
New Methods for the Prediction and Classification of Protein Domains

Jan Erik Gewehr



München 2007

New Methods for the Prediction and Classification of Protein Domains

Jan Erik Gewehr

Dissertation
an der Fakultät für Mathematik und Informatik
der Ludwig-Maximilians-Universität
München

vorgelegt von
Jan Erik Gewehr
aus Lübeck

München, den 01.10.2007

Erstgutachter: Prof. Dr. Ralf Zimmer
Zweitgutachter: Prof. Dr. Dmitrij Frishman
Tag der mündlichen Prüfung: 19.12.2007

Contents

1	Motivation and Overview	1
1.1	The Benefit of Protein Structure Prediction	1
1.2	Thesis Outline	3
2	Introduction to Protein Structure Prediction	7
2.1	Protein Structures and Related Databases	7
2.1.1	From Primary to Tertiary Structure	7
2.1.2	Structure-Related Databases	8
2.2	Assignment of Protein Domains	10
2.2.1	An Old Problem	10
2.2.2	Comparison of Domain Assignments	11
2.3	Structure Prediction Categories	13
2.3.1	Comparative Modeling	13
2.3.2	Fold Recognition	14
2.3.3	Ab Initio	15
2.4	Community-Wide Efforts	15
2.4.1	Community-Wide Experiments	15
2.4.2	Structural Genomics and Structure Prediction	16
2.5	Alignment Methods	16
2.5.1	Sequence Alignment	16
2.5.2	Alignment Methods used in this Work	19
3	Selection of Fold Classes based on Secondary Structure Elements	23
3.1	Introduction	23
3.2	Material	25
3.2.1	Training and Test Data	25
3.2.2	Quoted Methods	27
3.3	Preselection of Fold Classes	28
3.3.1	Secondary Structure Element Alignment (SSEA)	28
3.3.2	Selection Strategies based on SSEA	29
3.4	Refinement with Profile-Profile Alignment	31
3.5	Results	31
3.5.1	Preselection Performance on ASTRAL25	32

3.5.2	Fold Recognition Accuracy	33
3.5.3	Speed-Up Evaluation	34
3.6	Discussion	34
4	AutoSCOP: Unique Mapping of Patterns to SCOP	37
4.1	Introduction	38
4.2	Material	40
4.2.1	InterPro and its Member Databases	40
4.2.2	ASTRAL Asteroids and Family HMMs	41
4.2.3	Training Data	41
4.2.4	Test Data	41
4.3	The AutoSCOP Approach	42
4.3.1	Motivation	42
4.3.2	Unique Patterns	42
4.3.3	Extension: Pattern Combinations	44
4.3.4	AutoSCOP*: Inclusion of Further Data Sources	45
4.4	Results	45
4.4.1	Mapping of Training Domains	45
4.4.2	Prediction of SCOP 1.67 Domains	47
4.4.3	Comparison of InterPro Entries and AutoSCOP Mappings	51
4.4.4	Fold Prediction of CAFASP4 Targets	52
4.4.5	Performance in the Sequence Twilight Zone	54
4.4.6	Using AutoSCOP* as a Filter	54
4.5	AutoPSI DB	55
4.5.1	AutoPSI Database Content and Methods	55
4.5.2	Towards Large-Scale Protein Domain Prediction	59
4.6	Conclusion	61
5	SSEP-Domain: Template-Based Protein Domain Prediction	65
5.1	Introduction	66
5.2	The Domain Prediction Pipeline	67
5.2.1	Preliminaries	69
5.2.2	Step 1: Finding Potential Domain Boundaries	69
5.2.3	Step 2: Scoring of Domain Regions	72
5.2.4	Step 3: Combining Multiple Domain Regions	75
5.3	Results	76
5.3.1	CAFASP 4 and CASP 6 Results	76
5.3.2	Current Version under CAFASP Conditions	77
5.3.3	Evaluation of InterPro and Combination with AutoSCOP	86
5.3.4	SSEP-Align: An Extension towards Structure Prediction	87
5.3.5	Other Possible Extensions	89
5.4	Two Years After: CASP 7 and its Lessons	90
5.4.1	Analysis of CASP 7 Results	90

5.4.2	Using Alternative Definitions for SCOP Domains	91
5.4.3	Discontinuous Domains	94
5.5	Independent Applications and Evaluations	95
5.6	Discussion	96
6	Environment-Specific Alignment Computation and Scoring	99
6.1	The QUASAR Framework	100
6.1.1	Methods	100
6.1.2	Use Cases	102
6.2	Optimized Score Combinations	103
6.2.1	Preliminaries	103
6.2.2	Generation of Linear Combinations	105
6.2.3	Results	106
6.3	Optimized Matrices for Alignment Ranking	108
6.3.1	Comparison Matrices	108
6.3.2	Range-Adaptive Genetic Algorithm	109
6.3.3	Results	111
6.4	Optimized Profile-Profile Alignments	112
6.4.1	Training and Test Data	113
6.4.2	Modified Genetic Algorithm	113
6.4.3	Results	114
6.5	Discussion	115
7	Additional Tools	117
7.1	Vorolign: Structural Alignment and SCOP Classification Prediction	117
7.2	Representation of Protein Information in ProML	119
7.3	BioWeka: Extending the Weka Framework for Bioinformatics	121
7.3.1	Motivation	121
7.3.2	The Weka Framework	122
7.3.3	The BioWeka Library	124
7.3.4	Example Applications	126
7.3.5	Discussion	128
8	Concluding Remarks	129
	Acknowledgements	146

List of Figures

1.1	Introduction: Thesis Overview	6
2.1	Background: Homology-based Protein Structure Prediction	14
3.1	Preselection: Overview	26
3.2	Preselection: Preselection Performance	30
3.3	Preselection: Fold Recognition Accuracy on Three Evaluation Sets.	35
4.1	AutoSCOP: Pattern-Class Graph.	43
4.2	AutoSCOP: Pattern Combinations.	43
4.3	AutoSCOP: Screenshot of the AutoPSI Database.	56
4.4	AutoSCOP: Annotation Process for the AutoPSI Database.	57
4.5	AutoSCOP Example: 1a0p.	60
4.6	AutoSCOP Example: 1jwlc.	60
4.7	AutoSCOP Example: 1a79a.	60
5.1	SSEP-Domain: Overview	68
5.2	SSEP-Domain: Domain Length Histogram	70
5.3	SSEP-Domain: Alignment Score Histogram	74
5.4	SSEP-Domain: Overlap Score Example	83
5.5	SSEP-Domain: Results Plot	85
6.1	QUASAR: Overview	101
6.2	Comparison of GA Matrices with Well-Known Matrices.	112
7.1	Vorolign: Similarity Computation	118
7.2	BioWeka: Overview	124

List of Tables

4.1	AutoSCOP: Quantitative Analysis of Unique InterPro Patterns.	45
4.2	AutoSCOP: Coverage Analysis on Training Data.	46
4.3	AutoSCOP: Influence of InterPro databases.	46
4.4	AutoSCOP: Sensitivity and Specificity	49
4.5	AutoSCOP: Reduced Training Data	51
4.6	AutoSCOP: Non-Trivial Targets	53
5.1	SSEP-Domain: Correctly Predicted Targets	80
5.2	SSEP-Domain: Specificity	81
5.3	SSEP-Domain: Average Overlap Score	84
5.4	SSEP-Domain: SSEP-Align Results	88
5.5	SSEP-Domain: Comparison with SSEP-Domain* on CAFASP 4 and CASP 7	94
5.6	SSEP-Domain: Algorithmic Ingredients	96
6.1	Evaluated Scoring Schemes and Corresponding Parameter Settings	107
6.2	Results of Optimized PPA Parameters	114

Summary

Proteins play a central role in organisms as they perform many important tasks in their cells. Accordingly, the better we understand how proteins are built, the better we can deal with many common diseases. In particular, information on structural properties of proteins can give insight into the way they work and how mutations, for instance, may affect their operability. Such knowledge can therefore help and influence modern medicine and drug development.

This work is situated in the field of protein structure prediction. Here, the aim is to determine a three-dimensional structure from a protein's amino acid sequence. Depending on their quality, the resulting structure models can be used for a variety of purposes. At the moment there are millions of known protein sequences but only tens of thousands of known protein structures available. As it cannot be expected that it will be possible to experimentally determine a structure for each protein in the near future, protein structure prediction is an important task of current bioinformatics research.

In particular, so-called *template-based* modeling can result in very good predicted structures. Here, given a *target* sequence with unknown structure, a simple modeling approach (1) searches for similar sequences with known structures (so-called *templates*), (2) computes mappings from the target sequence to the template sequences (so-called *alignments*), (3) takes the atom coordinates from the mapped parts of the templates for the structure model, and (4) refines the model.

Protein structures can be classified into hierarchies. Prominent examples for such hierarchies are the SCOP and CATH databases. The units used for such classifications are so-called *protein domains*, which are parts of a protein that are able to (depending on the definition) fold independently, for instance, or to fulfill an independent function. In particular, the domain content of a protein, i.e. the contained domains and their positions, is important for the final function of a protein as well as for the biological processes it is used in and the molecules it interacts with.

The focus of this work is on methods for the prediction of protein domains and their structural classifications. Protein structure prediction can benefit from such predictions, as, once a structural classification is known, it is easier to find suitable templates and the probability is higher to obtain a good model. Furthermore, in many cases, additional properties can be derived from the knowledge of a structural classification, such as a potential function. For the experimental solution of structures it can also be of interest to know the contained domains, as it may be easier to solve the domains individually.

The methods described in the following are a new approach for quick selection of potential fold classes (Preselection and Refinement, [Gewehr et al., 2004]), a new method for fast and specific prediction of structural classifications using known sequence patterns (AutoSCOP, [Gewehr et al., 2007a]) and a corresponding database of predicted classifications (AutoPSI, [Birzele et al., 2008]) which contains more than two million sequences, a new and template-based protein domain prediction method (the SSEP-Domain approach, [Gewehr and Zimmer, 2006]), as well as a software (QUASAR, [Birzele et al., 2005]) and a new method for optimized alignment ranking and computation with respect to structural quality. In addition, we describe a new structural alignment method (Vorolign, [Birzele et al., 2007]), an XML schema for the representation of knowledge on protein structures (ProML), and an extension library for the well-known Weka machine learning framework, which contains bioinformatics-specific methods and data formats (BioWeka, [Gewehr et al., 2007b]).

These approaches provide important contributions for the protein structure prediction process: A new sequence can be split into domains using SSEP-Domain; these domains can be classified using Preselection and AutoSCOP. If the sequence is part of the public databases, structural classifications may already be available via the AutoPSI database. Having aligned a target to templates, with QUASAR and the corresponding optimization methods, good alignments can be either selected or newly computed. If the structure is known, structural alignment and a search for similar structures can be done with Vorolign. Gained information can be stored using ProML.

Protein structure prediction will remain an essential task for many years. Improvements of the prediction process allow to produce more and better structure models, which are steps towards the overall aim of finding a structure for each protein sequence. Our evaluations show that the proposed methods and tools can be used for this purpose and also provide a good basis for future research in this direction.

Zusammenfassung

Die Funktionsfähigkeit der in unseren Zellen enthaltenen Proteine spielt für unsere Gesundheit eine wesentliche Rolle. Dementsprechend besteht ein großes Interesse daran, die Mechanismen zu verstehen, nach denen Proteine aufgebaut sind und ihre Funktionen erfüllen. Insbesondere die Kenntnis der dreidimensionalen Struktur eines Proteins und der Effekt eventueller Mutationen auf diese Struktur können Hinweise und Ansatzpunkte für die Medikamentenentwicklung liefern.

Diese Arbeit ist im Bereich der Proteinstrukturvorhersage angesiedelt. Darunter versteht man die Aufgabe, für Proteinsequenzen mit unbekannter Struktur möglichst gute Modellstrukturen zu erzeugen, die dann entsprechend ihrer Qualität zu unterschiedlichen Zwecken herangezogen werden können. Die Notwendigkeit, Proteinstrukturen vorherzusagen anstatt sie experimentell aufzulösen, entsteht aus der Tatsache, daß die Anzahl der bekannten Proteinsequenzen um Größenordnungen höher ist (in den aktuellen Datenbanken finden sich Millionen) als die Anzahl der bekannten Strukturen (im Moment weniger als 50.000), und daß die experimentellen Prozesse zur Auflösung einer Struktur sowohl relativ langwierig als auch kostspielig sind. Es ist nicht zu erwarten, daß es in nächster Zukunft möglich sein wird, eine Struktur für jede Proteinsequenz zu finden, ohne in großem Maße Vorhersagemethoden einzusetzen.

Insbesondere die sogenannte *template-basierte* Modellierung liefert qualitativ hochwertige Strukturmodelle. Methoden, die in diese Kategorie fallen, machen es sich zunutze, daß ein ähnliches Protein mit bekannter Struktur existiert. Gegeben ein *Target*, ein Protein mit bekannter Sequenz aber unbekannter Struktur, läßt sich ein vereinfachter Ablauf einer solchen Modellierung wie folgt darstellen: (1) man sucht nach geeigneten Kandidaten (sog. *Templaten*), (2) man erstellt Abbildungen des Targets auf die Template (sog. *Alignments*), (3) man überträgt die entsprechenden Koordinaten aus der bekannten Struktur und (4) man verfeinert das Modell.

Proteinstrukturen lassen sich klassifizieren und in Hierarchien einteilen. Beispiele für solche Klassifizierungen sind die Datenbanken SCOP und CATH. Die Einheiten, die solchen Hierarchien zugrunde liegen, sind sogenannte *Proteindomänen*, Teile eines Proteins, die, je nach Definition, z.B. unabhängig eine Struktur ausbilden können oder eine eigene Funktion erfüllen. Insbesondere ist die Domänenstruktur, d.h. die enthaltenen Domänen und ihre Positionen, wesentlich für die finale Funktion eines Protein, sowie für die biologischen Prozesse, an denen es teilhat, und die Moleküle, mit denen es interagiert.

Der Fokus dieser Arbeit liegt auf Methoden zur Vorhersage von Proteindomänen und ihrer strukturellen Klassen. Der Nutzen für die Proteinstrukturvorhersage liegt darin, daß man, wenn eine strukturelle Klassifikation mit großer Konfidenz vorhergesagt werden kann, leichter gute Template finden kann und damit die Wahrscheinlichkeit erhöht, am Ende ein gutes Modell zu erhalten. Darüberhinaus lassen sich in vielen Fällen weitere Eigenschaften von hinreichend genauen strukturellen Klassifikationen ableiten, wie z.B. eine mögliche Funktion. Für die experimentelle Bestimmung von Strukturen ist es ebenfalls von Vorteil, die vorhandenen Domänen in einem Zielprotein zu kennen, da diese unter Umständen einfacher separat aufzulösen sind.

Die hier vorgestellten Methoden umfassen einen Ansatz zur schnellen Vorselektion potentieller Strukturklassen (Preselection and Refinement, [Gewehr et al., 2004]), einen neuen Ansatz zur schnellen und hochspezifischen Vorhersage von Strukturklassen auf der Basis bekannter Sequenzmotive (AutoSCOP, [Gewehr et al., 2007a]) und eine dazugehörige Datenbank von Vorhersagen, die Millionen bekannter Sequenzen umfaßt (AutoPSI DB), eine neue und schnelle templat-basierte Methode zur Proteindomänenvorhersage (SSEP-Domain, [Gewehr and Zimmer, 2006]), und eine Software (QUASAR, [Birzele et al., 2005]) sowie eine Methode zur optimierten Bewertung und Erstellung von Alignments in Hinblick auf die strukturelle Qualität der resultierenden Modelle. Darüberhinaus werden zusätzliche, neue Werkzeuge für die Forschung im Bereich der Proteinstrukturen eingeführt: (1) eine Methode zum Alignment von Proteinstrukturen (Vorolign, [Birzele et al., 2007]), (2) ein XML Schema zur Speicherung und Bereitstellung von Wissen über Proteine und Proteinmengen (ProML), und (3) eine neue JAVA-Bibliothek, die das bekannte Weka System für maschinelles Lernen um grundlegende Bioinformatikmethoden und -datenformate erweitert (BioWeka, [Gewehr et al., 2007b]).

Diese Methoden liefern wichtige Beiträge für den Proteinstrukturvorhersageprozeß: Eine neue Sequenz kann mittels SSEP-Domain in Domänen zerlegt werden. Diese können mit Preselection und AutoSCOP in die bekannten strukturellen Klassifikationen eingeordnet werden. Handelt es sich bei dem Target um eine bereits bekannte Sequenz aus den öffentlichen Datenbanken, besteht zudem die Möglichkeit, Regionen mit potentiellen strukturellen Klassifikation direkt über die AutoPSI-Datenbank zu erhalten. Nachdem man Alignments gegen geeignete Template erstellt hat, lassen sich QUASAR und die darauf aufbauenden Optimierungsmethoden anwenden, um gute Alignments zu erkennen oder neu zu erstellen. Ist eine Struktur bekannt, kann Vorolign zur Ähnlichkeitssuche zu den strukturell eingeordneten Strukturen eingesetzt werden. Informationen über bekannte Proteine können mittels ProML gespeichert werden, um sie dann mit BioWeka weiter zu untersuchen.

Die Vorhersage von Proteinstrukturen wird noch für viele Jahre ein wesentlicher Bestandteil der Bioinformatik bleiben. Schrittweise Verbesserungen des Vorhersageprozesses erlauben es, mehr und bessere Modellstrukturen zu erstellen, und damit dem Ziel, eine gute Struktur für jedes bekannte Protein zu finden, ein wenig näher zu kommen. Unsere Auswertungen zeigen, daß die vorgestellten Methoden und Werkzeuge nutzbringend einsetzbar sind und damit eine gute Basis für weitere Forschung in diesem Bereich darstellen.

Chapter 1

Motivation and Overview

1.1 The Benefit of Protein Structure Prediction

Proteins play a central role in organisms as they perform many important tasks in their cells. Accordingly, the better we understand how proteins are built, the better we can deal with many common diseases. In particular, information on structural properties of proteins can give insight into the way they work and how mutations, for instance, may affect their operability. Such knowledge can therefore help and influence modern medicine and drug development.

Protein Structure Prediction

Public databases contain millions of protein sequences (currently the UniProt database alone contains more than 4.5 million entries), but the number of publicly available protein structures is smaller by about two orders of magnitude (the PDB database of protein structure currently contains about 45,000 structures). The protein structure prediction community aims at developing methods for the prediction of the structure of a protein from its sequence, with the long term goal to provide a structure for each available protein or gene. Good examples for the effort spent in structure prediction and determination are the CASP experiments (Critical Assessment of techniques of protein Structure Prediction, [Moult, 2005]) as well as structural genomics (see chapter 2). Another example are databases like the SWISS-MODEL repository [Kopp and Schwede, 2006] and MODBASE [Pieper et al., 2006] which currently contain up to 4.3 million predicted models based on automated prediction pipelines.

What Can be Achieved?

In a Science publication of 2001, Baker and Sali describe the potential of protein structure models for different applications depending on the accuracy of the model when compared to the true structure [Baker and Sali, 2001]. This illustrates the many benefits of protein structure prediction, even when the model accuracy is not high enough for drug design pur-

poses. Structure models with an accuracy of 1.0 Å RMSD (root mean square deviation) for the main chain atoms, which is in the range of the deviation of native structures themselves (low-resolution X-ray or medium-resolution NMR), can be used for the study of catalytic mechanisms or the design and improvement of ligands. Such structures can be modeled if proteins with identical (or almost identical) sequences and already known structures exist. With decreasing accuracy, i.e. with an increasing RMSD and a decreasing coverage of the main chain, the tasks that are still possible range from docking of macromolecules to the refinement of NMR structures. Even with low accuracy but roughly correct structures, for certain regions of a protein, it may still be possible to assign functional sites or find functional relationships between proteins.

With the ever increasing number of available structures, the possibility for a target protein with yet unresolved experimental structure to have a similar protein with known structure will increase. For decades, researchers have been working on methods for modeling the structure of the target in such cases, using so-called *template-based modeling*, where template means a protein with known structure that is similar to the target. However, the quality of many predicted structures is still not sufficient for many purposes in drug development. One reason for this is the complexity of the structure prediction task, since it involves many steps from the target sequence to the final model, each of which is an interesting problem in its own right. The problems to be solved in template-based protein structure prediction and modeling include the search for suitable templates, the recognition of the overall topology of the protein, the assignment of structurally, evolutionary or functionally independent parts of the protein, the alignment of its sequence with the sequences of the available templates, the assignment of coordinates to the atoms of the protein (often based on an alignment), the modeling of flexible regions or regions where no similar, known structure was found, and the refinement of the resulting models.

Protein Domains and Structural Classifications

”Complexity in biology has evolved through modification and recombination of existing building blocks instead of invention from scratch. In the protein world these building blocks have been termed domains and the identification and characterization of new domains and domain families is a major goal of protein science” [Heger and Holm, 2003]. In a more structure-oriented view, protein domains are usually defined either as recurrent evolutionary units, as independent folding units or as globular, more or less independent parts of a protein, and further definitions can be found in the literature (see chapter 2). Nonetheless, based on protein domains, hierarchical structure-based classifications like SCOP [Murzin et al., 1995] or CATH [Orengo et al., 1997] attempt to introduce order in the universe of protein structures by classifying them into tree-like hierarchies, so-called *structural classifications*. If the classification of a target protein domain is known, finding potential templates for modeling the structure can be done by searching for structures with the same classification; depending on the level of the known classification (the finer the better), it may also be possible to deduce further properties of the target.

1.2 Thesis Outline

The focus of this thesis is on methods for protein domain recognition and the prediction of their structural classifications. By concentrating on these tasks, the main objective is to enable researchers to deduce and use information about structural properties and structural neighborhoods as defined by hierarchies of protein domains like SCOP. In particular, the following chapters contain:

- **Chapter 2: Background knowledge.** In this chapter, we introduce the basics of protein structure prediction and its subtasks; further, we briefly describe databases and alignment methods which are used frequently in this work.
- **Chapter 3: Speeding up alignment-based protein fold recognition.** As more sophisticated alignment methods for fold recognition can become computationally expensive when large numbers of alignments have to be computed, we propose a method to speed-up the prediction process [Gewehr et al., 2004] that is based on a fast scan for potential fold classes based on a simple measure for potential topological similarity. Our approach yields a speed-up of about one order of magnitude while keeping a comparable fold recognition accuracy when combined with profile-profile alignment, a well-known method for fold recognition. This so-called *preselection* is used in slightly modified variants in both the Vorolign and the SSEP-Domain method (chapters 7 and 5, respectively).
- **Chapter 4: Fast and reliable prediction of structural classifications.** A new approach to the prediction of structural classifications of protein domains, which we called AutoSCOP [Gewehr et al., 2007a], deals with the prediction of a protein classification on different SCOP levels based on sequence patterns. AutoSCOP focuses on high specificity, such that it can be used either as a reliable standalone predictor or as an additional filter in combination with other prediction methods, and we indeed observe an improvement in accuracy when combined with individual alignment methods in our evaluation. Given the sequence patterns on an amino acid sequence, our approach can assign SCOP predictions in a matter of seconds.

As the necessary input data is available in a precomputed form for many of the available protein sequences, our approach allows for large-scale prediction of potential SCOP classifications. In joint work with Fabian Birzele, we built the AutoPSI database [Birzele et al., 2008], a database of SCOP predictions which contains consensus predictions of AutoSCOP and Vorolign based on structural alignments and additional information for many newly found protein structures as well as AutoSCOP predictions for millions of further amino acid sequences.

With respect to protein domain prediction, pattern locations and especially annotated structural classifications can further give hints on the existence of domains on target sequences, though the boundaries are often not very exact. Therefore, using AutoSCOP, with the annotation of structural classifications, we can also quickly derive potential SCOP domain occurrences on these sequences.

- **Chapter 5: Fast, homology-based protein domain prediction.** The former prediction approaches mainly work on protein domains, which are the units for structural classification, as described above. Given a target amino acid sequence with unknown structure, however, the domain content (the positions of individual domains on the sequence) is usually unknown and has to be predicted. As stated above, AutoSCOP can already give insights into the potential domain structure, but it is often not exact enough. Therefore, in order to provide a more exact domain prediction method (though also computationally more expensive than AutoSCOP), we apply techniques which have been proven to be useful, namely patterns and the alignment-based fold recognition, together with appropriate filtering and scoring methods in our SSEP-Domain server, which assigns potential domain regions to amino acid sequences on the basis of similarity to known domains [Gewehr and Zimmer, 2006]. In the CASP 6 and CAFASP 4 community-wide blind-test experiments, SSEP-Domain was ranked among the top performing servers for protein domain prediction.
- **Chapter 6: Optimized alignment scoring and computation.** Once templates have been selected, for building structure models using comparative modeling, an important task is selecting the best model alignment out of a pool of sequence-structure alignments for a target. In joint work with Fabian Birzele, we have recently developed the QUASAR system [Birzele et al., 2005], a software for alignment ranking and model selection. Using the infrastructure of this software package, we evaluate known alignment scores for this purpose and show how to optimize combinations of them. Based on genetic optimization, we propose a method for building new matrices for alignment scoring. Further, we show how to extend this approach towards the generation of fold-class specific parameters for profile-profile alignment computation, which are able to improve the quality of the resulting models as compared to the default parameters.
- **Chapter 7: Additional tools for protein research.** Both Preselection and AutoSCOP require only the target's amino acid sequence as input; however, if the target structure is known, it is possible to include this information into the prediction process, i.e. for assigning a potential SCOP classification to the target. Vorolign [Birzele et al., 2007], developed by Fabian Birzele in joint work with the author and Gergely Csaba, is a new structure-based fold recognition and structural alignment method, which is able to align protein structures also in case of inherent protein flexibility, making use of both the sequence of residues of the aligned proteins and their structural neighborhoods based on Voronoi tessellation.

BioWeka [Gewehr et al., 2007b], developed in joint work with Martin Szugat, is an extension to the Weka data mining framework [Witten and Frank, 2005] that introduces bioinformatics data formats and methods to Weka. In addition, we developed a new XML schema based on the original ProML language [Hanisch et al., 2002] for the description of proteins and protein sets, which can be used with BioWeka, for instance.

The methods and tools in this work contribute to improving subtasks of protein structure prediction. As shown in Figure 1.1, they fit directly into the context of protein structure prediction:

As described above, a generalized, domain-based protein structure prediction process may look as follows: (1) find the domains on the target, (2) align them to good templates, and (3) build and refine corresponding models. In the diagram, starting on the left path, i.e. on the upper left corner, we are given a target sequence, usually with unknown structure. This sequence can be split into potential domains with SSEP-Domain. Protein domains can then be assigned structural classifications with methods such as Preselection and AutoSCOP. Here, also BioWeka may be included in order to apply machine learning methods for fold recognition on particular protein features, for instance. With the AutoPSI database, for many protein sequences available in UniProt, predicted classifications and their locations based on patterns can already be looked up without further overhead. Once structural classifications have been assigned, it is possible to select suitable templates with similar or identical classifications for the modeling of these domain regions, starting with alignments. QUASAR aims at ranking those alignments as high as possible that are expected to result in a good structure model. As we will see in chapter 6, if we can be relatively sure of the structural classification of a target domain, in some cases the alignments can even be refined using the described approach for optimization of scoring matrices and realignment. Based on such alignments, structure models can be generated.

However, protein domains and structural classifications are also interesting for newly resolved protein structures, for instance for the understanding of protein evolution or the evaluation of protein fusion and fission events. In a second scenario starting in the upper right corner of the diagram, if the target structure is known, we can find domains on the basis of structural information using standard methods for this task and then include the Vorolign structural alignment server for assigning structural classifications. In particular, for this work, we make use of AutoSCOP and Vorolign for the AutoPSI database of predicted classifications in order to assign potential domains and corresponding SCOP classifications to those new PDB entries that have not been classified by SCOP yet.

Overall, in this thesis we describe methods that can predict structural classifications quickly, namely Preselection, AutoSCOP and Vorolign. With AutoSCOP and the SSEP-Domain method, we can further predict the domain content of many sequences, either coarse-grained but very fast, or more refined but still fast enough to be applied to larger numbers of targets. In addition, the methods described in chapter 6 can help improving the quality of structure models, and BioWeka and ProML can be used for further evaluation of protein properties using machine learning approaches.

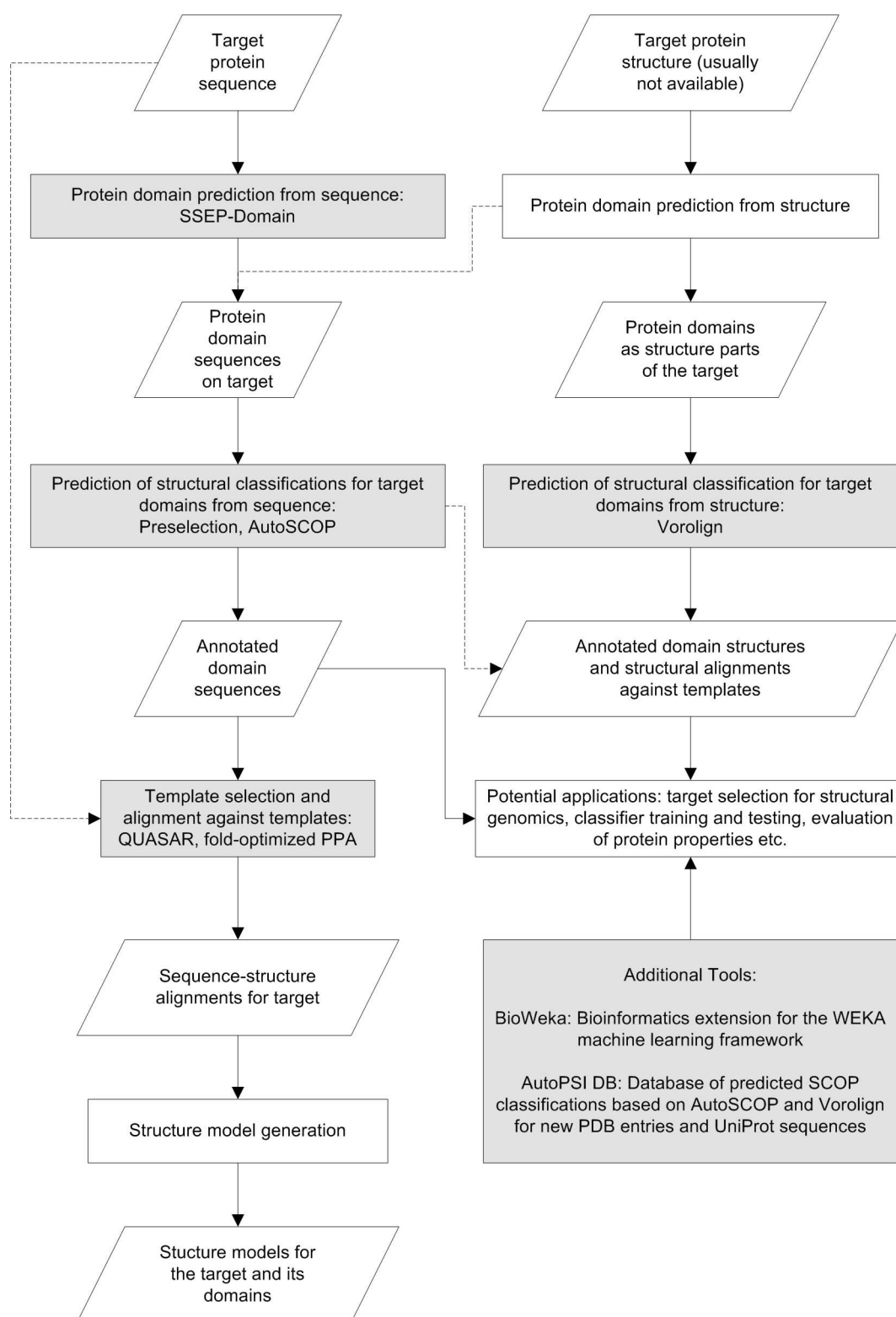


Figure 1.1: Overview of the methods described in this thesis and their contributions to template-based protein structure prediction. The light gray boxes mark those steps that contain algorithms and tools proposed in the following chapters, the other boxes are used to illustrate possible process flows in protein structure prediction. See section 1.2 for a detailed description of this diagram.

Chapter 2

Introduction to Protein Structure Prediction

This chapter contains general background knowledge for this thesis. We start with properties of protein structures, describe well-known databases and the assignment of protein domains, and end up with alignment methods, with the focus on methods and data used in this work.

2.1 Protein Structures and Related Databases

First, we introduce some of the structural properties of proteins. A more complete introduction can be found in "Introduction to Protein Structure" by Branden and Tooze [Branden and Tooze, 1999] (which is one of the main sources for this part). The second section describes important databases for researchers that work with protein structures. Here, besides the sources referenced in the text, [Bourne and Weissig, 2003] was used as an additional source.

2.1.1 From Primary to Tertiary Structure

A protein consists of *amino acids*, which are connected as a chain. As this chain is arranged in space, parts of a protein adopt certain local structural properties, whole proteins fold into their so-called *structures*, and several chains (even more than one protein) can connect and build new, more complex structures.

The so-called *primary structure* of a protein is defined as the order of its *amino acids* as determined by its genetically encoded sequence. Amino acids consist of a central carbon atom (C_α) and two connected groups, namely an amino group (NH_2) and a carboxyl group ($COOH$). Further, they have a so-called *side chain* attached to the C_α atom. In proteins, most of the naturally occurring amino acids (or *residues*) can be found in a standard alphabet of 20 amino acids which differ in the structure of their side chains. For this alphabet, there are two common representations: three-letter-representation (e.g. ALA for

Alanine) and one-letter-representation (e.g. A for Alanine) of the contained amino acids, the latter of which will be used in this work.

Amino acids are connected by so-called *peptide bonds*. Here, the carboxyl group of the first amino acid condenses with the amino group of the next amino acid to eliminate water [Branden and Tooze, 1999]. This results in the amino acid chains that make up the primary structure of a protein as described above.

A protein chain can usually fold by itself or together with other protein chains into a three-dimensional structure. Thereby, small parts of a chain build certain, often-observed local structures. When using the term *secondary structure*, one assigns local structure types to the contained amino acids. For this purpose, we make use of a three letter alphabet {C,E,H}, where C stands for *coil*, E stands for *extended*, and H stands for *helix*. This alphabet is also used e.g. by the well-known protein secondary structure prediction software Psipred [Jones, 1999b], which is important for many of the methods described in this work. Helices or α *helices* are parts of amino acid chains that contain a helix-like structure with 3.6 residues per turn and hydrogen bonds between residues n and $n + 4$. Other helix types can be formed with hydrogen bonds between n and $n + 3$ or $n + 5$ (namely the 3_{10} helix and the π -helix). The second type, extended regions, denote parts of the chain that are in an almost fully extended conformation. A combination of these regions where the individual parts lie either parallel or antiparallel to each other as β *strands* is called a β *sheet*. The term *coil* in this context is used for parts of the chain that contain any other local structure that is not helix or strand.

The *tertiary structure* of a protein is the three-dimensional structure its amino acid chains fold into. Finally, one speaks of the *quarternary* structure, when proteins exist as subunits that then bond with other subunits to build more complex structures.

2.1.2 Structure-Related Databases

There exist many databases in the field of protein structures and protein annotations, each of which has special properties that make it useful for certain tasks. Some of these databases, which are described below, have become very popular or, in case of e.g. the PDB, have become nearly inevitable for certain types of data. Here, we concentrate on those databases that are important for the following chapters.

The PDB

The protein data bank (PDB) [Berman et al., 2000] (<http://www.pdb.org>) is both an ancient and maybe the most important database for structural bioinformatics. New protein structures are made available in the PDB after they have been solved and published, and therefore the PDB provides the basis for most structure prediction research. It was started at the Brookhaven National Laboratory in 1971 and is thus one of the earliest community-wide databases of biological data [Bourne and Weissig, 2003]. The file format used by the PDB contains information about the source, the sequence and the three-dimensional coordinates of a protein structure among other information. For instance, a number of

method-specific details can be described, like experimental conditions and data collection information. Further, the website of the PDB offers derived information such as the domain content of a protein as assigned by different sources.

SCOP/ASTRAL and CATH

SCOP and CATH use protein domains (see next section) as classification units and order them in hierarchies, i.e. in tree-like structures with certain similarity criteria at each level of a tree.

SCOP is short for Structural Classification of Proteins. The database was set up in 1995 by Murzin and coworkers [Murzin et al., 1995]. The corresponding website can be found at <http://scop.mrc-lmb.cam.ac.uk>. Its classifications are mainly made manually. The SCOP hierarchy contains four main levels: class, fold, superfamily and family. Further, some sublevels of family are available. Families are supposed to contain clearly evolutionary related domains which can be grouped e.g. by sequence, structure or function similarity. The next higher level, the superfamily level, groups families which have common structures or functions and are believed to be evolutionary related. Sequence similarity within superfamilies can be much lower than within families, as structure is often more strongly conserved than sequence. The fold level groups superfamilies by so-called *core structures*, namely "the same secondary structure elements in the same arrangement with the same topological connections" [Bourne and Weissig, 2003]. Classes (the highest level) as used by SCOP are defined by the secondary structure element content of the domains, which becomes clear by looking at the full names: "all α ", "all β ", " α/β ", " $\alpha + \beta$ ", and some more. Besides coils, all α domains contain mostly helices, all β domains mostly sheets, α/β domains usually contain a sheet surrounded or flanked by helices, and $\alpha + \beta$ domains contain largely separated regions for helices and sheets.

An addition to SCOP is the ASTRAL compendium [Chandonia et al., 2004], which provides selections of SCOP domains filtered for different levels of sequence identity. Additional features of ASTRAL include Hidden Markov Models for SCOP families and coordinate files for each SCOP domain. ASTRAL is an important resource by itself, as it provides the SCOP data in an easily accessible way that makes possible many of the evaluations and applications that are described in the following chapters. ASTRAL is available at <http://astral.berkeley.edu>.

The name of the second, large domain compendium, CATH [Orengo et al., 1997] is an acronym for the levels of its hierarchy: Class, Architecture, Topology and Homology. In contrast to SCOP, some parts of CATH are automated, such as the definition of the class of a domain based on its secondary structure composition and packing. This level to some degree corresponds to the class level of SCOP, with slight differences. For instance, the two classes " α/β " and " $\alpha + \beta$ " were merged to a single class. The next level, architecture, does not consider connectivity but the orientation of secondary structures with respect to each other. It can be regarded as being situated above the level of folds in SCOP and contains architectures such as β barrels or α bundles. The fold of a domain is used for the definition of the third level, topology, which includes connectivity between secondary structures. The

next level, the homology level, then defines superfamilies of evolutionary related domains based on sequence, structure or function. Further, finer classes are e.g. families within superfamilies etc. Although the CATH domain assignment involves both automated and manual steps, the CATH database is often considered an expert-based database such as SCOP.

2.2 Assignment of Protein Domains

Protein domains, which have already been mentioned as basic units of SCOP and CATH, are subunits of a protein. Beyond this fact, there exist a number of different definitions which make the use of the term "protein domain" difficult. Nonetheless, domains are important for a number of reasons and in a number of different areas. For the crystallization of proteins, for instance, it is helpful to know which parts of a protein can fold independently, as these parts may be easier to crystallize independently, too. Similarly, in structure modeling and structure prediction, it may happen that one domain of a new protein sequence with unknown structure is very similar to a domain in a known protein structure A, whereas a second part is more similar to a domain belonging to a protein B; structure modeling may then be improved by handling domains independently. Also for predicting the function of a protein, it is helpful to be able to know which parts of the protein are similar to certain well-known functional subunits of proteins.

2.2.1 An Old Problem

In [Bourne and Weissig, 2003], Lorenz Wernisch and Shoshana J. Wodak review methods for the identification of domains in protein structures. They briefly cover the history of domain assignments starting in the early 1970s. According to the authors, the most popular concept of the earlier domain definition methods is based on a "globular" view, i.e. regarding domains as globular parts of proteins: it is usually assumed that "the atomic interactions within domains are more extensive than between domains". Often, domains were also considered as being stable on their own and possibly to fold independently. One problem with this approach is that globular or at least highly connected parts of a protein do not necessarily have to be comprised of contiguous sequence segments. Some possible reasons for so-called *discontinuous* domains are domain swapping or gene insertion events.

Wernisch and Wodak define basically two generations of methods. All first generation methods in this review have in common that they do not "consider the problem of optimally partitioning the protein 3D structure in its full generality", as they are based on the order of residues and use continuous segments of the sequence, at least as starting points. They identify those methods as the second generation methods that are free of such restrictions, which are often influenced by other disciplines like physics, statistics or graph theory.

Popular approaches which are currently used by the PDB database as additional information on protein structures are Domain Parser (DP) [Xu et al., 2000] and Protein Domain Parser (PDP) [Alexandrov and Shindyalov, 2003], both of which are able to de-

tect and define continuous as well as discontinuous structural domains based on a given protein structure in PDB format. Further, one of the earliest large-scale efforts to define and classify domains in a hierarchically way and provide the results as a resource for the community is FSSP [Holm and Sander, 1997], which is built using the DALI software [Holm and Sander, 1996] for structure-based domain recognition.

2.2.2 Comparison of Domain Assignments

William R. Taylor and Andras Aszodi name some reasons for the difficulties of finding a structure-based domain definition in chapter 7 of their book "Protein Geometry, Classification, Topology and Symmetry" [Taylor and Aszodi, 2004]. These are extensive interfaces between domains, which pose the problem of finding an appropriate level of granularity, and also the cases of discontinuous domains, which are in general harder to detect than continuous ones. Such reasons can result in differing domain definitions for a single protein for different experts or methods. A recent study by Veretnik et al. [Veretnik et al., 2004] compares protein domain definitions from some popular sources and analyzes the agreements and disagreements. Overall, the authors identify five possible classes of domain definitions, namely (in their words):

1. "Regions that display a significant level of sequence homology;"
2. "a minimal part of the gene that is capable of performing a function;"
3. "a region of the protein with an experimentally assigned function;"
4. "parts of structures that have significant structural similarity;"
5. and "compact spatially distinct units of protein structure."

For definitions one and four, similarity is often measured in comparison to similar sequences/structures, i.e. these definitions concentrate on conserved regions of the sequences/structures. The complete list illustrates the diversity of the existing definitions, as these five are apparently quite different from each other. Especially the latter three of these definitions are widely used and have thus been evaluated by Veretnik and coworkers.

The methods used for their setup have been categorized as follows: (1) *expert methods* such as SCOP, CATH and the annotation provided by the authors of crystal structures (AUTHORS) [Islam et al., 1995], and (2) the *algorithmic methods*, namely DALI [Holm and Sander, 1996], Domain Parser (DP) [Xu et al., 2000] and Protein Domain Parser (PDP) [Alexandrov and Shindyalov, 2003].

For their test set, the authors find that between 80 and 90% of the assignments of the different methods are in agreement with respect to the assigned number of domains, and each of the methods has its inherent advantages and drawbacks. They also find some tendencies:

- **Single domain vs. multi-domain chains.** The assigned number of domains varies significantly between methods. E.g. SCOP tends towards low numbers (81% single domains as compared to AUTHORS with only 69.5%), whereas DALI outputs large numbers of domains in comparison.
- **Continuous vs. discontinuous domains.** Again, e.g. SCOP shows a high number of continuous assignments (97%), whereas some of the other methods result in approximately 10-15% discontinuous domain assignments.

The evaluation of domain boundaries also shows that, when using AUTHORS as reference, the agreement on domain boundaries based on the overlap of the assigned domains is good. Based on the chosen domain overlap measure, especially SCOP shows the smallest number of chains that disagree on domain boundaries for both an 80% overlap threshold and a 95% overlap threshold (see Table 2 in [Veretnik et al., 2004]).

Further, in a more recent evaluation, Holland and coworkers [Holland et al., 2006] compared automated methods including PDP and DP to an expert consensus from SCOP, CATH and AUTHORS. Their results confirm that different methods have different tendencies: in particular, PDP, which reached 85% correct assignments, clearly tends towards predicting too many domains, whereas DP, which reached 77% correct assignments, tends towards too few predicted domains with respect to the experts.

These evaluations make us aware of the problem that there is no perfect protein domain definition standard for our experiments and methods. So far, the results of a domain assignment method will depend to some degree on the domain definition that is used as well as, if applicable, on the basis of its training data (e.g. SCOP or CATH).

For this work, we mostly concentrate on SCOP domain definitions, as the SCOP database is one of the major, expert-curated sources for domain definitions. According to its' authors, "A domain is defined as an evolutionary unit, in the sense that it is either observed in isolation in nature, or in more than one context in different multi-domain proteins" [Lo Conte et al., 2002]. In other words, SCOP uses domains as recurrent structural units. For structural classifications and their prediction, we believe that this is a good notion, as this definition inherently contains additional information about parts occurring together in all observed cases. This does not make it necessary to find all possible subparts of such a unit individually in a template search, for instance. Once a significant similarity to a SCOP domain has been found, the whole recurrent unit has been identified and can be used for subsequent steps such as structure modeling under the assumption that indeed all subparts of the SCOP domain always occur together. Further, a lot of groups use SCOP domains (in [Bourne and Weissig, 2003] Reddy and Bourne observed that "SCOP is the most cited resource for classifying proteins"), which makes it possible to compare results with other methods on the same standard.

Nonetheless, we are well aware of the problem of differing domain definitions. In our protein domain prediction method SSEP-Domain (chapter 5) we also made use of SCOP domains. Veretnik et al. state that "When SCOP assigns the number of domains correctly, it also assigns the domains correctly," and indeed independent evaluations confirmed that

our method is quite accurate in boundary placement. However, we found that in some cases SCOP domains disagree with other experts' point of view, who may emphasize the structural properties such as globularity. This agrees with Veretnik et al.'s observations. In the corresponding chapter, we therefore discuss the differences when using SCOP and other sources of domain assignments such as CATH, PDP and DP against the background of protein domain detection on the CAFASP 4 and CASP 7 targets.

