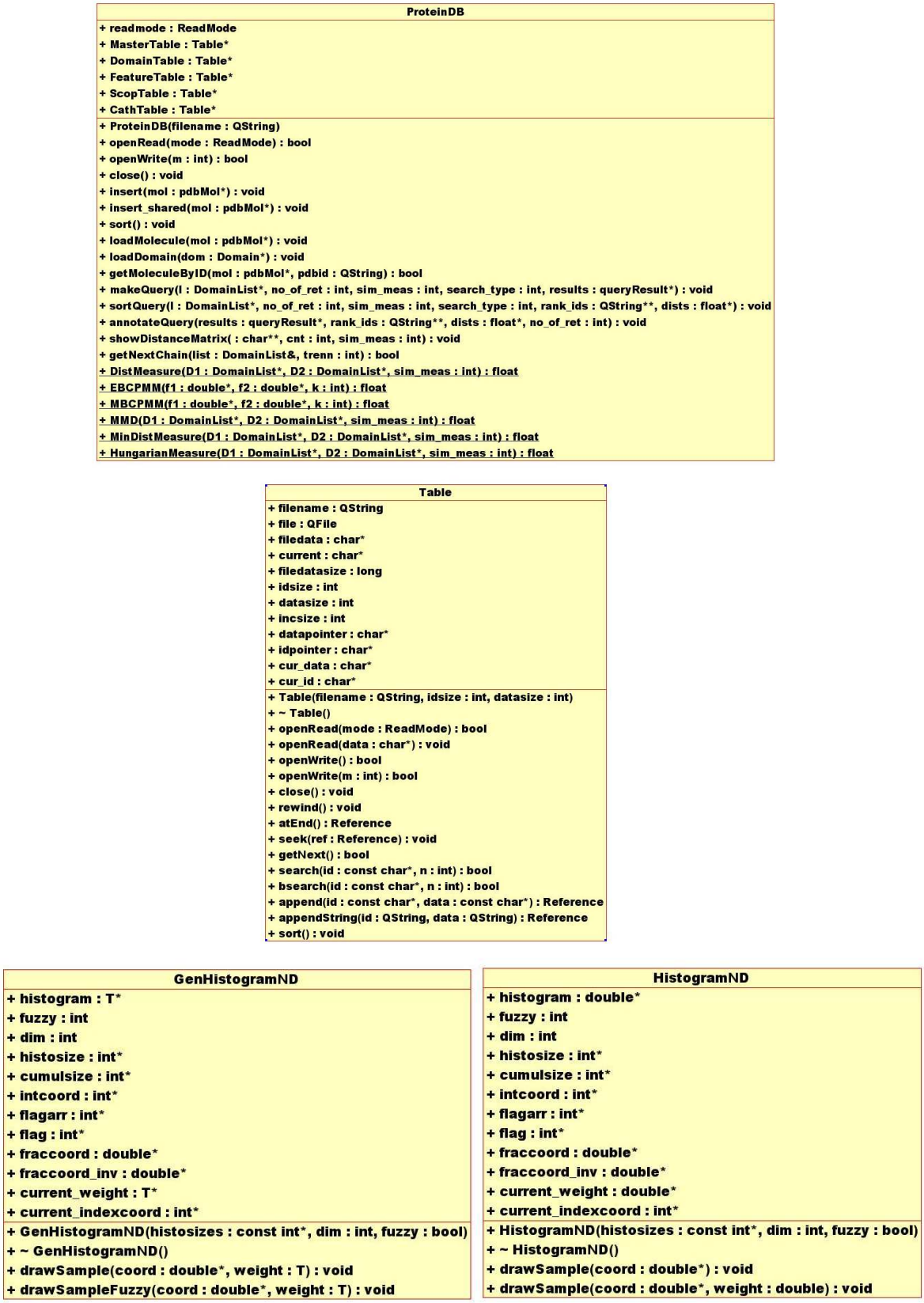

Figure C.2: The Protein Similarity Search by Structural Features (PSF) classes used in the programming framework are depicted.

Bibliography

[1] J. Berg, J. Tymoczko, and L. Stryer. Biochemistry, 5th edition. http://bcs.whfreeman.com/biochem5, 2002.

[2] I. Bronstein, K. Semendjajew, G. Musiol, and H. Mühlig. *Taschenbuch der Mathematik*. Verlag Harri Deutsch, 2000.

[3] H. Burkhardt and S. Siggelkow. *Invariant features in pattern recognition - fundamentals and applications. In Nonlinear Model-Based Image/Video Processing and Analysis*. John Wiley and Sons, 2001.

[4] O. Carugo and S. Pongor. Protein fold similarity estimated by a probabilistic approach based on c_α-c_α distance comparison. *J. Mol. Biol.*, 315:887–898, 2002.

[5] R. Gilad-Bachrach, A. Navot, and N. Tishby. Margin based feature selection - theory and algorithms. *Proc. 21st International Conference on Machine Learning*, 2004.

[6] D. Goldman, S. Istrail, and C. Papadimitriou. Algorithmic aspects of protein structure similarity. *Proceedings of the 40th IEEE Symposium on Foundations of Computer Science*, pages 512–522, 1999.

