

Development of Homology Modeling Techniques

Dissertation

der Fakultät für
Informations- und Kognitionswissenschaften
(Wilhelm-Schickard Institut für Informatik)
zur Erlangung des Grades eines
Doktors der Naturwissenschaften (Dr. rer. nat.)

vorgelegt von
Dipl.-Ing. Alexander Vasil Diemand
aus Zürich

**Tübingen
2006**

Tag der mündlichen Qualifikation: 21. Dezember 2006
Dekan: Prof. Dr. Michael Diehl
1. Berichterstatter: Prof. Dr. Andrei Lupas
2. Berichterstatter: Prof. Dr. Oliver Kohlbacher

Acknowledgments

Many people have influenced this work in various ways. I want to thank those unnamed ones that have provided feedback over the Internet and their feature requests have especially shaped the functionality of the iMolTalk server.

I thank Prof. Gerald Stranzinger, ETH Zürich, for having supported my first steps into bioinformatics. Dr. Nicolas Guex has introduced me to homology modeling and his computer program SwissPDBViewer, for which I am still very grateful. Big thanks go to Dr. Holger Scheib for continued support and for sharing his experience. I want to thank my co-supervisor, Prof. Oliver Kohlbacher, University of Tübingen, for his time and advice. Finally, I want to deeply thank my supervisor, Prof. Andrei Lupas, Max-Planck-Institute for Developmental Biology in Tübingen, for giving me the opportunity to work in his department and for introducing me to the fascinating world of bioinformatics analyses. He has been a constant source of inspiration and his suggestions and advice enabled me to proceed. All colleagues in the department are cordially recognized for their support, especially Sergej Duranovic, Dr. Michael Habeck and Dr. Johannes Söding. Further, I thank Dr. Kristin Koretke for her advice.

I must express my gratitude to my family for their support, patience, love and encouragement.

Es gibt ein Ziel, aber keinen Weg;
was wir Weg nennen, ist Zögern.
(Franz Kafka)

Zusammenfassung

Im Rahmen dieser Arbeit beschäftigte ich mich mit rechnerbasierten Methoden zur Analyse von Proteinstrukturen und deren Modellierung. Proteine werden nach Vorlage ihrer Sequenz, des Gens, in der Zelle hergestellt und bilden räumliche Strukturen aus, die Bedingung sind für die Ausübung biologischer Funktion durch das Protein. Ausgehend von der Proteinsequenz, basiert die rechnergestützte Strukturvorhersage für ein Protein auf der Suche nach signifikanten Homologien zu anderen Proteinen mit bekannter Struktur. Da die direkte experimentelle Untersuchung von Proteinen, z.B. deren Strukturbestimmung, sehr aufwändig ist, bieten Vorhersagemethoden, die auf Homologie basieren, praktische Alternativen. Sequenzanalysemethoden der Bioinformatik helfen uns Homologien zwischen Proteinen zu bestimmen, d.h. evolutionäre Beziehungen nachzuvollziehen, die auf gemeinsame Abstammung schliessen lassen. Anstrengungen der Strukturbiologie haben zusätzlich zum Ziel, Makromoleküle auf atomarer Auflösung zu erhellen und somit auch Einblicke in biochemische Reaktionen zu ermöglichen. Je mehr Strukturen gelöst wurden, umso augenfälliger wurde der Zusammenhang zwischen Sequenz und Struktur: Falls Proteine ähnliche Sequenzen aufweisen, dann sind ihre Faltungen im Allgemeinen auch ähnlich. Dieser Zusammenhang findet direkte Anwendung in der rechnergestützten Strukturvorhersage von homologen Proteinen. Gegenwärtig enthält die Protein Data Bank (PDB) insgesamt mehr als 80'000 Proteinstrukturen (verteilt auf ca. 39'000 Dateien), wovon etwa 70'000 signifikant homolog (mindestens 50% Sequenzidentität) zu anderen Strukturen sind. Das bedeutet, dass diese Datenbank auf Ebene der Proteinsequenzen eine etwa achtfache Redundanz aufweist. Ein extremes Beispiel hierfür sind die Strukturen von Antikörpern, welche in mehr als 2'000 Konformationen vorhanden sind. Für andere Proteinstrukturen sind keine Homologien nachweisbar; sie kommen bis jetzt nur einzeln vor. Die Analyse von Gruppen homologer Proteinstrukturen liefert wertvolle Informationen über mögliche Konformationen der Proteine. Die Struktur von Proteinen kann nicht als starr angesehen werden, sondern befindet sich in einem Zustand von mehreren bestimmten Konformationen. Solche Konformationszustände können mit der biologischen Funktion zusammenhängen und auch von aussen beeinflusst sein, z.B. dem Vorhandensein eines Liganden. Sequenzanalysen können keine Konformationsänderungen identifizieren. Deshalb lassen heute existierende Strukturvorhersagemethoden, die ausschliesslich auf Sequenzvergleichen aufbauen, alternative Konformationszustände einer Proteinfamilie ausser Acht. Am Beispiel der Modellierung von Aminotransferasen wird aufgezeigt, dass die Qualität der Modelle deutlich verbessert werden kann, wenn den unterschiedlichen Konformationen dieses Proteins Rechnung getragen wird. Auch in Zukunft wird die Anzahl experimentell gelöster Strukturen weiter ansteigen, jedoch werden nur wenige neue Faltungen definieren. Die meisten neu gelösten Strukturen werden homolog zu schon bekannten sein, können aber dadurch neue Konformationszustände aufzeigen. Aus diesem Grund muss der vergleichenden Proteinstrukturanalyse, wie sie in dieser Arbeit beschrieben wird, wachsende Bedeutung beigemessen werden.