2.3 Structure Prediction Categories

Proteins and their structures are determined by their corresponding genes. Therefore, if two genes have evolved from a common ancestor, one can use both the term *homologous genes* and, more importantly for this work, *homologous proteins*. For finding homologous proteins, many methods have been proposed. Mostly, these methods rely on sequence similarity between proteins, alone or in combination with other features. Homology as indicated by these factors is the basis for the differentiation of the structure prediction approaches into Comparative Modeling, Fold Recognition, and Ab Initio methods.

2.3.1 Comparative Modeling

Comparative modeling is a term that is often used for a structure modeling process, when significant sequence similarity has been observed. In a recent review, in agreement with John Moult's review of protein structure prediction [Moult, 2005], Krzysztof Ginalski, one of the most successful predictors in the CASP 6 structure prediction experiment, writes that the most reliable and accurate protein structure models still come from comparative modeling approaches when applicable [Ginalski, 2006]. The driving force behind template-based structure prediction in general and comparative modeling in particular is the hypothesis that two evolutionary related proteins (where we find sufficient sequence similarity) also have similar three-dimensional structures (an illustrating diagram of this view can be found in Fig. 2.1). Therefore, the usual steps for comparative modeling are

1. *Template Selection*: Given a databases of so-called *templates*, i.e. proteins of known structure, find the most similar templates for the target.
2. *Alignment*: Align target and templates such that the resulting alignments reflect the structurally similar regions for each target-template pair.
3. *Model Generation*: Based on the alignments, build structure models for the target.
4. *Model Selection*: Rank the resulting models such that the potentially best model can be selected.

The last step is closely related to the problem of automated servers e.g. in CASP and CAFASP (see below) to predict only a restricted number of models for a target (five, for instance), and to select a "first" model for both the benefit of a user and the evaluation

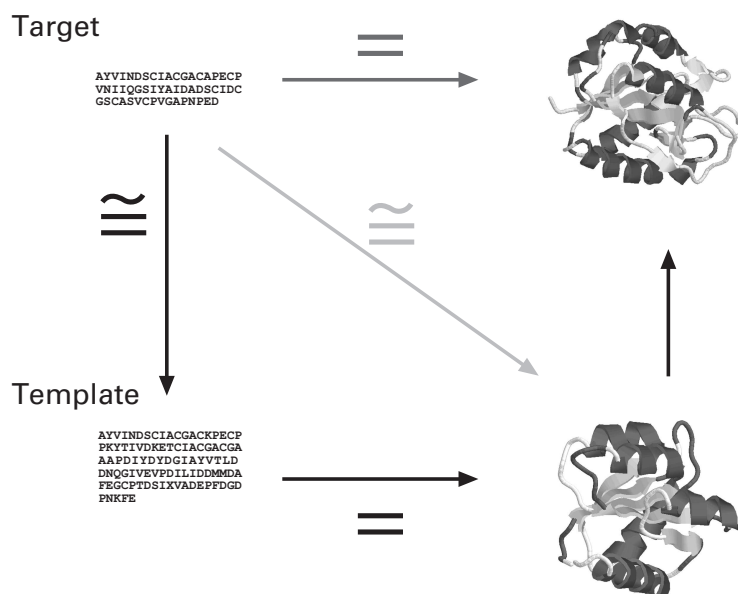


Figure 2.1: Homology-based approach to protein structure prediction (shown as black path). Instead of directly predicting the structure for a target (as it might be done by some ab initio methods), we select a template of sufficient sequence similarity and align it to the target. Then, under the assumption that sequence similarity yields structural similarity (shown as light gray path), we can transfer the coordinates of the template to the target on the basis of the aligned positions and subsequently refine the model.

of comparison experiments. Nonetheless, as Ginalski states, the "optimal use of structural information from available templates" and the correctness of the alignments are still the "most significant determinants of final model quality" [Ginalski, 2006].

2.3.2 Fold Recognition

Ginalski further categorizes comparative modeling cases as such in which templates can be found with standard methods such as PSI-BLAST [Altschul et al., 1997] or BLAST [Altschul et al., 1990]. The fold recognition category focuses on those cases where remote homologs can be found with newer, in some cases very elaborate methods, such as profile-profile alignment methods [von Öhsen and Zimmer, 2001, von Öhsen et al., 2003] or the alignment of so-called profile Hidden Markov Models [Eddy, 1998, Söding, 2005]. While nowadays the methods are very similar for both difficulty classes, the final implementations often differ depending on which area they focus on, e.g. by choosing to optimize their parameters on either closely related protein pairs or on remote homology cases.

2.3.3 Ab Initio

A whole new class of algorithms comes from the other side of the difficulty spectrum by focusing on structure prediction in cases where no suitable templates are available. These are called *ab initio* or *de novo* approaches. In this work, as we do not make use of *ab initio* methods, we will not discuss the corresponding methods and principles further.

2.4 Community-Wide Efforts

In order to assess and speed-up progress in protein structure prediction, some community-wide efforts have been established. Especially the CASP and CAFASP experiments are of importance for this work, as they provide independent assessments of protein structure prediction methods. Besides CASP/CAFASP, also Structural Genomics efforts are very important for research in structural biology depending on solved protein structures since they allow for concerted target searches and the solution of those targets which are deemed to be especially interesting.

2.4.1 Community-Wide Experiments

CASP¹ (Critical Assessment of Structure Prediction) is a large-scale community experiment, conducted every two years [Moult, 2005]. From 1994 to 2006, CASP has monitored the progress of the structure prediction approaches, both manual and automated. In 2004, already over 200 prediction teams from 24 countries participated in CASP 6. A similar experiment, CAFASP² (Critical Assessment of Fully Automated Structure Prediction), has been held five times so far, with focus on fully automated prediction servers [Fischer et al., 2003]. The procedure of such experiments is as follows: The organizers collect sequences for which structures will be solved in the near future and pass them on to the registered predictors. Human groups have some weeks and servers have a couple of days to submit their top models for each target. In 2004, besides structure prediction also categories like domain prediction, model refinement and model quality assessment have been added to CASP and CAFASP.

Especially important is the fact that the prediction setup described above is a blind-test setup, i.e. the true structure is truly unknown at the time the predictions have to be delivered, which allows for a fair comparison between methods. Further, independent assessors carefully analyze and rank the methods with respect to different aspects of their prediction performance, and thus provide an independent and widely acknowledged assessment of the progress and performance of the state-of-the-art approaches.

¹<http://predictioncenter.org>

²<http://www.bgu.ac.il/~fischer/CAFASP>

2.4.2 Structural Genomics and Structure Prediction

One excellent example of the effort spent on resolving more protein structures experimentally is the Structural Genomics Initiative of the National Institute of Health (NIH), a "worldwide initiative aimed at determining a large number of protein structures [...] in a high throughput mode [...]" (quoted from <http://sg.pdb.org/>). The term structural genomics stems from the aim to determine all protein structures for the available genomes, which may be regarded as the logical next step after determining all potential proteins derived from a genome. In other words, "the ultimate goal of structural genomics is to provide structures for all biological proteins" [Yan and Moult, 2005].

Nonetheless, even these efforts (combined with similar initiatives around the world) do not aim at directly solving the structures for all known protein sequences. In particular, as stated by Grabowski and coworkers, "the initial long-term goal of the Structural Genomics (SG) endeavor was to map all protein folds, so that the structures of virtually all proteins could be either found in the Protein Data Bank (PDB) or derived by computational methods" [Grabowski et al., 2007]. As described above, since evolutionary related proteins have similar structures, comparative modeling methods can be used to obtain a structure for any protein with a detectable evolutionary relationship to another protein with an experimental structure [Yan and Moult, 2005]. Even knowing that the accuracy of such comparative models is usually not as high as that of a high-quality X-ray structure, such structure models can still be useful, as discussed in chapter 1 (see also [Baker and Sali, 2001]).

2.5 Alignment Methods

As we have seen, in protein structure prediction, the ultimate question to answer is how to determine the three-dimensional structure of a protein from its sequence. In order to be able to tackle these tasks, especially one category of tools has become very popular, namely *alignment methods*.

Alignment methods generate so-called *alignments*. These are used to find similarities between protein sequences, which in turn can be used to find and map appropriate templates to a structure prediction target. For instance, most alignment methods used in this work generate mappings between the amino acids of two or more proteins based on different properties, which can then be utilized to assign coordinates to the matched amino acids of the target on the basis of the known coordinates for the match partners in a template sequence and its corresponding structure. Besides protein structure prediction, comparing amino acid sequences is also important for other tasks in molecular biology, as it may give insights into evolutionary relationships, in some cases functional similarity, and more.

2.5.1 Sequence Alignment

The most simple case of aligning two amino acid sequences by using only the order and the type of amino acids has been investigated for decades. In the following, we start with

this case and, based on it, explain techniques like profile-profile alignment and secondary structure element alignment.

Representation of Alignments

In order to find such similarities between amino acid sequences, it is necessary to find a suitable representation for protein sequences. Firstly, one needs an alphabet of amino acids:

Definition (Amino Acid Alphabet). The *alphabet of amino acids* Σ_A in its one-letter representation is defined as $\{A, C, D, E, F, G, H, I, K, L, M, N, P, Q, R, S, T, V, W, Y\}$, based on the 20 most usual amino acids in protein sequences. Sometimes this alphabet is combined with an additional character "X", which stands for any other, non-standard amino acid.

As stated in the introduction to this section, given two sequences $S_1, S_2 \in \Sigma_A^*$ built from the alphabet of standard amino acids, the aim of sequence alignment is to derive a measure of similarity (or, in some cases, difference) between these sequences, the *alignment score*, and to *align* the sequences such that an amino acid from S_1 is either mapped to an amino acid from S_2 or left unmapped (and vice versa). Thereby, in case of sequence alignments, the sequential order of amino acids is preserved, i.e. it is not possible to map the residue at position 1 of S_1 to the residue at position 2 of S_2 as well as the residue at position 2 of S_1 to the residue at position 1 of S_2 in the same alignment:

Definition (Extended Alphabet) The *extended alphabet of amino acids* $\Sigma_{A,-}$ is defined as $\Sigma_A \cup \{-\}$, where the so-called *gap symbol* "-" stands for "unmatched".

Definition (Alignment) A *pairwise alignment* of two amino acid sequences $S_1, S_2 \in \Sigma_A^*$ is defined as a tuple $\{A_1, A_2\}$, where $A_1 \in \Sigma_{A,-}^l$ results from S_1 after insertion of gap symbols (and A_2 from S_2 analogously). Here, $l \in \mathbb{N}$ denotes the length of the alignment, as both extended sequences A_1, A_2 must have the same length. *Aligned positions* $i \in \{1, \dots, l\}$, i.e. positions where neither extended sequence contains a gap symbol, can either be *matches* (i.e. $A_1[i] = A_2[i]$) or *mismatches* ($A_1[i] \neq A_2[i]$). At an *unaligned position* or *gap position*, either A_1 or A_2 contains the gap symbol "-". Please note that it is not allowed for both A_1 and A_2 to contain the gap symbol at the same position.

Scoring of Alignments

The basis for aligning amino acids is usually a so-called scoring matrix, which contains a score value for each pair of amino acids a and b based on observations such as point mutations or frequencies of occurring amino acid pairs obtained from superpositions of similar protein structures.

Stretches of unmatched residues are called *gap regions* or simply *gaps* and may occur in both A_1 or A_2 , as described above. In alignment methods which try to maximize a similarity score (in contrast to minimizing the difference score), gap positions are usually punished by a negative score. Further, one categorizes so-called *gap costs* as e.g. *linear* (i.e. each unmatched residue contributes the same score) or *affine*, where both a *gap open* score (the punishment for opening a new stretch of unmatched residues) and a *gap extend* score (the punishment for elongating a gap) are used.

Each alignment between two amino acid sequences can be assigned an overall score, the so-called *alignment score*, given a scoring matrix M and gap costs, by summing over the values in the matrix corresponding to the aligned positions and applying the gap costs depending on the distribution of gap symbols in the alignment. The computation of the gap punishment further depends on the alignment model that has been chosen, such as *global* or *local alignment* (see below).

Computation of Alignments

The computation of alignments usually means the search for the optimal alignment with respect to a certain alignment scoring scheme. For global alignment, the computation of the optimal alignment score with affine gap costs can be done by recursive equations, which can be found in a similar manner in many papers and books about bioinformatics. Here, we use a similar notation to Gusfield's in [Gusfield, 1997]:

Definition. Let i and $j \in \mathbb{N}$ denote positions on amino acid sequences S_1, S_2 . Define $E(i, j)$ as the maximum value of any alignment of prefix $S_1[1..i]$ with prefix $S_2[1..j]$ that ends with a gap in the extended sequence A_1 . Define $F(i, j)$ as the maximum value of any alignment that ends with a gap in the extended sequence A_2 . Define $G(i, j)$ as the maximum value of any alignment that ends with a match or mismatch. Finally, define $V(i, j)$ as the maximum value of $E(i, j), F(i, j)$ and $G(i, j)$.

The base cases of the necessary recurrence equations for global alignment can be written as

$$\begin{aligned} V(i, 0) &= E(i, 0) = -\text{gap}_{\text{open}} - i \text{gap}_{\text{extend}}, \\ V(0, j) &= F(0, j) = -\text{gap}_{\text{open}} - j \text{gap}_{\text{extend}}. \end{aligned}$$

The recurrences themselves are

$$\begin{aligned} V(i, j) &= \max\{E(i, j), F(i, j), G(i, j)\}, \\ E(i, j) &= \max\{E(i, j-1), V(i, j-1) - \text{gap}_{\text{open}}\} - \text{gap}_{\text{extend}}, \\ F(i, j) &= \max\{F(i-1, j), V(i-1, j) - \text{gap}_{\text{open}}\} - \text{gap}_{\text{extend}}, \\ G(i, j) &= V(i-1, j-1) + \text{score}(S_1(i), S_2(j)). \end{aligned}$$

Definition. The score of a global alignment given affine gap costs is the score obtained after having aligned both sequences S_1, S_2 completely according to the equations given above.

The aim of global alignment [Needleman and Wunsch, 1970] is to align the whole input sequences S_1, S_2 , and therefore each gap position is punished based on the assigned gap costs. For local alignment [Smith and Waterman, 1981], the aim is to find a local, highly similar region. Everything surrounding this region is ignored, and match/mismatch scores as well as gap costs apply only for positions in the corresponding part of the alignment. This means that the initialization as well as any other value computed in the recurrences is never below 0. So-called freeshift alignments are a special case. Here, it is assumed that a long part of the two sequences is similar, but can occur at different positions of the whole sequences. Therefore, leading and trailing gaps are free, and in between the mechanisms of global alignment are applied. Intuitively, this means that the two sequences can be slid along each other (or shifted) without costs.

Having computed the optimal score, one chooses one or more alignments that achieve this score by following the path that is given by the choices in the maximum operations backwards to the beginning of the recursion, as each choice defines either a match/mismatch between two residues or an insertion of a gap symbol in one of the two aligned sequences. Multiple optimal alignments are possible when the maximum operations can choose from equal values.

Alignments can usually be computed quite efficiently by a relatively simple technique called dynamic programming. Here, one makes use of the fact that many results in the recursions are computed over and over again in different instances and can thus be computed only once and then stored for efficiency. Then, as can be seen for the equations shown above, in a naive implementation the effort for alignment computation with affine gap costs is reduced to filling matrices E, F, G , and V , where the i and j in the equations correspond to the rows and columns of these matrices. The details of this technique are described in most of the available bioinformatics textbooks, including [Gusfield, 1997], and will not be discussed here.

Further, so far we have only introduced *pairwise* alignments, i.e. alignments between two protein instances. So-called *multiple* alignments, alignments between more than two proteins, require more elaborate techniques, e.g. to determine the order by which instances are incorporated in an incrementally growing alignment and so forth. In fact, the exact multiple alignment problem for the so-called sum of pairs-score has even been shown to be NP-complete [Wang and Jiang, 1994]. Prominent software tools using heuristics are ClustalW [Thompson et al., 1994] and T-Coffee [Notredame et al., 2000]. Again, the corresponding techniques are well-documented in the literature and will not be discussed here.

2.5.2 Alignment Methods used in this Work

In the final section of this chapter as well as of this part, we briefly describe the two alignment versions that will occur the most often in the next part, namely log average profile-profile alignment and secondary structure element alignment.

Profile-Profile Alignment

One recent breakthrough in structure prediction was achieved by the introduction of profiles. In a (sequence) profile as used in this work, a residue in an amino acid sequence at a position i is replaced by a vector $\in \mathbb{R}^{20}$, which contains the occurrence probability of each residue at this position in a multiple alignment of similar protein sequences. The alignment procedure works exactly as described above, with the only difference being the scoring of two aligned positions when, as it is the case for profile-profile alignment methods, vectors of occurrence frequencies are used on both the target and the template side during the alignment process. Well-known scoring functions used for profiles are discussed in [von Öhsen, 2005], such as

- **Dot product:** This method was proposed by [Rychlewski et al., 2000] and is probably the most simple way of scoring the coincidence of two vectors $\alpha, \beta \in \mathbb{R}^{20}$:

$$\text{score}_{\text{dotproduct}}(\alpha, \beta) = \sum_{i=0}^{20} \alpha_i \beta_i.$$

- **Average scoring:** This scoring function, which was used by ClustalW in a similar fashion, averages over the scores between residues:

$$\text{score}_{\text{avg}}(\alpha, \beta) = \sum_{i=0}^{20} \sum_{j=0}^{20} \alpha_i \beta_j \log \frac{p_{\text{rel}}(i, j)}{p_i p_j}$$

where $p_{\text{rel}}(i, j)$ denotes the observed frequency of the respective amino acid pair (i, j) in related sequences and $p_i p_j$ denotes the background occurrence frequency for this pair in random alignments.

Other methods discussed are an approach originally described by [Yona and Levitt, 2002], the COMPASS method [Sadreyev and Grishin, 2003] and the approaches proposed by [Panchenko, 2003].

Log Average Profile-Profile Alignment

Further, there is a scoring method which was shown to perform very well in comparison to other methods by Niklas von Öhsen and coworkers and which will be used several times in the following chapters. The *log average scoring function* looks very similar to the average scoring function, but contains a slight difference: the logarithm is taken after the averaging:

$$\text{score}_{\text{logavg}}(\alpha, \beta) = \log \sum_{i=0}^{20} \sum_{j=0}^{20} \alpha_i \beta_j \frac{p_{\text{rel}}(i, j)}{p_i p_j}.$$

This score has a number of advantages, which are discussed in [von Öhsen and Zimmer, 2001, von Öhsen et al., 2003], and was evaluated to be very accurate for fold recognition in the

Algorithm 1 Pseudocode of the pairwise SSEA algorithm.

- 1: Represent both target and template as sequences of contained secondary structure elements and discard leading and trailing coils. For instance, the secondary structure sequence CCCHHHHHCCCCCEEEEC becomes $([H,C,E],[5,6,4])$, i.e. a helix of length 5, a coil of length six and a strand of length 4.
 - 2: Align the two sequences using dynamic programming with zero gap costs. H-H, C-C and E-E are scored with the minimum length of the two aligned elements. H-C and E-C (or vice versa) are scored with half the minimum length, and H-E (or vice versa) is scored with zero.
 - 3: Normalize the score by dividing the raw score by the mean length of the two (trimmed) sequences, i.e. without leading and trailing coils.
-

CAFASP 3 experiment, where the corresponding fold recognition server (Arby) was ranked among the top servers for single domain targets [von Öhsen et al., 2004].

For Arby, an addition has been made to the log average function, namely the introduction of secondary structure profiles. Here, the procedure is essentially the same as for sequence profiles, but now two scores are computed (one for the sequence-based similarity and one for the secondary structure-based similarity) which are combined linearly:

$$\text{score}_{\log\text{avg}} = c_{aa}\text{score}_{aa} + c_{sec}\text{score}_{sec}$$

where $_{aa}$ denotes the sequence part and $_{sec}$ denotes the secondary structure part with their corresponding weights $c_{aa}, c_{sec} \in \mathbb{R}$.

Secondary Structure Element Alignment (SSEA)

Given two protein structures, one particular class of structural alignment methods performs a comparison of types and arrangements of α helices and β strands, including the ways these *secondary structure elements* are connected [Mount, 2001]. In many cases these elements are represented as vectors in space (including relative position, type, direction and length) and can thus be compared much easier than three-dimensional coordinates for each residue or even atom. Popular tools that use such secondary structure elements are VAST and SARF [Madej et al., 1995, Alexandrov, 1996].

In the next chapter, we will deal only with secondary structure annotations derived from the amino acid sequence of a target. In this situation, still the order, the type and the lengths of the elements can be used. *Secondary structure element alignment* (SSEA) for secondary structure sequences may therefore to some degree reflect topological similarity between proteins, though with a clearly reduced knowledge base.

The SSEA algorithm we employ for speeding-up PPA-based fold recognition (chapter 3), Vorolign (chapter 7) and SSEP-Domain (chapter 5) is a very simple method using three-state secondary structure representations. A similar procedure to the one described here was proposed by Theresa Przytycka and coworkers in 1999 [Przytycka et al., 1999]. Liam J. McGuffin and David Jones then adopted this approach and found in their evaluation

[McGuffin et al., 2001], that SSEA is able to find similar folds more accurately than other methods in their comparison (including simple sequence as well as secondary structure alignment).

The version that works best in their comparison, and which is therefore used in this work, is described in Algorithm 1. In short, each protein is represented as a sequence of its secondary structure elements (with annotated lengths). The alignment is then done via dynamic programming using a very simple scoring function based on the element types and lengths as described in the pseudocode of Algorithm 1.

Parameters used in this thesis

If not stated otherwise for the corresponding evaluations, all profiles and secondary structure predictions were generated using Psipred and PSI-BLAST [Altschul et al., 1997] against an NR database [Wheeler et al., 2000] of nonredundant protein sequences obtained in April 2004, using 5 iterations for PSI-BLAST (as proposed by [Schäffer et al., 2001]).

For the combination of sequence and secondary structure profiles, we use both the software and the parameters obtained from Niklas von Öhsen, shown here in the notation of the PPA software:

```
common.target.aa.convertermatrix=blosum62
common.target.ss.convertermatrix=KawabataN00
score.gapinsertion=14.7426772777912
score.gapextension=0.36945751321605
score.alimode=global
score.psc.ss.scale=0.695214057254826
score.psc.aa.scale=2.885390082
```

All matrices and classes as used in these parameters were already part of the original PPA software.

Chapter 3

Selection of Fold Classes based on Secondary Structure Elements

Alignments are among the most powerful tools for finding similar proteins to a target in a database of templates and therefore are relevant also for all subsequent tasks such as protein structure modeling. Also for the prediction of the fold class of a protein domain, the so-called *fold recognition task*, alignments are a heavily used tool, as they allow measuring similarity between new and known domains. However, one of the main drawbacks of many of the more sophisticated alignment-based fold recognition approaches is their relatively low speed while, given the growing number of available templates, one often has to find efficient means of selecting useful templates.

A solution to this problem proposed here is a two-stage approach, which uses a simple and thus very fast alignment method to discard a large part of the template database based on assumed topological dissimilarity before employing a more specific but also much slower method in the second stage. We call this approach "Preselection and Refinement".

In this chapter, we present an updated evaluation of this concept which has been presented in a previous stage in [Gewehr et al., 2004], having included a newer and more powerful refinement method, namely log average alignment on both sequence and secondary structure profiles. Our evaluations show that it is indeed possible to speed-up the recognition process over using methods such as profile-profile alignment alone while achieving a similar fold recognition accuracy. Further, the preselection idea has been integrated in different variants into the Vorolign method (chapter 7) and the SSEP-Domain method (chapter 5).

3.1 Introduction

Similar to the protein structure prediction task called *fold recognition*, there exists also a protein classification problem with the same name. Here, the aim is not to deliver a good coordinate model for a target, but the protein classification (namely the *fold*) the target structure will most probably belong to. In other words, for the former task, we

would produce a coordinate model for a target based on (remote) homology, whereas for the latter task it suffices to name the fold class, e.g. "a.1" when using SCOP. In the following, when we use the term "fold recognition", we mean the second problem, i.e. the classification task.

Recent approaches for tackling the fold recognition problem follow two major directions, namely the application of machine learning methods and the application of alignment methods. Representatives of the first direction are the methods by [Ding and Dubchak, 2001] (neural networks and support vector machines (SVMs)) and [Chinnasamy et al., 2004] (tree-augmented naïve Bayesian classifiers). Examples for alignment-oriented methods are GenTHREADER [Jones, 1999a] (sequence-profile alignment, evaluation by energy potentials) and the MANIFOLD approach [Bindewald et al., 2003] (sequence and secondary structure alignments combined with enzyme codes).

While there exist successful methods for this task such as profile-profile alignment, these methods often require considerable computational effort. For speeding up this process, we propose a two-stage approach, which uses a sensitive and fast alignment method to discard a large part of the template database before applying a more specific but also much slower method in the second stage. The approach is based on topological similarity of proteins and protein domains. In particular, given a protein structure, by *topology* we mean the sequence of the contained secondary structure elements (the helices, sheets and coils), their relative positions and orientations in space and the observed contacts between these elements.

Initially, all that is available in a fold recognition setup is the sequence of a target protein, therefore it is not possible to directly compare topologies between the target and the available templates. However, given a good secondary structure prediction for the target, we can approximate topology by using only the sequences of secondary structure elements without knowledge about contacts. Therefore, we can formulate the main hypothesis for this chapter as follows:

Working Hypothesis. Since fold membership is based on the topology of a protein structure, it is possible to select potential fold classes for a target based on the sequence of secondary structure elements.

We still have to define what we mean with *secondary structure element* in this context, i.e. when the structure of protein is unknown, as it is the case for our targets:

Definition (Secondary Structure Element). Given a predicted secondary structure sequence (i.e. a sequence over the alphabet $\{C, E, H\}$), the corresponding secondary structure elements are all contiguous stretches of identical symbols.

For instance, the sequence CCCEECCHHHHCCEECC contains three non-coil elements, namely two strands of length two and one helix of length four, with four surrounding coil regions with lengths 3,2,2,2, in sequential order.

In order to find fold classes quickly that contain topologically similar templates, we

select template fold classes based on the secondary structure elements in the target sequence as predicted by Psipred [Jones, 1999b] using a fast alignment method designed for such cases.

Once a number of potential template fold classes have been selected, often approximate topological similarity based on only the order and length of secondary structure elements is not enough to discriminate between them, as shown for the SSEA method in the results section. At this stage, we therefore change to a finer level of description by using profile-profile alignment (PPA) on both sequence and secondary structure profiles. This method is much slower than the preselection step in finding a single, final predicted class for a target but very accurate in comparison to other fold recognition methods.

An overview of the approach is shown in Figure 3.1. 1) It quickly preselects potential classes. 2) It rescores the selected classes using the second, more expensive measure for selecting the finally predicted class. As our evaluation will show, this idea allows for a reduction of computation time by about one order of magnitude as compared to PPA alone while achieving comparable results in fold recognition.

3.2 Material

3.2.1 Training and Test Data

We use three different data sets for this chapter, one well-known "difficult" set, one newly compiled "intermediate" set, and one well-known, "easy" set:

1. **CATH_MJ**: The first set was introduced by [McGuffin and Jones, 2002]. It contains 542 nonredundant domains based on CATH [Orengo et al., 1997] version 1.7 and is divided into a subset of 252 "known" domains which have at least one other match in this set, and 290 "unique" domains, i. e. domains which have folds unique with respect to this set. In order to compare our method to the results of [Bindewald et al., 2003], we used their approach by selecting the set of known folds as targets and the complete set as templates, excluding identical hits. For comparison purposes we used the classifications given by CATH V2.4 as described in [Bindewald et al., 2003]. It should be noted that, using this CATH version, we can find matching partners with respect to the CATH topology level for only 241 of the set of known domains. For the evaluation, we nonetheless keep all 252 domains as reference number for 100% accuracy.
2. **ASTRAL25**: The second set was compiled from the ASTRAL [Chandonia et al., 2004] subset with less than 25% sequence identity based on SCOP version 1.65¹. We performed leave-one-out tests on all fold classes containing at least 2 members (3999 domains in 441 fold classes). This set was used for training our approach (we evaluated the percentage of selected folds on this set) for two reasons: First, we have no

¹provided by <http://astral.berkeley.edu>

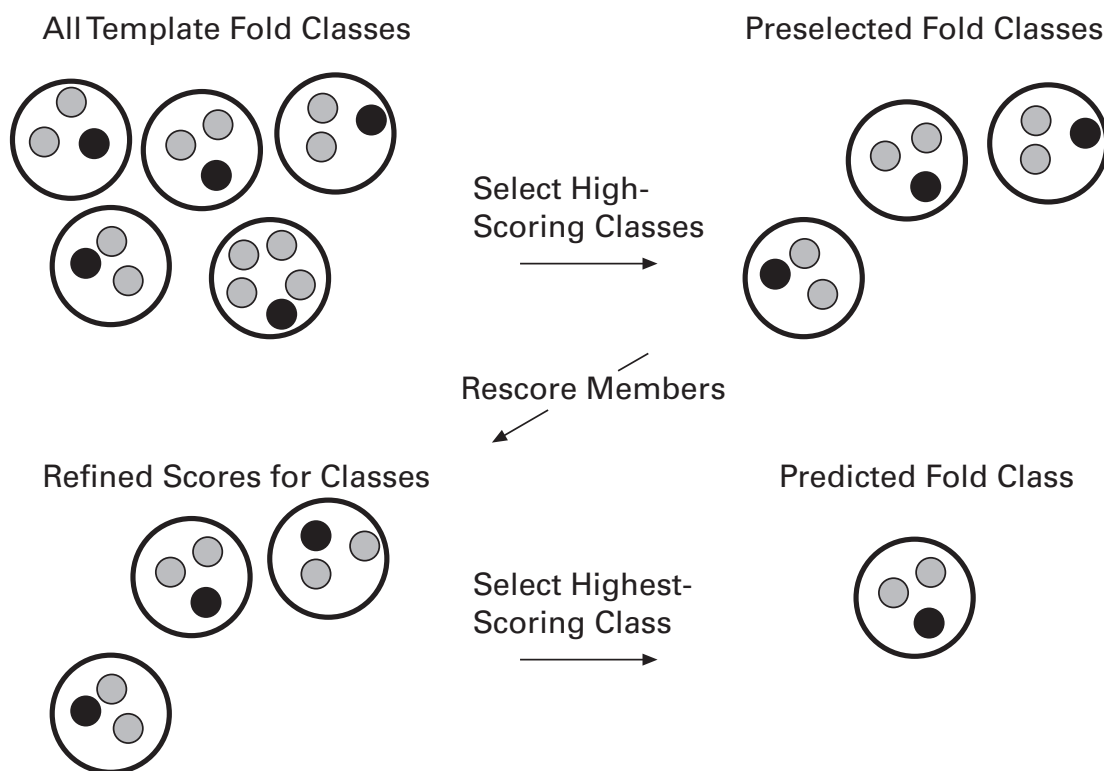


Figure 3.1: Graphical overview of the preselection and refinement approach. At first, all templates are scored with a fast but sensitive method and potential fold classes are selected. Using only these classes, all remaining templates are rescored in the refinement step with a more selective approach and the class with the highest-scoring template is selected for the final prediction.

comparison with other methods on this dataset (in contrast to the two other sets), and second, the ASTRAL set is the the most similar set to the template sets usually used for structure prediction and fold recognition by methods competing e.g. in the CASP experiment. Therefore, a high performance on the ASTRAL set is desirable especially when considering to apply preselection for fold recognition in prediction methods such as SSEP-Domain, for instance, which also uses ASTRAL as template database.

3. **SCOP_DD**: The third set is the test set provided by [Ding and Dubchak, 2001]. It contains 386 SCOP domains in 27 SCOP folds. This set is known to contain (distant) homologs [Bindewald et al., 2003], a fact that leads to higher recognition rate for such target-template pairs. We again follow the MANIFOLD procedure by performing leave-one-out tests on the test set only (Silvio C. E. Tosatto, personal communication).

For this updated evaluation of the preselection approach, sequence and secondary structure profiles as well as secondary structure predictions were generated in the same manner as for the SSEP-Domain method, for instance, which makes use of the preselection approach to speed up protein domain prediction (see section 2.5.2 for details).

It should be noted that 26% of the targets in the SCOP_DD set are contained in the ASTRAL25 set, i.e. 100 of the 386 domains are also used in the ASTRAL set. However, the set is much smaller and the conditions are very different to the ASTRAL set. From the CATH_MJ set, 36.5% of the used protein chains in the test set are also used in the ASTRAL data (92 of 252). Nonetheless, the setup is again very different from the ASTRAL data: no cross-validation is used, the set is much smaller and the domain definitions were taken from CATH instead of SCOP. Therefore, using the SCOP_DD and CATH_MJ sets as test sets allows for a fair comparison with the methods quoted for these sets.

3.2.2 Quoted Methods

For the sets obtained from the literature, we are able to compare our results directly to the accuracy values reported for other methods:

- **MANIFOLD (MF)**. The MANIFOLD method [Bindewald et al., 2003] is the most interesting comparison, since it also makes use of secondary structure element alignment. The results are combined with PDB-BLAST and enzyme code similarity by training a two-layer neural net for weighing the three contributions.
- **PDB-BLAST (PB)**. From [Bindewald et al., 2003] we quote their results for the PDB-BLAST method [Rychlewski et al., 2000] which generates PSI-BLAST profiles [Altschul et al., 1997] for each target and then aligns them to all template sequences.
- **GenTHREADER (GT)**. From [McGuffin and Jones, 2002] we used the results for GenTHREADER, an approach introduced by [Jones, 1999a] which uses a sequence profile-based algorithm and subsequently analyzes the alignments by using energy potentials.
- **BAYESPROT (BP)**. BAYESPROT utilizes tree-augmented naïve Bayesian classifiers. Here, we quote the results from [Chinnasamy et al., 2004].
- **Ding and Dubchak (DD)**. Ding and Dubchak studied support vector machines and neural nets for fold recognition. The results are quoted from the original paper of 2001 [Ding and Dubchak, 2001].

Since these results were not recomputed, it should be noted that there are small differences in the setup between our approach and the quoted methods. We use Psipred [Jones, 1999b] predictions while, for the Ding and Dubchak set, MANIFOLD makes use of consensus secondary structure predictions as described by [Albrecht et al., 2003]. Furthermore, since we made use of an NR version of April 2004 to compute our profiles, these will differ slightly from the profiles generated by Bindewald et al. for MANIFOLD. The final revision of their paper was in August 2003.

3.3 Preselection of Fold Classes

The first question that arises is how to select suitable fold classes from the template data. Intuitively, we can describe this problem as

Preselection of Fold Classes. Given template protein domains in n_{tot} classes, select a fraction of n classes out of n_{tot} such that the number of necessary templates is reduced significantly while keeping the correct class as often as possible within the selection of potential fold classes.

Based on the working hypothesis, we make use of the secondary structure elements contained in the target structure to find out which protein domains are similar to our target and which are not.

3.3.1 Secondary Structure Element Alignment (SSEA)

One alignment method based on secondary structure elements which is very well suited for this task is the so-called *secondary structure element alignment (SSEA)*. This method has been shown to compare favorably against direct secondary structure alignment methods [McGuffin et al., 2001]. Here, two proteins are represented as the sequences of their secondary structure elements and then aligned using dynamic programming based on the types and the lengths of the elements (see Algorithm 1). This matches our idea, as the topology of a protein is assumed to be related to the sequence of its elements, and thus using SSEA gives us a measure of supposed topological similarity between two proteins (or protein domains in this evaluation). Since SSEA is fast, in order to select potential fold classes, we can align a target against all available templates, assign the score of the highest scoring template in a fold class to the respective class and order the template fold classes respectively.

Whether to use Psipred or DSSP on the Template Side

On the target side, we can only make use of secondary structure predictions (in our case generated by Psipred), but on the template side, we have the option of using either direct secondary structure annotations (derived from a protein structure itself) obtained from DSSP [Kabsch and Sander, 1983], for instance, or secondary structure predictions. On the ASTRAL25 data, we evaluated which version would achieve higher accuracy.

We find that, if we use DSSP annotations on the template side and Psipred predictions on the target side, our prediction accuracy (when using only the top hit for each target) drops to 45% as compared to 54% for Psipred vs. Psipred. This effect can be attributed to the differences between these two methods, i.e. using the same method on both sides (target and template) allows finding similar templates, even if the predicted secondary structure for the template is not necessarily as correct as possible. In other words, as we make similar mistakes on both sides, it is beneficial to also use predictions on the

template side. Using DSSP on the template side and Psipred on the target side instead simply leads to larger differences and, in some cases, clearly different secondary structure element content between a target and suitable templates, such that their similarity cannot be recognized by SSEA.

Although this is not possible in a real-world prediction setup, we also compared to using DSSP on both sides, which results in an accuracy of 60% (i.e. six percentage points more than Psipred on both sides). This indicates that better secondary structure predictions, i.e. more accurate in terms of structural properties, also seem to result in better prediction accuracy for SSEA.

3.3.2 Selection Strategies based on SSEA

Approach 1: Relative Number of Folds

Once all potential fold classes have been assigned a score using SSEA, it is necessary to discard most of them in order to achieve the desired speed-up of the subsequent profile-profile alignment step. Simply using a fixed number of classes is not advisable, as the number of available classes can vary significantly depending on the setup, i.e. on the available template database (see e.g. the difference between Ding and Dubchak’s and the ASTRAL set). For this evaluation, we chose the next simple solution in selecting the top $n\%$ of available classes instead, i.e. choosing the number of selected classes relatively to the number of classes in the template database.

We therefore evaluated the number of times the correct fold was found within the top $n\%$ and the average number of templates needed on the ASTRAL dataset for increasing n (see Fig. 3.2). The final value of $n = 5$, i.e. selecting the top 5% of fold classes for further processing, was chosen as a reasonable tradeoff between speed and accuracy: We computed the relative accuracy gain as the gain in accuracy divided by the increase in the average number of templates per step for $n = [1 : 9]$ with a step size of 1. As expected, the relative accuracy gain falls with increasing n ; however, we find a local maximum for $n = 5$ (see Fig. 3.2, lower panel), which we decided to use as our threshold. Further, the stepwise accuracy gains after $n = 5$ are below two percent. For databases containing only few different fold classes, we defined a minimum number of selected fold classes of 5. On ASTRAL25, selecting the top 5% of the classes contains the correct class in 88% of the cases while discarding 95% of all available template classes.

Interestingly, the cases where this preselection approach misses the correct fold cannot be mapped clearly to certain features of the corresponding targets: These targets lie in 239 fold classes (which range in size from 2 members to 175 members with respect to this dataset), they range in length from 28 amino acids to 740 amino acids, and, as shown in section 3.5.1, they are not restricted to only few secondary structure elements.

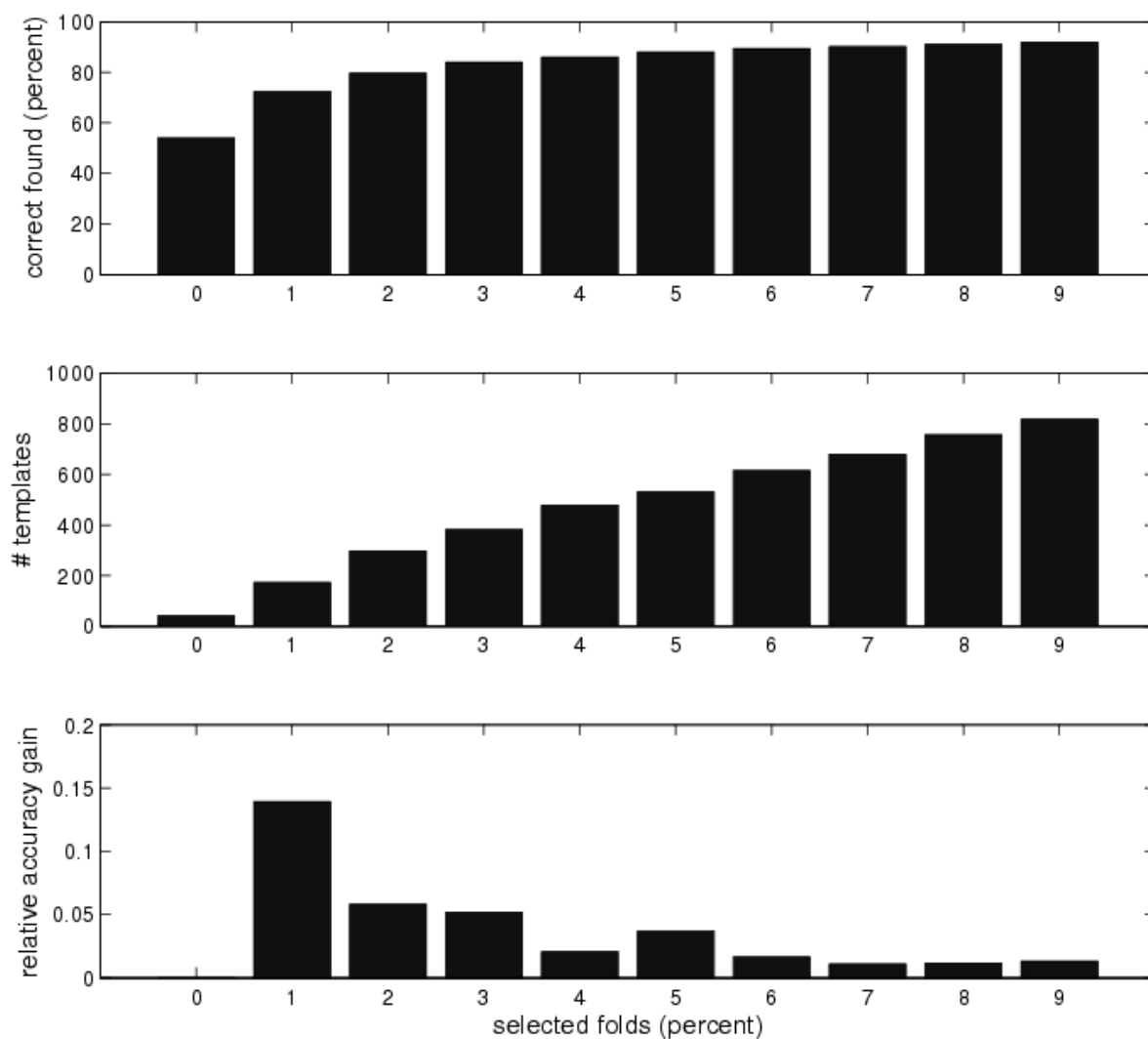


Figure 3.2: Evaluation of fold selection for the first approach: On the x-axis we find the relative fraction of selected fold classes, on the y-axis (1) the number of cases where the correct fold class is within the selected classes is shown (upper panel), (2) the average number of templates out of all 3999 templates in the ASTRAL25 set we have to align the target to (middle panel), and (3) the relative accuracy gain per step (the accuracy gain for a step divided by the increase in the number of templates for the same step). A local maximum in the otherwise more or less monotonically decreasing relative accuracy gain found at $n = 5$, i.e. selecting 5% of the folds, was chosen as the threshold for our method.

Approach 2: Threshold-Based Preselection

Alternatively, it is also possible to select those fold classes which have a member whose score difference to the top-scoring template overall is less than a threshold t . We computed the number of times the correct fold was found, including a fold whenever the top-scoring member of the fold was within a certain score distance to the top hit for a target. We evaluated different thresholds and chose one for which we found an identical performance to our first preselection approach (the relative number of folds) with respect to accuracy: using all folds with members having a score distance of less than this threshold (0.09417) to the top hit, we also find the correct fold in our selection for 88% of the targets.

As for the previous approach, we find the wrong predictions in 224 folds which contain 2 to 175 members, and the corresponding targets also vary strongly in lengths (from 24 residues to 740 residues). For this approach, we find that indeed those targets with few secondary structure elements are more difficult to predict than targets with many secondary structure elements (see 3.5.1).

3.4 Refinement with Profile-Profile Alignment

Given a number of preselected classes, the second step deals with the problem of refining the selection to finally predict only one fold class for the target:

Refinement: Selection of the Final Fold Class. Given m_{sel} template protein domains in n classes, select one class for the final prediction with as high accuracy as possible.

For finding the final prediction, we employ log average profile-profile alignment (PPA) on the templates of the preselected fold classes. This approach is computationally much more expensive than SSEA (the sequences are longer, as PPA aligns residue-wise, and at each position profile vectors have to be evaluated instead of single elements). While in our previous study [Gewehr et al., 2004] we made use only of sequence profiles, for the final evaluation presented in this chapter we selected a more recent and also more powerful PPA, namely the log average profile-profile alignment method used by the authors of the Arby structure prediction server [von Öhsen et al., 2004], which uses global alignment on both sequence and secondary structure profiles. In a comparison with PPA variants using sequence profiles only, we could confirm that Arby outperforms all previously used versions on our ASTRAL 25 dataset. The parameters for the Arby alignment approach have been optimized by its authors independently and were used without modification in our evaluation for this chapter (see section 2.5.2).

3.5 Results

As the aim of our study is to speed-up fold recognition, in the following we will evaluate different methods including our own with respect to *fold recognition accuracy*, i.e. the

number of correct predictions divided by the size of the test set, given as percentage. Further, we will have a closer look at the *preselection accuracy*, i.e. the number of targets for which the correct fold was included by a preselection approach divided by the number of all targets in percent.

3.5.1 Preselection Performance on ASTRAL25

For comparison, we configured both preselection approaches (relative number of fold classes and threshold-based selection) such that they find the correct fold in their selection in exactly the same number of cases (see above): When used as described above, both find the correct fold in their selection for 88% of all targets, whereas the first approach uses a fixed number of 22 folds and the second approach visits 24 folds on average.