[7] I. Guyon and A. Elisseeff. An introduction to variable and feature selection. *Journal of Machine Learning Research*, (4):1157–1182, 2003.

[8] B. Haasdonk, A. Halawani, and H. Burkhardt. Adjustable invariant features by partial haar-integration. In *Proceedings of the 17th International Conference on Pattern Recognition (ICPR)*, volume 2, pages 769–774, 2004.

[9] B. Haasdonk, A. Vossen, and H. Burkhardt. Invariance in kernel methods by haar-integration kernels. In *Proceedings of the 14th Scandinavian Conference on Image Analysis*, pages 841–851, 2005.

[10] L. Holm and J. Park. Dalilite workbench for protein structure comparison. *Bioinformatics*, 16(6):566–567, 2000.

[11] L. Holm and C. Sander. Mapping the protein universe. *Science*, 273:595–602, 1996.

[12] L. Holm and C. Sander. Dictonary of recurrent domains in protein structures. *Proteins*, 33:88–96, 1998.

[13] L. Holm and C. Sander. Touring protein fold space with dali/fssp. *Nuc. Acids*, 26:316–319, 1998.

[14] C.-D. Huang, C.-T. Lin, and N. R. Pal. Hierarchical learning architecture with automatic feature selection for multiclass protein fold classification. *IEEE Trans. on Nanobioscience*, 2(4):221–232, 2003.

[15] W. Kabsch. A solution for the best rotation to relate two sets of vectors. *Acta Cryst.*, A(32):922–923, 1976.

[16] K. Kira and L. Rendell. A practical approach to feature selection. *Proc. 9th International Workshop on Machine Learning*, pages 249–256, 1992.

[17] H. W. Kuhn. The hungarian method for the assignment problem. *Naval Res. Logist. Quart*, pages 83–98, 1955.

[18] G. Lancia, R. Carr, B. Walenz, and S. Istrail. Optimal pdb structure alignments: a branch-and-cut algorithm for the maximum contact map overlap problem. *RECOMB*, pages 193–202, 2001.

[19] W. Miller. Topics in harmonic analysis with application to radar and sonar. In *IMA Volumes in Mathematics and its Applications*, 1991.

[20] A. Murzin, S.E.Brenner, T.Hubbard, and C.Chothia. Scop: a structural classification of proteins database for the investigation of sequences and structures. *J. Mol. Biol.*, 247:536–540, 1995.

[21] C. Orengo, J. S. J. D. S. M. Michie, A.D., and J. Thornton. Cath- a hierarchic classification of protein domain structures. *Structure*, 5(8):1093–1108, 1997.

[22] R. Osada, T. Funkhouser, and B. C. und David Dobkin. Matching 3d models with shape distribution. In *Proceedings Shape Modeling International*, 2001.

[23] M. Reisert. Using irreducible group representation for invariant 3d shape description. *DAGM Berlin*, 2006.

[24] M. Reisert and H. Burkhardt. Second order 3d shape features: An exhaustive study. *accepted for publication in Computers and Graphics, Special Issue on Shape Reasoning and Understanding (publication date: April 2006)*, 30(2), 2006.

[25] P. Rogen and H. Bohr. A new family of global protein shape descriptors. *Mathematical Biosciences*, 182(2):167–181, 2003.

[26] P. Rogen and B. Fain. Automatic classification of protein structure by using gauss integrals. *Proc.Nat.Sci USA*, 100(1):119–124, 2003.

[27] O. Ronneberger, H. Burkhardt, and E. Schultz. General-purpose Object Recognition in 3D Volume Data Sets using Gray-Scale Invariants. In *Proceedings of the International Conference on Pattern Recognition*, Quebec, Canada, Sept. 2002.

[28] O. Ronneberger, J. Fehr, and H. Burkhardt. Voxel-wise gray scale invariants for simultaneous segmentation and classification. In *Proceedings of the 27th DAGM Symposium, Vienna, Austria*, 2005.

[29] M. Schael. Invariant texture classification using group averaging with relational kernel functions. In *Texture 2002 the 2nd international workshop on texture analysis and synthesis*, pages 129–134, 2002.

[30] L. Setia, J. Ick, and H. Burkhardt. Svm-based relevance feedback in image retrieval using invariant feature histograms. In *Proc. of the IAPR Workshop on Machine Vision Applications*, pages 542–545, 2005.

[31] P. Shilane, P. Min, M. Kazhdan, and T. Funkhouser. The princeton shape benchmark. In *Shape Modeling International, Genova, Italy*, 2004.

[32] S. Siggelkow, M. Schael, and H. Burkhardt. Simba - search images by appearance. In *Proceedings of the 23rd DAGM Symposium*, pages 9–16, 2001.

[33] W. Taylor, A. May, N. Brown, and A. Aszodi. Protein structure: geometry, topology and classification. *Reports on Progress in Physics*, 64:517–590, 2001.

[34] M. Temerinac. Invariant feature extraction for protein fold recognition. Student Project temerina@informatik.uni-freiburg.de, May 2005.

List of Figures

List of Tables