Die vorliegende Arbeit ist in drei Teile gegliedert. Im ersten Teil wird die rechnerische Umgebung MolTalk beschrieben. Ein Vergleich mit anderen Programmierbibliotheken ähnlicher Ausrichtung ergab, dass MolTalk in Hinsicht auf Geschwindigkeit und Speichernutzung für die Interpretation von Strukturdaten im PDB-Format deutliche Vorteile

le besitzt. Beide Eigenschaften sind Grundvoraussetzungen für die auf MolTalk aufbauenden Anwendungen: die Strukturdatenbank MTDB und die relationale Sequenz-zu-Struktur-Suchmethode MBSIS. Am Ende des ersten Teils, wird der Strukturanalyse-Webserver iMolTalk vorgestellt. Die neuartige Integration von Strukturanalysen, Homologiemodellierung und Datenbankzugriff in einem interaktiven Webserver machen iMolTalk zu einem einzigartigen und wertvollen Dienst, der viele Wissenschaftler in der Molekularbiologie anspricht, die sich mit Makromolekülen und deren Strukturen beschäftigen. Im zweiten Teil wird auf die Auswahl von Templaten in der Homologiemodellierung eingegangen. Zuerst wird PDBAlert vorgestellt, ein Softwareagent, der regelmässig die Neueinträge der PDB gegen Sequenzen und Modelle von iMolTalk-Benutzern vergleicht. Neu identifizierte Homologien werden den Benutzern als E-Mail mitgeteilt. Danach folgt eine Beschreibung der Evaluation von möglichen Templaten, welche zwar homolog sind, aber die verschiedenen Konformationen des gleichen Proteins darstellen und deshalb nicht durch Sequenzvergleiche nachweisbar sind. Mit Protopolis entwickelte ich eine Anwendung für die vergleichende Proteinstrukturanalyse. Homologe Strukturen werden dabei zueinander verglichen und gemäss der berechneten Strukturähnlichkeit gruppiert. Überlagerungen von Strukturen solcher Gruppen können anschliessend visualisiert werden, um mögliche Konformationszustände zu identifizieren. Der Vergleich von Annotationen zwischen Untergruppen kann helfen, Hypothesen zu bilden, um die strukturellen Unterschiede zu erklären. So ergab die Analyse von 3'514 Gruppen homologer Strukturen mit mehr als 50% Sequenzidentität, dass mehr als 2'500 Gruppen eine deutliche strukturelle Variabilität aufzeigen. In 101 Fällen scheint die Annotation über die Methode der Strukturbestimmung (Röntgenstrukturanalyse oder Kernresonanzspektroskopie) den grössten strukturellen Unterschied zu erklären. Allerdings sind diese strukturellen Unterschiede nicht biologisch relevant. In anderen Fällen, kann die gemessene Variabilität zum Beispiel durch das Vorhandensein eines Liganden erklärt werden (z.B. Methotrexat in Dihydrofolatreduktase).