We evaluated whether it was necessary to introduce a "special treatment" for targets with few predicted elements, which we expected to be harder to predict than those with more elements. For this reason, we exemplarily selected all targets with only one or two secondary structure elements (excluding coils) from our ASTRAL25 set. On this data, the first approach (using the top 5% of fold classes) still contains the correct fold in 85% of the cases (as compared to 88% for all targets). In contrast, the threshold-based version shows a clearly reduced performance, containing the correct class in only 51% in its selection. This shows that especially the first approach is applicable also for targets with few secondary structure elements.

For both approaches, we observe that with increasing number of secondary structure elements also the preselection accuracy increases: Using only targets with more than 20 secondary structure elements, the first approach selects the correct fold in 91% of the cases and the second approach nearly reaches 99%.

Overall, although they were tuned to the same preselection accuracy on all targets, the relative number of folds works much better on few secondary structure elements than the threshold, and the threshold is better for very high numbers of secondary structure elements. When used in combination, i.e. using at least 22 folds and running until the threshold is reached, it is possible to capture the good parts of both approaches. Then, in 91.5% of all cases we find the correct fold in our selection. However, using an average of 34 folds, this combination is actually comparable to the first approach alone when simply using the top 8% instead of the top 5% of folds. And indeed, for the top 8% of folds, we would have achieved a very similar preselection accuracy of also about 91%.

Apparently, there is a tradeoff between fold recognition accuracy and speed-up. Using the individual approaches or the combination of the two, an increased number of folds or a less restrictive threshold will increase the preselection accuracy but in turn include more potential folds. On the other hand, as we will see on the CATH_MJ set in the next subsection, the restriction to only few fold classes by SSEA can in some cases even improve accuracy over PPA alone. In addition, the threshold-based preselection depends much stronger on the properties of a prediction setup (the expected sequence similarity between the template classes, for instance) than the first approach: When trained on a set with low similarity between template classes and then used on a set with high sequence similarities

between template classes (and thus smaller score differences), the threshold will probably find many more folds than expected from the training data, and vice versa. In contrast, the relative number of folds can be expected to yield a speed-up on most datasets independently of the contained sequence similarities, as long as it does not happen that a very large part of the templates is concentrated in just a few of the available template classes. This illustrates that, as we have seen, the application of SSEA can help concentrating on potential fold classes in fold recognition setups, but it will depend on the intended application how to choose the approach and the corresponding parameters.

3.5.2 Fold Recognition Accuracy

In this subsection, we combine preselection with subsequent refinement using PPA for fold recognition. In direct comparison, the characteristics of the first approach seem better suited for this purpose than those of the second, as it does not depend on the number of secondary structure elements to work well, whereas the threshold-based version has considerable problems in the presence of only few secondary structure elements. Further, the combination of both approaches increases the number of folds over the first approach by more than 50% while only resulting in a few percent better preselection accuracy. In the following, we therefore use the relative number of folds as defined by the first approach as an exemplary choice of preselection method for the purpose of fold recognition: We select the top 5% of fold classes with SSEA, and we subsequently apply PPA to predict a single, final fold class for a target.

The fold recognition accuracy for this approach as well as our comparison methods on the two benchmark sets and on the training set is shown in Figure 3.3. All values were rounded to full percentages. The difficulty level of the benchmark sets decreases from left to right as indicated by the accuracy of the methods for each set.

- **CATH_MJ:** For the most difficult set we find that sequence based methods perform poorly (PDB-BLAST: 13%, GenThreader: 14%). Secondary structure element alignment achieves 32% accuracy and PPA achieves 38% accuracy. Nonetheless, on this set, the combination with SSEA can further increase prediction accuracy to 41%, in comparison to 34% for MANIFOLD [Bindewald et al., 2003]. The reason for this improvement is that, when only very low sequence similarities to sequences of the same fold are given (as in this set), PPA finds only very low scores against all templates. Then it is possible for unrelated templates to gain a slightly higher PPA score than a remotely related template by accident, for instance because of a few similar residues, although the overall topology may be completely different. On this set, for some cases, the restriction of the available folds by preselection prevented PPA from running into such traps. In such difficult situations, confidence measures such as score gaps [Sommer et al., 2002] may be used to abstain from a prediction completely and apply other methods instead when available. However, for this test set, the score differences between the first ranked and the second ranked fold are usually small, and therefore such an approach might significantly reduce PPA's sensitivity.

- **ASTRAL25:** Here, with only 54% accuracy, secondary structure element alignment achieves significantly less hits than PPA with 79%. The combination of both yields 76%, this time decreasing accuracy by about 3%.
- **SCOP_DD:** On the easier benchmark set containing distant homologs we find that our approach achieves 83% accuracy as compared to 75% for MANIFOLD, achieving 24% more fold recognition accuracy than the recently published BAYESPROT and even 27% more than the machine learning methods proposed by Ding and Dubchak. Again the best result is obtained by PPA alone with 84%, whereas secondary structure element alignment achieves 73%.

We find that, by speeding up the fold recognition process using preselection, we can obtain a similar performance to using PPA directly (CATH_MJ: +3%, ASTRAL25: -3%, SCOP_DD: -1%). On all three sets, both PPA and the combination of preselection and PPA clearly outperform their comparison methods.

3.5.3 Speed-Up Evaluation

In a runtime evaluation of the used implementations on an Intel Xeon DP with 2.8 Ghz, SSEA was more than a hundred times faster than PPA, with up to between 10^3 and 10^4 alignments per second as compared to 10 to 100 alignments per second for global PPA with both secondary structure and sequence profiles in our setup. This shows that SSEA is faster than PPA by about two orders of magnitude. Therefore, the speed-up achieved by preselection can indeed be considered relative to the number of discarded templates.

When using the top 5% of folds, under the assumption that we discard about 95% of the templates by discarding 95% of the fold classes, we therefore can expect a speed-up of 95% (20-fold). In fact, the real speedup depends on the selected classes. For the ASTRAL25 dataset, the average number of templates per fold class is about 9, whereas the maximum number is 175. Interestingly, the median is 4, and the distribution shows that only about 100 (i.e. about 25%) of the fold classes actually have more members than 9 in our set. Nonetheless, if we align against each template of the selected fold class, this distribution results in a true speedup as measured by the number of templates for each target of about 87%, i.e. 8-fold, as we have to use PPA against 532 templates on average instead of all 3998 of the ASTRAL set.

3.6 Discussion

We have introduced a simple way of combining two powerful alignment methods for fold recognition, namely profile-profile alignment and secondary structure element alignment (SSEA). We select potential fold classes according to their potential secondary structure topology and then rescore these classes using profile-profile alignment (PPA).

For an exemplary fold recognition setup, we used a strategy that selects the top 5% of available fold classes in the template data as ranked by SSEA scores descendingly. Direct

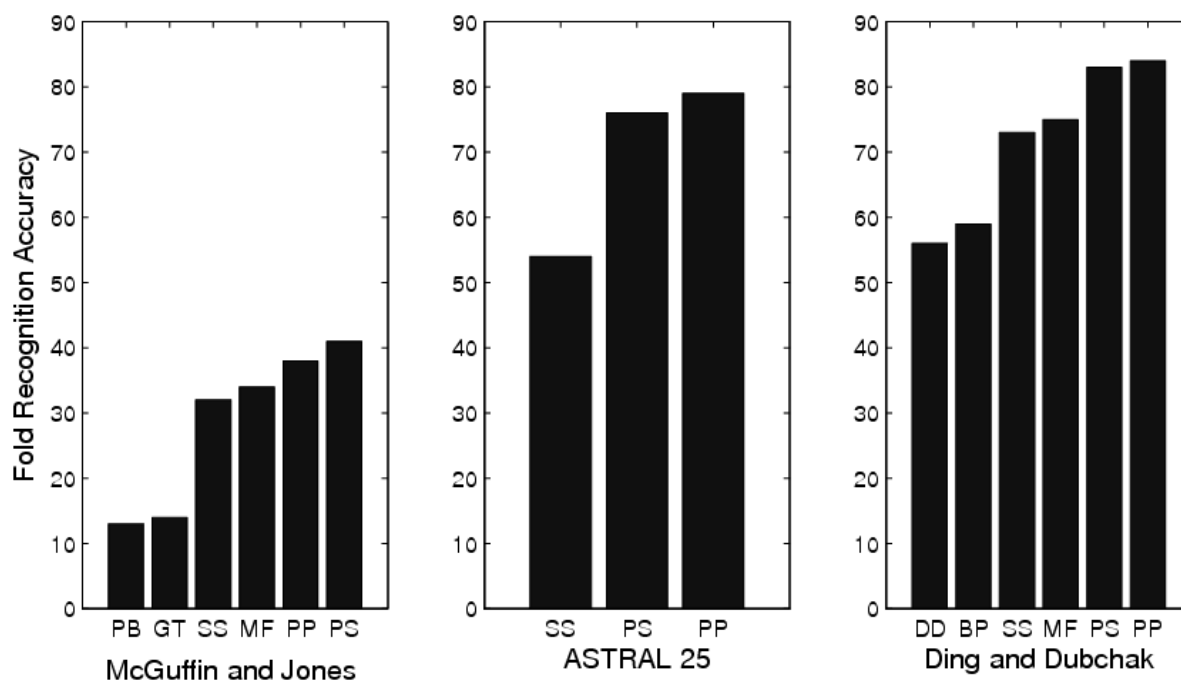


Figure 3.3: Fold recognition accuracy on three benchmark sets. Method labels are PS (the proposed combination of Preselection and PPA), PP (profile-profile alignment using both sequence and secondary structure profiles), BP (BAYESPROT), DD (Ding and Dubchak), GT (GenThreader), MF (MANIFOLD), PB (PDB-BLAST), and SS (secondary structure element alignment). The values for PB and MF were obtained from Bindewald et al. [Bindewald et al., 2003], the value for BP from Chinnasamy et al. [Chinnasamy et al., 2004], the value for DD from Ding and Dubchak [Ding and Dubchak, 2001], and the value of GT from McGuffin et al. [McGuffin and Jones, 2002]. For MF only the mean values were shown.

comparison to other fold recognition methods confirms that this approach is well capable of predicting the fold class of a target: On two well-known benchmark sets obtained from the literature both PPA alone as well as the sped-up combination with SSEA outperform published results of other methods. Especially interesting is the improvement of accuracy for difficult targets, i.e. targets for which we do not find close homologs in the template set, as it is the case for the McGuffin and Jones dataset. This shows that SSEA can actually guide PPA towards topologically similar templates and may thus be considered a useful advisor in the absence of significant sequence similarity, i.e. for more difficult targets in fold recognition setups. Overall, on all three benchmark sets, our combination of preselection with PPA achieves comparable performance to PPA alone, winning a few percent accuracy on one set and losing a few percent accuracy on another.

In comparison to using PPA alone, we have shown that a significant speed-up could be achieved by using SSEA to select fold classes before using PPA, as intended. For our example approach, this speedup could be estimated to be about 95% (i.e. 20-fold) under the assumption of an equal distribution of the number of templates over all template fold classes: The SSEA procedure is very fast and we discard 95% of the fold classes in our template data. On our training data (the ASTRAL 25 set), where such a distribution is not given, we can still realize a speedup of 87% (i.e. 8-fold).

To conclude, our proposed speed-up for fold recognition works well, and it is possible to tune the tradeoff between speed and accuracy by choosing the parameters as required by a particular application. Variants of our preselection idea were successfully integrated in the methods described in chapters 5 (SSEP-Domain) and 7 (Vorolign), in the first case again in combination with PPA, and in the second case in combination with structural alignments. This made it possible for the corresponding servers to provide accurate predictions in reasonable time, i.e. in minutes instead of hours.

Chapter 4

AutoSCOP: Unique Mapping of Patterns to SCOP Classifications and Application to Fold Recognition

In the previous chapter, we have described an alignment-based approach to the fold recognition problem, which can generally be defined as the task of predicting the correct fold of a new protein sequence and, of course, whether the fold is a so-called new fold and has therefore not been classified yet. Especially the latter case is interesting and would require an additional threshold or any other rejection criterion when using an alignment method such as profile-profile alignment, for instance. Further, domain hierarchies like SCOP do not stop on fold level but also make classifications on finer levels such as superfamilies or families. Therefore, we define the problems of *family recognition* and *superfamily recognition* analogously as the problem of assigning the correct family/superfamily to a target sequence.

In this chapter, we describe a new approach to fold, superfamily and family recognition of protein domain sequences (AutoSCOP), which makes use of the available motif and HMM databases for protein sequence annotation. In particular, we map patterns (i.e. hits from motif, profile and HMM searches) to SCOP classifications and then use this mapping to predict the SCOP classification of a target domain sequence. Thereby, the aim of AutoSCOP is to combine the ability to make predictions also for interesting targets (i.e. targets with relatively low sequence identity) with a reasonably high specificity, in order to be not only applicable as a standalone method but also for finding and correcting potential errors of other methods in combined approaches.

Since recognition problems inherently have to find similarities between template and target instances, databases based on sequence patterns such as those contained in the InterPro [Mulder et al., 2003] collection provide a wealth of knowledge on significant regions in amino acid sequences that have been defined for a lot of different applications. As our evaluations show, including this knowledge can clearly contribute to the prediction of SCOP classifications both being applied individually and in combination with well-known alignment-based methods.

AutoSCOP is joint work with Volker Hintermair, who performed initial evaluations of unique patterns in his bachelor's thesis supervised by the author. The description of the methods and the evaluations presented in this chapter are based on our paper on AutoSCOP which appeared in *Bioinformatics* in 2007 [Gewehr et al., 2007a]. In addition, we describe the AutoPSI database [Birzele et al., 2008] of predicted SCOP classifications for PDB and UniProt [Bairoch et al., 2005] entries, which is joint work with Fabian Birzele. This database makes use of both AutoSCOP and the Vorolign structural alignment method (see chapter 7).

4.1 Introduction

The method proposed in the following (AutoSCOP) is a straight-forward approach for SCOP classification prediction (or simply *SCOP prediction*) of protein domain sequences. However, the aim of this chapter is not only to come up with a good new standalone predictor but instead with a method that can be combined with already existing, well-performing methods for SCOP prediction. We aim at building a method that is highly specific and is at the same time able to make predictions for non-trivial cases, i.e. cases with low sequence identity to the available template sequences, for instance. One possible application for AutoSCOP is therefore to be used as a filter before applying other methods, i.e. all highly confident predictions are caught and all others are passed on for further processing.

Our data source are sequence patterns as provided by various databases. The AutoSCOP approach allows for the integration of this data into a single SCOP prediction framework. For an exemplary evaluation, InterPro [Mulder et al., 2003] provides us with a collection of useful databases including Pfam [Bateman et al., 2004] and SUPERFAMILY [Gough and Chothia, 2002]. Though e.g. SUPERFAMILY uses structure information explicitly for generating libraries of hidden Markov models (HMMs), on the target or query side we only make use of the sequence and do not need the corresponding structure.

Our approach can be used for any collection of pattern or feature databases, with the InterPro compendium being a convenient example for such a collection which was already applied by other approaches with different prediction aims. We use InterPro patterns in our protein domain prediction method SSEP-Domain ([Gewehr and Zimmer, 2006], see chapter 5). InterPro has further proven to be a valuable resource for EC number prediction using association rule mining [Chiu et al., 2006]. [Artamonova et al., 2005] have evaluated and successfully used association rules to improve sequence annotation which includes InterPro patterns among other data. In a recent study [Brezellec et al., 2006], the mapping of Pfam annotations to organism-specific proteins was found to be useful for the identification of certain genes with potential link to DNA maintenance. Especially interesting for SCOP predictions is the mapping between SCOP families and Pfam patterns as investigated by [Zhang et al., 2005]. This mapping showed that there is a general agreement between these databases, but there are still areas of disagreement as well as unmapped SCOP domains.

Given the latter result, it is obvious that Pfam patterns can be used for SCOP pre-

diction, but it is necessary to discard the disagreeing pattern occurrences and fill the gaps resulting from the unmapped domains with patterns from further data sources. Our approach, which is based on what we call unique mappings from patterns to SCOP classifications, exploits this idea with respect to highly specific SCOP prediction using multiple databases: For maximizing specificity, we introduce a strict criterion for the acceptance of a mapping between a pattern occurrence and a SCOP classification that allows us to discard all mappings that do not clearly match the SCOP hierarchy. We assign a pattern to a SCOP superfamily, for instance, whenever this pattern occurs only in members of this superfamily and nowhere else. Such patterns we call *unique patterns*. For these mappings, increasing the number of included databases simultaneously increases the coverage on the training data: The number of training sequences we could assign a SCOP class to using our mappings rises from 64.7% for Pfam alone to 86.2% for all InterPro member databases on family level. On superfamily and fold levels, we achieve a coverage of 99%.

The assignment of patterns to SCOP classifications was trained on the ASTRAL compendium [Chandonia et al., 2004]. The predictive power was evaluated in a blind-test like scenario using three different sets: (1) the complete difference set between two ASTRAL versions (which contains many "easy" predictions due to high sequence identities), (2) a more difficult set with low sequence identities which was used for structure alignment evaluation by [Birzele et al., 2007], and (3) the CAFASP4 targets. We made use of an InterPro version that was released before the ASTRAL domains we used in our test set, such that the contained HMMs, profiles and regular expressions could not have been trained on the SCOP classifications used for testing.

We evaluated the power of our method when applied as a filter by combining it with log average profile-profile alignment (PPA, [von Öhsen et al., 2003]). The combination was tested on the second, more difficult data set. Here, although we do not make use of the target structure, we could achieve results that are comparable even to structure alignment methods. Further, we observe an improvement over the best structure-based method on this set (Vorolign, [Birzele et al., 2007]) when we combine Vorolign with our method, similarly to the combination with PPA. On the third set, the CAFASP4 targets, we find that we can contribute SCOP predictions for about half of the targets with classifications available in the latest SCOP release.

Albeit being simple, our unique patterns are a quite powerful tool for SCOP prediction. The inclusion of unique pattern combinations does not significantly improve performance over unique patterns alone but helps a bit on family level. A possible reason for this is the high co-occurrence of patterns from different databases: for instance, a test domain may be classified correctly by patterns from different databases simultaneously. The extensibility of AutoSCOP was demonstrated by including also ASTRAL HMMs trained on SCOP families into our approach, which increased sensitivity on family level on the complete difference set significantly (already without pattern combinations).

We provide a web server for AutoSCOP where users can submit their sequences and obtain SCOP predictions. In addition, precomputed SCOP predictions for PDB and UniProt are available from the AutoPSI database.

4.2 Material

4.2.1 InterPro and its Member Databases: HMMs, Profiles and Regular Expressions for Protein Sequence Annotation

InterPro is a compendium of databases which include sequence patterns that range from e.g. functionally important motifs as stored in PROSITE up to structural domains trained on SCOP superfamily definitions (SUPERFAMILY). InterPro annotations can be found in many protein resources, as they give hints on e.g. the evolutionary or functional context of areas on amino acid sequences. In particular, the InterPro version used for this evaluation (v7.2) contains the following databases:

- **Pfam** [Bateman et al., 2004]: In Pfam, protein domain families have been represented as multiple alignments (one for each family). Then, profile hidden Markov models have been built from these alignments using the HMMer¹ software by S.R. Eddy. Alignment of a query sequence against the database of families is again done with HMMer.
- **PIR Superfamily** [Wu et al., 2004]: In PIRSF (or PIR Superfamily), proteins are classified by their evolutionary relationships and combined in HMMs.
- **PRINTS** [Attwood, 2002]: PRINTS contains so-called protein family "fingerprints", i.e. motifs which are used in combination to detect members of protein superfamilies. The search is done by FingerPRINTScan [Scordis et al., 1999].
- **ProDom** [Bru et al., 2005]: For ProDom, which contains protein domain families, InterProScan uses BlastProDom.pl (by Florence Servant, flo@ebi.ac.uk) based on BLAST [Altschul et al., 1990] to scan target sequences for these families.
- **PROSITE** [Hulo et al., 2004]: PROSITE contains both regular expressions for significant amino acid patterns, which are searched for by ScanRegExp (by W. Fleischmann, Wolfgang.Fleischmann@ebi.ac.uk) and Ppsearch (Fuchs, R. 1994), and profiles for protein families with higher sequence divergence, which are searched with pfsan from the Pftools package (by Philipp.Bucher@isrec.unil.ch).
- **SMART** [Letunic et al., 2004]: SMART (a Simple Modular Architecture Research Tool) annotates genetically mobile domains. The corresponding HMMs were built on manually optimized alignments.
- **SUPERFAMILY** [Gough and Chothia, 2002]: The SUPERFAMILY database represents SCOP superfamilies by groups of HMMs.
- **TIGRFams** [Haft et al., 2003]: TIGRFAMS contains HMMs for curated multiple alignments of protein families.

¹<http://hmmer.janelia.org>

We used the InterProScan program [Quevillon et al., 2005] against the InterPro 7.2 databases for searching InterPro patterns on the amino acid sequences in our training and test data.

4.2.2 ASTRAL Asteroids and Family HMMs

Between SCOP releases, the ASTRAL team provides predicted domains from PDB chains since the latest ASTRAL release (Asteroids), which are updated on a weekly basis. The prediction process makes use of BLAST against ASTRAL, HMMs for SCOP families and superfamilies and HMMs from the Pfam-A database. We used Asteroids as a comparison and also included ASTRAL’s family HMMs in our prediction method.

4.2.3 Training Data

For computing pattern mappings, it is necessary to define a training set based on SCOP which is as complete as possible, in order to find as many pattern matches as possible. All patterns that are not found in the training data cannot be used for prediction as they cannot be assigned to a SCOP class. Therefore, we chose the ASTRAL compendium based on SCOP 1.65 [Chandonia et al., 2004]. We make use of the atom-based entries as provided by the corresponding sequence file, which can be obtained at <http://astral.berkeley.edu>. This set contains 50979 domains as defined by the SCOP database after exclusion of so-called genetic domains (which are defined to be comprised of parts from different protein chains).

4.2.4 Test Data

For testing the predictive power of the AutoSCOP approach, three test sets are used, each of which gives a hint on the behavior of our method in a particular setup:

1. **Complete Difference Set:** We computed the difference set between ASTRAL 1.65 and 1.67 under exclusion of genetic domains. This yields 10039 domains classified in 536 SCOP folds, 804 SCOP superfamilies and 1251 SCOP families. Global sequence alignment against the ASTRAL 95 subset of SCOP 1.65, a representative subset filtered for 95% sequence identity, shows that about 50% of these test domains have more than 95% sequence identity to the training set, i.e. many of the contained targets are easy to predict.

It should be noted that 458 of these folds contain only one superfamily. However, AutoSCOP’s prediction accuracy for those superfamilies belonging to folds with more than one superfamily in the test set was found to be comparable to the overall prediction accuracy on all levels.

2. **Non-Trivial Difference Set:** Since the complete set contains many trivial targets, as a second set, we also used the subset of non-trivial targets. This set was generated as described for the Vorolign evaluation [Birzele et al., 2007]: It contains all domains

which have at most 30% sequence identity and at least 30 identically aligned residues with one of the templates, and which belong to a SCOP family that is represented by at least one template in ASTRAL 1.65. This filter results in 979 domains, which can give a good hint on how AutoSCOP behaves on difficult but still homologous targets. The 979 domains are classified in 129 different folds, 169 different superfamilies, and 208 different families.

3. **CAFASP 4 Set:** In addition, AutoSCOP's fold recognition performance was tested on the 58 targets of the CAFASP 4 community-wide blind test experiment for protein structure prediction, which contain many difficult cases (new families, superfamilies and even folds).

4.3 The AutoSCOP Approach

As stated in the introduction, our aim is to build a filter method, by which we mean a component of a larger fold recognition system that yields highly specific results when possible and abstains otherwise. A similar aim has the SCOPmap approach [Cheek et al., 2004], where a number of alignment methods including both sequence-based and structural alignments are combined for SCOP superfamily prediction of new protein structures. However, for SCOPmap, knowledge of the target structure is essential, whereas we concentrate on a setup where the structure of the target is still unknown. Further, where the SCOPmap approach calibrates the thresholds for the individual methods such that specificity is maximized, we keep standard parameters but exclude patterns that do not match the InterPro hierarchy, as described in the following.

4.3.1 Motivation

It was observed before for the mouse secretome [Grimmond et al., 2003], that some InterPro domains as well as SUPERFAMILY predictions were exclusively found in secretome proteins and that such occurrences might be used as an alternative approach to identifying putative secretome proteins. Another example, the DomainSieve approach [Brezellec et al., 2006] searches for Pfam patterns that occur only in certain organisms. The authors of the PANDORA system [Kaplan et al., 2003], a web-based tool for automatic representation of keyword-based biological knowledge associated with sets of proteins based on graphical analysis, suggest to analyze protein sets as given by GO [Camon et al., 2003] or SCOP by studying shared keywords.

4.3.2 Unique Patterns

In a similar fashion, we assign those patterns that occur in only one subtree of the classification hierarchy (in our training data) with respect to a particular SCOP level as so-called *unique patterns* to the corresponding SCOP subtree. Thus, for instance, a superfamily may

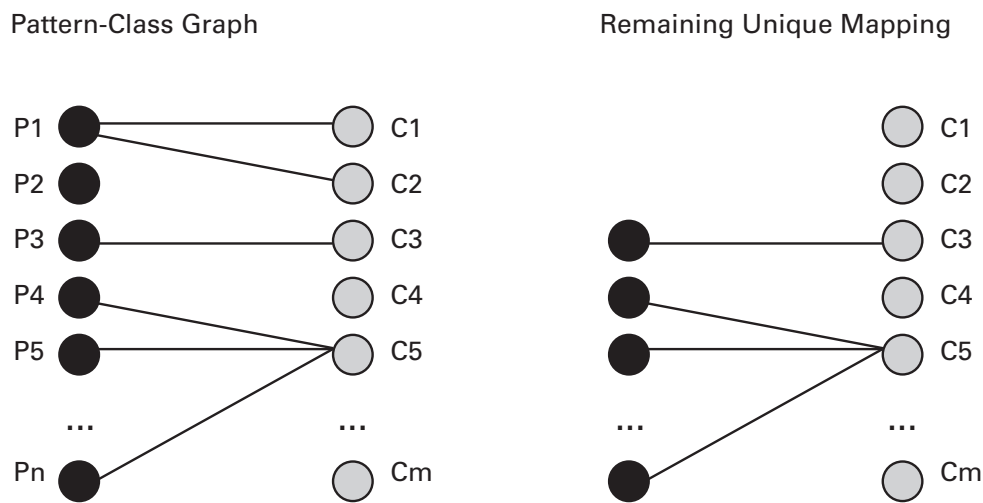
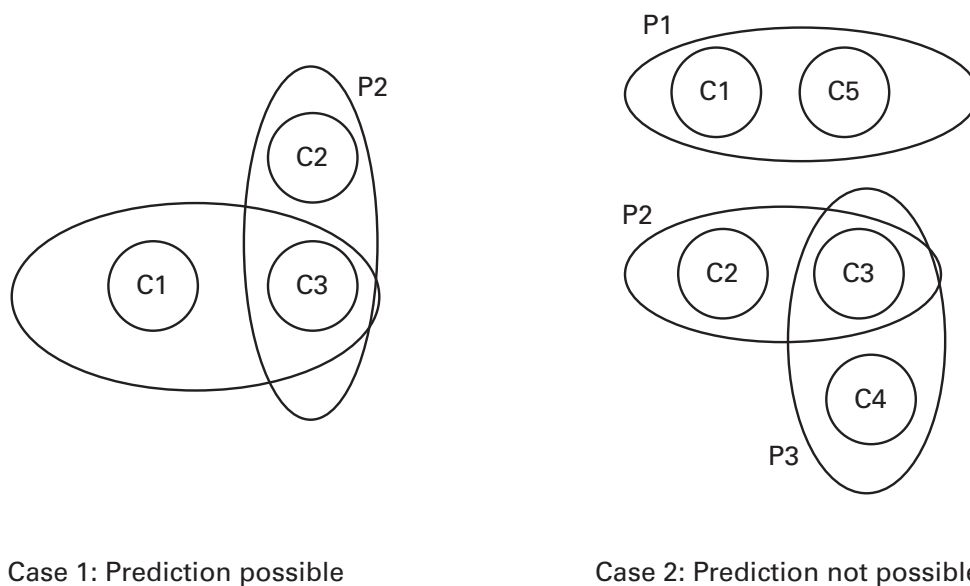


Figure 4.1: Illustration of the concept of unique patterns: Given the Pattern-Class graph for a SCOP level l shown on the left panel, we keep only unique patterns (i.e. patterns P_i with exactly one adjacent edge) for our final mapping of patterns to SCOP classifications. In this example, some patterns are not unique (e.g. P_1 and P_2) and some SCOP classifications have become unpredictable on this level (e.g. C_1 and C_2).



Case 1: Prediction possible

Case 2: Prediction not possible

Figure 4.2: Illustration of the second stage of AutoSCOP: If no unique match is found using unique patterns, the intersection of all possible assignments resulting from common patterns is built. If this intersection contains exactly one possible classification, a prediction is made (case 1), otherwise AutoSCOP abstains, as in case 2, where the intersection for patterns P_1 , P_2 , and P_3 is empty.

be described by a set of unique patterns, each of which covers a subset of the superfamily’s members.

Given a sequence with unknown classification, for prediction we again use InterProScan to detect patterns. We then compare the found patterns to our database of unique patterns. If any unique pattern as defined on the training data is found, we assign the corresponding classification to the sequence. In cases where no unique patterns are found or we find unique patterns with differing SCOP classification assignments, we cannot assign a SCOP classification at this stage.

Let P denote the set of all patterns that have been found in the training data. For a classification task, let C define a set of classes. In particular, for SCOP prediction, let $l \in \{\text{fold, superfamily, family}\}$ denote a SCOP level and C_l denote the set of SCOP classifications on level l , e.g. $C_{\text{fold}} = \{”a.1”, ”a.2”, \dots\}$.

Definition: Pattern-Class Graph. The *pattern-class graph* $G_l = (V_l, E_l)$ for a level l is defined as a bipartite graph using patterns P and classifications C_l as nodes V_l . An edge $e \in E_l \subseteq P \times C_l$ exists between a pattern $p \in P$ and a class $c \in C_l$ iff pattern p occurs in at least one member (i.e. one protein domain sequence) of class c .

Definition: Unique Patterns. A pattern $p \in V_l$ is called a *unique pattern* for a pattern-class graph G_l iff $\text{degree}_{G_l}(p) = 1$, i.e. p has exactly one adjacent edge in G_l . We define

$$P_l^* = \{p \in P | \text{degree}_{G_l}(p) = 1\}$$

as the set of unique patterns on pattern-class graph G_l .

Thus, we obtain functions $f_l^* : P_l^* \mapsto C_l$ that return the corresponding classification for a unique pattern (see Fig. 1 for an illustration). For domain sequences, we can now define prediction functions $f_l : \text{Sequences} \mapsto C_l$ that map a sequence to a SCOP classification on level l if f_l^* maps all unique patterns found on the sequence to the same classification, and we abstain otherwise.

4.3.3 Extension: Pattern Combinations

In an additional step, we also include unique combinations of *common patterns* (patterns that are not unique with respect to the chosen SCOP level), but only if no unique patterns are found. We analyze combinations of common patterns by searching for consensus classifications. Each common pattern occurs in a number of different SCOP classifications on the chosen hierarchy level. If the intersection of the sets of possible classifications for all found common patterns contains exactly one remaining classification assignment, we predict this classification for the target. If the intersection is empty or contains more than one possible classification, we abstain from a prediction (see Figure 4.2).

Database	Detected Patterns	Unique (Fold)	Unique (Superfamily)	Unique (Family)
Pfam	2031	1981	1979	1825
PIRSF	68	68	68	68
PRINTS	755	674	673	649
ProDom	520	500	500	475
PROSITE	1188	1087	1080	999
SMART	419	405	401	351
SUPERFAMILY	1226	1207	1196	873
TIGRFAMs	493	488	488	481
total	6700	6410	6385	5721

Table 4.1: Quantitative analysis of unique InterPro patterns on the training data. The highest value for each column is printed in boldface.

4.3.4 AutoSCOP*: Inclusion of Further Data Sources

In order to show the extensibility of the AutoSCOP approach, in the following we also included predictions made by HMMs trained on SCOP families as provided by ASTRAL for SCOP version 1.65. This is a logical extension because of the relatively low coverage on the family level using InterPro data alone (see Table 4.2). Predictions were made using HMMer 2.3.2 (S.R. Eddy, <http://hmmerr.janelia.org>) against the complete HMM library. For each target, the top hit was used like any InterPro pattern using an e-value threshold of 0.1, which is proposed as a useful cutoff in HMMer’s user’s guide. We will refer to AutoSCOP including ASTRAL’s family HMMs as *AutoSCOP** in the results section.

4.4 Results

4.4.1 Mapping of Training Domains

Table 4.1 shows the number of unique patterns for each individual database in our training data. In fact, most of the patterns (6410 of 6700, 95%) are unique on at least fold level, which is an indicator for the high quality of the database scan results. Here, we can also assess the performance of our pattern-based approach on the training data (see Table 4.2): With unique patterns alone we can correctly assign folds, superfamilies and families to 99.12%, 98.99% and 86.20% of all domains, respectively. While unique combinations of common patterns do not clearly improve coverage on fold and superfamily levels, they can contribute nearly three percent on family level.

Database	Coverage (Fold)	Coverage (Superfamily)	Coverage (Family)
Pfam	85.07%	84.96%	64.70%
PIRSF	1.46%	1.46 %	1.46%
PRINTS	30.80%	30.79%	28.45%
ProDom	23.40%	23.40%	20.99%
PROSITE	43.90%	43.57%	37.43%
SMART	25.19%	25.10%	19.93%
SUPERFAMILY	96.50%	94.60%	30.33%
TIGRFAMs	10.69%	10.69%	10.47%
total	99.12% (99.34%)	98.99% (99.27%)	86.20% (88.97%)

Table 4.2: Coverage analysis on training data based on unique InterPro patterns. The results after inclusion of pattern combinations (AutoSCOP) are shown in brackets. The highest value for each column was highlighted.

Method	Family Sens	Family Spec	Superfam. Sens	Superfam. Spec	Fold Sens	Fold Spec
AutoSCOP	84.20%	96.77%	94.40%	98.17%	93.36%	98.13%
w/o Pfam	74.57%	97.06%	92.77%	97.73%	92.33%	98.11%
w/o PIR	84.19%	96.77%	94.40%	98.17%	93.36%	98.13%
w/o PRINTS	82.96%	96.65%	94.51%	98.17%	93.47%	98.14%
w/o PRODOM	83.60%	96.80%	94.31%	98.16%	93.27%	98.13%
w/o PROSITE	82.78%	97.13%	94.56%	98.19%	93.47%	98.16%
w/o SMART	77.70%	96.52%	94.39%	98.14%	93.35%	98.11%
w/o SUPERFAM.	79.76%	97.17%	84.49%	99.06%	83.26%	99.08%
w/o TIGRFAMs	82.79%	96.64%	94.40%	98.17%	93.36%	98.13%

Table 4.3: Influence of InterPro databases as measured by the prediction performance of AutoSCOP (i.e. our approach on InterPro data) after leaving out individual databases. This analysis shows that many predictions are made by more than one database. As in the coverage analysis, again SUPERFAMILY and Pfam have the highest impact on the final method. The lowest value for each column was highlighted.

4.4.2 Prediction of SCOP 1.67 Domains

In our test set, we find 97 new SCOP folds, 163 new SCOP superfamilies and 326 new SCOP families. On fold level, 433 of the 10039 domains in our test set belong to new folds and are therefore considered as "new" in our framework, i.e. targets for which we do not have a template with the same fold in our database. Accordingly, the remaining 9606 domains are considered "known", i.e. targets for which a correct prediction would be possible. On superfamily level, we have 601 "new" and 9438 "known" domains, and on family level, we have 1167 "new" domains and 8872 "known" domains. We evaluate prediction accuracy on the ASTRAL difference set for each SCOP level by means of

- **sensitivity:** the number of correct predictions on "known" domains divided by the number of all "known" domains, and
- **specificity:** the number of correct predictions divided by the number of all predictions, including wrongly predicted "new" domains.

Individual Contributions of InterPro Databases

It is interesting to see how the individual contributions of the databases differ (Table 4.2). On fold level, the SUPERFAMILY database already covers 96.5% of all domains (the best individual result on fold level). A similar performance can be observed on superfamily level with 94.6%. However, on family level, SUPERFAMILY only achieves about 30% coverage, whereas here Pfam achieves the best result with about 65%. This shows that some patterns are good for certain levels of the SCOP hierarchy (e.g. SUPERFAMILY for fold and superfamily), but none performs best on all SCOP levels.

Table 4.3 shows the performance of AutoSCOP on the test data after leaving out individual databases. As could be expected from the coverage analysis, SUPERFAMILY and Pfam are especially important. Some databases are nearly completely covered by the remaining InterPro members for our purpose. PRINTS and PROSITE are interesting, as these databases increase performance on family level but slightly decrease performance on superfamily and fold level. For our approach we kept all InterPro member databases, but leaving out e.g. the latter two may be an option when the focus lies on higher levels of the SCOP hierarchy.

Comparison with Reference Methods

Table 4.4 shows sensitivity and specificity of our pattern-based predictions on all three evaluated levels of the SCOP hierarchy. For the family level, unique patterns on InterPro data achieve a specificity of more than 96%, for superfamily and fold more than 98%. With respect to sensitivity, this approach performs best on fold and superfamily level, achieving values of more than 93% and 94%, respectively. As expected from the mapping results, the less complete coverage of the family level is reflected in the prediction performance for SCOP families in a sensitivity of only about 80%.

The inclusion of pattern combinations has practically no effect on fold and superfamily predictions, but, on family level, slightly increases sensitivity and slightly decreases specificity. More importantly, the inclusion of ASTRAL Family HMM predictions (AutoSCOP*) rises sensitivity up to 95% on family level even for unique patterns alone and also slightly improves performance on superfamily and fold level while keeping high specificity.

In this evaluation, in addition to the different AutoSCOP variants, we also used a number of reference methods on the same data to compare our results against. These methods were applied in our setup as follows:

- **Asteroids.** Asteroids comes as a FASTA file containing annotated regions on protein chains which are not necessarily identical to the final ASTRAL domains on the corresponding chains (i.e. our test domains). However, ideally a test domain and the corresponding Asteroids region should be identical in sequence. Therefore, for mapping Asteroids regions to domains we aligned all Asteroids sequences against the test domains using BLAST and transferred superfamily assignments (when available) whenever the e-value of a match to a domain on the same chain was below an e-value cutoff that ensures a clear similarity between two regions of the same original amino acid sequence. The chosen threshold ($3e-14$) is the minimum e-value achieved by aligning randomly selected subsequences of length 29 taken from test domains against the test data (which contains the original, full-length sequences). It thus reflects coverage of about 30 residues or more in significant parts of a sequence, which is similar to one of the criteria we applied for filtering the Vorolign test set.

Using this filter, we obtain 86.35% sensitivity and 99.39% specificity on the test data, as shown in Table 4.4. If we relax this criterion, i.e. if we do not use an e-value cutoff, we can actually improve sensitivity to 86.87% while keeping a specificity of 99.39%, as we then allow shorter overlaps and can capture also the few very short regions contained in Asteroids; nonetheless, this difference is small, and in both versions, Asteroids' sensitivity is clearly below all other methods except Pfam in our setup.

- **ASTRAL Family HMMs.** We used HMMer to align the test domains against ASTRAL's family HMMs based on ASTRAL 1.65. We accepted predictions using an e-value of 0.1 as an upper bound, a threshold described as appropriate by HMMer's user's guide.
- **BLAST.** For comparison, we used BLAST in its default parameters and applied an e-value threshold such that the specificity is comparable to AutoSCOP's specificity on fold level. In particular, we transferred the SCOP classifications only of matches below an e-value of 0.2; everything else was considered an abstention.
- **PSI-BLAST.** For PSI-BLAST, we used an NR database of April 2004 together with the ASTRAL 1.65 as training data. We performed five iterations using an inclusion threshold of 0.001 and used the best ASTRAL hit in the last iteration. As for BLAST, we chose an e-value threshold such that the specificity on fold level was comparable to AutoSCOP, namely 7.0, and transferred the SCOP classifications only when a

Method	Family Sens	Family Spec	Superfam. Sens	Superfam. Spec	Fold Sens	Fold Spec
AutoSCOP*	95.25%	96.28%	94.96%	98.07%	93.85%	98.13%
AutoSCOP _U *	95.01%	96.43%	94.95%	98.08%	93.85%	98.13%
AutoSCOP	84.20%	96.77%	94.40%	98.17%	93.36%	98.13%
AutoSCOP _U	80.34%	96.92%	94.39%	98.19%	93.36%	98.13%
Asteroids	-	-	86.35%	99.39%	-	-
Family HMMs	91.45%	96.74%	-	-	-	-
BLAST	93.59%	96.26%	89.65%	98.09%	88.17%	98.19%
Pfam	61.14%	97.67%	78.57%	99.10%	77.40%	99.12%
PSI-BLAST	95.57%	93.06%	94.31%	97.69%	93.07%	98.13%
SUPERFAMILY	27.22%	97.06%	89.31%	97.86%	90.22%	98.20%
<i>Vorolign</i>	<i>95.91%</i>	<i>89.52%</i>	<i>96.60%</i>	<i>95.91%</i>	<i>96.66%</i>	<i>97.68%</i>

Table 4.4: Upper Part: Sensitivity and specificity on different SCOP levels on the complete ASTRAL 1.67-1.65 difference set, for AutoSCOP and AutoSCOP* as well as for both methods restricted to unique patterns only (U). Lower Part: Reference methods. As a reference for the prediction quality of AutoSCOP, we used ASTRAL’s Asteroids (version 1.65-040809) predictions on superfamily level (Asteroids’ most complete level). In total, we were able to find Asteroids regions on the same chain for 97.5% of all test domains using BLAST. Sensitivity for Asteroids was computed relatively to the number of domains coming from known superfamilies in this set. Further, we included the predictions made by the ASTRAL Family HMMs based on ASTRAL 1.65 directly (using an upper e-value bound of 0.1) and computed individual unique pattern results for the two InterPro databases with the highest coverage (SUPERFAMILY and Pfam). We added PSI-BLAST and BLAST predictions with e-value cut-offs that give comparable specificity to AutoSCOP on fold level (7.0 and 0.2, respectively). For PSI-BLAST, we used 5 iterations and an inclusion threshold of 0.001 against an NR database downloaded from the NCBI in April 2004 (which was available at the same time as SCOP 1.65) together with our training set, using the best ASTRAL hit below the threshold in the last iteration as prediction. For BLAST, we directly used our training data as database. In addition, the structure alignment method Vorolign was applied using its default settings. The highest sequence-based value for each column was highlighted.

match was below this threshold. Please note that 7.0 is a rather high threshold, which shows that, on the complete test set, PSI-BLAST does not make many errors at all. This also explains the high sensitivity of PSI-BLAST in Table 4.4.

- **Pfam and SUPERFAMILY.** In addition, we used our unique pattern approach restricted to Pfam or SUPERFAMILY patterns only. In other words, we use a complementary approach to the one shown in Table 4.3 by concentrating on the accuracy when using only one database in AutoSCOP.
- **Vorolign.** Finally, in order to achieve something like an upper bound to what is possible on the data, we used the structure alignment method Vorolign (see chapter 7) in its default setting. As it is not sequence- but structure-based, in the table the corresponding results are written in italics.

Comparison with our reference methods shows that AutoSCOP* achieves the highest sensitivity of all compared sequence-based methods on superfamily and fold level (its sensitivity being only second to the structure alignment method Vorolign). On family level, where sequence similarity is most important, AutoSCOP*, PSI-BLAST and Vorolign are close together and achieve sensitivities of 95.25%, 95.57% and 95.91%, respectively, but with clearly lower specificity for Vorolign and PSI-BLAST as compared to AutoSCOP*. Asteroids and Pfam achieve higher specificity but clearly lower sensitivity.

False Assignments of Domains from new Classifications

Errors often result from the assignment of test domains from "new" classifications to "known" classifications. Using unique InterPro patterns on fold level, we make 170 false assignments, 115 of which are wrongly assigned new fold domains (67.64%). On superfamily level, of the 164 false assignments, 135 fall into this category (82.31%). On family level, of the 226 false assignments, we have 192 targets from new families (84.95%). We correctly abstain from a prediction for 73.44% of the test domains belonging to new folds (318 of 433), 77.53% of the test domains belonging to new superfamilies (466 of 601) and 83.54% of the test domains belonging to new families (975 of 1167).

On superfamily level, we further analyzed the wrong assignments made by unique InterPro patterns from new superfamilies to already existing superfamilies (135): about 70% have corresponding PSI-BLAST hits with e-values less than $1e-5$, and nearly 50% have PSI-BLAST hits with e-values less than $1e-20$. This shows that, as judged by sequence similarity, many of these assignments are reasonable. All errors with clear PSI-BLAST hits could be attributed to changes in the classification or in the domain definition between ASTRAL 1.65 and newer versions.

Prediction Rates using Reduced Training Data

As the computation of InterPro matches for the whole ASTRAL dataset is quite time-consuming, we also evaluated the prediction performance of AutoSCOP* using the ASTRAL25 (which is smaller than the whole training dataset by more than one order of

Training Set	Family Sens	Family Spec	Superfam. Sens	Superfam. Spec	Fold Sens	Fold Spec
Whole Training Set	95.25%	96.28%	94.96%	98.07%	93.85%	98.13%
ASTRAL 95	95.18%	96.28%	94.91%	98.07%	93.80%	98.13%
ASTRAL 25	94.72%	96.06%	94.34%	97.97%	93.24%	98.09%

Table 4.5: AutoSCOP*'s sensitivity and specificity on different SCOP levels on the complete ASTRAL 1.67-1.65 difference set using reduced training data sets. We find that, for the ASTRAL 95 set (filtered for 95% sequence identity), which is less than one fifth of the whole set, there is nearly no loss in sensitivity and specificity. Even for the ASTRAL 25 (filtered for 25% sequence identity), which contains less than one tenth of the original data, we observe a loss of less than one percent in both sensitivity and specificity.

magnitude, containing only 4326 domains with annotated patterns) and the ASTRAL95 (which contains 9386 domains with annotated patterns) as training data. The corresponding prediction rates are given in Table 4.5. We find that the ASTRAL95 with nearly as good results as the whole dataset may be an interesting alternative to using all available data. The ASTRAL25 as a further reduced set still performs well with only about 0.5 percent less sensitivity on all three levels. Nonetheless, for the following evaluations, we make use of the whole training data in order to achieve the best possible performance for our method given our data sources (InterPro and Family HMMs).

4.4.3 Comparison of InterPro Entries and AutoSCOP Mappings

InterPro itself groups individual database patterns together as InterPro entries based on identified sequences in UniProt. Further, it delivers curated information including an abstract on the associated proteins, literature references and links to relevant member databases. Such entries sometimes also contain curated links to structural classifications as additional annotations, but their generation is not aimed at unique SCOP hits.

An example, which illustrates the possible differences between AutoSCOP mappings and annotated InterPro entries, is IPR000191 ("Formamidopyrimidine-DNA glycolase"), where we have 4 patterns (PD003689, PF01149, PF06831, and TIGR00577) and three SCOP links (a.156.1.2, b.113.1.1, and g.39.1.8). With AutoSCOP we can directly assign a.156.1.2 to PF06831 and b.113.1.1 to PF01149, whereas we indeed find PD003680 in all three different SCOP families and therefore do not map this pattern.

Having in mind that many individual patterns often occur together, for AutoSCOP we investigated whether it would be beneficial to use these entries instead of the direct results of the member databases by exchanging the individual pattern identifiers with the InterPro entry identifiers. On the training data, we found that using this variant of the AutoSCOP approach decreases coverage clearly: we lose about 14% on each SCOP level (family: 75.71% instead of 88.97%, superfamily: 85.21% instead of 99.27%, fold: 85.28% instead of 99.34%). On the test data, we obtain similar results. When concentrating on

the IPR entries, we only obtain 70.52% sensitivity and 96.61% specificity on family level, 78.91% sensitivity and 98.62% specificity on superfamily level, and 78.10% sensitivity and 98.61% specificity on fold level. This means that we also lose 14-15% sensitivity on each SCOP level as compared to AutoSCOP (see Table 4.4 for comparison), and therefore using the individual patterns is better for our purpose.

Further, we evaluated whether AutoSCOP can infer information not already contained in the curated InterPro database: If we use those IPR entries with single annotated SCOP links, i.e. those entries that should clearly map to a SCOP classification, we can keep our specificity but clearly reduce sensitivity to 52.56% on family level, 65.13% on superfamily level and 65.04% on fold level, which is reduction compared to AutoSCOP of nearly 30 percentage points. Correspondingly, we observe a further reduced coverage on the training data: 59.34% on family level, 71.23% on superfamily level and 71.97% on fold level. We find that many of these annotations do not agree with our criteria although only a single SCOP link is annotated: Most of the links are to SCOP families. If we directly use these family level links as predictions on the test data, we achieve a sensitivity of 61.47% with a specificity of only 82.03%. Thus, in many cases the links pointed to a wrong SCOP family; and indeed, of these annotations, only 84% are actually unique on this level in our training data.

Our automatically generated mappings are different from the curated IPR entries and much more suited for our purpose. As we could find, InterPro’s curated SCOP links are not necessarily unique with respect to our criteria, even if we concentrate on those InterPro entries which contain exactly one structure link. With respect to predictions, we clearly benefit from the higher resolution we gain by using database patterns individually as well as from the direct, level-specific mapping resulting from our unique pattern approach.

4.4.4 Fold Prediction of CAFASP4 Targets

CASP [Moult, 2005] and CAFASP [Fischer et al., 2003] are community-wide blind test experiments for protein structure prediction. Our domain prediction method SSEP-Domain [Gewehr and Zimmer, 2006] (which includes an InterProScan run on a target protein) participated in CAFASP4 during 2004 [Saini and Fischer, 2005]. We analyzed the InterPro pattern occurrences that were found by SSEP-Domain during this experiment. The databases for this evaluation were chosen such that we make use of data already available before the beginning of CAFASP4, i.e. the predictions were made in a blind-test-like setup.

For 46 of the 58 CAFASP4 targets we can find SCOP annotations in ASTRAL 1.71. For 23 of these targets we can make AutoSCOP predictions on the InterPro data (50%). Of these 23, 21 are completely correct (91.3%), including two two-domain targets for which AutoSCOP finds the correct fold for both domains. One target is a new fold but is wrongly predicted as belonging to a known fold. For the remaining target, two possible folds were found, one of which is correct.

Method	Sensitivity Family	Sensitivity Superfam.	Sensitivity Fold
AS* + PPA	89.0	92.7	96.2
PSI-BLAST + PPA	82.1	89.2	93.2
PPA	81.5	88.3	92.2
Asteroids + PPA	-	88.2	-
AutoSCOP* (AS*)	69.7 (97.0)	85.3 (99.6)	88.6 (99.9)
PSI-BLAST	75.3 (89.6)	79.9 (95.0)	83.1 (98.9)
Asteroids	-	32.3 (99.7)	-
AS* + Vorolign	92.6	95.1	99.3
Vorolign	86.4	92.4	97.7
CE	84.6	91.9	94.1

Table 4.6: Comparison of AutoSCOP* (AS*) with other methods on the Vorolign test set. This set contains 979 non-trivial test domains from "known" families which have less than 30% sequence identity on at least 30 residues with the template set. Values are given in percentages. Methods were ordered by sensitivity on fold level descendingly. Best results for each part are highlighted. For each test sequence, the combination with other methods (AS* + method) was computed by using AutoSCOP* to obtain a prediction first and, if AutoSCOP* abstained, using the prediction of the method using in the combination. Combinations of PSI-BLAST and Asteroids as filters with PPA were computed analogously. For comparison purposes, in the lower part, the results of the two structure alignment methods Vorolign and CE are shown. For some methods, which were able to abstain from a prediction, specificity is shown in brackets. For all other methods, on this set, sensitivity equals specificity, since there are no new families and hence the total number of predictions equals the total number of possible correct assignments. Sensitivity for all combinations, AutoSCOP*, PPA, and PSI-Blast was computed with respect to the whole Vorolign set (979 domains). Results for CE and Vorolign were quoted from [Birzele et al., 2007]. As on the complete difference set, Asteroids' sensitivity was computed on those ASTRAL domains for which we found Asteroids regions on the same chain using BLAST (903 domains).

4.4.5 Performance in the Sequence Twilight Zone

We compared our approach to well-known alignment methods using the test set of non-trivial targets defined in [Birzele et al., 2007]. Vorolign and CE results were also quoted from [Birzele et al., 2007]. For global profile-profile alignment on both sequence and secondary structure profiles (PPA) using the parameters described in section 2.5.2, we aligned the target domains against the ASTRAL 25 compendium (Version 1.65) as a representative template set as described by Birzele et al., but without using Vorolign’s secondary structure element-based filtering. PSI-BLAST hits were computed as described for the complete difference set (Table 4.4). For all alignment methods, the classification of the highest scoring template was used as the predicted classification of the target sequence.

Table 4.6 shows the results. We find that AutoSCOP* performs better on superfamily and fold than on family level. On these SCOP levels, sensitivity is slightly worse than for PPA. When using only InterPro patterns (AutoSCOP), we lose 0.4% on superfamily and fold level and 8.1% on family level as compared to AutoSCOP*. Further, AutoSCOP* achieves specificity rates of 99.9% (fold), 99.6% (superfamily) and 97.0% (family) due to being able to abstain from predictions.

4.4.6 Using AutoSCOP* as a Filter

For evaluation of AutoSCOP*’s ability as a filter, we combine AutoSCOP* with profile-profile alignment as follows: We predict the SCOP classification using AutoSCOP*. For all abstentions, we then align the corresponding targets against the ASTRAL 25 using PPA as described above. The corresponding results are shown in Table 4.6 (*AS* + PPA*).

We find that, in combination, we achieve about 4% improvement over the best individual method on fold and superfamily level, and about 7% on family level. We also find that using AutoSCOP* as a filter clearly increases accuracy over using PSI-BLAST or Asteroids as a filter. Comparison with the results of structure alignment methods on the same test set as an upper bound to accuracy shows that our combination outperforms the well-known CE method [Shindyalov and Bourne, 1998] on all levels and the best structure alignment method in our comparison (Vorolign, [Birzele et al., 2007]) on both superfamily and family level. In this setup, the inclusion of Astral Family HMM predictions only slightly improves accuracy: using AutoSCOP instead of AutoSCOP* in combination with PPA we get 0.6% less accuracy on family level and identical accuracy on superfamily and fold level, as most of the additional predictions are covered by PPA.

Using AutoSCOP* together with Vorolign, e.g. for the purpose of assigning a classification to a newly resolved structure, we achieve a clear improvement over Vorolign alone. In other words, on this set, we can correct some false assignments made by structure alignments. Again, using AutoSCOP instead of AutoSCOP* in combination with Vorolign decreases accuracy only by up to 0.6% (on family level).

Both combinations show that AutoSCOP* indeed works well as a filter. AutoSCOP*’s high specificity apparently still allows for a sensitivity that can improve accuracy in the combinations with well-known methods over using these methods individually.

4.5 The AutoPSI Database - Bridging the Gap between SCOP and PDB and more

With the AutoSCOP method, we have introduced a unique mapping from patterns to SCOP classifications based on InterPro as well as other patterns which can be used for highly-specific SCOP classification prediction of new domain sequences. Using AutoSCOP we can now assign SCOP classifications to unclassified PDB entries by running InterPro on their chains, for instance, and assigning predictions based on the locations of the matched patterns. In addition, given precomputed InterPro data for millions of UniProt sequences [Bairoch et al., 2005] available for download from InterPro, it is possible to assign our annotated structural classifications to regions on these sequences. For UniProt sequences, our unique mappings extend the meta-level InterPro entries which can also contain structural annotations, as we aim at a unique, level-specific and pattern-wise mappings to SCOP: In section 4.4.3, we achieved both clearly higher sensitivity and clearly higher specificity for our approach.

Furthermore, in cases where the structure is known, we can split a structure into domains using PDP or similar tools and then apply Vorolign as described in chapter 7 to assign SCOP classifications based on the similarity found to already classified SCOP domains.

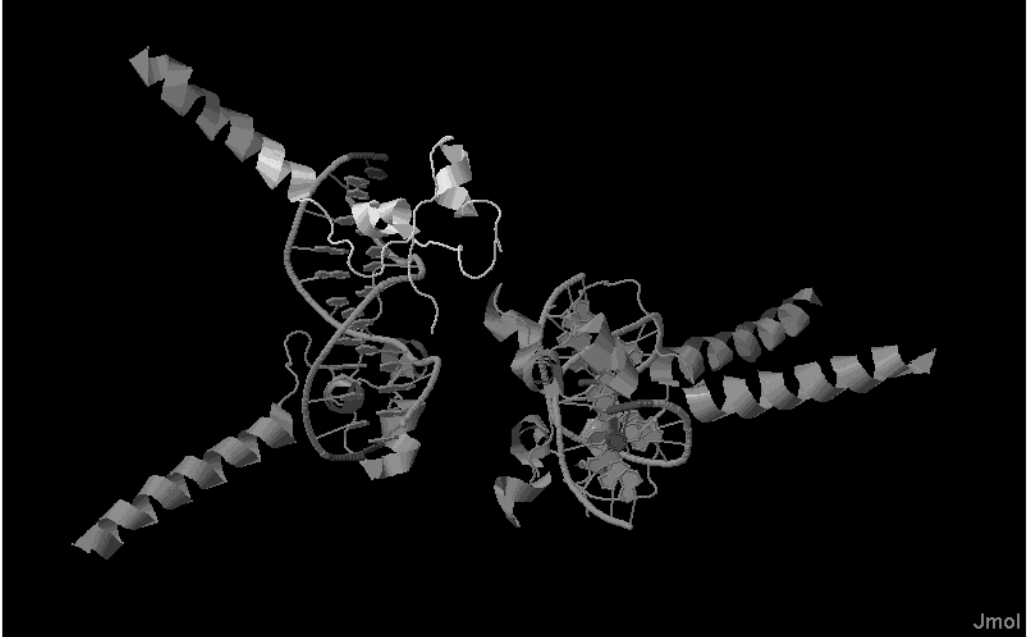
Using both of these tools, in joint work with Fabian Birzele, we developed a database of annotated and predicted SCOP classifications which contains PDB entries as well as UniProt sequences and thus bridges the gap between SCOP and PDB and even extends SCOP's reach to UniProt [Birzele et al., 2008]. Users can search for sequences by keyword and then browse the annotations and predictions. We deliver a simple consensus which gives direct access to the agreeing and disagreeing predictions. Covering large parts of UniProt together with using a new and different approach focusing on high specificity, this database can help to further clarify relationships between proteins in the protein sequence-structure space.

4.5.1 AutoPSI Database Content and Methods

Protein Content: The protein data available in our database is based on PDB and UniProt. In particular, we provide all PDB and UniProt sequences for which we could make SCOP classification predictions or find already existing SCOP annotations using any of our methods.


Pattern Content and AutoSCOP: The AutoSCOP method computes InterPro patterns that are unique for a particular SCOP classification, i.e. that occur only in a particular superfamily, for instance. As training data, for this version of our database, the ASTRAL 1.69 distribution was scanned with InterProScan on the InterPro databases of version 12.1. All PDB sequences in the database were scanned in the same way and the found InterPro patterns and their locations were stored. Therefore, for each PDB sequence, we can show

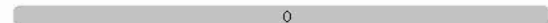
Structure Visualisation





Restrict to chain C Select none

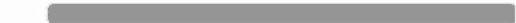
Locations in 2D View


Protein: 


PDP Domains:  0


Consensus (use mouse-over): 


Vorolign: d1pyfa1 (g.38.1.1): 


AutoSCOP: PF00172(g.38.1.1): 

AutoSCOP: PR00054(g.38.1.1): 

AutoSCOP: PS00463(g.38.1.1): 

AutoSCOP: PS50048(g.38.1.1): 

AutoSCOP: SM00066(g.38.1.1): 

AutoSCOP: SSF57701(g.38.1.1): 

Do Not Show Individual Predictions

Click on the SCOP, PDP, Vorolign or AutoSCOP bars to colour the corresponding structure parts in the structure view (if available).

Detailed Information

Chain Description PDP Domains Vorolign Predictions **Unique Patterns**

Unique Patterns

Name	Location	Classification
PF00172	4-46	g.38.1.1
PR00054	5-11 12-18 24-34	g.38.1.1
PS00463	5-36	g.38.1.1
PS50048	5-38	g.38.1.1
SM00066	1-47	g.38.1.1
SSF57701	3-45	g.38.1.1

Click on an item to obtain links to SCOP and the corresponding pattern database.

Figure 4.3: Screenshot of the AutoPSI detail view for 2er8C.

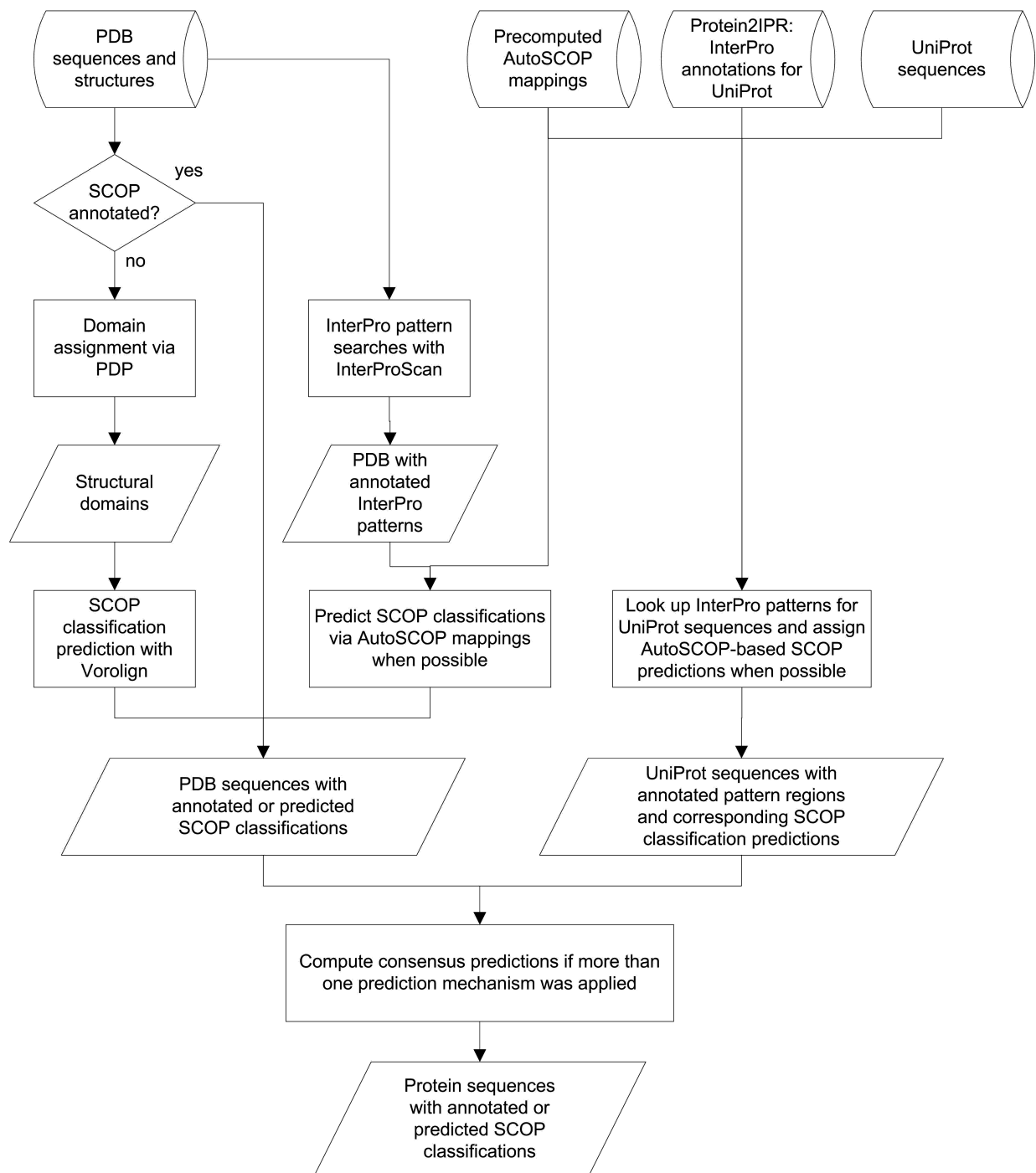


Figure 4.4: Workflow of the annotation process for the AutoPSI database.

the location of the patterns and, based on AutoSCOP, the assigned SCOP classification. For UniProt, we made use of the corresponding Protein2IPR.dat file downloaded from InterPro, which is a precomputed database of pattern occurrences on UniProt sequences. Again, whenever we find a pattern for a UniProt sequence, we can show its location and annotate a classification when such a mapping is possible.

Structures, Domains and Vorolign: In addition, if the structure is known (as it is the case for PDB entries), we annotate the structural domains. If available, we make use of the SCOP definitions directly. If not, we use PDP on the structure to assign potential structural domains and then run the Vorolign structural alignment method for these potential domains against a template database of structural domains (the ASTRAL 1.69), in order to predict the SCOP classification based on structural similarity.

Consensus Predictions: In cases where we have multiple predictions, we compute a simple consensus between the corresponding SCOP classifications (residue-wise by choosing the finest level of agreement between them). As an example, if we have two predictions, namely a.1.1 from position 1 to 200 and a.1.1.2 from position 50 to 150, our consensus yields a prediction of a.1.1 for the regions 1-49 and 151-200, and a prediction of a.1.1.2 for positions 50 to 150. Further, in the user interface, very short regions are not displayed in order to make the consensus parts easier to grasp by visual inspection.

Using the Web Interface of AutoPSI DB

In the entry dialog, a user can enter a search term for a search either on PDB IDs, UniProt IDs, and the annotations given by PDB and UniProt. This will result in a list of found sequences and PDB chains which can then be browsed and selected for a detailed inspection.

The detail view for a sequence which opens as a new tab on the website then contains a summary of the available information: If available, a JMOL² applet opens where the structure can be examined in a 3D view. A second, schematic view shows the protein, its domains and the consensus predictions. By clicking on the button "Show Individual Predictions" a user can get a larger picture which contains also patterns and Vorolign predictions and their locations on the sequence (see Figure 4.3). A click on the bars in the picture results in coloring of the corresponding parts in the structure visualization. Below the overview picture, a tabbox containing entries for the sequence itself and the outputs of the different predictions mechanisms and annotations provides further information. Clicking on list items will open a popup with links to external databases.

The individual predictions obtained from AutoSCOP-annotated InterPro patterns (for both UniProt and PDB) as well as the Vorolign predictions for PDB entries can be downloaded from the database homepage as flat files.

²<http://jmol.sourceforge.net/>

4.5.2 Towards Large-Scale Protein Domain Prediction

As we have seen in the previous evaluations, unique patterns as mapped by AutoSCOP are strong indicators for SCOP classifications and therefore for contained protein domains. For protein chains, the question is whether the correct domain boundaries differ from the pattern locations. Different examples for PDB chains annotated with both SCOP domains and the corresponding pattern locations can be found in Figures 4.5, 4.6 and 4.7.

Predicted Domain Content in ASTRAL 1.69-1.71 Difference Set

For our first analysis of the predicted domain content, i.e. the predicted classifications without considering the locations, we used the difference set (with respect to chains) between ASTRAL 1.71 and ASTRAL 1.69 as test set. Here, we have about 3000 chains with both annotated SCOP domains obtained from ASTRAL 1.71 and annotated AutoSCOP patterns based on ASTRAL 1.69. For 92%, SCOP and AutoSCOP agree in both the number of folds and the assigned fold classifications. In the remaining chains, we find that in most of the cases either SCOP contained all assignments of AutoSCOP or vice versa. Not all of such predictions have to disagree with the assigned SCOP domains: The first case can occur when no unique pattern was found for one of the SCOP domains, and the second case can occur when SCOP does not assign domains for regions of the target where patterns are found, as shown in the examples.

Analysis of Pattern Boundaries

For an analysis of the correlation between pattern boundaries and ASTRAL domains, we used the predicted pattern occurrences on all ASTRAL 1.65 domains. For this set, we find that, when averaging over all patterns of all used InterPro databases, we achieve a coverage of only 51.7%, i.e. a pattern covers only about half of a domain sequence on average. If we discard members from PRINTS and PROSITE, since these databases contain many short-ranged patterns, we can improve this value to 86.7%.

In particular, we analyzed the average coverage of patterns with respect to the InterPro databases: ProDom 75.3%, Pfam 80.6%, PIRSF 99.1%, PRINTS 10.0%, PROSITE 31.7%, SMART 75.6%, SUPERFAMILY 98.4%, and TIGRFams 94.2%. What we can learn from this evaluation is that in many cases using pattern matches in the domain prediction will "underestimate" the extent of a domain region.

What this evaluation does not show is whether some patterns tend to "overestimate" the extent of a SCOP domain on a chain. We tested the agreement between SCOP and pattern locations on about 50000 PDB chains from ASTRAL 1.69 (about 75% single-domain and about 25% multi-domain). We define a pattern location as different from a SCOP domain whenever we observe an overlap of more than 10 residues at the same time as a non-overlapping region of more than 10 residues. We observe that on nearly 20% of all chains we find pattern locations that exhibit such differences to SCOP domains. Therefore, the pattern regions can be regarded as good hints for domain occurrences but not necessarily as sufficient indicators for the concrete domain locations.



Figure 4.5: In this example (1a0p₋), we can see the case when AutoSCOP abstains, as there were patterns for the first domain of the sequence but these patterns were not found to be unique. For the second domain, however, we can assign the correct family. Here, the consensus, which is shown in the interface as a tooltip via mouse-over, assigns the family to a correct location based on PF005589 and leaves out the borders with the remaining superfamily annotations based on SSF56349 as they are too short to be displayed.



Figure 4.6: For 1jwlc we find patterns in regions without annotated SCOP domains. The consensus finds the SCOP domain as c.93.1.1, but spares a short part of it as here we find an overlap with a differing prediction.

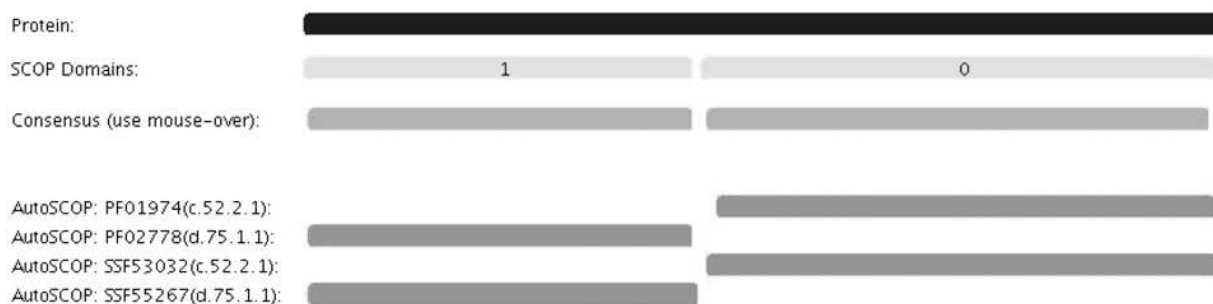


Figure 4.7: This example, 1a79a, shows a very good prediction: both domains are predicted correctly on family level, and the corresponding locations are also nearly identical.

Fold-Based Domain Consensus and Evaluation on CAFASP 4 Data

However, a combination of pattern regions may be a better indicator than each pattern itself, especially if PROSITE and PRINTS are involved. As a simple means to combine individual pattern locations into a final domain prediction for a protein chain, we used a fold-based consensus. The underlying algorithm is as follows: (1) merge all patterns of the same fold as long as there are less than 20 residues between them, and (2) abstain for all regions where differing folds are annotated.

We tested our fold consensus in a blind-test like setup using our predicted patterns from the CAFASP 4 experiment, which allows us to compare against our specialized SSEP-Domain protein domain prediction method, which will be described in the next chapter. As mentioned in section 4.4.4, of the 46 CAFASP 4 targets which we could find in the ASTRAL 1.71, for 23 we can make AutoSCOP predictions. We analyzed the corresponding pattern occurrences on these targets and compared them to the CAFASP 4 domain assignments. With respect to the predicted domain number, when compared to CAFASP on the 23 targets, AutoSCOP is in 18 cases correct and in 5 cases wrong. Using SCOP as a reference, AutoSCOP is in 21 cases correct and two predictions are wrong. In all cases, AutoSCOP and SSEP-Domain agreed in the predicted domain number. We further compared the average overlap between the predicted domain regions and the CAFASP 4 definitions using the score described in section 5.3.2 (subsection "Overlap Score"). We find that, on the 23 targets, the AutoSCOP patterns achieve an average overlap score of about 84% as compared to 91% for our SSEP-Domain approach, i.e. the predicted locations are good, but not completely correct, as could be expected from the evaluations given above.

4.6 Conclusion

AutoSCOP is a simple yet effective sequence-pattern-based approach to SCOP prediction on different SCOP levels. For the domains in the test set with known folds/superfamilies we achieve sensitivity values of more than 93%. Here, especially the specificity values of about 98% are striking. On the Vorolign test set, AutoSCOP even achieves specificity rates of up to 99.9% (fold level). This means that, if a prediction is made, it is indeed very reliable. A test on CAFASP4 targets also shows that the predictions made by AutoSCOP can provide useful information in blind-test protein structure prediction scenarios.

The combination with profile-profile alignment underlines the potential of the AutoSCOP approach by improving the sensitivity of our predictions over the best individual method by about four percentage points. On family and superfamily level, this combination even outperforms the structure alignment methods in our comparison. AutoSCOP can be used as a filter for template selection and fold or superfamily recognition in addition to alignment-based recognition methods.

The inclusion of unique pattern combinations does not significantly improve the performance over unique patterns alone. One possible reason for this is the high redundancy between the InterPro member databases. In Table 4.3 we observe that, with the excep-

tion of Pfam and SUPERFAMILY, leaving out one database does not change the performance very much, especially on superfamily and fold level. Even after leaving out the SUPERFAMILY database, we still observe sensitivity values well above 80% for these levels. Nonetheless, using all found patterns is beneficial for our approach: we could show that using the higher-level InterPro entries, for instance, clearly decreased the performance of our method.

The low coverage and the low sensitivity of AutoSCOP on family level can be explained by the focus of the pattern databases, many of which concentrate on less fine-grained similarities (an obvious example is the SUPERFAMILY database). This implies that many patterns that are unique on coarse levels can be found in more than one SCOP family, and therefore the prediction of the family is not possible. It seems that most patterns work best on superfamily level, which also explains the similar performance of AutoSCOP on superfamily and fold level, as all unique patterns on superfamily level have to be unique on fold level by definition. Thus, especially for the family level, inclusion of specialized data sources such as ASTRAL Family HMMs is useful.

One problem for AutoSCOP as well as for many other SCOP predictors is the handling and recognition of domains belonging to new folds, superfamilies or families. Many predictions for such cases could be traced back to changes in the SCOP versions. However, sometimes we observe only low sequence identities, and in such cases it remains difficult to discriminate between known and new classifications. Discarding such targets can increase specificity but comes with the loss of many good predictions in the twilight zone of sequence identities. This is an interesting point for future development.

The proposed method can easily be extended by including sequence patterns from other data sources, which we have shown here by including predictions from ASTRAL Family HMMs. AutoSCOP is further applicable to any protein domain hierarchy, with SCOP being one very popular example. For the time between releases of such hierarchies, reliable predictions of potential protein classifications are important also for proteins with already available structures. The combination with Vorolign (Table 4.6) shows that there is potential to detect and avoid errors in assignments made on the basis of structure alignments. AutoSCOP may also be a useful additional component for systems like SCOPmap [Cheek et al., 2004] that combine both sequence- and structure-based predictors into a larger system.

If an InterProScan run is necessary, the runtime of AutoSCOP was found to be about half the runtime of a PPA run (with included profile generation for the target), but slightly longer than a Vorolign scan as described in [Birzele et al., 2007], using up to a few minutes per target on an AMD Athlon XP with 1.8 Ghz. If annotated patterns are available the whole AutoSCOP process is mainly reduced to a database lookup which can be done in a few seconds.

Further, for the computation of the training data, which may be time-consuming, we could show that it is possible to still achieve good results with strongly filtered training data (e.g. the ASTRAL 25, which is filtered for 25% sequence identity); this means that the effort needed for generating the training data can be reduced by more than one order of magnitude (in our case) if necessary without a large reduction in accuracy.

Structurally classified protein sequences and structures are a useful basis for research in protein structure prediction. We therefore provide the AutoPSI database of precomputed, predicted SCOP classifications for new PDB entries as well as for about two million UniProt sequences, which is available at <http://www.bio.ifi.lmu.de/AutoPSIDB>. It is a resource that can help in two ways: researchers with an interest in specific proteins may get a clue on structural classification and, associated with this, possible further properties such as a general function; method developers can use the database to derive and compare larger scale data for their purpose.

With respect to domain prediction, the AutoSCOP method and its derived database also offer first insights into the domain structure of these sequences. It is known that InterPro pattern locations can to some degree reflect the locations of protein domains (an evaluation of the performance of InterPro as a domain predictor was done by the assessors of the CAFASP 4 experiment and will be discussed in the next chapter). With AutoSCOP, we can annotate our highly-specific SCOP classifications to each pattern we have already seen in our training data, i.e. in the available SCOP/ASTRAL sequences. With these annotations, we can combine the pattern locations in a consensus which can then give a clear picture of the regions where the corresponding SCOP classifications are located, even if such a consensus region is comprised of many short PROSITE patterns, for instance. This can help users to get a direct impression on potential SCOP domain locations. Our evaluations show that AutoSCOP annotated patterns when available performed well on the CAFASP 4 data in a blind-test like setup, which confirms their applicability. Therefore, this approach can also be regarded as a step towards large-scale protein domain recognition.

In addition to our precomputed results, on our website we provide a web server at <http://www.bio.ifi.lmu.de/autoscop>, which applies AutoSCOP* on domain sequences. Here, users can directly submit their sequences in order to obtain SCOP predictions.

Chapter 5

SSEP-Domain: Template-Based Protein Domain Prediction

The methods presented so far (Preselection and AutoSCOP) deal mostly with protein domains and the corresponding classifications. However, given a target protein of unknown structure, also the domain content, i.e. the partition of the target into structural domains, is unknown. Here, AutoSCOP can give good hints but is not necessarily accurate with respect to the correct domain boundaries.

In this chapter, we present the SSEP-Domain protein domain prediction approach, which is based on the application of secondary structure element alignment and profile-profile alignment in combination with InterPro pattern searches. As we could show for Preselection and Refinement in chapter 3, secondary structure element alignment (SSEA) allows rapid screening for topologically similar and therefore potential domain regions while profile-profile alignment provides us with the necessary specificity for selecting significant hits. Including InterPro patterns, which have turned out to be a valuable resource for the AutoSCOP approach, we can also find regions on the target sequence that share similarities to known family or superfamily definitions which are not necessarily based on structural homologs.

In the CAFASP 4 experiment, SSEP-Domain performed well and was placed in the top group of domain prediction algorithms. Since then, we have introduced some changes that both significantly speed-up the procedure and slightly improve the performance. The description of the method and its evaluation on CAFASP 4 data presented here is an extended version of our journal contribution on the SSEP-Domain approach that appeared in *Bioinformatics* [Gewehr and Zimmer, 2006].

In addition, a newer evaluation on CASP 7 data shows that SSEP-Domain as well as its template database (SCOP) tend towards defining too few domains on the target sequences with respect to the definitions of the CASP 7 assessors. We therefore describe a variant of SSEP-Domain that includes an alternative structure-based domain assignment for the template domains (SSEP-Domain*), i.e. a different view on protein domains than the SCOP standard, that performs better on multi-domain proteins when evaluated on the CASP 7 data.

5.1 Introduction

Domain assignment in proteins is an important subtask of structure prediction, as domains are usually considered the basic units for protein folding, evolution, and function [Heger and Holm, 2003, Vogel et al., 2005], and thus the decomposition of proteins into domains can help in areas such as functional classification, homology-based structure prediction, and structural genomics [Liu and Rost, 2003]. Since 2004, the CASP and CAFASP experiments have included a domain prediction subcategory into their evaluations, which confirms the importance of this task.

The algorithm described in this chapter, SSEP-Domain, predicts protein domains using the amino acid sequence of the target on the basis of alignments to known SCOP domains. Other recent approaches for domain recognition from sequence are also often alignment-based, such as ADDA [Heger and Holm, 2003], the Dompred-DomSSEA approach [Marsden et al., 2002], and DOMAINATION [George and Heringa, 2002a]. DO-PRO [von Öhsen, 2005] uses stochastic models on the alignment-based output of the Arby structure prediction server [von Öhsen et al., 2004]. Besides alignments, the basis for such approaches can also be machine learning methods as described for BIOZON in [Nagarajan and Yona, 2004] and PPRODO [Sim et al., 2005], statistics as in the DGS method [Wheelan et al., 2000], taxonomy [Coin et al., 2004] or clustering as in MKDOM [Gouzy et al., 1999] or DIVCLUS [Park and Teichmann, 1998]. Some approaches make inherently use of predicted structures, such as SnapDragon [George and Heringa, 2002b] or the Robetta servers, namely Robetta-Rosettadom [Kim et al., 2005] and Robetta-GINZU [Chivian et al., 2003], which are part of the evaluation in the results section.

Similar to the preselection approach described in chapter 3, for SSEP-Domain, secondary structure elements of proteins play an important role. Moreover, as the preselection-based speedup for fold recognition, also this approach is based on the observation that the fold class of a protein domain is often defined by the topology of its secondary structure elements (i.e. the elements and their lengths, the order of these elements, and the contacts between them). For protein domains, this means that their secondary structure element topology may be an indicator for fold class membership even if the sequence differs from all known members of the respective class. If the structure of a protein is unknown, we are still able to make use of its secondary structure elements and their order based on secondary structure prediction. Therefore, the comparison of subsequences of a target protein with known protein domains based on these features may reveal regions of the target that have the potential for being independent domains.

The DomSSEA protein domain prediction uses secondary structure element alignment (SSEA) for selecting PDB chains of potential templates. Other template-based methods such as Arby select many seed sequences (derived from scans by PSI-BLAST [Altschul et al., 1997], InterProScan [Zdobnov and Apweiler, 2001], predicted secondary structures, and more) and then apply elaborate and often costly alignment and scoring procedures such as threading or log-average profile-profile alignment (PPA). In SSEP-Domain, as we aim at both a fast and an accurate prediction method, we combine the quick SSEA with the accurate PPA, allow InterPro pattern regions directly as potential

domains, and include some speed-up and filtering techniques. Further, as we use single domains as templates, we can predict multi-domain targets independently of whether the specific domain combination is already contained in the protein data bank (PDB) or not.

We provide a web-based user interface for SSEP-Domain. Our service may also be used together with other methods as part of the domain prediction meta-server META-DP [Saini and Fischer, 2005].

5.2 The Domain Prediction Pipeline

Definition (Domain Prediction Task): We define the *Domain Prediction Task* as the problem of decomposing a target protein sequence into subsequences, each of which represents one protein domain of the target.

Like Nagarajan and Yona in [Nagarajan and Yona, 2004], we consider only continuous subsequences as domain regions. Possible extensions towards the prediction of discontinuous domain regions will be discussed at the end of this chapter.

A generic template-based domain prediction method using single-domain templates may consist of three consecutive steps: (1) It searches for potential domain boundaries on a target sequence, (2) it generates potential domain regions from these boundaries, and (3) it generates combinations of potential domain regions for a complete prediction for the target. Our approach follows these three steps, with two main objectives: We aim at providing a fast method, i.e. we want to perform each step efficiently, and of course we want to provide accurate predictions. In order to do so, we use a number of simple ideas:

We know that domain boundaries should lie in coil regions; therefore, we use only the centers of coil regions in the predicted secondary structure for our target and select some potential boundaries from them using a very quick, SSEA-based scoring. As these boundaries are few, we inspect all continuous subsequences between them which contain at least 50 residues in detail (we have less than ten such regions on average on the CAFASP4 test set). This inspection is done using an efficient but accurate template-based fold recognition method, namely Preselection and Refinement as described in chapter 3, concentrating only on templates with similar lengths. In addition, we directly add locations of Inter-Pro patterns found on the target as additional potential domain regions, without further processing them by alignments.

Based on the scores obtained for our regions, we use a simple scoring scheme for combinations of potential domains which evaluates and ranks all possible combinations of evaluated regions in only seconds. As our results show, we can predict the domains on a target accurately in comparison to other methods, using less than ten minutes on average on a single workstation, in contrast to Arby, for instance, which needs up to one day. A visualization of our ideas is given in Figure 5.1.

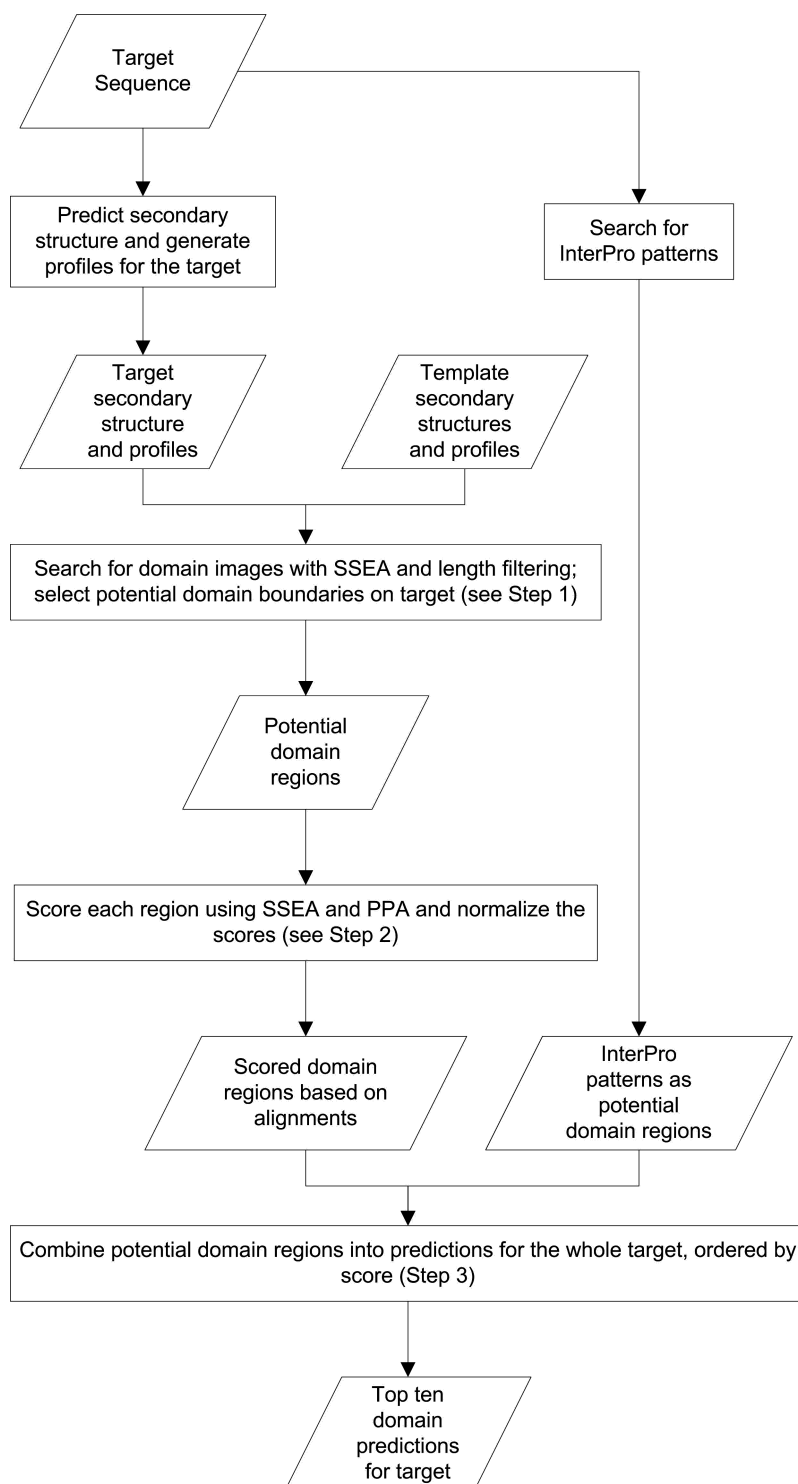


Figure 5.1: Visualization of the different stages of the SSEP-Domain pipeline.

5.2.1 Preliminaries

Target and Template Data

For each target sequence, we used PSIPRED [Jones, 1999b] and PSI-BLAST against the NR database of April 2004 as described in section 2.5.2. From these runs we obtain not only the secondary structure prediction, but also the PSI-BLAST sequence profile and the PSIPRED secondary structure profile of the target. The same was done for each domain in our template library.

We use the atom-based ASTRAL¹ compendium [Chandonia et al., 2004] based on SCOP [Murzin et al., 1995] version 1.65 (released in December 2003) and the corresponding subsets filtered for 95% and 25% sequence identity without genetic domains. Furthermore, the ASTRAL compendium provides us with the classification of the templates into fold classes. The template library *Domains* contains the ASTRAL 95 subset.

Besides the fact that the underlying SCOP database is expert-curated, another advantage of using ASTRAL/SCOP domains as templates is that SCOP's boundary placement was observed to be the most precise in comparison to other methods (see 2.2.2).

Parameter Calibration

Some parameters were fitted based on statistical evaluations on ASTRAL (length filter, significance filter, score normalization, and gap penalties) which will be described in the corresponding text parts. All other parameters were calibrated such that SSEP-Domain achieves high accuracy with respect to the predicted domain number together with a reasonably fast average prediction time on a training set of randomly chosen PDB chains available in ASTRAL 1.65.

5.2.2 Step 1: Finding Potential Domain Boundaries

Before going into details, though this already pertains to the description of the algorithms, we now define some sets which will make it easier to follow the description of the domain prediction process, namely

- **Centers:** The set of all centers of coil regions on the targets sequence with respect to the predicted secondary structure, with the exception of the leading and trailing coils. For those, we include the first and the last position of the target sequence instead.
- **Domains:** The template database of known domains used by our algorithm, namely the ASTRAL 95 after exclusion of genetic domains (i.e. domains which span different chains).
- **Images:** This set collects the highest-scoring representatives of the templates stored in the set *Domains* as found by SSEA (see below for details).

¹provided by <http://astral.berkeley.edu>

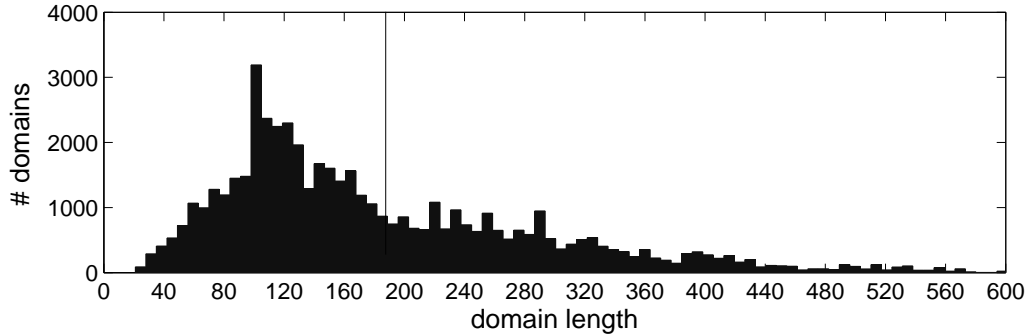


Figure 5.2: Histogram of domain lengths observed in the ASTRAL distribution. The histogram was cut at length 600 for better readability, though some domains in the ASTRAL 1.65 distribution are longer than 600. The vertical line shows the mean length of the whole distribution.

- **Regions:** This set contains all potential domain regions on the target sequence.

The first step of our method deals with finding positions on the target sequence where boundaries between domains may be located. We regard all centers of predicted coil regions on the target sequence t as potential boundaries. Since the number of boundaries may affect the complexity of the method quadratically (see Step 2), we employ a heuristic to select only a reasonably small number of these centers for further evaluation as described in Algorithm 2.

Finding Domain Images using SSEA

First of all, we collect all centers of predicted coil regions on t together with the start and the end of t in the set *Centers*. For each template domain sequence d in our template database *Domains*, we align d against all subsequences r_{ij} between coil centers c_i and $c_j \in$ *Centers* using SSEA.

Definition(Domain Image). We collect the highest-scoring r_{ij} for each template domain d as so-called *domain images* in the set *Images*.

In other words, all subsequences between coil centers are scanned for secondary structure similarity to the template domains using SSEA. The stored domain images for our template domains will be used to assign a score to each center and then to select potential domain boundaries based on these scores.

Length Filtering of Templates

Here, instead of aligning against all possible subsequences, we apply a simple length filter for selecting only subsequences of similar length for each template domain.

Definition(Length Filter): As a further criterion for finding domain images for a domain template d , in order to be evaluated, subsequences on the target may differ in length from $|d|$ by 5% at maximum. In the following we will write this property as $|d| \approx |r_{ij}|$.

As we directly make use of ASTRAL domains as templates, we chose the threshold of 5% based on a simple evaluation on all ASTRAL domain sequences of version 1.65: The distribution of the lengths of these domains is shown in Figure 5.2. Our analysis shows a mean length of 188 with a large standard deviation of 118. We further computed the mean coil length at either end of a domain according to DSSP [Kabsch and Sander, 1983] applied to the coordinate files provided by ASTRAL, which was found to be about 4.5 amino acids.

Using a threshold of 5%, for a potential region of length 188, we allow templates to differ by the average coil length at either end, i.e. by 9 amino acids. In addition, using a scaled threshold, we assume that with increased domain length the possible length differences between homolog domains are also increased.

Significance Filtering of Domain Images

For filtering out unlikely domain images, we compare the SSEA score of a hit $s_{\max}(d)$ against a threshold $s_{\text{thresh}}(d)$ derived from the all-against-all SSEA alignment score distribution of the fold class the template belongs to. These distributions were computed for each fold class by aligning all members against each other, based on ASTRAL 95. Only hits having a score higher than the mean of the corresponding distribution are accepted and thus added to the set of domain images (*Images*). For classes having only one member, we use the mean of all computed means as threshold.

Accumulative Boundary Scores

Now, given the set of domain images for our template domains, we can derive a score for each coil center, which will then enable us to select only the few most probable ones as potential domain boundaries for the next stages. In particular, the score of each of the top-scoring 100 accepted domain images is added to the corresponding coil centers, i.e. the score of a coil center c_i is the sum of the scores of all adjacent domain images in this set.

Definition(Potential Boundaries): For the next stages, we then select the ends of the target sequence as well as the 4 top scoring coil centers with respect to this accumulative score as domain boundaries.

The number of boundaries was determined in the parameter calibration process described in section 5.2.1.

Algorithm 2 Domain Boundary Search (Step 1)

```

1: // initialization
2: Centers ← centers of coil regions predicted on target t
3: Regions ← { $r_{ij} = t[c_i..c_j] \mid c_i, c_j \in \textit{Centers} \wedge c_i < c_j$ }
4: Images ← {}
5: Domains ← ASTRAL95
6:
7: // generation of domain images
8: for all template domains  $d \in \textit{Domains}$  do
9:   // get highest scoring region of similar length
10:   $s_{\max}(d) \leftarrow \max_{r_{ij} \in \textit{Regions} \wedge |r_{ij}| \approx |d|} \text{SSEA}(d, r_{ij})$ 
11:
12:  // significance filtering: score high enough?
13:  if  $s_{\max}(d) > s_{\text{thresh}}(d)$  then
14:    add corresponding region  $r_{ij}$  to Images
15:    with  $\text{score}(r_{ij}) \leftarrow s_{\max}(d)$ 
16:  end if
17: end for
18:
19: // accumulative scoring of coil centers
20:  $\forall c \in \textit{Centers} : \text{score}(c) \leftarrow 0$ 
21: for the top-scoring  $r_{ij} \in \textit{Images}$  do
22:   $\text{score}(c_i) \leftarrow \text{score}(c_i) + \text{score}(r_{ij})$ 
23:   $\text{score}(c_j) \leftarrow \text{score}(c_j) + \text{score}(r_{ij})$ 
24: end for
25: select the top-scoring coil centers

```

5.2.3 Step 2: Scoring of Domain Regions

Now that we have found potential boundaries, we can take a closer look at the subsequences of the target defined by these boundaries:

Definition(Domain Region): A potential *domain region* $r \in \textit{Regions}$ is defined as a subsequence of the target that starts and ends at potential boundaries and contains at least 50 residues.

In the following we evaluate these $r \in \textit{Regions}$ with respect to their similarity to the template domains using a more sophisticated alignment method, namely profile-profile alignment on both sequence and secondary structure profiles, in combination with a preselection approach as introduced in chapter 3 (see Algorithm 3).

Algorithm 3 Scoring of Domain Regions (Step 2)

```

1: Regions  $\leftarrow$  potential domain regions
2:
3: for all  $r \in \textit{Regions}$  do
4:   // score fold classes by highest-scoring members
5:   for all fold classes  $\textit{Fold} \subset \textit{Domains}$  do
6:      $\text{score}(\textit{Fold}) \leftarrow \max_{d \in \textit{Fold} \wedge |d| \approx |r|} \text{SSEA}(r, d)$ 
7:   end for
8:
9:   // select members of potential fold classes
10:   $D_{\text{top}} \leftarrow$  members of top-scoring fold classes
11:
12:  // score normalization for multiplicative scoring
13:   $\text{score}_{\text{raw}}(r) \leftarrow \max_{d \in D_{\text{top}} \wedge |d| \approx |r|} \text{PPA}(r, d)$ 
14:   $\text{score}_{\text{final}}(r) \leftarrow \text{score}_{\text{raw}}(r) / (10 \log |r|)$ 
15: end for

```

Alignment-based Region Scores

All fold classes are ranked by their highest-scoring member d (under the restriction that $|d| \approx |r|$) with respect to the SSEA scores against r (see Step 1), and the highest-scoring 20 classes are selected. In order to find distant homologs in the members of these classes with matching secondary structures and similar lengths (D_{top}), we align each of them with r using PPA on both sequence and secondary structure profiles. The largest score of these alignments is assigned as $\text{score}_{\text{raw}}(r)$ to r (see section 2.5.2 for implementation details and alignment parameters).

This selection strategy is based on the evaluation of preselection and refinement as shown in chapter 3: In the corresponding results, for two of three benchmark sets, a preselection of fold classes using SSEA sped up the prediction procedure significantly while only slightly decreasing accuracy, and on a third benchmark set preselection even increased accuracy. In order to be able to predict the domain architecture of a target protein sequence in reasonable time, we therefore included preselection in our approach. The number of templates classes (20) was also determined during the parameter calibration procedure described in section 5.2.1.

Score Normalization

Finally, we normalize the resulting raw score in order to obtain a representative score for each evaluated domain region:

Definition(Final Region Score): We compute the final score of a potential domain region r as

$$\text{score}_{\text{final}}(r) = \text{score}_{\text{raw}}(r) / (10 \log |r|).$$

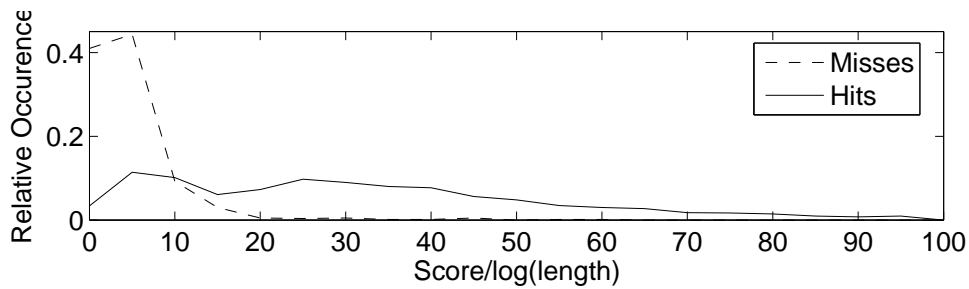


Figure 5.3: Score histogram of fold recognition evaluation on an ASTRAL subset. For better readability, the histogram was cut at a score of 100, though few hits exceeded this score. The optimal threshold for partitioning scores into hits and misses lies at about $\text{score}_{\text{raw}}(r)/\log|r| = 10$.

Since we assume that the raw scores grow stronger than logarithmically with increasing region length, the denominator penalizes shorter regions. The factor of 10 results from fold recognition experiments using the combined PPA scores divided by the logarithm of the corresponding domain lengths on an ASTRAL subset filtered for 25% sequence identity. In this evaluation, we find that the optimal threshold to discriminate between hits and misses is $\text{score}_{\text{raw}}(r)/\log|r| \approx 10$ (see Figure 5.3).

By dividing the region scores by $10 \log|r|$, we obtain scores where the border between correct and wrong predictions is at about 1.0. The reason for this transformation will become clear with the description of the next step in the SSEP-Domain pipeline, Step 3, where we score domain combinations by multiplying the individual scores. Then, a score of 1.0 is in fact a *neutral* score: Potential hits have a score above 1.0 and therefore augment the score of a combination of domain regions, potential misses are below 1.0 and therefore diminish the final scores.

Patterns as Domain Regions

At this stage, we also include InterPro patterns [Mulder et al., 2003] found by the InterProScan program [Zdobnov and Apweiler, 2001] on the target sequence as potential indicators for domain regions.

As observed by [Saini and Fischer, 2005], the output of InterProScan "is often ambiguous", and a user needs to analyze the InterProScan output visually. Nonetheless, based on the results of the protein domain prediction analysis in the previous chapter, we added InterPro patterns to the list of potential domain regions with the exception of PRINTS and PROSITE patterns. However, keeping in mind the observed differences between pattern and domain boundaries, the score of a single pattern is computed conservatively as 1.0 minus the e-value returned by InterProScan for the pattern. The maximum score of 1.0 allows patterns with highly confident hits to fill gap regions without affecting the overall

multiplicative score of a domain combination as described in Step 3. In other words, we regard patterns at best as neutral domain regions against the background of the score transformations for alignment-based region scores, as for our purpose the correlation between pattern and domain boundaries was not found to be clear enough to emphasize such regions over a neutral level.

Inclusion of AutoSCOP

At this point, a further option is to include the AutoSCOP method, either by emphasizing patterns that are unique for a fold class or by including predicted fold cases into the set of evaluated fold classes for a potential domain region. This option (both versions) will be discussed and evaluated in the results section (5.3.3) together with the influence of InterPro on the prediction process. Please note that (although this already refers to 5.3.3) inclusion of AutoSCOP did not result in differing predicted domain numbers on our test sets when compared to the original SSEP-Domain Step 2 as described above.

5.2.4 Step 3: Combining Multiple Domain Regions

Finally, for combining potential domain regions, we recursively generate all possible non-overlapping combinations of regions and patterns, score them and choose the top combinations as predictions.

Multiplicative Scoring

We score each combination c based on the scores obtained for the regions in the previous steps and gap penalties for unassigned parts on the target sequence:

$$s(c) = \prod_{i=1}^p s(r_i) \prod_{j=1}^{p+1} g_j,$$

where $r_i, i \in \{1, \dots, p\}$ denotes a participating region and g_i denotes the factor for the unassigned region between r_{i-1} and r_i , with g_1 being the gap at the beginning, and g_{p+1} being the gap at the end of the target sequence.

Gap Costs

We assume that gaps may only occur in coils. Furthermore, we assume that all known domains may be combined with each other independently of whether they occur in single- or multi-domain chains. Therefore, we analyzed the coil lengths at both ends of the DSSP [Kabsch and Sander, 1983] secondary structures of all ASTRAL domains. We do not penalize gaps of length less than 10 (see 5.2.2), and we allow only gaps shorter than the minimal domain length of 50. All gaps of length 10 to 49 are penalized with the empirically estimated probability of observing combined coils (the coil region at the end of the first domain plus the coil region at the beginning of the second domain) longer than 9.

This coarse-grained setup with only three different gap states (0..9, 10..49, and > 49) allows pattern boundaries to diverge from alignment boundaries within a range of the minimal domain length while favoring short gaps. If we find gaps after having scored the candidates for the final output, we elongate all regions equally until all gaps are closed. Thus, like many other predictors, we concentrate on boundaries between domains and do not predict inter-domain linker regions.

How to Handle Unknown Domains

This scoring scheme does not take into account unknown domains, i.e. domains that show no similarity to any of the template domains. For instance, think of a case where we find two very good hits, one at the beginning of a sequence and one at the end of a sequence, and a long stretch (longer than 50 residues) in between where no good template could be found. If at least a remote homology to a known template domain is found, the multiplicative score of the three parts (two with high scores and one with a low score) will most probably result in a good score for the overall prediction. However, if no template at all is found for the intermediate region, we have two options: (1) discarding this prediction, as such a long gap is not allowed, and (2) considering the intermediate stretch as an "unknown" domain. In its default configuration, SSEP-Domain uses the first option. In the results section we will also evaluate and discuss what can be done for the second option (see section 5.3.5). Please note that a case as described above did not occur in both the CAFASP 4 and the CASP 7 test data.

5.3 Results

5.3.1 CAFASP 4 and CASP 6 Results

The SSEP-Align structure prediction server maintained by Alessandro Macri (see also section 5.3.4) that participated in CASP 6 made use of SSEP-Domain in a first version for its domain predictions. It submitted domain predictions for 60 of the 63 evaluated targets along with the predicted protein structure models. SSEP-Align is ranked among the top ten predictors (including both human and server groups) for all criteria, the best result being rank 6 on a set of multi-domain targets. Among the servers, SSEP-Align is ranked fourth on all targets and third on multi-domain targets with respect to NDO score, a domain overlap measure used by the CASP 6 assessors [Tai et al., 2005]. These results are in accordance with the results we obtained for our independent server SSEP-Domain in a second, parallel experiment, namely CAFASP 4, which will be described below (see Table 5.3 for comparison). In addition, SSEP-Align submitted the top-scoring prediction for the difficult multi-domain target T0237 ([Tai et al., 2005]; not evaluated in CAFASP).

In parallel to the CASP 6 experiment, the CAFASP 4 experiment was performed. CAFASP 4 was held from May 2004 to September 2004, containing domain prediction as sub-category of the experiment. Here, SSEP-Domain participated as independent server

and was also ranked among the top five domain prediction servers². The best performance was observed on so-called homology targets, targets for which templates having a high sequence identity were available. It should be noted that the CAFASP evaluation, when compared to CASP, is based on different target sets, different domain definitions and differing evaluation methods.

5.3.2 Current Version under CAFASP Conditions

In the following section, we will concentrate on the CAFASP 4 evaluation, since there SSEP-Domain participated as individual server. Furthermore, CAFASP evaluated more servers that did not participate in CASP than vice versa.

Changes after CAFASP 4

At the beginning of the experiment, we detected domain boundaries by sliding a window of roughly the size of the current template domain along the target sequence. For each window position, we performed SSEAs similarly to the final method described in Step 1. InterPro patterns as additional domain regions were introduced shortly after the start of the experiments. Since the CAFASP version of SSEP-Domain needed up to several hours for one target, after the end of CAFASP we added length filtering in order to reduce the number of potential templates and replaced the exhaustive sliding window approach by the coil-center-based domain boundary search (see Step 1). Thus, the main difference between the current version and the CAFASP version is the speed of the predictions.

This speed-up can be understood by looking at the number of performed PPAs in Step 2, which are the most time-consuming part of SSEP-Domain. Naively implemented, each potential domain region would be aligned against more than 9000 templates in the ASTRAL95. The preselection of potential fold classes using the SSEA scores without length filtering reduces the number of alignments per potential region to about 11% of the original number of templates. The additional length filter then reduces the number of alignments per region to less than 2% of the number of available templates. So we achieve a speedup of two orders of magnitude due to preselection and length filtering, resulting in less than 10 minutes average runtime per target.

Further, the final version yields slightly different predictions as indicated by the performance analysis (see Table 5.1): two more targets are predicted correctly. One reason for this are the new, coil-centered boundaries. Using a sliding window as in the CAFASP version, there may be low-scoring predicted boundaries close to each other, while the new approach combines such blurred boundaries in the coil centers. This results in an accumulated score for each coil and thus a clearer picture of whether a coil may contain a linker region or not. In addition, the length filtering (see 5.2.1) improves these predictions by discarding domains that achieve good alignment scores but are not representative of the domain region under inspection due to the length differences.

²<http://cafasp4.bioinformatics.buffalo.edu/dp/update.html>

Experimental Setup

For this work, we evaluated the final version of SSEP-Domain under CAFASP 4 conditions in order to compare with our own CAFASP results as well as with the CAFASP performance of other servers. This means that the domain database we used as template data and for parameter calibration does not contain any of the test targets, since it was available before the start of CAFASP 4.

We quote the CAFASP 4 results from the official evaluation website (October 1st, 2005) for the following methods:

- **ADDA** [Heger and Holm, 2003]: The ADDA algorithm uses alignments derived from an all-against-all sequence comparison to define domains within protein sequences based on a global maximum likelihood model.
- **Armadillo** [Dumontier et al., 2005] uses an amino acid index (the domain linker propensity index DLI) to convert a protein sequence to a smoothed numeric profile from which domains and domain boundaries are deduced using z-score distributions.
- **BIOZON** [Nagarajan and Yona, 2004]: For the BIOZON approach, multiple sequence alignments are generated and several different measures are defined to quantify the information content of each position along the sequence. Combination of these measures into a single predictor is done using a neural network.
- **Dompred-DomSSEA** [Marsden et al., 2002], which uses SSEA to map a target to a template protein chain and then transfers the domain assignments from the template to the target, and another method from the same group called **Dompred-DPS** which was also entered into the CAFASP 4 experiment.
- **DOMPRO** [Cheng et al., 2005], which uses recursive neural networks on profiles, predicted secondary structure and predicted relative solvent accessibility.
- **DOPRO** [von Öhsen, 2005], which uses an approach based on stochastic models on the output of the fold recognition stage of the ARBY fold recognition server [von Öhsen et al., 2004], and which has been shown in [von Öhsen, 2005] to be more accurate than Arby in domain prediction.
- **GLOBPLOT** [Linding et al., 2003], an approach based on a running sum of the propensity of amino acids to be in an ordered or in a disordered state.
- **MATEO**, which was entered into CAFASP 4 by Matej Lexa.
- **Robetta-GINZU** [Chivian et al., 2003], which uses BLAST, PSI-BLAST, FFAS03 [Jaroszewski et al., 2005], the structure prediction meta-server 3DJury [Ginalski et al., 2003] and PFam-A (using HMMER) to detect putative domain regions in the query sequence.

- **Robetta-RosettaDOM** [Kim et al., 2005] searches for homologous regions with GINZU and, if nothing sufficient has been found, uses Rosetta [Bradley et al., 2005] to produce three-dimensional de-novo models, applies a structure-based domain boundary assignment to these models and finally chooses domain boundaries based on consistencies in their models in an approach similar to the one described by George et al. for SnapDragon [George and Heringa, 2002b].
- **CONSENSUS** [Saini and Fischer, 2005]: The CAFASP 4 domain prediction consensus server.
- **InterPro** [Mulder et al., 2003] as evaluated by CAFASP, for which an automated evaluation protocol for the InterProScan output was devised together with the EBI support team [Saini and Fischer, 2005].

To our CAFASP results we will refer as SSEP-CAFASP in the tables.

The CAFASP 4 test set contains 58 targets. Some servers had missing predictions during CAFASP 4, namely Armadillo (7), DomSSEA (5), DPS (5), GLOBPLOT (4), and BIOZON (1). For consistency reasons, in the tables we kept the values for all targets from the CAFASP 4 evaluation for sensitivity, specificity, and average overlap score. These count missing submissions as wrong, ignore them, or assign 0%, respectively. In addition, we computed the common subset of targets for which all servers sent predictions. This set is the basis for our rankings and plots. The following sets are used in our evaluation:

1. *CAFASP* contains all 58 targets, including those which were missed by some servers.
2. *Common* contains the 44 targets for which all servers submitted predictions (see above).
3. *Single* contains the 29 single-domain targets from the *Common* set.
4. *Two* contains the 15 two-domain targets from the *Common* set.

Sensitivity and Specificity

For our first evaluation, we concentrate on the predicted number of domains. This assessment does not penalize situations where predicted boundaries are far from being correct, as long as the number of predicted domains equals the native domain definition. We evaluate sensitivity and specificity of the predicted domain numbers, where sensitivity is defined as $TP/(TP + FN)$, and specificity is defined as $TP/(TP + FP)$. TP denotes the number of true positives, FP the number of false positives, and FN the number of false negatives, each with respect to the evaluated category (e.g. single-domain). Furthermore, in CAFASP 4, split-domain predictions were considered as wrong predictions for the sensitivity evaluation and left out for the specificity evaluation. Therefore, in addition to our CAFASP-like evaluation, we computed the corresponding values for the affected servers (RosettaDOM and GINZU) after including split-domain predictions with respect to the number of predicted domains (see below).

Server	CAFASP 58 targets	Single 29 targets	Two 15 targets	Common 44 targets
CONSENSUS	48	26	10	36
<i>SSEP-Domain</i>	48	28	8	36
SSEP-CAFASP	46	27	7	34
RosettaDOM	46	23	10	33
DOPRO	44	24	9	33
InterProScan	42	28	3	31
DOMPRO	41	25	6	31
DPS	36	24	7	31
GINZU	42	23	7	30
DomSSEA	38	26	4	30
GLOBPLOT	37	27	3	30
ADDA	38	26	3	29
MATEO	23	18	2	20
BIOZON	10	3	5	8
Armadillo	8	1	4	5

Table 5.1: Correct predictions on single-domain, two-domain, and all targets of the common subset of targets for which all servers submitted predictions. For comparison, the CAFASP values on all 58 targets are given (based on the values given on the official evaluation website for the numbers of correctly predicted targets on single- and two-domain targets). The percentage given for a predictor for a certain set is computed as the relative fraction of correct predictions in the corresponding set. Predictors were ranked on the common subset (*Common*) descendingly. SSEP-Domain shows the results of our final method in the reevaluation on the CAFASP data, SSEP-CAFASP gives the results of the preliminary approach in the original CAFASP evaluation.

Server	single-domain	two-domain	single & two
RosettaDOM	94% (36)	75% (16)	88% (52)
CONSENSUS	88% (42)	79% (14)	86% (56)
GINZU	92% (36)	69% (13)	86% (49)
<i>SSEP-Domain</i>	83% (47)	82% (11)	83% (58)
SSEP-CAFASP	84% (45)	73% (11)	82% (56)
DOPRO	88% (40)	64% (14)	81% (54)
InterProScan	75% (51)	67% (6)	74% (57)
DomSSEA	75% (44)	63% (8)	73% (52)
DOMPRO	76% (46)	50% (12)	71% (58)
GLOBPLOT	71% (48)	60% (5)	70% (53)
DPS	78% (36)	50% (16)	69% (52)
ADDA	73% (48)	33% (9)	67% (57)
MATEO	78% (27)	15% (13)	58% (40)
Armadillo	100% (4)	18% (22)	31% (26)
BIOZON	100% (4)	19% (31)	29% (35)

Table 5.2: Specificity of predictions (based on the values given on the official evaluation website), rounded to full percentages. All submitted predictions were used, so e.g. the single-domain class may contain up to 41 correct predictions (all available single-domain targets in CAFASP 4). As in CAFASP 4, missing, split-, and multi-domain predictions were not considered. For each server, we give the fraction of correct predictions within a certain class followed by the number of all predictions made by the server for this class shown in brackets, in order to be able to distinguish between servers with high specificity but low sensitivity and vice versa. For instance, for SSEP-Domain on the *two-domain* set, 82% (11) means that 9 out of 11 submitted two-domain predictions were correct. For consistency with the CAFASP 4 evaluation, for the *single & two* set, specificity is computed as the number of correct predictions divided by the number of single- and two-domain predictions. Predictors were ranked by specificity on all targets descendingly.

Table 5.1 shows the number of correct predictions of the CAFASP 4 participants together with the results of SSEP-Domain. With 48 of all 58 targets predicted correctly (82.8%), SSEP-Domain achieves the highest number of correct predictions of all individual servers. Only the CAFASP consensus method also achieves 48 correct predictions. Sensitivity evaluation on the *Common* set shows a similar picture: CONSENSUS and SSEP-Domain perform best, followed by SSEP-CAFASP, RosettaDOM, and DOPRO. While RosettaDOM, CONSENSUS, and DOPRO find more native two-domain proteins, SSEP-Domain achieves the highest number of correct predictions for single-domain proteins together with InterProScan.

Table 5.2 shows the corresponding specificity values. With 82%, SSEP-Domain achieves the highest specificity on two-domain targets. However, while we observe high overall sensitivity for SSEP-Domain, RosettaDOM, GINZU, and CONSENSUS achieve higher overall specificity on all targets (see also Figure 5.5, upper panel).

If we include split-domain predictions, we get different values for RosettaDOM and GINZU: RosettaDOM now achieves 35 correct predictions on *Common* and 86% specificity (of 56 counted predictions) on both single-domain and two-domain targets; GINZU achieves 32 correct predictions and 80% (of 55 counted predictions), respectively.

Overlap Score

The second major part of the CAFASP evaluation is the assessment of the correct boundary placement using a so-called *overlap score*. We follow the CAFASP evaluators in using the algorithm described in [Jones et al., 1998] for the predictions of the final version. The values for the CAFASP participants were taken from the official website. For this evaluation, split-domain predictions were included already in the original CAFASP 4 evaluation.

The algorithm for the computation of the overlap score is simple: assign the predicted domains to the reference domains such that the order is preserved and that the maximal overlap over all reference domains is reached. No predicted domain nor reference domain can be assigned to more than one match partner. Then sum over the overlapping positions for each reference domain and divide the resulting value by the overall number of residues. Figure 5.4 shows an example for the overlap score computation taken from [Jones et al., 1998]. Here, A and B denote the two domain assignments, and the overlap tables show the matching domains. The third part shows the computation of the final overlap score.

Table 5.3 shows the overlap scores for all evaluated servers on the different sets. SSEP-Domain achieves the highest score of all evaluated predictors on the *Single*, *Common*, and *CAFASP* sets. We observe an increase of average overlap score on the *CAFASP* set of about three percent for SSEP-Domain over the CAFASP predictions (91.9% to 88.9%) which can be explained to a large extent by the increased performance on single domain targets. Figure 5.5 (lower panel) shows all CAFASP 4 participants in a plot of sensitivity vs. average overlap score on the *Common* set.

A (i) Assignments

Residue	01	02	03	04	05	06	07	08	09	10	11	12	13	14	15	16	17	18	19	20
A	1	1	1	1	1	1	1	2	2	2	2	2	2	2	2	2	3	3	3	3
B	1	1	1	1	1	1	2	2	2	2	2	2	2	2	2	3	3	3	3	3

(ii) Overlap table:

	A1	A2	A3
B1	6	0	0
B2	1	8	0
B3	0	1	4

(iii) Overlap Score $\frac{6 + 8 + 4}{20} \times 100 = 90 \%$

B (i) Assignments

Residue	01	02	03	04	05	06	07	08	09	10	11	12	13	14	15	16	17	18	19	20
A	1	1	1	1	1	1	1	2	2	2	2	2	2	2	2	2	3	3	3	3
B	3	3	1	1	1	1	1	1	1	2	2	2	2	3	3	3	3	3	3	3

(ii) Overlap table:

	A1	A2	A3
B1	5	2	0
B2	0	4	0
B3	2	3	4

(iii) Overlap Score $\frac{4 + 5 + 4}{20} \times 100 = 65 \%$

Figure 5.4: Overlap score examples taken from [Jones et al., 1998]: In both examples (A and B), the overlaps are computed between the assigned domains for the two sequences as shown in the overlap tables. Based on the resulting values, the mapping between domains with the maximum overlap scores were chosen.

Server Name	CAFASP	Single	Two	Common
<i>SSEP-Domain</i>	91.9%	98.5%	77.3%	91.3%
CONSENSUS	91.0%	94.6%	81.1%	90.0%
GINZU	89.3%	93.4%	81.5%	89.3%
RosettaDOM	89.9%	92.5%	82.6%	89.1%
SSEP-CAFASP	88.9%	94.4%	76.7%	88.3%
DomSSEA	81.5%	95.5%	73.7%	88.0%
DOMPRO	87.7%	95.5%	70.2%	86.8%
DPS	78.2%	92.8%	73.4%	86.2%
ADDA	85.0%	93.5%	69.2%	85.2%
DOPRO	85.0%	88.0%	76.6%	84.1%
GLOBPLOT	75.4%	88.1%	64.5%	80.1%
InterProScan	76.0%	83.9%	61.2%	76.1%
MATEO	73.2%	78.7%	66.7%	74.6%
BIOZON	62.2%	61.3%	67.0%	63.3%
Armadillo	49.7%	49.6%	63.2%	54.2%

Table 5.3: Average overlap score of the CAFASP 4 predictors and SSEP-Domain for the *CAFASP*, *Single*, *Two*, and *Common* sets. Predictors were ranked by average overlap score on the *Common* set descendingly. For a description of the underlying algorithm, please see section 5.3.2.

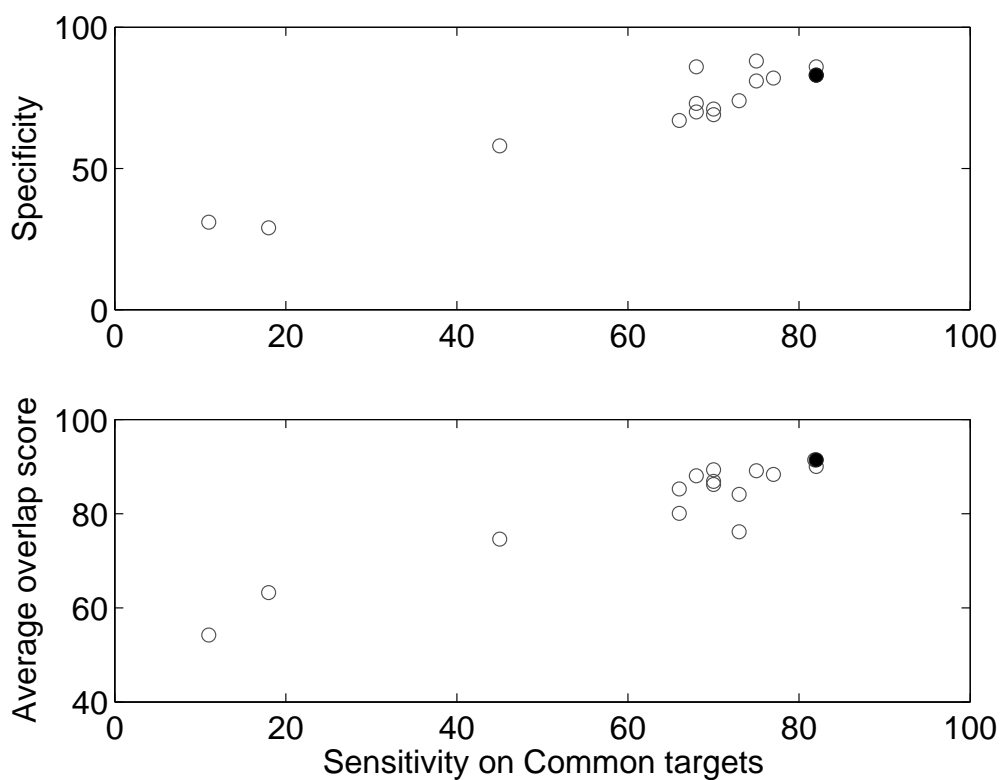


Figure 5.5: Sensitivity on the *Common* set vs. specificity on the *single & two* class (upper panel), and sensitivity vs. average overlap score (lower panel) on the *Common* set. Each "o" in gray represents one CAFASP participant. The results of SSEP-Domain in its final version are marked by a circle in black. SSEP-Domain achieves high sensitivity (81.8%, see Table 5.1), high average overlap score (91.3%, see Table 5.3), and good specificity (83%, see Table 5.2).

5.3.3 Evaluation of InterPro and Combination with AutoSCOP

Influence of InterPro on the CAFASP 4 data

Direct comparison between InterPro as evaluated by CAFASP 4 and SSEP-Domain on *CAFASP* shows that we gain about 16% in average overlap score and about 10% in sensitivity by combining InterPro with our alignment-based approach. For our evaluation, we used InterProScan on InterPro 7.2 (March 25th 2004), which contains member databases with dates ranging from September 2003 to March 2004. A pattern occurs as part of the highest-scoring domain combination for 19 of the targets, and patterns lead to different predictions with respect to the predicted domain number than alignments alone for two of the 58 targets. In both cases, the alignment-based prediction would have been wrong with respect to the CAFASP 4 assignment.

Combining SSEP-Domain and AutoSCOP

Given the found InterPro patterns, we computed the CAFASP results we would have achieved had we included the AutoSCOP matching into SSEP-Domain. The corresponding fold recognition evaluation can be found in chapter 4. We evaluated two possible ways of including AutoSCOP into our domain prediction mechanism:

1. When comparing SSEP-Domain directly with an SSEP-Domain version that would have used an AutoSCOP match whenever possible as domain region, we find that in none of the cases AutoSCOP would have changed the predictions. However, there are obvious problems when directly applying AutoSCOP to domain prediction. Some exemplary cases illustrate these problems:
 - **T0200:** For this target (length 255), we find two patterns, namely one Pfam pattern from 0-177 and one SUPERFAMILY pattern from 0-254, both of which are unique for the correct fold class. The target is single domain both by CAFASP 4 and by the ASTRAL 1.69 definition. For this target, when using only AutoSCOP, it would be difficult to say whether the Pfam pattern indicates that there might be a second domain at the end of the target (which would be wrong) or the SUPERFAMILY pattern is correct in matching the whole sequence (which is correct). Due to the gap mechanism chosen by SSEP-Domain, the SUPERFAMILY variant would be chosen, which matches the prediction made by alignments only.
 - **T0204:** This target is single domain according to CAFASP 4 and two-domain according to ASTRAL 1.69 with both domains belonging to the same fold class. AutoSCOP correctly finds this fold class, but with differing region matches for the corresponding unique patterns: SUPERFAMILY indicates two domains, PRODOM only finds one with a large gap, and TIGR finds one with a match that practically covers the whole sequence. In addition, a non-unique pattern match from PIR is found, which also covers the whole sequence. SSEP-Domain

and its gap mechanism chooses the single-domain variant (which matches the CAFASP assignment), and the alignment-only prediction as well as the SUPERFAMILY version would correctly find the ASTRAL assignment. Again, using AutoSCOP directly without SSEP-Domain, the choice between single- and two-domain would be difficult.

- **T0235:** Here, CAFASP says two-domain and ASTRAL 1.69 says single-domain. Both alignment-based prediction and patterns result in two-domain predictions. One domain is assigned the correct fold class (as judged by ASTRAL) from both AutoSCOP and alignments. For the remaining predicted domains, the alignment hit is weak, but AutoSCOP clearly assigns a different fold class based on a SUPERFAMILY match. When looking at the predicted boundaries, the CAFASP boundary is at position 120, the boundary induced by the SUPERFAMILY patterns is 90, and the alignment-based boundary is 149 (i.e. both patterns and alignments miss the "correct" boundary by about 30 residues).

These cases show that the patterns found by AutoSCOP may be used but are not sufficient to find domain predictions for the whole sequence of a target, as the pattern boundaries for different unique patterns can vary strongly, and, as shown above, pattern boundaries not necessarily correlate with domain boundaries.

2. A second variant of including AutoSCOP into SSEP-Domain is to simply add the assigned fold class to the set of selected fold classes in Step 2 whenever a domain region contains a unique pattern hit. Also this version does not alter the predictions made by SSEP-Domain on the CAFASP data, as in all cases the unique patterns corresponded to those predicted regions that had already been chosen for the final predictions by the original SSEP-Domain algorithm.

Both variants did not improve the predictions but instead can in some cases even lead to inherently unclear situations such as described above. For this reason, we decided not to include AutoSCOP directly into SSEP-Domain, as described for variant (1), but to include AutoSCOP as described for variant (2), i.e. as an additional advisor for fold class membership when applicable.

For fold recognition purposes, it should be noted that, of the 23 evaluated targets with AutoSCOP predictions and annotated ASTRAL/SCOP fold classes, in seven cases the correct fold was not found as top-scoring hit by SSEP-Domain (in all cases only weak alignment hits were found), but could be predicted by AutoSCOP. Further, there is no case where SSEP-Domain would have been able to correct a fold prediction made by AutoSCOP. This again shows that AutoSCOP is a useful tool for further analysis of the predicted regions such as the assignment of potential SCOP classifications.

5.3.4 SSEP-Align: An Extension towards Structure Prediction

For the CASP 6 and CAFASP 4 experiments, as already mentioned above, a straightforward extension of the SSEP-Domain method for structure prediction was examined.

Ranking	Position
CAFASP 4, top ten models, all targets, MaxSub	rank 35 of 70
CAFASP 4, top ten models, all targets, TM-Score	rank 38 of 70
CAFASP 4, top ten models, fold rec. targets, MaxSub	rank 25 of 68
CAFASP 4, top ten models, fold rec. targets, TM-Score	rank 34 of 68
CAFASP 4, top ten models, homology mod. targets, MaxSub	rank 40 of 69
CAFASP 4, top ten models, homology mod. targets, TM-Score	rank 39 of 69
TM-Score evaluation on CASP 6, top 5 models, all targets	rank 34 of 60
TM-Score evaluation on CASP 6, top 5 models, easy targets	rank 41 of 60
TM-Score evaluation on CASP 6, top 5 models, medium targets	rank 29 of 60
TM-Score evaluation on CASP 6, top 5 models, hard targets	rank 13 of 60

Table 5.4: Results of the SSEP-Align extension for structure prediction on CAFASP 4 and CASP 6 targets. The CAFASP 4 ranking makes use of both MaxSub [Siew et al., 2000] and TM-Score [Zhang and Skolnick, 2004] as quality measures, whereas the second evaluation, which includes few human predictors, is completely based on TM-Scores.

As this extension, the SSEP-Align server, was mainly implemented and maintained by Alessandro Macri, we will describe it only briefly. SSEP-Align used a simple protocol: (1) predict domains using SSEP-Domain, (2) use predicted domain regions to compute PPA alignments against the template database, and (3) use MaxSprout [Holm and Sander, 1991] for postprocessing of C- α models obtained by copying the coordinates in the alignments. Whenever the domain prediction did not result in a significant hit, PSI-BLAST results were included instead.

All evaluations find SSEP-Align somewhere in the middle of the field of participating automated predictors, being ranked in the main bulk of structure prediction methods. Exemplary results from CAFASP 4 (downloaded December 3rd, 2004) and a TM-Score based ranking of CASP 6 (downloaded from <http://bioinformatics.buffalo.edu/casp6> on December 3rd, 2004; link posted by Yang Zhang in November 2004 on <http://forcasp.org>) are shown in Table 5.4. We find that, in comparison, the performance is better on more difficult targets (the top result is rank 13 on hard targets as defined by the second ranking). This can be explained by the observation that for "easy" targets, significant differences between predictors can result from different model refinement steps (which the experimental SSEP-Align server did not use), whereas for medium or hard targets, the focus lies more on finding a good template and generating suitable alignments.

Some important problems recognized during the experiments were the quality of the models (even when a correct template had been found) and the ranking of the resulting models (in many cases, the top performing model was not recognized by the server and accordingly not submitted as first model). The ranking of alignments with respect to expected structural quality and the improvement of the final structural quality by tuning the alignment process itself will be discussed further in chapter 6.

5.3.5 Other Possible Extensions

Consensus Pattern Scoring

One possible extension that we evaluated are so-called *consensus patterns*. Here, we simply assigned consensus scores to regions found by pattern searches: When a pattern region was also found by other patterns (with a difference of maximal 5 residues at each end), we added the corresponding scores to the original pattern score. This allows to assess the consensus of pattern hits directly. For our test data, this does not alter the predictions. Nonetheless, we believe that the consensus score for a pattern region can give a good impression on the reliability of a predicted pattern region. Therefore, we included consensus pattern scoring as an additional option for our approach.

Inclusion of Unknown Domains

In our scoring mechanism, we do not allow unassigned regions of length 50, and the minimal score we assign to potential domain region is 0 (this happens very rarely but is possible), which in turn reduces the score of a domain prediction containing this region also to 0. This can potentially be harmful in cases where we have clear hits in combinations with a region that is scored with 0, as then the good hits may be missed because of an "unknown" and potentially new fold domain.

This problem is not easy to solve. One possible way would be to score each region by at least a minimum score, say 0.1 or 0.2. Another, more elaborate way, would be to also invoke an ab initio test for globality on potential regions, resulting in a hybrid method between the template-based SSEP-Domain and additional algorithm. In this case it would be necessary to find a suitable way to let such an ab initio derived score override a low PPA score without including too many false positives.

In SSEP-Domain, we evaluated the first version using both 0.1 and 0.2 as minimum score for all evaluated regions. This has the effect that, for a score of 0.2, for instance, if we have two good hits and one "unknown" domain in between, the multiplication of the score of the two hits will be relatively high, and another multiplier of 0.2 will only reduce this to an overall score of one fifth of the combined score of the two hits. As an example, if we have two hits of scores 5.0, we would achieve $5.0 * 5.0 * 0.2 = 5.0$ as overall prediction score, which has a reasonable probability of being chosen as final prediction. In our evaluations, introducing a minimum score of 0.2 had no effect on our predictions on the CAFASP 4 and, in fact, in anticipation of the next section, also no effect on the CASP 7 data; nonetheless it is a possible option.

The second option we did not pursue so far, but a clever integration of ab initio methods into our prediction process may be a good point for future research. However, as we can see in the tables presented above, where the best method classified as "ab initio" by its authors (DOMPRO) is clearly below SSEP-Domain on both single- and two-domain targets, such an integration may be difficult to parameterize in order not to lose accuracy.

5.4 Two Years After: CASP 7 and its Lessons

In order to further test the ability of SSEP-Domain, which was developed mainly in 2004, we submitted the server predictions as part of the LMU predictor group for the CASP 7 experiment, which was held from May 2006 to August 2006. The aim of participating in this evaluation was to again make use of independent assessors in a blind-test situation to figure out drawbacks and necessary extensions of our original algorithm. As expected, we found the situation in CASP 7 different for us in comparison to CAFASP 4, for reasons explained below, and therefore we will discuss and to some degree evaluate possible extensions of our algorithm based on the lessons learned in CASP 7.

5.4.1 Analysis of CASP 7 Results

We start with the discussion of the CASP 7 experiment. The only changes between the version used for computing the CAFASP 4 results and the one used two years later in CASP 7 are the databases: For CASP 7, we used ASTRAL 1.69 together with some CAFASP 4 targets as alignment templates and InterPro 12.1 as pattern database (in both cases the latest releases available at the beginning of CASP 7). However, in comparison with the CAFASP 4 situation, we find that in CASP 7 our approach had to face one major drawback: In CAFASP 4, discontinuous domains were not considered for the evaluation. This matched our approach, since SSEP-Domain is not able to predict discontinuous domains by definition. In contrast, in CASP 7, many targets have been assigned discontinuous domains. Thus, CASP 7 was a challenging evaluation for SSEP-Domain. The results are as follows:

- **All predicted targets:** Of 95 targets, 71 have been assigned the correct domain number according to the CASP 7 assessor's definition (74%).
- **Single domain targets:** On single domains, 65 of 68 are assigned correctly (95%) with a specificity of 76%.
- **Two-domain targets:** On two-domain targets, only 6 of 23 have been found by SSEP-Domain(26%) with a specificity of 60%.
- **Multi-domain targets:** Of the two three-domain targets, none is assigned correctly by SSEP-Domain (one is assigned two domains on the basis of a clear SCOP hit, and the second is assigned a single domain on the basis of a TIGR pattern hit).
- **Performance of InterPro patterns:** Of those 9 targets which were predicted wrongly without a clear SCOP hit, in three cases we do not find patterns: T0342, T0347, T0372. In another three cases InterPro patterns without alignments fall into at least one correct region of the target: T0321 (length: 251, Pfam pattern: 148-228, CASP 7: (1-96), (97-251)), T0334 (length: 530, Pfam pattern: 6-494, CASP 7: (3-528)), and T0386 (length: 299, Pfam pattern: 67-204, CASP 7: (13-218), (219-299)). In two cases InterPro patterns do not match the correct regions:

- T0299 (length 180): a Pfam hit covers 1-137, but the CASP 7 definition is (1-78,168-180) for domain 1, and (79-167) for domain 2.
- T0301 (length 395): a Pfam hit covers 5-391, but the CASP 7 definition is (1-182,378-395) for domain 1, and (187-377) for domain 2. Here, the alignments alone would have been much better with (1-168) and (169-395) as the predicted domains.

In the final case, T0356 (length 505), different patterns clearly disagree: here, we have a TIGR pattern (8-458), a Pfam pattern (12-437) and two SUPERFAMILY patterns (168-260, 327-446). The corresponding CASP 7 reference definition is (7-96,314-347), (122-313) and (348-467). The closest agreement here is between SUPERFAMILY and CASP, but still only two domains are more or less recognized with wide gaps in between.

- **Inclusion of AutoSCOP:** As for the CAFASP 4 evaluation (see 5.3.3), inclusion of AutoSCOP would not have improved the SSEP-Domain predictions with respect to CASP 7 assignments in any case: For all three targets where an improvement might have been possible (T0321, T0334 and T0386, see above), the found patterns could not be matched to a SCOP fold.

If we compare directly against the CASP 7 assignments, we fall short of the CAFASP 4 result with only about 75% correct assignments (as compared to about 82%) due to a worse performance on multi-domain proteins. In particular, SSEP-Domain tends to predict fewer domains on the targets in comparison to the CASP 7 assessors.

Based on this evaluation, we can identify two major issues for SSEP-Domain in the CASP 7 experiment: (1) the tendency towards too few domains, and (2) the algorithm not being able to predict discontinuous domains. We will therefore discuss a possible extension for SSEP-Domain that aims at reducing the impact of these drawbacks by including alternative domain definitions in the following.

5.4.2 Using Alternative Definitions for SCOP Domains

We evaluated all wrong single-domain predictions for which we had clear SCOP hits by aligning the predicted template structure against the target structure, if available. Of 16 cases found with SCOP hits over a neutral score of 1.0, we could find structures in the PDB for 14. On these 14 targets, the alignment with the structural alignment method Vorolign and the corresponding TM-Score as a measure of structural similarity (see also chapter 6) could confirm that the templates had been chosen correctly in almost all cases: For all we found a template with a TM-Score over 0.5, for 12 (85%) we found a template with a TM-Score over 0.7, and still for 7 (50%) we found a template with a TM-Score over 0.8. According to the authors of the TM-Score (which ranges between 0.0 and 1.0), a score of 0.4 is already a significant threshold for structural similarity. A score above 0.7 can be considered an indicator for clear structural similarity. Therefore, if we are conservative

with respect to the TM-Score, for 12 more targets our predictions can be considered as correct if we use SCOP definitions and not CASP 7 definitions. This means an increase in sensitivity of more than 12%, i.e. SSEP-Domain would then achieve a sensitivity on all targets of 86%. Correspondingly, also the sensitivity on two domain targets would be increased to 54%.

As a side-effect, this evaluation also shows that the scoring used by SSEP-Domain is a good indicator for structural similarity, as none of the regions with a score above neutral level (i.e. above 1.0) in this comparison was found together with a TM-Score below 0.4 in the corresponding structural alignment.

However, the main result of this evaluation is that in many cases SCOP hits disagreed with the CASP definitions by assigning too few domains. As a possible alternative, in the following, we will therefore evaluate automated domain assignments made by programs such as DomainParser (DP) and PDP, which are known to predict more domains on average than assigned by SCOP, on the SCOP template domains.

Evaluation of Automated Assignments

As a first evaluation, we computed PDP assignments for each SCOP domain in our template set. Of the 11950 domain structures downloaded from ASTRAL 1.69, PDP splits 2054 into smaller domains (17%). In 1361 cases (about 11%), the PDP assignments contain discontinuous domains. This shows a clear disagreement between ASTRAL and PDP with respect to domain definitions; apparently methods such as PDP make it possible to generate alternative, multi-domain predictions based on already existing SSEP-Domain predictions by further splitting SCOP domains.

In the same evaluation, DomainParser splits 1289 ASTRAL domains into smaller domains (10%), and in 602 cases these splits result in discontinuous domains (5%). It seems that DP is more moderate than PDP, in the sense that it lies somewhere between SCOP and PDP with respect to multi-domain assignments. This is in agreement with the observations made by [Holland et al., 2006] in their comparison of automated domain assignment methods.

We then evaluated the accuracy of both DomainParser and PDP on the CASP 7 targets with respect to the assigned domain numbers, in order to find out how well these methods agree with the CASP 7 assessors. On the 89 targets where a structure was found in the PDB, for DomainParser we find 8 differing assignments, and for PDP we find 11 differing assignments, i.e. an agreement of 91% and 87% with the CASP assessors, respectively. We cannot use SCOP on the CASP 7 data, as the targets are not available there yet.

On the CAFASP 4 data, however, we already know SCOP definitions for some of the targets. When compared with the CAFASP 4 domain definitions, we find that DP and SCOP have 10 disagreements and for PDP we have 11 disagreements on a set of 46 targets which we could find in ASTRAL 1.71. This means that we observe 78% similar definitions at maximum (SCOP and DP). In 9 of the observed ten cases, SCOP assigns too few domains with respect to CAFASP 4. PDP tends towards assigning too many domains (9 of 11 cases), and for DP both cases happen equally often (5 times).

In summary, both DP and PDP agree better with the CASP 7 assessors than with the CAFASP 4 assessors (up to 91% as compared to 78% similar assignments). Further, on CAFASP 4, we could again observe that SCOP tends to assign too few domains with respect to the assessors, whereas PDP tends to assign too many domains.

Inclusion of Other Assignments into SSEP-Domain: SSEP-Domain*

Apparently, PDP and DP provide interesting alternatives to SCOP definitions, especially with respect to the CASP 7 assessment. However, as observed by [Veretnik et al., 2004], SCOP is the most precise standard with respect to boundary placement, which is a property we would like to keep. When considering the precise boundary placement together with the results from our evaluations as well as from [Veretnik et al., 2004] (see chapter 2), we can infer that our main problem when using SCOP templates in structure-centered contexts such as CASP 7 is that SCOP regions are sometimes split further by other experts.

Therefore, one possible way to overcome SSEP-Domain's tendency towards few domains, e.g. in order to concentrate more on purely structure-based domains instead of SCOP domains, can be to use SSEP-Domain to find SCOP domain templates and then use alternative definitions for the found templates that are more likely to represent structural or even discontinuous domains.

We only consider this option when we have hits above neutral level, i.e. with scores above 1.0 (see Step 3). This can never reduce the number of predicted domains, but it is possible that one SCOP domain is split into two or more (even discontinuous) domains by other methods. This extension we call SSEP-Domain*.

In a first evaluation on the CAFASP 4 data (see Table Table 5.5), when we have templates with sufficient scores, we use both the Domain Parser (DP) and the PDP assignments for the domain region on the corresponding template structure. We observe that, while DP results in a slightly improved performance (one additional target is predicted correctly and the rest of the assignments remains untouched), PDP actually results in a worse performance with respect to the CAFASP 4 assignment, with a tendency towards multi-domain predictions.

The corresponding results on the CASP 7 data are also given in Table 5.5. We find that the use of DP assignments helps to increase the sensitivity on two-domains and also increases the general agreement with the CASP assessors. We find disagreements between SCOP and DP in 15 of 60 cases when our procedure was applicable. 9 times the prediction was improved with respect to the CASP assignment, 4 times it was made worse, and for two targets both variants were wrong, resulting in 76 correct predictions (80%). When using PDP instead of DP, our results are not as good as for DP: in 9 cases we get better, in 7 cases we get worse, and in 2 cases we change one wrong prediction into another, resulting in 73 of 95 targets predicted correctly (76%). Therefore, from both CAFASP 4 and CASP 7 it seems that DomainParser is the better solution for our purpose.

Method	Accuracy All targets	Sensitivity Single	Sensitivity Two
SSEP-Domain (CAFASP 4)	81%	96%	53%
SSEP-Domain*, PDP (CAFASP 4)	75%	86%	53%
SSEP-Domain*, DP (CAFASP 4)	84%	96%	60%
SSEP-Domain (CASP 7)	74%	95%	26%
SSEP-Domain*, PDP (CASP 7)	76%	85%	65%
SSEP-Domain*, DP (CASP 7)	80%	89%	65%

Table 5.5: Comparison of SSEP-Domain and SSEP-Domain* on CAFASP 4 (the Common set) and CASP 7 data. We find that, we can gain accuracy by including automated domain assignments, especially when using DP. Therefore, as an alternative with a tendency away from SCOP and towards smaller and more domain assignments, SSEP-Domain* based on DP may be used instead of SSEP-Domain alone.

Using CATH instead of SCOP

As a final test, we used CATH 3.0 (which is the latest version available before CASP 7), also reduced to 95% sequence identity. With these templates, we achieve one more correct prediction than SSEP-Domain using SCOP (i.e. without refinement by DP or PDP) with respect to the predicted domain number. However, the CATH-based predictions include more continuous two-domain predictions for targets which contain one domain surrounded by another (so-called discontinuous nested domains), i.e. these predictions are not correct either although the predicted number of domains is correct. Using SCOP, where we often find clear single-domain hits, we have the chance to find such architectures by applying DP or PDP, as shown in the next subsection. Using CATH, having assigned an incorrectly continuous two-domain prediction, this is not possible anymore. Therefore, for our purpose, SCOP as in SSEP-Domain as well as SCOP and DP as in SSEP-Domain* are a reasonable choice.

5.4.3 Discontinuous Domains

From the assessor’s talk at the CASP 7 conference, it seems that, especially with respect to discontinuous domains, the most promising approaches in CASP 7 built structure models first and then assigned domains based on these models, as can be done by programs like Domain Parser (DP) or Protein Domain Parser (PDP), for instance. This is the opposite direction of what SSEP-Domain wants to achieve, namely the quick prediction of domains from the sequence, in order to facilitate modeling.

Nonetheless, by using PDP or DP on the template side, as described for SSEP-Domain*, we not only improve our accuracy with respect to the predicted domain numbers, but we can now make discontinuous domain predictions for some of the CASP targets, although this can only be a first step towards the reliable prediction of discontinuous domains. In

particular, on the CASP 7 data, for SSEP-Domain* using PDP we have 12 discontinuous domain predictions, 9 of which meet discontinuous definitions made by the CASP assessors, 7 of which do reflect the assessor's definitions well. For DP, we have 8 discontinuous predictions, 7 of which meet discontinuous definitions made by the assessors, and all of these 7 are in agreement with the CASP definitions.

When taking a closer look at the data, in order to explain some of these differences, we find that, also for seven CASP 7 targets, our top templates lie in the SCOP superfamily c.108.1. Interestingly, this superfamily is described by SCOP as "contains an insert alpha+beta subdomain". Both DP and PDP when used as refinement of the SCOP definitions can capture five of these seven cases.

Other examples for targets with single domain predictions based on good SCOP single-domain hits whose descriptions allow for discontinuous multi-domain interpretations are T0323 (captured by both DP and PDP, fold class a.96, defined as single domain but "multihelical, consists of two all-alpha domains") and T0333 (captured by both DP and PDP, fold class c.87, defined as single domain but "consists of two non-similar domains"). The reason for such SCOP definitions is probably the underlying understanding of the term "domain", as SCOP defines domains as evolutionary units (see 2.2.2).

5.5 Independent Applications and Evaluations

SSEP-Domain is available as a web server. After the method had been published, some other groups have used it and to some extent evaluated it. Two recent papers that discuss SSEP-Domain are:

- [Kim and Patel, 2006]: In a recently published study SSEP-Domain was used as part of the proCC approach for protein structure classification and identification of novel protein structures by Kim and Patel. They observed that, in comparison to the SCOPmap approach [Cheek et al., 2004], "the SSEP-domain prediction method performs better than SCOPmap in identifying single domain chains." Kim and Patel further state that "on average 8 minutes" were "spent on the SSEP domain prediction web service" per target, which agrees with our own runtime evaluation as described above.
- [Sikder and Zomaya, 2006]: An independent comparison which includes SSEP-Domain has been described by Sikder and Zomaya. In this comparison, the following methods were used: SSEP-Domain, DomPro, DomPred, CHOP [Liu and Rost, 2004], Galzitskaya et al. [Galzitskaya and Melnik, 2003] and the two proposed methods, DomainDiscovery and Improved DomainDiscovery. Sikder and Zomaya observed that SSEP-Domain's "performance is superior for single and two-domain chains but inferior for three-domain or larger chains. SSEP-Domain also shows a precise placement of domain boundaries [...]" In particular, they found that the "SSEP-Domain method appears to be the most precise in placement of its boundaries."

Algorithm	Purpose
SSEA	Search for topological similarities; speeds up scoring
Coil Centers	Avoids close-by, low-scoring potential boundaries
Length filter	Restricts search to templates of similar length; improves precision and speeds-up the scoring
Score filter	Discards low-scoring SSEAs in Step 1
PPA	Final scoring of potential domain regions
Normalization	Prepares scores for multiplicative scoring
InterPro	Finds members of known sequence families

Table 5.6: Main algorithmic ingredients of SSEP-Domain and the corresponding purpose in the domain prediction pipeline. Details are given in the text in the corresponding sections.

5.6 Discussion

SSEP-Domain is an alignment-based approach to domain prediction (see Table 5.6 for an overview of the contained algorithms). We combine secondary structure element alignment and direct boundary placement to detect potential domain boundaries on a target sequence. Domain regions are deduced from these boundaries and an InterPro pattern search. They are evaluated using a combination of secondary structure element alignment and profile-profile alignment on both sequence and secondary structure. The combination of multiple domain regions is done using a simple recursive algorithm based on the scores of the individual regions, including InterPro patterns found on the target sequence. For this approach, we observe an average runtime of less than 10 minutes per target on the CAFASP set with a maximum of 18 minutes on an Intel Xeon with 2.8 Ghz. The evaluation of the influence of InterPro patterns shows that the combination of our alignment-based approach with InterPro patterns is indeed beneficial for domain prediction.

SSEP-Domain has been tested in the blind test scenario of CAFASP successfully, being part of the top group of domain predictors. Since features were added to the server during and after the experiment, we evaluated the final version under CAFASP 4 conditions. This gives us the opportunity to compare our results to the CAFASP predictors. SSEP-Domain performs well, achieving high sensitivity, high overlap scores, and good specificity. The final version yields the best overall accuracy of domain predictions as measured by overlap score due to an improved performance over the preliminary version. Direct comparison with other CAFASP participants shows that SSEP-Domain performs very well on single-domain proteins, but three of the other 14 methods (RosettaDOM, GINZU, and the CAFASP CONSENSUS meta-predictor) have higher overlap scores on two-domain proteins.

Recently, independent evaluations of the SSEP-Domain server confirmed that it is capable of placing domain boundaries quite precisely and works well in comparison to other methods (some of which have not been evaluated in this chapter) on single and two-domain targets [Sikder and Zomaya, 2006, Kim and Patel, 2006].

In 2006, two years after CAFASP 4, the SSEP-Domain algorithm was used on the CASP 7 targets. Here, SSEP-Domain performs worse than on CAFASP 4, especially on multi-domain targets, if we compare against the CASP 7 assessors' assignments. If we compare against clear SCOP hits, however, SSEP-Domain performs better than on the CAFASP 4 data. The different performance of SSEP-Domain with respect to CASP 7 and with respect to SCOP on the CASP 7 data matches the observations made by Veretnik et al [Veretnik et al., 2004], who find that SCOP tends towards few and continuous domains.

A simple combination of SSEP-Domain with DomainParser (SSEP-Domain*) improves sensitivity on multi-domain proteins as well as overall accuracy with respect to the official CASP 7 assignments. Although these results are positive, this combination should be handled with care, as it alters the SCOP definitions and is therefore a step back from the gold standard we used for training our method. Further, our evaluations show that different experts will assign to some degree different domains, and that SCOP, for instance, seems to agree better with the CAFASP 4 assessors than with the CASP 7 assessors.

From what we have seen in this chapter, we would expect any "gold standard" database of domain assignments to contain many entries that can be discussed and where different viewpoints will lead to differing domain assignments. Again, this agrees with Veretnik et al.'s observations, who conclude that "caution is recommended in using current domain assignments" [Veretnik et al., 2004].

So far, with the variants (1) SSEP-Domain and (2) SSEP-Domain*, we described methods that are (1) based on SCOP, and (2) revised with DP in order to account more for structural domains instead on evolutionary units, respectively. Both methods achieve good results for their respective scenarios (SSEP-Domain with respect to SCOP, and SSEP-Domain* with respect to the assessors' definitions in CAFASP 4 and CASP 7). An alternative to SSEP-Domain* and similar, automated approaches which contains manual intervention and will to some degree be driven by intuition would be to manually choose the domain assignment one likes best whenever the used sources (e.g. SCOP, PDP, DP, etc.) differ.

Another important issue which has become clear with the CASP 7 evaluation is the prediction of discontinuous domains. With respect to SCOP, not being able to detect discontinuous domains is not as much a hindrance as one might think, as SCOP assigns such definitions only in very few cases to PDB chains (see chapter 2). However, when trying to identify structural instead of evolutionary subunits of proteins, as the CASP 7 assessors probably did, discontinuous definitions obviously become far more frequent. At the moment, predicting such domains with SSEP-Domain is not possible, which is a drawback we share with most of the evaluated methods in this chapter. A first step towards the prediction of discontinuous domains has been proposed with the SSEP-Domain* variant, which is able to generate discontinuous domain predictions to some degree.

The SSEP-Domain server, which is based on SCOP domain templates, is available at <http://www.bio.ifi.lmu.de/SSEP>. On average, the computation time for a target is less than ten minutes (Kim and Patel observed about 8 minutes per target as the average response time in their independent evaluation of our server [Kim and Patel, 2006]), which is quite fast as some other methods may require hours or even days.

With AutoSCOP and SSEP-Domain we proposed useful methods for protein domain recognition for different purposes: very fast, but not necessarily very accurate; and still quite fast, but also as accurate as the state-of-the-art as measured by CAFASP 4. As an outlook, a combination of AutoSCOP and SSEP-Domain would be applicable to larger numbers of targets: Given many targets of interest, by first assigning structural classifications quickly with AutoSCOP as described in the previous chapter, which can be used as good initial positions for potential domains, we can identify the "clear cases"; then, for the reduced set of ambiguous or difficult targets, SSEP-Domain can be applied to refine AutoSCOP's annotations.

Chapter 6

Environment-Specific Alignment Computation and Scoring

As described in chapter 2, an important step to building a complete all-atom model is often to align a *target* sequence of unknown structure to a database of *template* sequences with known structures. On the basis of these alignments and the underlying known template structures, models are built and refined. In a blind-test-like setup, it is helpful to be able to select potentially good models already on the alignment level, before having to build each model. Further, similar to the identification of good models based on different templates, finding the best sequence-structure alignment in a pool of alignments even coming from the same source remains an interesting problem. This is being observed in the CASP and CAFASP experiments regularly, where it happens often that the model ranked best by the predictor groups themselves out of the, say, five submissions for a target is not the best of the five models in the final assessment. In this chapter, we want to find out whether alignment scores correlate well with the final model quality and how this correlation may be improved. We evaluate and optimize alignment scores with respect to the correlation to a quality measure on subsequently generated structure models. Thereby, the prediction setup using targets of unknown structure restricts our efforts to sequence-based alignment scores instead of including structural properties.

In addition, we optimize alignment parameters including scoring matrices for the log average profile-profile alignment approach. In particular, this optimization is based on the hypothesis that it is possible to tune parameters to specific *environments* for individual users and thus improve performance over parameters that have been tuned more conservatively to work well for most users instead. Thereby, we define the *environment* as those tools and databases a structure predictor will eventually concentrate on for his/her work, by which we mean the alignment programs, the scoring schemes, the template data, the evaluation mechanisms and everything else he/she has chosen to finally make the predictions. Default configurations of software as well as standard mechanisms are meant to work well for the average case but are not necessarily perfect for a specific environment, whereas including specific properties of an alignment computation process may help to improve prediction quality as long as these properties are kept unchanged.

We have developed an alignment-scoring software package called QUASAR which will be described in the first section of this chapter. The QUASAR framework is joint work with Fabian Birzele who implemented the main parts of the QUASAR package as a research student working for the author. The contents of the corresponding section are based on our Bioinformatics publication on QUASAR which appeared in 2005 [Birzele et al., 2005].

Based on the infrastructure provided by this package, in the subsequent sections, we analyze how well sequence-based alignment scores correlate with structure-based scores. We evaluate the performance of optimized linear combinations of well-known matrix-based scores for the task of alignment ranking in comparison to the individual performances of these matrices. The results show that combinations can be tuned to approximate the behavior of certain benchmark scores, based on suitable training data.

In order to adapt scoring to specific environments, we further generated optimized matrices for alignment ranking using a genetic algorithm adapted for this purpose. Finally, by slightly modifying this genetic algorithm, we show that it is possible to find optimized matrices and parameters for individual fold classes for profile-profile alignment; i.e., once the fold class of a target is known, it is possible to generate better alignments on average with respect to the resulting structure model by using fold-class specific parameters generated by the proposed procedure.

6.1 The QUASAR Framework

The QUASAR (QUALity of Sequence-Structure Alignments Ranking) system has been designed to fit two needs. First, it is a platform-independent and easily extendable software package for scoring and ranking sequence-structure alignments coming from different sources. Second, it aids the process of developing, benchmarking and optimizing new alignment quality measurements. The graphical user interface (GUI) of QUASAR provides quick access to each of the possible use cases and allows for visualization and comparison of the results as well as for configuration of all essential parts. QUASAR can also be used directly from the command-line.

6.1.1 Methods

Scoring Alignments

So-called *scoring schemes* represent alignment quality scores that are based on information that is available from amino acid sequences (e.g. predicted secondary structure) or that can be directly inferred from template structures. Scoring schemes provided by the system include several amino acid and secondary structure based exchange matrices (like PAM [Dayhoff et al., 1978] and [Luthy et al., 1991]), the two standard secondary structure fit measures Q3 and SOV [Zemla et al., 1999] as well as two contact-capacity-based scores [Berrera et al., 2003, Singer et al., 2002]. The number of available scoring schemes can be easily extended by implementing a Java interface, or, in the case of (amino acid exchange) scoring matrices, by adding a text file in a QUASAR specific format that contains the

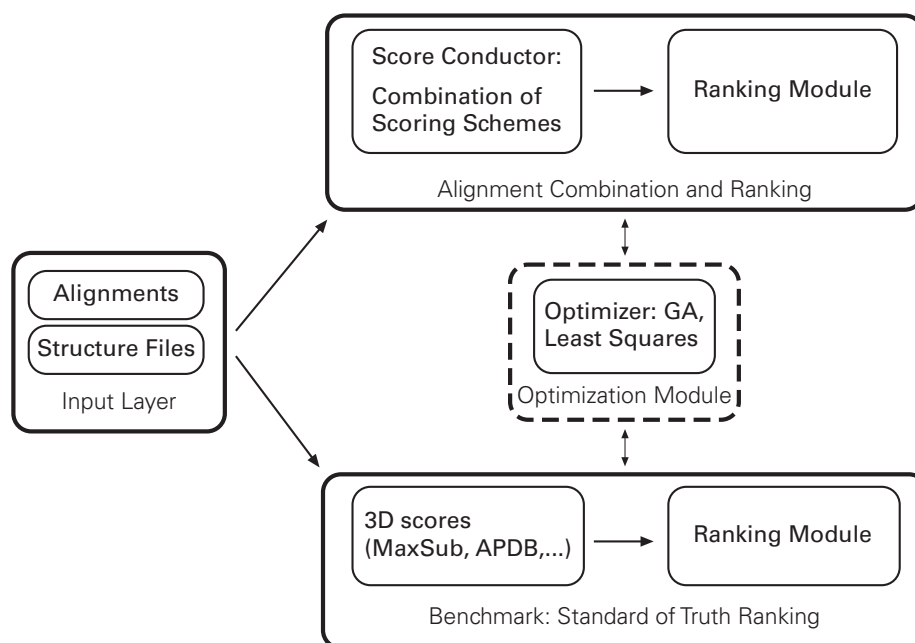


Figure 6.1: QUASAR reads protein alignments (input layer) and allows to evaluate the structural quality of the alignments according to built in and/or user programmed (Java) quality measures (ranking module). In addition, it supports to benchmark and optimize scoring functions, consisting of a weighted, linear combination of individual scoring schemes, with respect to a set of standard-of-truth (structural) alignments (optimization and benchmark modules).

matrix information. This provides a fast connection to matrix collections such as the AAIndex database [Kawashima et al., 1999], which has a very similar data format.

Combining Scores

With the so-called *score conductor*, the user can integrate several scoring schemes into one scoring function by combining the scores in a weighted sum (assigning user specified weights for the single scores), i.e. as a linear combination of the individual scores. In addition, by editing the configuration file, experienced users can build more complex, tree-like formulas using further operators like multiplication and division. Therefore, a user can test different combinations of scoring schemes with a minimal effort in order to improve the ranking quality over the performance of the single scores. The final quality score of every alignment is calculated by combining the single alignment scores according to the formula given in the configuration. Single scores can also be normalized to range between zero and one to combine scores with different magnitudes.

Benchmarking Scores

To help the user find a scoring function that gives the best possible results, QUASAR contains a number of structure-based quality scores like Touch, APDB [O’Sullivan et al., 2003], as well as re-implementations of MaxSub [Siew et al., 2000] and TM-Score as described by [Zhang and Skolnick, 2004]. For a given alignment benchmark set for which the structures of query and template proteins are known, QUASAR measures the correlation coefficient of the ranking resulting from the specified alignment score with a structure-based benchmark measure (e.g. RMSD). It is also possible to use a user-defined quality score as a reference by annotating it to the alignments. This makes it easy to compare the performance of an alignment score or a combination of scores to a given “standard-of-truth” without the need to implement the score in Java.

Optimizing Scores

The performance of a scoring function depends heavily on the weights which are assigned to the individual scoring schemes. Thus, QUASAR allows optimizing these weights with respect to a benchmark set of alignments with assigned or computed standard-of-truth scores (see above). So far, two optimization routines are available. One may invoke least-squares optimization or use a rudimentary genetic algorithm to explore the space of possible score combinations. The fitness of a combination of scoring scheme weights is evaluated with respect to a benchmark set. Such an optimization may also uncover the main ingredients of an already well-performing score combination by leaving out unnecessary scores.

Implementation

QUASAR is completely implemented in Java (version 1.4+). It is freely available for academic users as standalone and as a Java Web Start application. All scoring schemes, scoring functions, benchmark scores and optimization routines can be configured in an XML like configuration file that can be generated using the GUI.

6.1.2 Use Cases

Benchmarking and Optimization

First, we discuss an interactive use case: Given a new scoring scheme, e.g. a new scoring matrix, one builds a benchmark set of alignments and loads the data into QUASAR. In QUASAR, one explores the performance of the new scoring matrix in comparison to and in combination with in-built scores. The evaluation is done with respect to the standard-of-truth benchmark scores available in QUASAR and with help of the visualization panel. One further improves the ranking performance by combining well-performing schemes and optimizing their weights using QUASAR’s optimization routines. Now, one saves the configuration for future use of QUASAR from the command-line.

Automated Alignment Ranking

A second, non-interactive use case is the *ranking of sequence-structure alignments*. Here, one already has an optimized combination of scores together with the corresponding QUASAR configuration at hand. Given a set of different sequence-structure alignments for a target (e.g. to different template structures), one includes the call of QUASAR using the configuration file into the structure prediction process and is thus able to e.g. discard alignments on the basis of the previously optimized alignment score automatically.

6.2 Optimized Score Combinations

There exists many scoring matrices for pairwise protein sequence alignments, and each of these matrices again can be used with different gap parameters and normalization methods. In our first evaluation, we compute the correlation of a number of well-known matrix-based scores used with parameters found in the literature. As a first method, we propose least-squares optimization of combinations of these scores with respect to TM-Score in order to combine the strengths of the different matrices.

6.2.1 Preliminaries

TM-Score: A Structure-Based Benchmark Score

The template modeling score (TM-score) developed by [Zhang and Skolnick, 2004] is an interesting alternative to the assessment methods used e.g. in the CAFASP 4 structure prediction experiment. It compares protein structure templates with predicted full-length models by extending the approaches used in Global Distance Test (GDT) and MaxSub (which was used in CAFASP, for instance).

The motivation for the TM-score comes from the well-known shortcomings of the root mean square deviation (RMSD) as a measure of quality for protein structure models. Since the RMSD is independent of the alignment coverage (i.e. the number of aligned residues), a low RMSD on few aligned residues is not necessarily better than a slightly higher RMSD on significantly more aligned residues when one aims at producing a good full-length model instead of modeling only parts of a protein. The precise formula is as follows:

$$\text{TM-Score} = \max \left\{ \frac{1}{L_N} \sum_{t=1}^{L_T} \frac{1}{1 + \left(\frac{d_t}{d_0}\right)^2} \right\}.$$

Here, L_N is the length of the native structure, L_T the number of aligned residues, d_t the distance for aligned residue t and d_0 a normalization factor. The maximum operator means that the optimal spatial superposition is chosen with respect to the final TM-Score. For our evaluations, as there are no dedicated targets and templates, for each aligned protein domain pair, we chose the roles randomly. Here, the target defines the native structure and the aligned positions are taken from the template as model.

Alignment Scoring with Scoring Matrices

Scoring a sequence alignment given a scoring matrix is simple. For each aligned pair one reads a substitution score from the matrix. The final score is the sum over the scores of all aligned positions, with penalties for gap opening (opening a region of unaligned residues) and gap extension (elongating a gap region) (see chapter 2). As our benchmark score is normalized, we make use of so-called *bit scores*. Here, we simply divide the scores as computed above by the length of the corresponding alignment. This reduces the influence of the sequence lengths and increases the average correlation with the benchmark score over the raw scores in our evaluations.

For our evaluation, we make use of an exemplary selection of 15 well-known matrices which were available in the AAINDEX database and for which we could find parameters optimized with respect to structural alignments in the literature. Listed here with the names we will use for them in the following, these matrices are BLOSUM50 and BLOSUM62 [Henikoff and Henikoff, 1992], PAM250 [Dayhoff et al., 1978], the BlakeCohen matrix [Blake and Cohen, 2001], the Gonnet matrix [Gonnet et al., 1992], the Johnson matrix [Johnson and Overington, 1993], the Miyazawa matrix [Miyazawa and Jernigan, 1993], the Overington matrix [Overington et al., 1992], the Prlic matrices [Prlic et al., 2000], the Risler matrix [Risler et al., 1988], SM_Sausage and SM_Threader [Dosztanyi and Torda, 2001], the STROMA matrix [Qian and Goldstein, 2002], and Gonnet_P [Vogt et al., 1995]. The matrices, the used parameters and their sources are shown in Table 6.1.

Training and Test Data

As database for our evaluations we used the ASTRAL compendium version 1.65, namely the subset reduced to 95% sequence identities without genetic domains, i.e. without domains that have been defined by ASTRAL by combining parts from more than one chain. This set contains 797 different folds, 1288 different superfamilies and 2315 different protein families.

We built a dataset using all folds having at least two members in the ASTRAL95, such that for each domain in the database there is at least one potential template that has similar structural properties. For each of these fold classes, we then aligned all members against each other using global log average profile-profile alignment with standard parameters on both sequence and secondary structure profiles (see 2.5.2).

As some fold classes contain huge numbers of alignments (for instance b.1 induces more than 350000 alignments), for each class above 10000 alignments, we randomly sampled 10000 alignments from the available pool. This avoids heavy overrepresentation of classes such as b.1, for instance; however, large classes will still be overrepresented. For our setup, we accepted this fact, as some classes are very small, which would have reduced our datasets severely when used as the reference size; further, in most template sets based on SCOP/ASTRAL, there will be an unbalanced representation of fold classes which is also reflected in our set.

All folds were then divided into a training and a test set by randomly selecting classes for the test set until more than 50 percent of all alignments were covered. This yields about 100000 alignments for testing and about 100000 alignments for training. A few alignments were discarded during annotation of the TM-Scores, e.g. because of differences between the ASTRAL sequences and the residue sequences in the corresponding coordinate files.

In the final step, we discarded all alignments with a TM-Score below 0.4, which has been described as a statistically significant threshold for structural similarity by the developers of the TM-Score [Zhang and Skolnick, 2004, Zhou and Skolnick, 2007]. Therefore, in the final set of alignments, we have 25348 alignments for training and 32602 alignments for testing.

Correlation Coefficient as Fitness Function

We are interested in a good ranking of a set of alignments with respect to our benchmark score. In other words, when an alignment would be ranked above others based on the TM-Score, for instance, we also want it ranked above them by our score. As fitness function for our ranking optimization procedures, we therefore use the Pearson correlation coefficient as defined in eq. 6.1 to measure the correlation of a set of alignment scores X and the corresponding set of benchmark scores Y where (x_i, y_i) represents the respective matrix score x_i and benchmark score y_i of alignment i and \bar{x} and \bar{y} are the average values of the set X and Y respectively:

$$\frac{\sum_{i=1}^n (x_i - \bar{x}) * (y_i - \bar{y})}{\sqrt{\sum_{i=1}^n (x_i - \bar{x})^2 * \sum_{i=1}^n (y_i - \bar{y})^2}}. \quad (6.1)$$

All correlations were computed using QUASAR.

One remark has to be made when using the correlation to a structure-based benchmark score as a fitness function: There are known cases when the highest structure-based benchmark scores does not necessarily represent the best possible alignments, especially if inherent protein flexibility is involved. Rigid structure superposition evaluation on Calmodulin, for instance, will probably not be able to correctly assign high scores to good models, if the template structure is in an open state and the target structure (used for benchmarking) is in a closed state. In the following, as we are in need of a score that gives us a hint on the structural quality, we will nonetheless use the correlation with the TM-Score, which has been shown to be a good measure for model quality in many cases and is a widely accepted score.

6.2.2 Generation of Linear Combinations

In order to combine the strengths of individual scores, we combine individual scores in a weighted sum.

Score Conductors

Such a combination is called a *score conductor*. Given m participating scores

$$s_i : \text{Alignments} \mapsto \mathbb{R}, i \in \{1, \dots, m\}$$

for an alignment $a \in \text{Alignments}$, the score s_{combined} for a conductor $c = (\bar{w}, s_1, \dots, s_m)$ is defined as

$$s_{\text{combined}}(c, a) = \sum_{i=1}^m w_i s_i(a) = \bar{w}^T \bar{s}_a,$$

where w_i denotes the individual weight for a score s_i as given by the score conductor, and $\bar{w}^T \bar{s}$ denotes the scalar product of an m -dimensional weight vector with an m -dimensional score vector.

Linear Optimization

In order to obtain a good combination of scores, i.e. useful weights for our individual scores, we tune our score conductors to approximate the benchmark scores: Given a benchmark score $s^* : \text{Alignments} \rightarrow \mathbb{R}$ and a training alignment set A^* , we compute the so-called *least-squares* weight vector

$$\bar{w}^* = \arg \min_{\bar{w} \in \mathbb{R}^m} \sum_{a \in A^*} (\bar{w}^T \bar{s}_a - s^*(a))^2.$$

This is a well-known problem which can be solved efficiently and which has been implemented in many software packages. Here, the computation of least squares weights was done using QUASAR's optimization routine based on JAMA¹.

6.2.3 Results

We scored all alignments in the test set with all matrices and computed the Pearson correlation between the resulting scores and the TM-Score as our benchmark score. The results are shown in Table 6.1. The best matrix is the SM_THREADER matrix with 0.7933. Also the standard matrices (PAM and BLOSUM) do quite well in comparison, their best result being 0.7615 (BLOSUM62). Another interesting observation is the high impact of the gap costs (all configurations were proposed in the literature): the Miyazama matrix ranges between 0.6611 and 0.7518 in its different setups. This underlines the importance of optimal parameters as well as the fact that such optimal parameters may vary with the situation.

We computed the least-squares combination of all matrix configurations used above on the set of training alignments. The resulting combination achieves a correlation of 0.8275 on the test data, which is an increase of 3.5 percentage points in comparison to the best individual scoring matrix as observed above. Apparently, the combination of matrices is useful for our task, as it can better approximate the ranking behavior of the TM-Score than any of the individual matrices in our setup.

¹<http://math.nist.gov/javanumerics/jama/>

Scheme	Correlation	GO	GE	Parameters in
BlakeCohen	0.7328	-20.2	-3.0	[Qian and Goldstein, 2002]
BlakeCohen	0.7254	-17.0	-2.0	[Blake and Cohen, 2001]
Blosum50	0.7538	-6.0	-2.0	[Prlic et al., 2000]
Blosum62	0.7422	-5.5	-0.8	[Prlic et al., 2000]
Blosum62	0.7481	-12.0	-1.0	[Qian and Goldstein, 2002]
Blosum62	0.7452	-8.4	-0.9	[Qian and Goldstein, 2002]
Blosum62	0.7615	-3.4	-3.0	[Qian and Goldstein, 2002]
Gonnet	0.7539	-8.5	-0.8	[Prlic et al., 2000]
Gonnet	0.76	-12.0	-1.0	[Qian and Goldstein, 2002]
Gonnet	0.7468	-14.2	-0.2	[Qian and Goldstein, 2002]
Gonnet	0.7614	-3.0	-2.9	[Qian and Goldstein, 2002]
Gonnet	0.7467	-14.0	-0.2	[Vogt et al., 1995]
Johnson	0.7388	-9.5	-1.2	[Prlic et al., 2000]
Johnson	0.7205	-31.0	-28.0	[Blake and Cohen, 2001]
Miyazawa	0.6906	-12.7	-0.52	[Dosztanyi and Torda, 2001]
Miyazawa	0.6611	-11.5	-0.22	[Dosztanyi and Torda, 2001]
Miyazawa	0.7518	-9.3	-0.66	[Dosztanyi and Torda, 2001]
Miyazawa	0.7305	-13.6	-1.18	[Dosztanyi and Torda, 2001]
Overington	0.7458	-12.0	-1.0	[Qian and Goldstein, 2002]
Overington	0.7351	-9.5	-0.5	[Qian and Goldstein, 2002]
Overington	0.7585	-3.6	-2.5	[Qian and Goldstein, 2002]
Pam250	0.7541	-10.0	-1.0	[Prlic et al., 2000]
Pam250	0.7556	-12.0	-1.0	[Qian and Goldstein, 2002]
Prlic	0.7515	-7.0	-0.6	[Prlic et al., 2000]
Prlic2	0.7412	-19.0	-0.8	[Prlic et al., 2000]
Risler	0.7835	-3.0	-0.2	[Prlic et al., 2000]
Risler	0.783	-5.0	-0.1	[Vogt et al., 1995]
SM_Sausage	0.4912	-10.9	-0.08	[Dosztanyi and Torda, 2001]
SM_Sausage	0.4885	-3.8	-0.51	[Dosztanyi and Torda, 2001]
SM_Sausage	0.4832	-4.9	-0.01	[Dosztanyi and Torda, 2001]
SM_Sausage	0.5064	-6.7	-1.69	[Dosztanyi and Torda, 2001]
SM_Threader	0.7897	-16.4	-0.4	[Dosztanyi and Torda, 2001]
SM_Threader	0.7884	-12.7	-0.22	[Dosztanyi and Torda, 2001]
SM_Threader	0.7887	-15.2	-0.24	[Dosztanyi and Torda, 2001]
SM_Threader	0.7933	-15.1	-1.13	[Dosztanyi and Torda, 2001]
Stroma	0.7785	-16.2	-1.1	[Qian and Goldstein, 2002]
Gonnet_P	0.7636	-6.0	-0.8	[Vogt et al., 1995]

Table 6.1: Evaluated scoring schemes and corresponding parameter settings taken from the literature. GO stands for "gap open" and GE stands for "gap extend". The highest values for each correlation column are highlighted.

6.3 Optimized Matrices for Alignment Ranking

As we have seen, it is possible to combine the strengths of well-known matrices in a linear combination. Such matrices, however, are not necessarily well suited for the environment they are used in. The hypothesis for this section is that, if the environment (i.e. the alignment method, the template database and perhaps additional features) for producing the alignments is known, it should be possible to generate optimized scoring matrices that can deal with the properties of these alignments.

Training has to be done on a set of training alignments that can reflect the behavior of the underlying method, and then one can expect, if the test data comes from the same method, that the resulting matrices can perform well on previously unseen alignments. The algorithm described in this section is meant for approximation of the ranking behavior of the TM-Score, using "bit scores", i.e. normalized alignment scores as described above. Nonetheless, it is general enough to optimize scoring matrices given a representative training set and a fitness function that corresponds to the desired behavior.

6.3.1 Comparison Matrices

Many studies exist which have proposed amino acid scoring matrices for differing purposes and with differing methodologies (see 6.2.1 for the list of matrices used in this section), some of them aiming at alignment quality with respect to structural alignments as reference data. Such matrices are especially interesting for us, as we expect them to reflect the structural quality of alignments (i.e. the structural quality of the resulting models) better than others. For comparison with our approach, we chose prominent studies of the latter type, whose matrices were available to us via the AAIndex database and which are also the sources for the optimized parameter settings for other well-known matrices as used in this chapter:

- **[Vogt et al., 1995]:** In the study by Vogt, Etzold and Argos, different matrices and parameter combinations are tested on a set of amino acid sequences matched by superposition of known topologies. The authors find "relatively similar results for the top scoring matrices, a preference for global alignment, and the importance of matrix modification and optimized gap penalties." This strengthens our point made at the beginning of this chapter, namely that it is possible (and sometimes important) to use optimized parameters for certain situations instead of the average, standard parameters, although these do work good in many cases. From this study, we took the highest scoring matrix ("Gonnet_P" in the table).
- **[Prlic et al., 2000]:** Prlic, Domingues and Sippl derived matrices "based on superimpositions from known protein pairs of similar structure" using a formalism based on the observed occurrence frequencies of aligned amino acid pairs. They compared them with other previously published matrices. Their results confirm that their matrices can be used for comparisons of distantly related sequences. We included the derived matrices into our own study ("Prlic", "Prlic2" in the table).

- **[Blake and Cohen, 2001]:** Blake and Cohen also built a new set of amino acid interchange matrices from structural superposition data based on log-odds probability ratios. They find both improved pairwise alignments as well as an increase in fold recognition accuracy. We used the top performing matrix ("BlakeCohen" in the table).
- **[Dosztanyi and Torda, 2001]:** Dosztanyi and Torda use low resolution force fields to derive amino acid substitution matrices. They computationally mutated residues and collected their contribution to the total score; based on the position-wise averages of these values, the substitution matrices were compiled. We included both the "SM_THREADER" and the "SM_SAUSAGE" matrix in different configurations in our evaluation.
- **[Qian and Goldstein, 2002]:** Qian and Goldstein computed a matrix they called STROMA by using a downhill simplex optimization and a RMSD-based merit function. We included the "STROMA" matrix, which was found to generate more accurate alignments than other compared matrices by their authors.

Further, in a recent study, Torda et al. [Torda et al., 2004] generated optimized substitution matrices for their WURST structure prediction server using Qian and Goldstein's approach, but with the difference that they included a structure-based scoring term used by the WURST server into their optimization procedure. As they state, their "substitution matrix is not a general substitution matrix, but rather a numerical creation, fitted to the influence of the structural score function"; in other words, their matrix is not meant to work on its own but was adjusted to the combination of scores used by the WURST server. For this reason, in our evaluations, we concentrated on Qian and Goldstein's optimization results instead (namely the STROMA matrix) [Qian and Goldstein, 2002], which are based on the same principal procedure but not restricted to use with Wurst's structure-based scores.

6.3.2 Range-Adaptive Genetic Algorithm

For our optimization approach, we employ a so-called genetic algorithm. The principle behind genetic algorithms (GAs) for optimization problems is in analogy to evolution: Populations of individual solutions to a posed problem are gathered, the best solutions are selected, combined and eventually mutated. After each generation, bad performing solutions are discarded and new potential solutions are generated via the mechanisms described below. This process is repeated until convergence or a limiting criterion has been reached. For a good introduction into GAs, please see [Mitchell, 1998].

In the case of scoring matrices, it is desirable to represent individual solutions as m -dimensional vectors of floats. Unfortunately, the main problem in using GAs for such a continuous optimization problem is the necessity of employing a very large number of chromosomes in the population, which demands extensive computer resources. One possible solution to this problem is to adaptively narrow the range of values for each parameter with

Algorithm 4 Pseudocode of the GPSSA algorithm.

```

1: boundaries ← initializeBoundaries()
2: for  $i = 1 : 100$  do
3:   population ← generateNewPopulationWithinBoundaries(boundaries)
4:   while generations < 500 AND stopping criterion not satisfied do
5:     evaluate fitness of every population member
6:     perform tournament selection
7:     perform mating and crossover
8:     perform mutation
9:   end while
10:  boundaries ← updateBoundaries(population)
11: end for

```

increasing number of generations. This concept, genetic algorithms with parameter-space size adjustment (GAPSSAs), was described and successfully applied to the problem of model parameter determination of the optical constants of metals by Djuricic and coworkers [Djuricic et al., 1997].

In this subsection, we propose a modification of this approach for optimizing the entries of an amino acid substitution matrix with respect to alignment ranking and model selection. The fitness function of this optimization method is again the Pearson correlation as defined in Equation 6.1 based on bit scores. The algorithm was used as follows:

Adaptation of the GAPSSA algorithm

An amino acid substitution matrix assigns a score to every pair of amino acids (m, n) which scores the substitution of amino acid m by amino acid n and vice versa. Since scoring matrices are usually symmetrical, a matrix consists of 210 values together with two additional parameters representing gap open and gap extend penalties. Therefore, a population member is a vector $\bar{w} \in \mathbb{R}^m$, with $m = 212$. Without loss of generality we assume that the initial matrix values are in the range of $[-1, 1]$ for these matrices.

As shown in Algorithm 4, the GAPSSA consists of two loops, namely an inner and an outer loop. In the inner loop, we use a classical genetic algorithm as described below. More interesting is the restriction of the parameter-space size done at the end of each inner loop: As described by [Djuricic et al., 1997], we narrow the boundaries for each parameter based on the corresponding average values in the population:

$$upperbound(k) = upperbound(k) - c(upperbound(k) - average(k))$$

$$lowerbound(k) = lowerbound(k) + c(average(k) - lowerbound(k)).$$

Here, k determines the parameter and the factor $c \in [0, 1]$ thereby regulates the speed of the convergence in this process. Before the beginning of a new inner loop, the best member is kept and the population is filled with new random members using the new boundaries. Our criteria for termination of an inner loop are either that the top-scoring member has not

changed for 50 generations or that we have reached a maximum number of 500 generations. For the outer loop, we used a maximum number of 100 inner loops in our setup.

Inner Loop Genetic Algorithm

In the selection phase, 50% of a population are selected by pairwise tournament selection. Random member pairs are formed and the member with the better fitness survives and enters the next generation, while the other chromosome is removed from the population. If the two members have the same fitness, the surviving member is chosen randomly.

After selection the rest of the population is filled with children of the surviving population members. For mating, two random population members are chosen uniformly from the population and their offspring is generated by a crossing over of the two chromosomes. In a crossover event, for each parameter of a child, we choose randomly between the corresponding values from its parents. This variant is referred to as "multiple-point" crossover.

The best solution is duplicated, and the copy replaces one randomly chosen child generated by the mating procedure. Then, with the exception of the original copy of the currently best solution, any parameter of any member in the population may mutate with a certain probability. Just like when initializing a new population member, the mutated value of the gene is determined according to the formula given above, i.e. within the current boundaries set for the parameter.

6.3.3 Results

Using a mutation rate of 0.4, 100 outer loops, 500 inner loops and a population size of 100 we performed 15 runs on the training data, each of which took about one day on a single personal computer. Figure 6.2 shows the results of the optimization procedure as a box plot. Overall the matrices optimized by the genetic algorithm perform well in comparison with the known matrices, though the best individual performance on the test data is still observed for the SM_THREADER Matrix: The top five matrices on the training data lie in the range of 0.7735 and 0.7909. This shows that the optimization procedure yields matrices which perform comparably to the best known matrices in our comparison.

Further, one usually wants to find a single matrix-parameter combination that works well. Given our setup, the choice would have to be made on the training data. Accordingly, in order to select only one matrix, we can choose the best matrix with respect to correlation on the training data. This yields a correlation of 0.7883 on the test data for the best GA matrix (the third best result of the GA matrices on the test data, which is still only outperformed by the SM_Threader matrix of all known matrices in the comparison).

For comparison, if we select the best known matrix on the training data by correlation (the Gonnet-Matrix with gap open costs of -3 and gap extend costs of -2.9), we achieve only a correlation of 0.7614 on the test data. Again, this shows that GA matrices perform comparably to known matrices, and as it is usually not known beforehand what the best matrix is (i.e. one has to choose a matrix on the training data), can even outperform those known matrices that perform best on the training data.

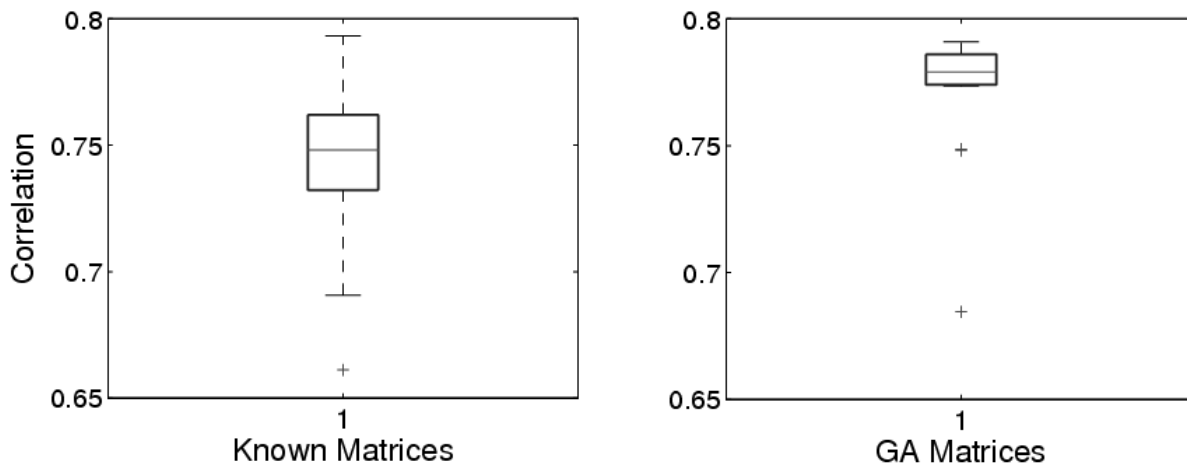


Figure 6.2: Comparison of GA matrices with well-known matrices. On the left side, a boxplot is shown for the well-known matrices in their different parameter configurations. On the right side, a corresponding boxplot is shown for the matrices generated by the genetic algorithm. For each column, the upper line gives the maximum value, the box is between 75% and 25% of the ordered results, and the lower line shows the minimum. The line in the middle of the box shows the median of the plotted data points. In order to show the main parts better, we cut off some outliers with lower correlations from the well-known matrices and set the range on the y-axis to [0.65, 0.8].

In addition, we included our GA matrices into our previous approach, the linear combination of matrices. Here, the least-squares combination of the GA matrices with all known matrices yields a correlation of 0.8299 on the test data, which is slightly better than the least-squares combination of known matrices only.

6.4 Optimized Profile-Profile Alignments

In the previous chapters, we have shown that in many cases the fold class of a target can be predicted from its sequence. In fact, for the AutoSCOP methods as described in chapter 5, the fold level is covered to more than 90% on the test data. However, when aligning a target with a template, usually standard parameters are used that work reasonably well for basically all possible fold classes, e.g. the Blosum62 matrix and the Kawabata matrix for log-average profile-profile alignment.

At this point, our hypothesis is that the knowledge of the correct fold can be used to improve the quality of the resulting alignments by training parameters specifically to fold environments. Therefore, in this section, we adapt the genetic algorithm described above to produce optimized matrices (sequence and secondary structure), gap costs and weighting factors for log-average profile-profile alignments for exemplary fold classes. The

aim is to make use of knowledge obtained from methods such as AutoSCOP, which predict the fold class with high specificity, to improve the alignment procedure and therefore to improve the expected quality of corresponding structure models.

A similar concept has been evaluated by [Vilim et al., 2004], who derived fold-specific substitution matrices for protein classification, also using a genetic algorithm. However, their so-called Class Attribute Substitution Matrices (CLASSUMs) approach, which was applied to the lambda and kappa subgroups of the immunoglobulin superfamily, is based on finding a distance measure between sequences in contrast to the usual task of recognizing similarity and includes a sequence-position-specific term which either includes or leaves out a position for the classification score. Second, their approach is aimed at classifying proteins for which an overall class (e.g. the superfamily) is already known into subgroups, whereas our approach described below has a different optimization goal: it is aimed at improving the alignment quality for alignments between members of known folds.

Another similar study has recently been published by [Sommer et al., 2006], who make use of non-optimal alignments to improve the structural quality of structure models. Here, the selection of models is done using a support vector machine based on the output of the VICTOR/FRST model quality assessment tool (i.e. its underlying potentials). The method proposed in the following is different from Sommer et al.'s approach as (1) it does not build structure models before selection, and (2) actually results in a fixed parameter setting for PPA. Once the genetic algorithm has finished, we propose to make use of the optimized parameters if there is a clear improvement on the test data for a particular fold class.

6.4.1 Training and Test Data

We randomly chose 8 exemplary fold classes, two for each major SCOP class, namely a.3, a.118, b.34, b.47, c.6, c.3, d.15, and d.19. For each of these fold classes, we extracted all domains from the ASTRAL 95, Version 1.65, and split them into two halves, one for training and one for testing. From the training sets, we then computed all-against pairs, and randomly selected 100 of them. For testing, we computed all possible alignments from the second half of the fold members. We have an average of 789 alignments per test set (maximum: 1770 alignments, minimum: 351 alignments).

6.4.2 Modified Genetic Algorithm

The modifications introduced to the genetic algorithm are simple:

- **Representation:** our chromosomes now consist of matrix entries for the sequence-based matrix, matrix entries for the secondary structure-based matrix, gap open and gap extension costs and weighting factors for the sequence-based score and the secondary structure-based score.
- **Diagonal entries:** matrix entries are initially drawn randomly from $[-1,1]$. In addition, in order to speed-up the process, we increased the initial weights of the diagonal

Fold	opt. PPA	PPA	Δ	#Sign. Changed	#Neg	#Pos
a.3	0.471	0.438	0.032	10.6%	0.7%	9.9%
a.118	0.288	0.294	-0.006	4.3%	2.8%	1.5%
b.34	0.353	0.347	0.006	13.2%	4.7%	8.5%
b.47	0.570	0.532	0.038	20.4%	2.2%	18.1%
c.26	0.336	0.319	0.016	9.1%	1.9%	7.1%
c.3	0.446	0.456	-0.010	10.2%	8.1%	2.1%
d.15	0.313	0.301	0.012	8.7%	2.8%	5.8%
d.19	0.558	0.505	0.053	31.0%	0.9%	30.0%
Avg	0.416	0.399	0.017	13.4%	3.0%	10.3%

Table 6.2: Average TM-Scores after optimizing PPA parameters using a modified genetic algorithm. Differences are shown in bold face whenever the optimization process leads to better results on the test data. In addition, the number of significantly changed TM-Scores (a difference of more than 0.1 TM-Score) as compared to the original PPA parameters on the test alignments is given together with the relative number of positive changes (where optimized parameters lead to better TM-Scores by more than 0.1) and the relative number of negative changes (where the original parameters performed better by more than 0.1).

entries of the matrices by drawing from $[0.8, 2]$. Before being used in the PPA procedure, all matrices are normalized such that the entries sum up to 1.0.

- **Other parameters:** All other parameters such as the weights for the sequence and the secondary structure-based parts are initially drawn from $[0, 20]$.

We used a population size of 50 and a mutation rate of 0.4. As each population member has to be evaluated by generating profile-profile alignments and then computing the TM-Score, the complete optimization procedure takes considerably longer than the genetic algorithm proposed in the previous section (up to several days on a single CPU). This is the main reason why we used only 8 exemplary fold classes in this evaluation instead of all available fold classes (about 800) of the ASTRAL distribution.

6.4.3 Results

For each fold class, we ran our genetic algorithm five times and then chose the parameters that performed best on the training data for the evaluation on the test set. The average changes of the mean-length normalized TM Score are given in Table 6.2. Further, we evaluated the number of alignments for which the TM-Score was in- or decreased by more than 0.1 (we refer to these alignments as *significantly changed alignments* in the following).

Within our 8 randomly chosen fold classes, we find that for three of them (a.3, b.47 and d.19) there is a clear improvement in average TM-Score of 0.032, 0.038 and 0.053, respectively, whereas we observe only slight increases for three classes and slight decreases for the remaining two classes (within a range of 0.02 difference in TM-Score between the

original and the newly generated parameters). With respect to individual alignments, we find that, on the test sets for each fold class, on average (over the fold classes, weighing each class equally) 13.4% of the alignments have been significantly changed, 10.3% being "better" and 3.0% being "worse". In other words, on average, nearly 4 of 5 significantly changed alignments have been improved. In the best case (d.19), 29 out of 30 significantly changed alignments have become better with respect to TM-Score.

Keeping in mind that the genetic algorithm starts more or less without any knowledge and that the PPA parameters have already been optimized for fold recognition purposes in general, this shows that it is possible to generate optimized parameters from random initial choices, at least for some of the available fold classes. By splitting the domains and the corresponding alignments into training and test data, it is further possible to select those fold classes where optimized parameters might be useful in future applications and where not, though this would require quite some CPU time when applied large-scale.

The GA parameters for this evaluation were chosen intuitively, and the training sets as well as the numbers of runs were small due to the runtime requirements. Therefore, in a larger experiment using larger training sets, larger populations and more runs, it may be possible to improve over the results shown in Table 6.2.

6.5 Discussion

In the first section of this chapter, we have described the QUASAR package, a simple alignment ranking software which allows for combination and, to some extent, optimization of alignment scores with respect to structure-based benchmark scores. The software is available at <http://www.bio.ifi.lmu.de/QUASAR> together with documentation, a tutorial and examples. It has been developed on a Linux/Unix system, but, being implemented in JAVA, can also be used on other platforms.

QUASAR may be of interest for users who work with alignments and want to either evaluate them with structure-based scores or rank them according to an optimized score combination that will hopefully rank "good" alignments (with respect to the structural quality of the resulting coordinate model) above "bad" alignments. Both situations occur in CASP-like environments: the former for training and evaluation of the servers and methods being developed, and the latter for finding those alignments and models that are finally used as predictions.

Subsequently, two similar optimization problems have been discussed: (1) alignment scoring and ranking for selection of potentially good structure models on the alignment level, and (2) the optimization of fold-class specific parameters for improving the quality of profile-profile alignment.

For the first problem, two different methods have been proposed, namely least-squares optimized linear combination of well-known scoring matrices and the genetic optimization of new scoring matrices. We find that the first approach works well in our comparison, as the scores obtained from linear combinations show an improved correlation with our benchmark score. Here, more elaborate combination techniques allowing more operators

than only addition and weighting may be an interesting starting point for future research. The second method, the genetic optimization of scoring matrices and their gap parameters produces competitive matrices to the well-known matrices in our comparison. Though no individual GA matrix performed better than the best known matrix, inclusion of the GA matrices into a combination of all evaluated matrices could further improve accuracy.

For the second problem, we modified the genetic algorithm such that it optimizes all necessary parameters for PPA. The results show that, for some cases, a clear improvement of alignment quality with respect to TM-Score can be reached. For this problem, the optimization procedure itself is time-consuming, but has to be done only once for each fold class in order to obtain the new parameters.

Overall, the evaluations in this chapter lead to two conclusions. First, given enough computing power, it is to some degree possible to optimize parameters for alignment ranking or alignment generation for specific environments (i.e. alignment procedures, template data etc.). Second, however, it is also obvious that the currently used matrices and their parameters already perform well and that one has to choose carefully when to use adapted parameters and when not, especially in the second case (alignment computation with optimized parameters).

Chapter 7

Additional Tools for Protein Domain Representation and Classification

In addition to the methods described in the previous chapters, in cooperation with colleagues, some additional tools have been developed that can contribute to research in protein structure prediction, three of which are the topics of this chapter. The first two will be described only briefly, namely the Vorolign structural alignment server [Birzele et al., 2007] and the ProML Schema for representation of proteins and protein sets. The third, the BioWeka library [Gewehr et al., 2007b] that extends the Weka data mining framework with bioinformatics data formats and methods, is described in more detail.

7.1 Vorolign: Structural Alignment and SCOP Classification Prediction

So far, we have described methods that predicted SCOP classifications based on a protein domain's amino acid sequence (and derived information) under the assumption that the target's structure is unknown. If we know the structure of the target, prediction accuracy can be improved by including this information into the prediction process. For the AutoSCOP method, this was shown by combining AutoSCOP with a structural alignment method which will be described in this section, namely the Vorolign method.

Vorolign is mainly the work of Fabian Birzele (in joint work with the author and Gergely Csaba), therefore we will keep this section short and present only an overview which, together with the previous chapters, completes our efforts in SCOP classification prediction based on different types of data (sequences, patterns and finally structures). More details about Vorolign can be found in the corresponding publication [Birzele et al., 2007]. In this work, Vorolign was used as an additional predictor for the AutoSCOP database, which provides predicted SCOP classifications using both AutoSCOP and Vorolign for new PDB entries.

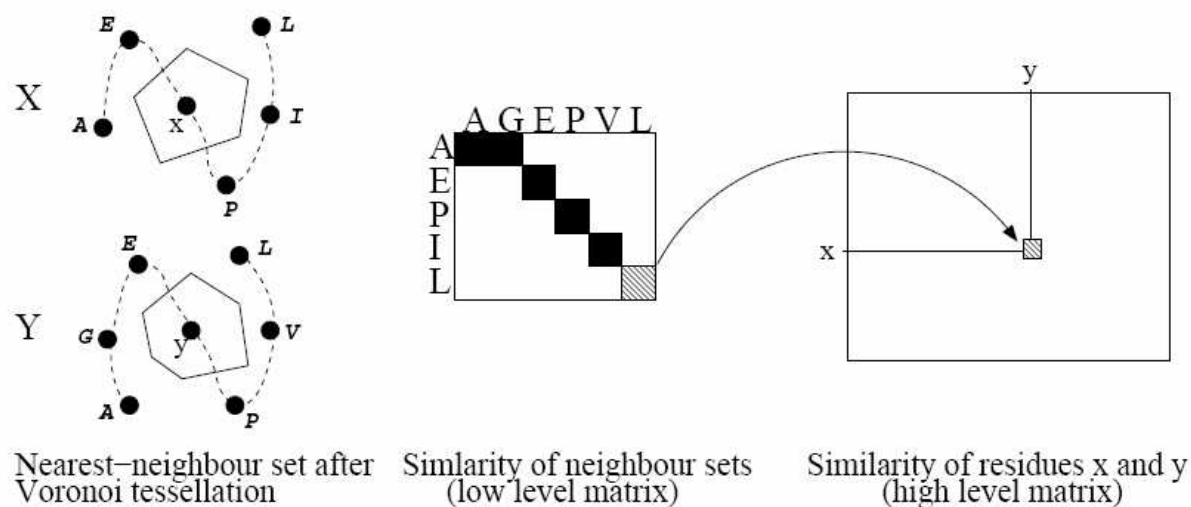


Figure 7.1: The similarity computation method of the Vorolign approach (taken from [Birzele et al., 2007]): Given their neighbors (left panel), Vorolign computes the similarity between two residues via dynamic programming (middle panel, low level matrix). These individual similarities are then used for the overall alignment as shown in the right panel (high level matrix).

The Vorolign Approach

The task of alignment and comparison of protein structures has been investigated for several years. When considering the proposed methods, it is usually possible to categorize them into one of the following two classes: (1) methods treating proteins as rigid (such as DALI [Holm and Sander, 1996] and CE [Shindyalov and Bourne, 1998]) and (2) methods allowing flexibility in proteins (such as FATCAT [Ye and Godzik, 2003]). Especially for the former class, the alignment criterion is often the root mean square deviation (RMSD) of the resulting superimposition, which may be regarded either locally (i.e. over certain, similar parts of the aligned structures) or globally (over the complete structures). Belonging to the latter class, the Vorolign method allows for flexible alignment of protein structures based on the following concept:

Vorolign assumes that two structurally similar residues are also similar with respect to their environment in the corresponding structures. Thereby, the environment of a residue is defined by its neighbors in the Voronoi tessellation of its structure. Given these neighbors, for the contained residues, similarity is computed using dynamic programming based on both amino acid as well as secondary structure exchange scores, and using these similarity scores between residues, an overall alignment is computed again using dynamic programming (see Figure 7.1). For details, we refer to [Birzele et al., 2007].

Summary of Results and Conclusion

Vorolign allows for two different applications: (1) Scans for family members, i.e. aligning a new structural domain against templates in order to predict its SCOP classification, and (2) flexible, pairwise as well as multiple alignment of protein structures. This first application includes a speed-up step very similar to the Preselection method described in chapter 3, as only the top 5% of templates with respect to SSEA score are considered by Vorolign.

For family recognition, the results show that Vorolign performs better than the structural alignment method CE and significantly better than sequence-based methods such as profile-profile alignment while reaching a good structural alignment quality (see Table 2 in [Birzele et al., 2007]). Multiple alignment examples show that indeed structurally similar regions can be mapped onto each other even between "open" and "closed" conformations of Calmodulin-like proteins, for instance.

In conclusion, Vorolign is a powerful structural alignment tool that extends our efforts in SCOP classification into the domain of new protein structures.

7.2 Representation of Protein Information in ProML

The large number of proteins and the corresponding data (millions of sequences and several thousands of structures, each annotated with further features and keywords) makes it necessary to find a way to represent instances such as proteins and protein sets for the purpose of finding similarities or differences between them as well as for browsing and searching the corresponding protein space. In this section, we briefly describe an XML application for this purpose, which is a rebuild of the Protein Markup Language (ProML, [Hanisch et al., 2002]) as an XML Schema. The schema in its version 1.0, as described here, is joint work with Martin Szugat. Alessandro Macri and Arthur Zimek participated in the development of earlier versions.

Why another XML Application?

The eXtensible Markup Language (XML¹) has become a standard tool for data description and communication in the areas of biology and chemistry. Accordingly, there exist several XML-based languages capable of representing proteins. Most of them are basically XML representations of the underlying databases (like PIR, InterPro, SwissProt and PDB), being capable of describing not only proteins but also database-specific annotations, thus emphasizing the aspect of complete data storage. This often leads to language elements unnecessary for the most common computational needs and largely increased document size. On the other hand, languages like BioML² are more general, aiming at e.g. hierarchical descriptions of complete organisms. This yields less preciseness than most protein

¹<http://www.w3.org/XML>

²<http://www.proteometrics.com/BIOML/>

related tasks require, lacking e.g. the possibility to communicate coordinates. Finally, it may be important for describing protein data to be able to include not only single proteins but also sets of proteins together with set-based properties like multiple alignments.

The ProML Schema, Version 1.0

To cope with these demands, we developed the *ProML* schema, which is both slim and modular, as it is based on a library of more common XML elements for computational biology called *BioSchemas*³ (which initially arose in the process of rebuilding ProML and is now maintained by Martin Szugat). Further, ProML can easily be extended by including further XML schemas from other namespaces.

The hierarchic syntax tree of ProML splits between single proteins and protein sets at the top node. While a protein is viewed pretty traditional as a structural unit, a protein set carries only references to the single proteins with the main part of the data describing the characteristics of this collection, e.g. a certain protein class resulting from a ProML query.

A protein document allows three different views, that is primary, secondary and tertiary structure. Each of these views holds subviews, consisting of the raw data which defines the view and constraints over this data. Some exemplary fields that are implemented in ProML 1.0 are the amino acid sequence in single letter code, secondary structure sequences by source (DSSP or PSIPRED, for instance), InterPro patterns with locations and further annotations, atomic coordinates, structural classifications, general residue contacts, and disulfide bonds.

Protein sets in ProML are usually generated by application of a given constraint, for example by filtering for the SCOP-tag to generate the set representing a certain SCOP family. ProML 1.0 supports mainly sequence features as protein set properties, such as multiple alignments of the contained proteins' sequences coming from different sources and InterPro pattern profiles.

Summary

The ProML schema allows users to describe proteins and protein sets including alignments, patterns, predicted classifications and more. As an XML application, ProML can be easily parsed and has been included into BioWeka, for instance, simply by providing a corresponding stylesheet for the XMLXSLLoader. ProML 1.0, stylesheets and additional information are available at <http://www.bio.ifi.lmu.de/2005/proml/>. ProML is now being maintained by Gergely Csaba who is working on an updated and extended Schema combination for use with web services.

³<http://www.bioschemas.org>

7.3 BioWeka: Extending the Weka Framework for Bioinformatics

The package described in this section is not an integral part of any of the previously described methods. Started as an evaluation and education project, the BioWeka library [Gewehr et al., 2007b] has nonetheless grown into a size and applicability for many basic tasks which makes it interesting for quick prototyping and data analysis before developing new, sophisticated applications. Further, the integration with Weka allows unexperienced users to perform basic bioinformatics tasks together with machine learning tasks on a single platform. In the following, we will describe the features of the underlying Weka machine learning framework [Witten and Frank, 2005] and the BioWeka library as well as two example applications.

Large parts of BioWeka have been implemented during the bachelor's thesis of Martin Szugat supervised by the author. Currently, the project is still maintained by both Martin Szugat and the author. The following content is partly based on a publication concerning BioWeka which appeared recently [Gewehr et al., 2007b].

7.3.1 Motivation

The tremendous amount of biological data which is nowadays available leads to the application of data mining methods for tasks like classification and clustering. The aim of these tools are to provide testable models, i.e. simplified abstractions, that allow for predictions of the behavior of the underlying systems. In a recent review under the title "Machine Learning in Bioinformatics" [Larranaga et al., 2006], Pedro Larranaga and coworkers present modeling methods as well as optimization methods together with applications in the fields of genomics, proteomics, systems biology, evolution and text mining.

In the case of *supervised classification*, given a set of instances divided into classes, classifiers are trained on the available training data (e.g. labeled examples) and are then used for predicting labels/classes of new instances. Larranaga et al. give examples for supervised classification problems in all fields listed above, including gene finding, secondary structure prediction, prediction of protein subcellular location, modeling of signal-response cascades, and protein/gene identification in text. This wide range of applications shows how established machine learning has become in bioinformatics. However, the available data is often not stored in the necessary feature-based representation for data mining applications. For instance, the well-known protein structure prediction server GenTHREADER [Jones, 1999a] computes alignment scores in a first step and then combines these with other features using a neural network. Other applications like ECLAT [Friedel et al., 2005] generate features from biological sequences by counting codons.

The popular data mining framework Weka offers a broad variety of useful tools for machine learning purposes. Our BioWeka project which is described in this section extends the Weka framework with additional bioinformatics functionalities to make applications and features as described above easily accessible from this standard machine learning tool.

Such extensions can be combined with the built-in functionalities of Weka (see Figure 7.2 for an overview of the interplay between Weka and BioWeka). This enables the user to employ all the useful facilities Weka has to offer together with well-known bioinformatics data formats and algorithms in a consistent way on a single platform.

The BioWeka website (<http://www.bioweka.org>) contains documentation, a tutorial and additional information on BioWeka. The distribution can be downloaded from SourceForge.net via a link given on the BioWeka website. It contains Weka, BioWeka and a couple of additional packages. As BioWeka is an open source project, users can easily integrate their own methods into the library.

7.3.2 The Weka Framework

Weka (the Waikato Environment for Knowledge Discovery) is a project pursued by the computer science department of the University of Waikato with the overall aim "to build a state-of-the-art facility for developing machine learning (ML) techniques and to apply them to real-world data mining problems,"⁴ which is well-known in the bioinformatics community [Frank et al., 2004]. The software is available from the Weka website, free for download and distributed under the GNU General Public License⁵. At the time of this thesis, the book version of the software is 3.4 (which is used in the latest release of the Weka book [Witten and Frank, 2005]), and the developers version is 3.5. For our purpose, we make use of the book version, i.e. Weka 3.4.

The User Interfaces

Having downloaded, installed and started Weka, a user can choose between three different interfaces, each of which has a special focus:

1. **The Explorer** is a basic interface which consists of panels for data handling, classification, rule extraction, clustering and analysis of the results.
2. **The Experimenter** is a more elaborate interface. Here it is possible to define setups that contain multiple datasets as well as classifiers, for instance, and thus to make large comparisons and experiments with Weka.
3. **The Knowledge Flow** interface provides the user with a graph-based experiment design facility. Components can be drawn onto a canvas, they can be configured, and connections between the components define the way the data flows through these components.

In addition, all components of Weka can be used from the command-line as individual JAVA classes. A workflow can be set up by simply using the output of one component as input for the next by building a pipeline (which is possible because of the universal ARFF format within Weka).

⁴<http://www.cs.waikato.ac.nz/ml/index.html>

⁵<http://www.gnu.org/copyleft/gpl.html>

Data Handling in Weka

Many data mining tools (as well as most of the databases used nowadays) represent their data as relations. Here, WEKA is no exception: the so-called *Attribute-Relation File Format* (ARFF) is used by its components to interchange data. All data that comes in and out of WEKA has to be transformed into or out of the ARFF format; however, once the data is correctly formatted, it can be used by more or less all of Weka's components (at least with respect to the syntax).

Another way to input and output data is to make use of relational databases, which are supported by Weka. Data can be read from databases, new databases can be set up and also all results can be stored again in a database. However, as Weka does not work with multiple relations at once, it is important to join all necessary data in only one table (or one ARFF file, respectively) to be able to use it in the Weka environment.

Weka's way of manipulating data is to apply so-called *filters* to ARFF-formatted relations. Basically, a filter is a component that takes ARFF-formatted data as input, manipulates the data and outputs it again in the ARFF format. A variety of filters are available which perform tasks such as the conversion of attribute data types or the deletion of attributes.

Available Data Mining Algorithms

Once the data has been prepared, it can be used as input for one of the algorithms of the large collection of data mining methods available in Weka. The following list gives an impression of this collection without listing all available components. Among the available classifiers, we can find many standard methods such as regression functions, neural network variants, decision trees, rule learners, association rule mining, and support vector machines. Further, classifiers can be stacked and combined with each other by so-called meta classifiers. For clustering, some well-known methods for hierarchical and conceptual clustering are included.

Two facts that should be noted about Weka is that some of the contained implementations have considerable memory requirements and are not necessarily the fastest implementations available. Nonetheless, for exploration and prototyping with small evaluation sets, Weka is very well suited.

Experiment Design, Evaluation and Analysis Facilities

As mentioned in 7.3.2, e.g. in the Experimenter GUI it is possible to design larger data mining experiments instead of testing each dataset and classifier by hand in the Explorer GUI. However, already in the Explorer users can choose between different validation methods such as cross validation or percentage splits into training and test data. Several analysis mechanisms are provided which include significance tests, the output of confusion matrices and the graphical analysis of classification results, for instance.

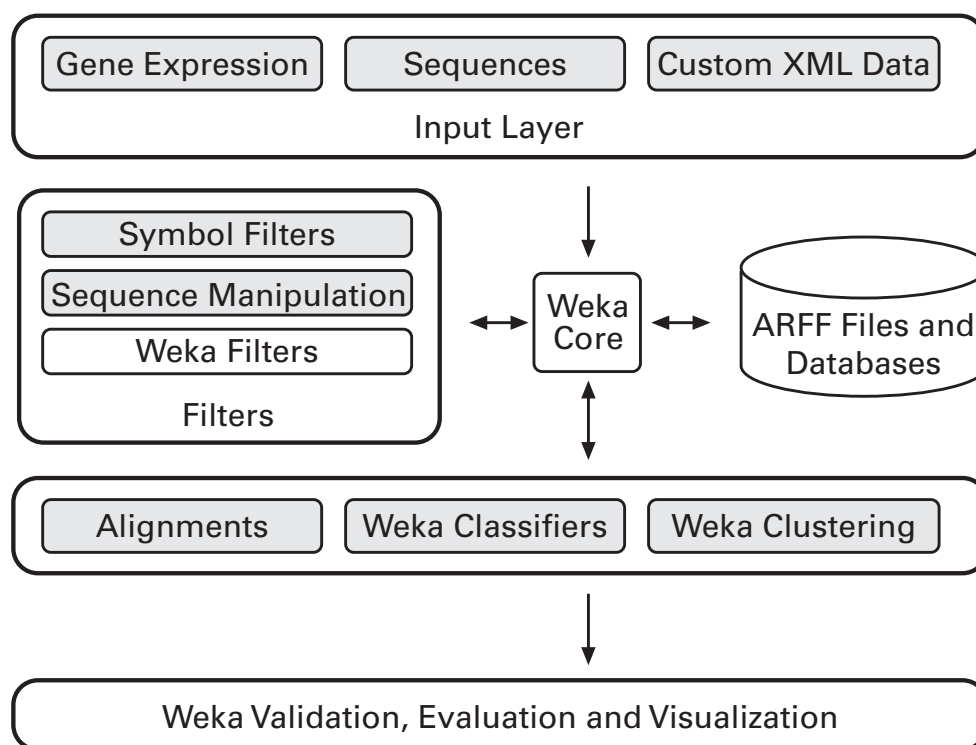


Figure 7.2: BioWeka Overview: BioWeka offers input filters for many well-known bioinformatics file formats as well as the possibility to import custom XML formats. Further, BioWeka adds a number of symbol-based filters and sequence transformations as well as the possibility to align sequences in Weka and generate classifications from the resulting scores. The BioWeka extensions are shown in light gray.

7.3.3 The BioWeka Library

Weka is a useful collection of tools for data mining tasks, but is mainly based on the ARFF format and does not provide many other input formats. In order to combine Weka with bioinformatics, in our BioWeka project, we have two main goals, (1) providing loaders for typical bioinformatics data formats, and (2) adding mechanisms to handle such bioinformatics data (such as alignments for sequence data).

Input Formats

BioWeka contains an input layer for converting well-known formats into ARFF (and vice versa for some formats). So far, the following data formats are supported:

- **MAGE-ML** [Spellman et al., 2002] and CSV compatible formats for gene expression data,

- **FASTA** [Pearson and Lipman, 1988], **EMBL** [Kulikova et al., 2004], **Swiss-Prot** [Bairoch and Boeckmann, 1991], and **GenBank** [Benson et al., 1993] for the storage of biological sequences in ASCII files.
- **InterProScan** [Zdobnov and Apweiler, 2001] for the annotation of sequence patterns.
- **ProML**, an XML Schema which allows the description of proteins and protein sets, as described briefly in the previous subsection.

In addition to these formats already provided by BioWeka, users can easily extend BioWeka by adding their own converters. Custom XML formats can be incorporated into BioWeka using XSL stylesheets. Another possibility is to store the data directly in ARFF format, of course.

Bioinformatics Extensions

Besides its loaders, BioWeka contains new filters for

- annotating symbol properties (e.g. hydrophobicity of amino acids),
- annotating symbol counts (e.g. codon frequencies),
- translating sequences from one alphabet into another (e.g. DNA to RNA),
- manipulating sequences (e.g. cutting sequences after their first stop codon),
- generating different sequence frames (e.g. the open reading frames of a DNA sequence).

For the annotation of symbol properties BioWeka supports the Amino Acid Index database [Kawashima et al., 1999]. Symbol counting also considers ambiguous symbols (e.g. X), overlapping symbol groups and pseudo counts.

Another large part of BioWeka enables users to align amino acid sequences or secondary structure sequences with each other using different alignment methods, including BLAST, PSI-BLAST and JAligner [Moustafa, 2006]. For alignment-based classification, a couple of different evaluation mechanisms are provided (e.g. by selecting the class with the highest average alignment score or the class with the highest single alignment score). Further, custom alignment score evaluation schemes can be plugged in.

Using BioWeka

One has to download both the Weka and the BioWeka distribution and include the Weka JAR in the CLASSPATH variable for BioWeka. The BioWeka startup script provides access to Weka as well as BioWeka. For the BLAST and PSI-BLAST classifiers, one also needs a BLAST installation. In the Explorer GUI, for instance, one can import the new data formats listed above using BioWeka's converters and apply BioWeka's filters and classifiers.

Contributing to BioWeka

BioWeka is an open and ongoing project. It is licensed under the GNU General Public License to ensure that any contributions made to the BioWeka project are free to anyone. We encourage bioinformatics developers and users of Weka to participate in the further development of BioWeka by contributing code, exemplary data sets, or practical knowledge.

Extending the BioWeka framework with custom components is as easy as writing a single Java class. By inheriting from one of the many already existing BioWeka classes the coding effort is minimal. These implementations can then be combined with the existing BioWeka classes.

To improve the collaboration between the BioWeka users the BioWeka web site provides an open Wiki (based on the popular MediaWiki software of the Wikipedia Foundation) that can be edited by any registered user. The Wiki contains an end-user documentation of the BioWeka components, tutorials and a knowledge base. In addition, there are several mailing lists and user forums.

7.3.4 Example Applications

Example 1: Machine Learning on SSEA Scores

As an example how to benefit from the BioWeka library, we describe a small analysis we performed before finalizing the preselection approach proposed in chapter 3. For this analysis, we make use of the ASTRAL 25 dataset as described in section 3.2.1. Previous studies have shown that secondary structure element alignment (SSEA) is a useful tool for finding topologically similar templates (see chapter 3), and for our approach we align a target against all available templates with SSEA in order to find potential fold classes. The aim of this small test is to find out how to make use of these scores in a suitable way for our prediction task.

We can load the dataset into Weka using BioWeka's FASTA loader and then apply secondary structure element alignment (SSEA) in BioWeka's AlignmentScorer filter to obtain the SSEA scores for each sequence against all other sequences in the set. Using five-fold cross-validation, with BioWeka's AlignmentScoreClassifier we can now easily evaluate different evaluation options. With the MaxScoreEvaluator, i.e. using the top-scoring template for a prediction, we achieve 53% correctly classified instances on fold level (in a leave-one-out test, this approach leads to 54%). The AverageScoreEvaluator (which uses the class with the highest average score as prediction) reaches only 22% which can be explained by the fact that many of the larger fold classes contain domains that differ significantly and that therefore reduce the average scores for such classes. As a nice side-effect we can use Weka's analysis facilities, which allow for inspection of the correlation between the scores obtained by aligning against particular instances of the set, which in turn allows for visually measuring the similarity between such instances.

As we have the scores from an all-vs.-all alignment, a further question is whether it is useful to simply use the alignment scores for each instance as vectors and then apply a learner on these vectors, in order to make use of the context provided by the vector

contents instead of using only the maximum score for classification. For this purpose, we can drop all String attributes from our set and end up with the sequence lengths, the alignment scores and the annotated, nominal class values. On this reduced set, we can evaluate basic machine learning classifiers in order to get a feeling for their applicability as well as of the performance of the MaxScoreEvaluator in comparison. The ZeroR classifier (which simply predicts the largest class in the set in all cases) as the most basic test leads to only 4% correctly classified instances. Using the OneR learner, a simple rule learner, we reach 11% accuracy. Also a more sophisticated method, the support vector machine (SVM) implementation provided by the LibSVM [Fan et al., 2005] package (which is included in the BioWeka distribution) yields only 13% accuracy on this data when used in its default configuration with radial basis functions, and Weka's J48 classifier, which builds a decision tree, yields 20%. The JRIP rule learner achieves 24% accuracy, and the default LibSVM using a linear kernel yields 45% accuracy.

Although we have not varied the parameters for these machine learning algorithms, we can already see that subsequent application of machine learning algorithms will have at least difficulties to reach the performance of the original alignment method with the MaxScoreEvaluator. Further, subsequent methods require additional effort in comparison to simply using the maximum score, which in turn could reduce the desired speed-up. From this evaluation, we decided that using SSEA scores directly and ordering template classes by the maximum score per class is a reasonable and efficient choice for the ASTRAL 25 dataset, and we applied it accordingly in our methods as presented in chapter 3, for instance. Though some of the learners took hours for finishing their runs because of the relatively large dataset, for the user everything could be done with only a few clicks in the Weka GUI, as SSEA and the score evaluators are part of the BioWeka library.

Example 2: Reimplementation of ECLAT

In Martin Szugat's bachelor thesis, the applicability of BioWeka was confirmed by an exemplary reimplementation of the ECLAT method [Friedel et al., 2005]. ECLAT is an approach to classify DNA with respect to its origin, in order to be able to discriminate between plant DNA and pathogen DNA, for instance. It applies support vector machines to perform classification based on codon usage differences, which involves steps such as the calculation and normalization of codon frequencies and the generation of open reading frames.

The reimplementation of the whole method in BioWeka took about 650 lines of code as compared to the 1260 lines of code for the original implementation which was kindly provided by Caroline Friedel, and it provides a nice graphical user interface based on the Weka software. For a complete description of this reimplementation, please see the bachelor's thesis and its addendum, both of which are available from <http://www.bioweka.org>.

7.3.5 Discussion

In bioinformatics research, often (newly developed) classifiers have to be compared to other, well-known classifiers. In order to use many methods, it is often necessary to deal with many different input and output formats and even to implement a customized evaluation framework around the corresponding programs.

Weka is a well-known framework that offers many standard machine learning methods. BioWeka makes it easy to use a number of data formats relevant for bioinformatics with Weka. Everything from classification to validation can be done with such data without further overhead using the standard workflow in Weka. To handle such data properly, some bioinformatics-specific methods have been integrated into Weka via BioWeka. In addition, the multifactor dimensionality reduction of the Weka-CG project [Moore et al., 2005] and the Weka LibSVM project [EL-Manzalawy and Honavar, 2005] come with the distribution. Tutorials on how to use BioWeka with sequences as well as how to import gene expression data formats are available online at <http://www.bioweka.org>.

To conclude, the integration of bioinformatics methods and other useful tools into Weka allows users to perform many bioinformatics standard tasks without the overhead of parsing data formats or writing code that combines different software packages. Developers can make use of BioWeka's abstract classes and interfaces in order to prototype and test new algorithms. Again, this reduces the overhead of writing converter as well as evaluation classes and allows to concentrate directly on the methods. Comparison with many other methods can be done directly in BioWeka. Finally, BioWeka is highly configurable and available free of charge.

Chapter 8

Concluding Remarks

Protein structure prediction and related aspects such as structural genomics will remain an important part of bioinformatics and computational biology, as the ultimate solution to predicting a structure from the sequence alone seems nowhere in sight. However, gradual improvements in methods as well as increasingly fast algorithms can provide better and more models and may thus continually extend the possibilities for drug development and other research purposes. Improving intermediate steps of the protein structure prediction process is therefore a step towards the overall goal of finding a structure for every known protein and all possible benefits which would result from such a situation.

Homology-based protein structure prediction can in many cases be broken down into a number of subtasks, which include the search for potential domains on a target sequence, the search for good templates, the computation of alignments to these templates and the generation and refinement of corresponding structure models based on these alignments. In this work, we concentrated on the recognition of protein domains on a target and the prediction of corresponding structural classifications, which are important steps for template searches and other applications including target selection for structural genomics. Further, we developed new methods for optimized alignment ranking and computation with respect to structural quality.

Summary

The first new approach described in this thesis can speed-up alignment-based fold recognition (Preselection, chapter 3, [Gewehr et al., 2004]). Our results show that, when used in combination with log average profile-profile alignment (PPA), which was shown to be very accurate for this purpose, we can be faster than PPA alone by about one order of magnitude while keeping a comparable accuracy.

Our second method, AutoSCOP, works either as an independent predictor or as an additional filter not only to predict SCOP families, superfamilies and folds but also to avoid or detect errors (chapter 4, [Gewehr et al., 2007a]). AutoSCOP is fast and reliable, as our results could confirm: When used as standalone predictor, AutoSCOP already achieves both high sensitivity and high specificity on the difference set between two ASTRAL ver-

sions. When used as a filter in combination with well-known methods, we could clearly improve accuracy over the individual methods.

One important property of the AutoSCOP approach is that its input data is available in a precomputed form for many of the currently available protein sequences, and thus for most sequences predictions can be done via a simple database lookup. This enables us to provide predictions for all these sequences in a quick scan, which makes AutoSCOP a useful tool for purposes such as structural genomics. In particular, in a joint project with Fabian Birzele, given precomputed data from the InterPro database as well as our own data on PDB sequences, we provide the AutoPSI database [Birzele et al., 2008] of SCOP predictions based on both AutoSCOP and Vorolign (see below) for thousands of new, unclassified PDB entries as well as two million UniProt/TrEMBL sequences (chapter 4). Further, given the locations of the annotated regions, we can use our AutoSCOP hits as initial guesses for domain locations on these sequences; therefore, AutoSCOP is a quick way to find potential locations of SCOP domains on target sequences, which can easily be applied in a large scale, especially if precomputed pattern data is available.

Preselection and AutoSCOP work on protein domains, and in general the recognition of domains on a new protein sequence is a common step to find good starting regions for protein crystallization as well as for protein structure prediction. Although AutoSCOP gives us the possibility to assign regions with potential structural classifications, the corresponding boundaries are often not accurate, and some domains may be missed. We thus proposed a new algorithm for template-based protein domain recognition from a protein sequence (SSEP-Domain, chapter 5, [Gewehr and Zimmer, 2006]) which was ranked among the top domain prediction servers in the community-wide CAFASP 4 experiment. Our results as well as independent evaluations confirmed that SSEP-Domain works well in both predicting the number of domains on a protein chain and correct placement of the domain boundaries.

In chapter 5, we further discussed the influence of the underlying template databases on the accuracy of the predictions. By including alternative domain definitions, we are able to provide predictions based on different sources for template domain assignments, depending on the intended purpose. As SSEP-Domain is still quite fast (less than ten minutes per predicted sequence on average), it can be used to refine predictions where no AutoSCOP hits have been found or the boundaries should be refined, for instance. When used together, AutoSCOP and SSEP-Domain can help both researchers that are interested in particular targets as well as researchers that are more interested in larger scale evaluations of protein domain predictions and the locations of structural classifications.

For a different part of the protein structure prediction pipeline, as the quality of the alignments is still one of the most important factors influencing the final quality of a predicted structure, we developed the QUASAR software (chapter 6, [Birzele et al., 2005]), which facilitates the ranking of sequence-structure alignments on the basis of combinations of well-known scoring matrices. In addition, based on QUASAR, we implemented and evaluated optimization methods that allow for improved correlations between alignment scores and structure-based benchmark scores. Using a genetic algorithm, we were further able to show that we can improve the structural quality of our alignments (i.e. the quality

of the straight-forwardly generated model structures resulting from them) by tuning parameters and scoring matrices to specific fold classes. The approaches are general enough to be applied also for other fitness functions than the correlation with a benchmark score, which was chosen in this work.

The last chapter, chapter 7, contains additional tools for protein structure prediction. We briefly introduced a structural alignment method using Voronoi cell-based contact definitions (Vorolign, [Birzele et al., 2007]). Due to its underlying algorithm, Vorolign can capture flexibilities in protein structures better than any rigid-body superimposition method. Further, being both fast and accurate, it provides a structure-based means for SCOP classification prediction that complements our sequence-based approaches (Preselection and AutoSCOP).

We described an XML schema for the storage and handling of protein data (ProML). Finally, we introduced BioWeka [Gewehr et al., 2007b], an extension of the well-known machine learning framework Weka which includes bioinformatics functionalities and data formats (including ProML) and allows for the application of standard machine learning and data mining procedures to bioinformatics data. Based on two examples, we could show how both small evaluations as well the development of larger prediction algorithms can be done easily with BioWeka.

The presented methods and tools can contribute to the structure prediction process in various ways (and they do, as we have shown in the evaluation sections of the different chapters): Given a new target in a CASP-like situation (i.e. only the amino acid sequence), it is possible to find potential domains using SSEP-Domain and then predict SCOP classifications using Preselection or AutoSCOP. Given a predicted SCOP classification (and thus a number of associated templates), alignments can be generated and ranked with QUASAR and its extensions. The additional tools can help to handle the data or deduce additional features of a target.

Further, the AutoPSI database already contains millions of predictions for many proteins in many genomes, and the AutoSCOP method itself works in minutes, when patterns have to be searched, and seconds, when precomputed data is available. As such data is often available, AutoSCOP can easily be applied in a large scale. Besides AutoSCOP, also SSEP-Domain, which provides predictions in a matter of minutes, is fast enough to be applied in a larger scale, and it can be used directly or to refine or correct initial hints on a domain structure of a target obtained from AutoSCOP or the AutoPSI database. For instance, the SSEP-Domain server has recently been used by the "parasitic nematode genomics" group of Makedonka Mitreva from the St. Louis Genome Sequencing Center for several thousand predictions. In this project, the aim is to identify highly conserved sequences across all Nematode species, so-called Nematode-specific Multi-species Conserved Sequences (NMCS). SSEP-Domain was used to predict the number and locations of potential domains on ESTs based on homology to the SCOP templates.

Outlook

While the methods presented here are already useful in their current states, they can be extended towards many directions. Interesting points for future research include the disagreement of domain definitions coming from different sources and the recognition of discontinuous domains on the amino acid sequence, as discussed in chapter 5.

Regarding the first problem, the differing points of view when using the term "protein domain" make it difficult to simply provide a consensus, in our opinion, as most definitions have a justification, and a consensus will result in a blurred view somewhere in between the individual views. Therefore, the most useful way of using such definition may be to be clear about the purpose of an experiment and choose the definition that matches this purpose best.

For discontinuous domains, which often result from a pure structure-oriented viewpoint, the SSEP-Domain algorithm as well as many other domain predictors could be improved by finding efficient means of recognizing parts of discontinuous domains and how they belong together already on a sequence. So far, it seems that the most accurate algorithms that can predict such domains obtain predictions by generating model structures for the target and then assigning domains using structure-oriented algorithms. With respect to speeding-up domain predictions for purposes such as fast, genome-wide screening, however, it might be desirable to avoid the model building step.

As another possible extension, the combination of the template-based method SSEP-Domain with an *ab initio* approach, which could cover also new fold domains, would be interesting. Such a hybrid approach could perhaps improve the accuracy of the predictions in cases where no (remotely) similar templates for the contained domains are available.

The AutoSCOP method will benefit from the integration of additional data sources, as we could demonstrate exemplarily by including ASTRAL's family HMMs in chapter 4. The framework is simple enough to include all kinds of predicted sequence regions, and corresponding databases and prediction algorithms are still being developed towards new aims by many research groups. Including this accumulated knowledge may further increase both accuracy and coverage of the AutoSCOP approach as well as the corresponding AutoPSI database.

Further, tools such as BioWeka need the acceptance and feedback of the research community, and therefore we chose to make BioWeka available as an open source project and thus open to contributions from its users. The more different methods and datasets are made available in such a project by independent researchers, the better it will be possible for new users to apply and extend it for their own tasks.

In conclusion, SSEP-Domain, AutoSCOP and Vorolign are available as web servers, the AutoPSI database is publicly available both via web interface and as flat files, and QUASAR and BioWeka are available as software packages. We therefore provide both useful solutions to known problems as well as a good basis for future research in the area of protein structure prediction and structural genomics, which will continue to be challenging and necessary tasks.

Bibliography

- [Albrecht et al., 2003] Albrecht, M., Tosatto, S. C. E., Lengauer, T., and Valle, G. (2003). Simple consensus procedures are effective and sufficient in secondary structure prediction. *Protein Engineering*, 16(7):459–462.
- [Alexandrov and Shindyalov, 2003] Alexandrov, N. and Shindyalov, I. (2003). PDP: protein domain parser. *Bioinformatics*, 19(3):429–430.
- [Alexandrov, 1996] Alexandrov, N. N. (1996). Sarfing the PDB. *Protein Eng*, 9(9):727–732.
- [Altschul et al., 1990] Altschul, S., Gish, W., Miller, W., Myers, E. W., and Lipman, D. J. (1990). Basic local alignment search tool. *J Mol Biol*, 215(3):403–410.
- [Altschul et al., 1997] Altschul, S. F., Madden, T. L., Schäffer, A. A., Zhang, J., Zhang, Z., Miller, W., and Lipman, D. J. (1997). Gapped BLAST and PSI-BLAST: a new generation of protein database search programs. *Nucleic Acids Research*, 25(19):3389–3402.
- [Artamonova et al., 2005] Artamonova, I. I., Frishman, G., Gelfand, M. S., and Frishman, D. (2005). Mining sequence annotation databanks for association patterns. *Bioinformatics*, 21(Suppl.3):iii49–iii57.
- [Attwood, 2002] Attwood, T. K. (2002). The PRINTS database: a resource for identification of protein families. *Brief Bioinform*, 3(3):252–263.
- [Bairoch et al., 2005] Bairoch, A., Apweiler, R., Wu, C. H., Barker, W. C., Boeckmann, B., Ferro, S., Gasteiger, E., Huang, H., Lopez, R., Magrane, M., Martin, M. J., Natale, D. A., O'Donovan, C., Redaschi, N., and Yeh, L.-S. L. (2005). The universal protein resource (UniProt). *Nucleic Acids Res*, 33(Database issue):D154–D159.
- [Bairoch and Boeckmann, 1991] Bairoch, A. and Boeckmann, B. (1991). The SWISS-PROT protein sequence data bank. *Nucleic Acids Res*, 19 Suppl:2247–2249.
- [Baker and Sali, 2001] Baker, D. and Sali, A. (2001). Protein structure prediction and structural genomics. *Science*, 294(5540):93–96.

- [Bateman et al., 2004] Bateman, A., Coin, L., Durbin, R., Finn, R. D., Hollich, V., Griffiths-Jones, S., Khanna, A., Marshall, M., Moxon, S., Sonnhammer, E. L. L., Studholme, D. J., Yeats, C., and Eddy, S. R. (2004). The Pfam protein families database. *Nucleic Acids Res*, 32(Database issue):D138–D141.
- [Benson et al., 1993] Benson, D., Lipman, D. J., and Ostell, J. (1993). GenBank. *Nucleic Acids Res*, 21(13):2963–2965.
- [Berman et al., 2000] Berman, H., Westbrook, J., Feng, Z., Gilliland, G., Bhat, T., Weissig, H., Shindyalov, I., and Bourne, P. (2000). The protein data bank. *Nucleic Acids Research*, 28:235–242.
- [Berrera et al., 2003] Berrera, M., Molinari, H., and Fogolari, F. (2003). Amino acid empirical contact energy definitions for fold recognition in the space of contact maps. *BMC Bioinformatics*, 4(1):8.
- [Bindewald et al., 2003] Bindewald, E., Cestaro, A., Hesser, J., Heiler, M., and Tosatto, S. C. E. (2003). MANIFOLD: Protein fold recognition based on secondary structure, sequence similarity and enzyme classification. *Protein Engineering*, 16(11):785–789.
- [Birzele et al., 2007] Birzele, F., Gewehr, J. E., Csaba, G., and Zimmer, R. (2007). Vorolign–fast structural alignment using Voronoi contacts. *Bioinformatics*, 23(2):e205–e211.
- [Birzele et al., 2005] Birzele, F., Gewehr, J. E., and Zimmer, R. (2005). QUASAR–scoring and ranking of sequence–structure alignments. *Bioinformatics*, 21(24):4425–4426.
- [Birzele et al., 2008] Birzele, F., Gewehr, J. E., and Zimmer, R. (2008). AutoPSI: A database for automatic structural classification of protein sequences and structures. *Nucleic Acids Research*, Accepted.
- [Blake and Cohen, 2001] Blake, J. D. and Cohen, F. E. (2001). Pairwise sequence alignment below the twilight zone. *J Mol Biol*, 307(2):721–735.
- [Bourne and Weissig, 2003] Bourne, P. E. and Weissig, H. (2003). *Structural Bioinformatics*. Wiley & Sons.
- [Bradley et al., 2005] Bradley, P., Malmström, L., Qian, B., Schonbrun, J., Chivian, D., Kim, D. E., Meiler, J., Misura, K. M. S., and Baker, D. (2005). Free modeling with Rosetta in CASP6. *Proteins*, 61 Suppl 7:128–134.
- [Branden and Tooze, 1999] Branden, C. and Tooze, J. (1999). *Introduction to Protein Structure*. Taylor & Francis.
- [Brezellec et al., 2006] Brezellec, P., Hoebeke, M., Hiet, M.-S., Pasek, S., and Ferat, J.-L. (2006). DomainSieve: a protein domain-based screen that led to the identification of dam-associated genes with potential link to DNA maintenance. *Bioinformatics*, 22(16):1935–1941.

- [Bru et al., 2005] Bru, C., Courcelle, E., Carrère, S., Beausse, Y., Dalmar, S., and Kahn, D. (2005). The ProDom database of protein domain families: more emphasis on 3D. *Nucleic Acids Res*, 33(Database issue):D212–D215.
- [Camon et al., 2003] Camon, E., Magrane, M., Barrell, D., Binns, D., Fleischmann, W., Kersey, P., Mulder, N., Oinn, T., Maslen, J., Cox, A., and Apweiler, R. (2003). The gene ontology annotation (GOA) project: implementation of GO in SWISS-PROT, TrEMBL, and InterPro. *Genome Res*, 13(4):662–672.
- [Chandonia et al., 2004] Chandonia, J.-M., Hon, G., Walker, N. S., Conte, L. L., Koehl, P., Levitt, M., and Brenner, S. E. (2004). The ASTRAL compendium in 2004. *Nucleic Acids Res*, 32(Database issue):D189–D192.
- [Cheek et al., 2004] Cheek, S., Qi, Y., Krishna, S. S., Kinch, L. N., and Grishin, N. V. (2004). SCOPmap: automated assignment of protein structures to evolutionary super-families. *BMC Bioinformatics*, 5:197.
- [Cheng et al., 2005] Cheng, J., Sweredoski, M., and Baldi, P. (2005). DOMpro: Protein domain prediction using profiles, secondary structure, relative solvent accessibility, and recursive neural networks. *Data Mining and Knowledge Discovery*, page to appear.
- [Chinnasamy et al., 2004] Chinnasamy, A., Sung, W. K., and Mittal, A. (2004). Protein structure and fold prediction using tree-augmented naïve Bayesian classifiers. In Altman, R., Keith, A., Hunter, L., Jung, T., and Klein, T., editors, *Pacific Symposium on Biocomputing 2003*, volume 9, pages 387–398.
- [Chiu et al., 2006] Chiu, S.-H., Chen, C.-C., Yuan, G.-F., and Lin, T.-H. (2006). Association algorithm to mine the rules that govern enzyme definition and to classify protein sequences. *BMC Bioinformatics*, 7:304.
- [Chivian et al., 2003] Chivian, D., Kim, D. E., Malmstrom, L., Bradley, P., Robertson, T., Murphy, P., Strauss, C. E., Bonneau, R., Rohl, C. A., and Baker, D. (2003). Automated prediction of CASP-5 structures using the Robetta server. *Proteins*, 53:524–533.
- [Coin et al., 2004] Coin, L., Bateman, A., and Durbin, R. (2004). Enhanced protein domain discovery using taxonomy. *BMC Bioinformatics*, 5:56.
- [Dayhoff et al., 1978] Dayhoff, M. O., Schwartz, R., and Orcutt, B. C. (1978). A model of evolutionary change in proteins. *Atlas of Protein Sequence and Structure*, 5:345–352.
- [Ding and Dubchak, 2001] Ding, C. H. Q. and Dubchak, I. (2001). Multi-class protein fold recognition using support vector machines and neural networks. *Bioinformatics*, 17(4):349–358.
- [Djurisic et al., 1997] Djurisic, A. B., Elazar, J. M., and Rakic, A. D. (1997). Genetic algorithms for continuous optimization problems - a concept of parameter-space size adjustment. *J. Phys. A: math. Gen.*, 30:7849–7861.

- [Dosztanyi and Torda, 2001] Dosztanyi, Z. and Torda, A. E. (2001). Amino acid similarity matrices based on force fields. *Bioinformatics*, 17(8):686–699.
- [Dumontier et al., 2005] Dumontier, M., Yao, R., Feldman, H. J., and Hogue, C. W. (2005). Armadillo: domain boundary prediction by amino acid composition. *J Mol Biol*, 350(5):1061–73.
- [Eddy, 1998] Eddy, S. R. (1998). Profile hidden Markov models. *Bioinformatics*, 14(9):755–763.
- [EL-Manzalawy and Honavar, 2005] EL-Manzalawy, Y. and Honavar, V. (2005). *WLSVM: Integrating LibSVM into Weka Environment*. <http://www.cs.iastate.edu/~yasser/wlsvm>.
- [Fan et al., 2005] Fan, R. E., Chen, P. H., and Lin, C. J. (2005). Working set selection using the second order information for training SVM. *Journal of Machine Learning Research*, 6:1889–1918.
- [Fischer et al., 2003] Fischer, D., Rychlewski, L., Dunbrack, R. L., Ortiz, A. R., and Elofsson, A. (2003). CAFASP3: the third critical assessment of fully automated structure prediction methods. *Proteins*, 53 Suppl 6:503–516.
- [Frank et al., 2004] Frank, E., Hall, M., Trigg, L., Holmes, G., and Witten, I. H. (2004). Data mining in bioinformatics using Weka. *Bioinformatics*, 20(15):2479–2481.
- [Friedel et al., 2005] Friedel, C. C., Jahn, K. H. V., Sommer, S., Rudd, S., Mewes, H. W., and Tetko, I. V. (2005). Support vector machines for separation of mixed plant-pathogen EST collections based on codon usage. *Bioinformatics*, 21(8):1383–1388. Evaluation Studies.
- [Galzitskaya and Melnik, 2003] Galzitskaya, O. V. and Melnik, B. S. (2003). Prediction of protein domain boundaries from sequence alone. *Protein Sci*, 12(4):696–701.
- [George and Heringa, 2002a] George, R. A. and Heringa, J. (2002a). Protein domain identification and improved sequence similarity searching using PSI-BLAST. *Proteins*, 48(4):672–681.
- [George and Heringa, 2002b] George, R. A. and Heringa, J. (2002b). SnapDRAGON: a method to delineate protein structural domains from sequence data. *J Mol Biol*, 316(3):839–851.
- [Gewehr et al., 2007a] Gewehr, J. E., Hintermair, V., and Zimmer, R. (2007a). Auto-SCOP: Automated prediction of scop classifications using unique pattern-class mappings. *Bioinformatics*, 23(10):1203–1210.
- [Gewehr et al., 2007b] Gewehr, J. E., Szugat, M., and Zimmer, R. (2007b). BioWeka—extending the weka framework for bioinformatics. *Bioinformatics*, 23(5):651–653.

- [Gewehr et al., 2004] Gewehr, J. E., von Oehsen, N., and Zimmer, R. (2004). Combining secondary structure element alignment and profile-profile alignment for fold recognition. In Giegerich, R. and Stoye, J., editors, *German Conference on Bioinformatics 2004*, pages 141–149. Gesellschaft für Informatik.
- [Gewehr and Zimmer, 2006] Gewehr, J. E. and Zimmer, R. (2006). SSEP-Domain: protein domain prediction by alignment of secondary structure elements and profiles. *Bioinformatics*, 22(2):181–187.
- [Ginalski, 2006] Ginalski, K. (2006). Comparative modeling for protein structure prediction. *Curr Opin Struct Biol*, 16(2):172–177.
- [Ginalski et al., 2003] Ginalski, K., Elofsson, A., Fischer, D., and Rychlewski, L. (2003). 3D-Jury: a simple approach to improve protein structure predictions. *Bioinformatics*, 19(8):1015–1018.
- [Gonnet et al., 1992] Gonnet, G. H., Cohen, M. A., and Benner, S. A. (1992). Exhaustive matching of the entire protein sequence database. *Science*, 256(5062):1443–1445.
- [Gough and Chothia, 2002] Gough, J. and Chothia, C. (2002). SUPERFAMILY: HMMs representing all proteins of known structure. SCOP sequence searches, alignments and genome assignments. *Nucleic Acids Res*, 30(1):268–272.
- [Gouzy et al., 1999] Gouzy, J., Corpet, F., and Kahn, D. (1999). Whole genome protein domain analysis using a new method for domain clustering. *Computers and Chemistry*, 23:333–340.
- [Grabowski et al., 2007] Grabowski, M., Joachimiak, A., Otwinowski, Z., and Minor, W. (2007). Structural genomics: keeping up with expanding knowledge of the protein universe. *Curr Opin Struct Biol*, 17:347–353.
- [Grimmond et al., 2003] Grimmond, S. M., Miranda, K. C., Yuan, Z., Davis, M. J., Hume, D. A., Yagi, K., Tominaga, N., Bono, H., Hayashizaki, Y., Okazaki, Y., Teasdale, R. D., R. I. K. E. N. GER Group, and G. S. L. Members (2003). The mouse secretome: functional classification of the proteins secreted into the extracellular environment. *Genome Res*, 13(6B):1350–1359.
- [Gusfield, 1997] Gusfield, D. (1997). *Algorithms on Strings, Trees, and Sequences*. Cambridge University Press.
- [Haft et al., 2003] Haft, D. H., Selengut, J. D., and White, O. (2003). The TIGRFAMs database of protein families. *Nucleic Acids Res*, 31(1):371–373.
- [Hanisch et al., 2002] Hanisch, D., Zimmer, R., and Lengauer, T. (2002). ProML—the protein markup language for specification of protein sequences, structures and families. *In Silico Biol*, 2(3):313–324.

- [Heger and Holm, 2003] Heger, S. and Holm, L. (2003). Exhaustive enumeration of protein domain families. *J Mol Biol*, 328(3):749–67.
- [Henikoff and Henikoff, 1992] Henikoff, S. and Henikoff, J. G. (1992). Amino acid substitution matrices from protein blocks. *Proc Natl Acad Sci U S A*, 89(22):10915–10919.
- [Holland et al., 2006] Holland, T. A., Veretnik, S., Shindyalov, I. N., and Bourne, P. E. (2006). Partitioning protein structures into domains: why is it so difficult? *J Mol Biol*, 361(3):562–590.
- [Holm and Sander, 1991] Holm, L. and Sander, C. (1991). Database algorithm for generating protein backbone and side-chain co-ordinates from a c alpha trace application to model building and detection of co-ordinate errors. *J Mol Biol*, 218(1):183–194.
- [Holm and Sander, 1996] Holm, L. and Sander, C. (1996). Mapping the protein universe. *Science*, 273(5275):595–603.
- [Holm and Sander, 1997] Holm, L. and Sander, C. (1997). Dali/FSSP classification of three-dimensional protein folds. *Nucleic Acids Res*, 25(1):231–234.
- [Hulo et al., 2004] Hulo, N., Sigrist, C. J. A., Saux, V. L., Langendijk-Genevaux, P. S., Bordoli, L., Gattiker, A., Castro, E. D., Bucher, P., and Bairoch, A. (2004). Recent improvements to the PROSITE database. *Nucleic Acids Res*, 32(Database issue):D134–D137.
- [Islam et al., 1995] Islam, S. A., Luo, J., and Sternberg, M. J. (1995). Identification and analysis of domains in proteins. *Protein Eng*, 8(6):513–525.
- [Jaroszewski et al., 2005] Jaroszewski, L., Rychlewski, L., Li, Z., Li, W., and Godzik, A. (2005). FFAS03: a server for profile–profile sequence alignments. *Nucleic Acids Res*, 33(Web Server issue):W284–W288.
- [Johnson and Overington, 1993] Johnson, M. S. and Overington, J. P. (1993). A structural basis for sequence comparisons. an evaluation of scoring methodologies. *J Mol Biol*, 233(4):716–738.
- [Jones, 1999a] Jones, D. T. (1999a). GenTHREADER: An efficient and reliable protein fold recognition method for genomic sequences. *J Mol Biol*, 287(4):797–815.
- [Jones, 1999b] Jones, D. T. (1999b). Protein secondary structure prediction based on position-specific scoring matrices. *J Mol Biol*, 292(2):195–202.
- [Jones et al., 1998] Jones, S., Stewart, M., Michie, A., Swindells, M. B., Orengo, C., and Thornton, J. (1998). Domain assignment for protein structures using a consensus approach: Characterisation and analysis. *Protein Science*, 7(2):233–242.

- [Kabsch and Sander, 1983] Kabsch, W. and Sander, C. (1983). Dictionary of protein secondary structure: pattern recognition of hydrogen-bonded and geometrical features. *Biopolymers*, 22(12):2577–637.
- [Kaplan et al., 2003] Kaplan, N., Vaaknin, A., and Linial, M. (2003). PANDORA: keyword-based analysis of protein sets by integration of annotation sources. *Nucleic Acids Res*, 31(19):5617–5626.
- [Kawashima et al., 1999] Kawashima, S., Ogata, H., and Kanehisa, M. (1999). AAindex: amino acid index database. *Nucleic Acids Res.*, 27:368–369.
- [Kim et al., 2005] Kim, D. E., Chivian, D., Malmstrom, L., and Baker, D. (2005). Automated prediction of domain boundaries in CASP6 targets using Ginzu and RosettaDOM. *Proteins*, page to appear.
- [Kim and Patel, 2006] Kim, Y. J. and Patel, J. M. (2006). A framework for protein structure classification and identification of novel protein structures. *BMC Bioinformatics*, 7:456.
- [Kopp and Schwede, 2006] Kopp, J. and Schwede, T. (2006). The SWISS-MODEL repository: new features and functionalities. *Nucleic Acids Res*, 34(Database issue):D315–D318.
- [Kulikova et al., 2004] Kulikova, T., Aldebert, P., Althorpe, N., Baker, W., Bates, K., Browne, P., van den Broek, A., Cochrane, G., Duggan, K., Eberhardt, R., Faruque, N., Garcia-Pastor, M., Harte, N., Kanz, C., Leinonen, R., Lin, Q., Lombard, V., Lopez, R., Mancuso, R., McHale, M., Nardone, F., Silventoinen, V., Stoehr, P., Stoesser, G., Tuli, M. A., Tzouvara, K., Vaughan, R., Wu, D., Zhu, W., and Apweiler, R. (2004). The EMBL Nucleotide Sequence Database. *Nucleic Acids Res*, 32(Database issue):27–30.
- [Larranaga et al., 2006] Larranaga, P., Calvo, B., Santana, R., Bielza, C., Galdiano, J., Inza, I., Lozano, J. A., Armañanzas, R., Santafé, G., Pérez, A., and Robles, V. (2006). Machine learning in bioinformatics. *Brief Bioinform*, 7(1):86–112.
- [Letunic et al., 2004] Letunic, I., Copley, R. R., Schmidt, S., Ciccarelli, F. D., Doerks, T., Schultz, J., Ponting, C. P., and Bork, P. (2004). SMART 4.0: towards genomic data integration. *Nucleic Acids Res*, 32(Database issue):D142–D144.
- [Linding et al., 2003] Linding, R., Russell, R. B., Neduva, V., and Gibson, T. J. (2003). GlobPlot: Exploring protein sequences for globularity and disorder. *Nucleic Acids Research*, 13:3701–3708.
- [Liu and Rost, 2003] Liu, J. and Rost, B. (2003). Domains, motifs and clusters in the protein universe. *Curr Opin Chem Biol*, 7:5–11.
- [Liu and Rost, 2004] Liu, J. and Rost, B. (2004). Sequence-based prediction of protein domains. *Nucleic Acids Res*, 32(12):3522–3530.

- [Lo Conte et al., 2002] Lo Conte, L., Brenner, S. E., Hubbard, T. J. P., Chothia, C., and Murzin, A. G. (2002). SCOP database in 2002: refinements accommodate structural genomics. *Nucleic Acids Res*, 30(1):264–267.
- [Luthy et al., 1991] Luthy, R., McLachlan, A. D., and Eisenberg, D. (1991). Secondary structure-based profiles: use of structure-conserving scoring tables in searching protein sequence databases for structural similarities. *Proteins*, 10(3):229–239.
- [Madej et al., 1995] Madej, T., Gibrat, J. F., and Bryant, S. H. (1995). Threading a database of protein cores. *Proteins*, 23(3):356–369.
- [Marsden et al., 2002] Marsden, R. L., McGuffin, L. J., and Jones, D. T. (2002). Rapid protein domain assignment from amino acid sequence using predicted secondary structure. *Protein Science*, 11(12):2814–2824.
- [McGuffin et al., 2001] McGuffin, L. J., Bryson, K., and Jones, D. T. (2001). What are the baselines for protein fold recognition? *Bioinformatics*, 17(1):63–72.
- [McGuffin and Jones, 2002] McGuffin, L. J. and Jones, D. T. (2002). Targeting novel folds for structural genomics. *PROTEINS: Structure, Function, Genetics*, 48(1):44–52.
- [Mitchell, 1998] Mitchell, M. (1998). *An Introduction to Genetic Algorithms*. MIT Press.
- [Miyazawa and Jernigan, 1993] Miyazawa, S. and Jernigan, R. L. (1993). A new substitution matrix for protein sequence searches based on contact frequencies in protein structures. *Protein Eng*, 6(3):267–278.
- [Moore et al., 2005] Moore, J. H., Barney, N., and Holden, T. (2005). *The Weka-CG Project*. <http://www.epistasis.org/weka-cg-project.html>.
- [Moult, 2005] Moult, J. (2005). A decade of CASP: progress, bottlenecks and prognosis in protein structure prediction. *Curr Opin Struct Biol*, 15(3):285–289.
- [Mount, 2001] Mount, D. W. (2001). *Bioinformatics: Sequence and Genome Analysis*. Cold Spring Harbor Laboratory Press, U.S.
- [Moustafa, 2006] Moustafa, A. (2006). *JAligner: Open source Java implementation of Smith-Waterman*. <http://jaligner.sourceforge.net/>.
- [Mulder et al., 2003] Mulder, N. J., Attwood, T. K., Bairoch, A., Barrell, D., Bateman, A., Binns, D., Biswas, M., Bradley, P., Bork, P., Bucher, P., Copley, R. R., Courcelle, E., Das, U., Durbin, R., Falquet, L., Fleischmann, W., Griffiths-Jones, S., Haft, D., Harte, N., Hulo, N., Kahn, D., Kanapin, A., Krestyaninova, M., Lopez, R., Letunic, I., Lonsdale, D., Silventoinen, V., Orchard, S. E., Pagni, M., Peyruc, D., Ponting, C. P., Selengut, J. D., Servant, F., Sigrist, C. J. A., Vaughan, R., and Zdobnov, E. M. (2003). The InterPro database, 2003 brings increased coverage and new features. *Nucleic Acids Research*, 31(1):315–318.

- [Murzin et al., 1995] Murzin, A. G., Brenner, S. E., Hubbard, T., and Chothia, C. (1995). SCOP: A structural classification of proteins database for the investigation of sequences and structures. *J Mol Biol*, 247(4):536–540.
- [Nagarajan and Yona, 2004] Nagarajan, N. and Yona, G. (2004). Automatic prediction of protein domains from sequence information using a hybrid learning system. *Bioinformatics*, 20(8):1335–1360.
- [Needleman and Wunsch, 1970] Needleman, S. B. and Wunsch, C. D. (1970). A general method applicable to the search for similarities in the amino acid sequence of two proteins. *J Mol Biol*, 48(3):443–453.
- [Notredame et al., 2000] Notredame, C., Higgins, D. G., and Heringa, J. (2000). T-Coffee: A novel method for fast and accurate multiple sequence alignment. *J Mol Biol*, 302(1):205–217.
- [Orengo et al., 1997] Orengo, C. A., Michie, A. D., Jones, S., Jones, D. T., Swindells, M. B., and Thornton, J. M. (1997). CATH - a hierarchical classification of protein domain structures. *Structure*, 5:1093–1108.
- [O’Sullivan et al., 2003] O’Sullivan, O., Zehnder, M., Higgins, D., Bucher, P., Grosdidier, A., and Notredame, C. (2003). APDB: a novel measure for benchmarking sequence alignment methods without reference alignments. *Bioinformatics*, 19:215i–221i.
- [Overington et al., 1992] Overington, J., Donnelly, D., Johnson, M. S., Sali, A., and Blundell, T. L. (1992). Environment-specific amino acid substitution tables: tertiary templates and prediction of protein folds. *Protein Sci*, 1(2):216–226.
- [Panchenko, 2003] Panchenko, A. R. (2003). Finding weak similarities between proteins by sequence profile comparison. *Nucleic Acids Res*, 31(2):683–689.
- [Park and Teichmann, 1998] Park, J. and Teichmann, S. A. (1998). DIVCLUS: an automatic method in the GEANFAMMER package that finds homologous domains in single- and multi-domain proteins. *Bioinformatics*, 14(2):144–150.
- [Pearson and Lipman, 1988] Pearson, W. R. and Lipman, D. J. (1988). Improved tools for biological sequence comparison. *Proc Natl Acad Sci U S A*, 85(8):2444–2448.
- [Pieper et al., 2006] Pieper, U., Eswar, N., Davis, F. P., Braberg, H., Madhusudhan, M. S., Rossi, A., Marti-Renom, M., Karchin, R., Webb, B. M., Eramian, D., Shen, M.-Y., Kelly, L., Melo, F., and Sali, A. (2006). MODBASE: a database of annotated comparative protein structure models and associated resources. *Nucleic Acids Res*, 34(Database issue):D291–D295.
- [Prlic et al., 2000] Prlic, A., Domingues, F. S., and Sippl, M. J. (2000). Structure-derived substitution matrices for alignment of distantly related sequences. *Protein Eng*, 13(8):545–550.

- [Przytycka et al., 1999] Przytycka, T., Aurora, R., and Rose, G. D. (1999). A protein taxonomy based on secondary structure. *Nat Struct Biol*, 6(7):672–682.
- [Qian and Goldstein, 2002] Qian, B. and Goldstein, R. A. (2002). Optimization of a new score function for the generation of accurate alignments. *Proteins*, 48(4):605–610.
- [Quevillon et al., 2005] Quevillon, E., Silventoinen, V., Pillai, S., Harte, N., Mulder, N., Apweiler, R., and Lopez, R. (2005). InterProScan: protein domains identifier. *Nucleic Acids Res*, 33(Web Server issue):W116–W120.
- [Risler et al., 1988] Risler, J. L., Delorme, M. O., Delacroix, H., and Henaut, A. (1988). Amino acid substitutions in structurally related proteins. A pattern recognition approach. determination of a new and efficient scoring matrix. *J Mol Biol*, 204(4):1019–1029.
- [Rychlewski et al., 2000] Rychlewski, L., Jaroszewski, L., Li, W., and Godzik, A. (2000). Comparison of sequence profiles. Strategies for structural predictions using sequence information. *Protein Science*, 9(2):232–241.
- [Sadreyev and Grishin, 2003] Sadreyev, R. and Grishin, N. (2003). Compass: a tool for comparison of multiple protein alignments with assessment of statistical significance. *J Mol Biol*, 326(1):317–336.
- [Saini and Fischer, 2005] Saini, H. K. and Fischer, D. (2005). Meta-DP: domain prediction meta-server. *Bioinformatics*, 21(12):2917–2920.
- [Schäffer et al., 2001] Schäffer, A. A., Aravind, L., Madden, T. L., Shavirin, S., Spouge, J. L., Wolf, Y. I., Koonin, E. V., and Altschul, S. F. (2001). Improving the accuracy of PSI-BLAST protein database searches with composition-based statistics and other refinements. *Nucleic Acids Res*, 29(14):2994–3005.
- [Scordis et al., 1999] Scordis, P., Flower, D. R., and Attwood, T. K. (1999). Finger-PRINTScan: intelligent searching of the PRINTS motif database. *Bioinformatics*, 15(10):799–806.
- [Shindyalov and Bourne, 1998] Shindyalov, I. N. and Bourne, P. E. (1998). Protein structure alignment by incremental combinatorial extension (CE) of the optimal path. *Protein Engineering*, 11:739–747.
- [Siew et al., 2000] Siew, N., Elofsson, A., Rychlewski, L., and Fischer, D. (2000). Max-Sub: an automated measure for the assessment of protein structure prediction quality. *Bioinformatics*, 16(9):776–785.
- [Sikder and Zomaya, 2006] Sikder, A. R. and Zomaya, A. Y. (2006). Improving the performance of DomainDiscovery of protein domain boundary assignment using inter-domain linker index. *BMC Bioinformatics*, 7 Suppl 5:S6.

- [Sim et al., 2005] Sim, J., Kim, S. Y., and Lee, J. (2005). PPRODO: Prediction of protein domain boundaries using neural networks. *Proteins*, 59:627–632.
- [Singer et al., 2002] Singer, M. S., Vriend, G., and Bywater, R. P. (2002). Prediction of protein residue contacts with a PDB-derived likelihood matrix. *Protein Eng*, 15(9):721–725.
- [Smith and Waterman, 1981] Smith, T. F. and Waterman, M. S. (1981). Identification of common molecular subsequences. *J Mol Biol*, 147(1):195–197.
- [Söding, 2005] Söding, J. (2005). Protein homology detection by HMM-HMM comparison. *Bioinformatics*, 21(7):951–960.
- [Sommer et al., 2006] Sommer, I., Toppo, S., Sander, O., Lengauer, T., and Tosatto, S. C. E. (2006). Improving the quality of protein structure models by selecting from alignment alternatives. *BMC Bioinformatics*, 7:364.
- [Sommer et al., 2002] Sommer, I., Zien, A., von Ohsen, N., Zimmer, R., and Lengauer, T. (2002). Confidence measures for protein fold recognition. *Bioinformatics*, 18(6):802–812.
- [Spellman et al., 2002] Spellman, P. T., Miller, M., Stewart, J., Troup, C., Sarkans, U., Chervitz, S., Bernhart, D., Sherlock, G., Ball, C., Lepage, M., Swiatek, M., Marks, W. L., Goncalves, J., Markel, S., Iordan, D., Shojatalab, M., Pizarro, A., White, J., Hubley, R., Deutsch, E., Senger, M., Aronow, B. J., Robinson, A., Bassett, D., Stoeckert, C. J. J., and Brazma, A. (2002). Design and implementation of microarray gene expression markup language (MAGE-ML). *Genome Biol*, 3(9).
- [Tai et al., 2005] Tai, C.-H., Lee, W.-J., Vincent, J. J., and Lee, B. (2005). Evaluation of domain prediction in CASP6. *Proteins*, 61 Suppl 7:183–192.
- [Taylor and Aszodi, 2004] Taylor, W. R. and Aszodi, A. (2004). *Protein Geometry, Classification, Topology and Symmetry*. Institute of Physics Publishing.
- [Thompson et al., 1994] Thompson, J. D., Higgins, D. G., and Gibson, T. J. (1994). CLUSTAL W: improving the sensitivity of progressive multiple sequence alignment through sequence weighting, position-specific gap penalties and weight matrix choice. *Nucleic Acids Res*, 22(22):4673–4680.
- [Torda et al., 2004] Torda, A. E., Procter, J. B., and Huber, T. (2004). Wurst: a protein threading server with a structural scoring function, sequence profiles and optimized substitution matrices. *Nucleic Acids Res*, 32(Web Server issue):W532–W535.
- [Veretnik et al., 2004] Veretnik, S., Bourne, P. E., Alexandrov, N. N., and Shindyalov, I. N. (2004). Toward consistent assignment of structural domains in proteins. *J Mol Biol*, 339(3):647–678.

- [Vilim et al., 2004] Vilim, R. B., Cunningham, R. M., Lu, B., Kheradpour, P., and Stevens, F. J. (2004). Fold-specific substitution matrices for protein classification. *Bioinformatics*, 20(6):847–853.
- [Vogel et al., 2005] Vogel, C., Teichmann, S., and Pereira-Leal, J. (2005). The relationship between domain duplication and recombination. *J Mol Biol*, 346:355–365.
- [Vogt et al., 1995] Vogt, G., Etzold, T., and Argos, P. (1995). An assessment of amino acid exchange matrices in aligning protein sequences: the twilight zone revisited. *J Mol Biol*, 249(4):816–831.
- [von Öhsen, 2005] von Öhsen, N. (2005). *A Novel Profile-Profile Alignment Method and Its Application in Fully Automated Protein Structure Prediction*. Shaker Verlag.
- [von Öhsen et al., 2003] von Öhsen, N., Sommer, I., and Zimmer, R. (2003). Profile-profile alignment: A powerful tool for protein structure prediction. In Altman, R. B., Dunker, A. K., Hunter, L., Jung, T. A., and Klein, T. E., editors, *Pacific Symposium on Bio-computing 2003*, pages 252–263. World Scientific Publishing Co. Pte. Ltd., Singapore.
- [von Öhsen et al., 2004] von Öhsen, N., Sommer, I., Zimmer, R., and Lengauer, T. (2004). Arby: automatic protein structure prediction using profile-profile alignment and confidence measures. *Bioinformatics*, 20(14):2228–2235.
- [von Öhsen and Zimmer, 2001] von Öhsen, N. and Zimmer, R. (2001). Improving profile-profile alignment via log average scoring. In Gascuel, O. and Moret, B. M. E., editors, *Algorithms in Bioinformatics, First International Workshop, WABI 2001, Aarhus, Denmark, August 2001, Proceedings*, volume 2149 of *Lecture Notes in Computer Science*, pages 11–26. Springer-Verlag Berlin Heidelberg New York.
- [Wang and Jiang, 1994] Wang, L. and Jiang, T. (1994). On the complexity of multiple sequence alignment. *J Comp Biol*, (1):337–348.
- [Wheelan et al., 2000] Wheelan, S. J., Marchler-Bauer, A., and Bryant, S. H. (2000). Domain size distributions can predict domain boundaries. *Bioinformatics*, 16(7):613–618.
- [Wheeler et al., 2000] Wheeler, D. L., Chappay, C., Lash, A. E., Leipe, D. D., Madden, T. L., Schuler, G. D., Tatusova, T. A., and Rapp, B. A. (2000). Database resources of the national center for biotechnology information. *Nucleic Acids Res*, 28(1):10–14.
- [Witten and Frank, 2005] Witten, I. H. and Frank, E. (2005). *Data Mining: Practical machine learning tools and techniques. 2nd Edition*. Morgan Kaufmann, San Francisco.
- [Wu et al., 2004] Wu, C. H., Nikolskaya, A., Huang, H., Yeh, L.-S. L., Natale, D. A., Vinayaka, C. R., Hu, Z.-Z., Mazumder, R., Kumar, S., Kourtesis, P., Ledley, R. S., Suzek, B. E., Arminski, L., Chen, Y., Zhang, J., Cardenas, J. L., Chung, S., Castro-Alvear, J., Dinkov, G., and Barker, W. C. (2004). PIRSF: family classification system at the protein information resource. *Nucleic Acids Res*, 32(Database issue):D112–D114.

- [Xu et al., 2000] Xu, Y., Xu, D., Gabow, H. N., and Gabow, H. (2000). Protein domain decomposition using a graph-theoretic approach. *Bioinformatics*, 16(12):1091–1104.
- [Yan and Moulton, 2005] Yan, Y. and Moulton, J. (2005). Protein family clustering for structural genomics. *J Mol Biol*, 353(3):744–759.
- [Ye and Godzik, 2003] Ye, Y. and Godzik, A. (2003). Flexible structure alignment by chaining aligned fragment pairs allowing twists. *Bioinformatics*, 19 Suppl 2:II246–II255.
- [Yona and Levitt, 2002] Yona, G. and Levitt, M. (2002). Within the twilight zone: A sensitive profile-profile comparison tool based on information theory. *J Mol Biol*, 315(5):1257–1275.
- [Zdobnov and Apweiler, 2001] Zdobnov, E. M. and Apweiler, R. (2001). InterProScan—an integration platform for the signature-recognition methods in InterPro. *Bioinformatics*, 17(9):847–848.
- [Zemla et al., 1999] Zemla, A., Venclovas, C., Fidelis, K., and Rost, B. (1999). A modified definition of Sov, a segment-based measure for protein secondary structure prediction assessment. *Proteins*, 34(2):220–223.
- [Zhang et al., 2005] Zhang, Y., Chandonia, J.-M., Ding, C., and Holbrook, S. R. (2005). Comparative mapping of sequence-based and structure-based protein domains. *BMC Bioinformatics*, 6:77.
- [Zhang and Skolnick, 2004] Zhang, Y. and Skolnick, J. (2004). Scoring function for automated assessment of protein structure template quality. *Proteins*, 57(4):702–710.
- [Zhou and Skolnick, 2007] Zhou, H. and Skolnick, J. (2007). Ab initio protein structure prediction using chunk-TASSER. *Biophys J*, to appear.

Acknowledgements

This thesis has been a very large part of my life during the last years, and I want to express my gratitude to the many people who made it possible, even if I cannot mention everyone here.

First of all, I want to extend my warmest thanks to my supervisor Prof. Dr. Ralf Zimmer for giving me the opportunity to write my thesis in the field of structural bioinformatics. He always encouraged and supported me, and without his experience and inspiring ideas this work would not have been possible.

I am very grateful to Fabian Birzele and Martin Szugat for our joint projects, as well as to Joannis Apostolakis, Gergely Csaba, Volker Hintermair, Alessandro Macri, Niklas von Öhsen and Florian Sohler for the productive cooperations. It was a great experience for me to work with them, and I thank all of them for their enthusiasm, their support and the fun we had.

I want to thank the Deutsche Forschungsgemeinschaft (DFG) for funding parts of this thesis under grant PROSEQO II (Zi 616/2).

Further, I am grateful to Prof. Dr. Dmitrij Frishman for reviewing this thesis, and Prof. Dr. Hans-Peter Kriegel and Prof. Dr. Volker Heun for agreeing to participate in my dissertation committee.

Finally, I am indebted to my parents, my friends, and, in particular, Tanja Lederer for being patient with me and giving me time and support in all stages of this exciting project.

Curriculum Vitae

Jan Erik Gewehr was born on March 8th, 1977 in Lübeck, Germany. He attended primary school in Sereetz and high school in Bad Schwartau, receiving his high school degree in 1996. From 1996 to 1997, he worked as a male nurse for disabled persons at the Marli Werkstätten in Lübeck. In October 1997, he entered the University of Lübeck, studying Computer Science with a minor in Bioinformatics and Biomathematics. In February 2003, he received his diploma degree. He entered the Ludwig-Maximilians University (LMU) in March 2003 as a research assistant in the Teaching and Research Unit for Practical Informatics and Bioinformatics, headed by Prof. Dr. Ralf Zimmer.

Publications:

1. Fabian Birzele*, Jan E. Gewehr* and Ralf Zimmer. AutoPSI: A Database for Automatic Structural Classification of Protein Sequences and Structures. *Nucleic Acids Research*, accepted.
* Authors contributed equally
2. Jan E. Gewehr, Volker Hintermair and Ralf Zimmer. AutoSCOP: Automated Prediction of SCOP Classifications using Unique Pattern-Class Mappings. *Bioinformatics*, 23, 1203-1210, 2007.
3. Jan E. Gewehr*, Martin Szugat* and Ralf Zimmer. BioWeka - Extending the Weka Framework for Bioinformatics. *Bioinformatics*, 23, 651-653, 2007.
* Authors contributed equally
4. Fabian Birzele, Jan E. Gewehr, Gergely Csaba and Ralf Zimmer. Vorolign - Fast Structural Alignment using Voronoi Contacts. *Bioinformatics*, 23, e205-e211, 2007.
5. Martin Szugat, Jan E. Gewehr and Cordula Lochmann. *Social Software - Blogs, Wikis & Co.* Entwickler.Press, 2006, ISBN 3-939084-09-3.
6. Jan E. Gewehr and Ralf Zimmer. SSEP-Domain: Protein Domain Prediction by Alignment of Secondary Structure Elements and Profiles. *Bioinformatics*, 22, 181-187, 2006.
7. Fabian Birzele*, Jan E. Gewehr* and Ralf Zimmer. QUASAR - Scoring and Ranking of Sequence-Structure Alignments. *Bioinformatics*, 21, 4425-4426, 2005.
* Authors contributed equally

8. Florian Sohler and Jan E. Gewehr. Inference of Developmental Transcription Factor Activities in *Drosophila Melanogaster*. Proceedings of the Moscow Conference on Computational Molecular Biology (MCCMB) 2005.
9. Jan E. Gewehr, Niklas von Öhsen and Ralf Zimmer. Combining Secondary Structure Element Alignment and Profile-Profile Alignment for Fold Recognition. R. Giegerich and Jens Stoye (eds.): German Conference on Bioinformatics (GCB) 2004, GI Lecture Notes in Informatics P-53, 141-148, 2004.
10. Jan T. Kim, Jan E. Gewehr and Thomas Martinetz. Binding Matrix: A Novel Approach for Binding Site Recognition. *Journal of Bioinformatics and Computational Biology*, 2, 289-307, 2004.
11. Thomas Martinetz, Jan E. Gewehr and Jan T. Kim. Statistical Learning for Detecting Protein-DNA Binding Sites. D. C. Wunsch II, M. Hasselmo, and K. Venayagamoorthy (eds.): Proc. of the IEEE International Joint Conference on Neural Networks (IJCNN) 2003, IEEE Press, 2940-2945, 2003.