Im dritten Teil wird die Homologiemodellierung ausgeweitet auf die Modellierung von Proteinkomplexen mittels einer Kombination von abgeleiteten strukturellen Randbedingungen und Protein-Protein Docking. Als eine biologisch wichtige Anwendung dieses Ansatzes wird die Modellierung von Ringstrukturen von AAA+ Proteinen vorgestellt. Das Ableiten von strukturellen Randbedingungen von bekannten Ringstrukturen bildete dabei die Basis der Multimermodellierung. In weiteren Schritten wurden diese dann im Docking der Monomere angewendet. Dies führte zu Strukturmodellen von Oligomeren, welche die biologische Funktion von AAA+ Proteinen besser erklären können. Für das Strukturmodell des Apoptosoms schlagen wir eine Umordnung von Domänen im Ring vor, konsistent mit den abgeleiteten Strukturbedingungen.

Synopsis

The focus of this thesis was on computer-aided protein structure analysis and homology modeling. Proteins are produced in the cell according to their sequences, which are encoded in their genes. Moreover, biological function of proteins depends on their structure. Computer-aided structure prediction is based on statistically significant homology detection applying sequence comparison between a model protein and proteins with known structure. As the direct study of proteins *in vitro* and *in vivo* requires laborious experiments, prediction methods relying on homology offer practical alternatives. In this context, bioinformatics methods for sequence analysis can identify homologies between related proteins, which have evolved from a common ancestor. Moreover, structural biology addresses the elucidation of macromolecular structures at atomic resolution and provides insight into the molecular basis of biochemical reactions. The more structures were solved experimentally, the more it became apparent that proteins with similar sequences predominantly share similar structural architectures (folds). An immediate application thereof is computer-aided modeling of protein structures by homology. Currently, the publicly available Protein Data Bank (PDB) contains more than 80,000 protein structures (organized in over 39,000 files). However, for nearly 70,000 entries at least one homologous structure can be significantly identified, which in other words corresponds to an eightfold redundancy of this database. As an example, more than 2,000 structures of antibodies are present in the database. In contrast, some protein structures only occur as singletons. Important information about protein conformation is revealed from analyzing groups of homologous protein structures. Protein structure cannot be regarded as a rigid object, rather it exists in one defined conformational state that is related to biological function and can depend on external effects, e.g. the presence of a ligand. Because there is no signal for conformational changes at the level of sequence, sequence analyses fail to detect them. Consequently, today's structure prediction methods, which rely on sequence homology detection for modeling, may overlook alternative conformational states in a protein family. As shown in this work for the remodeling of aminotransferases, information about protein conformation can lead to better homology models. In the future, even more protein structures will be solved experimentally, yet only a few will show new and unrelated folds. Therefore, the majority of new structural data will be redundant with respect to sequence. As a result, comparative structural analyses in homology modeling, as introduced in this work, will gain in importance.

This thesis consists of three parts. In the first part, the computational environment MolTalk is introduced. In a comparison to other programming libraries, which serve similar tasks, MolTalk was shown to be very fast in loading and interpreting PDB-formatted files and its memory requirements were medium. These properties are key for the structural database system MTDB and the relational sequence-to-structure system MBSIS. At the end of this first part, our structure analysis web-server iMolTalk is presented. The novel integration of structural analyses, homology modeling and database access make iMolTalk a unique and valuable service to scientists in molecular biology, who work on macromolecules and their structures.

In the second part, template selection in homology modeling is addressed in more detail.

First, I describe PDBalert, a software agent, which periodically compares sequences and models of iMolTalk users against the released structures from PDB and reports new homologies by e-mail. Second, the problem of evaluating putative templates is addressed. Although these templates are homologous, they might represent different conformations, which cannot be detected by sequence comparison. As an application for comparative structural analysis I developed Protopolis, which exhaustively compares homologous structural chains and clusters them according to structural similarity. The resulting groups can then be superimposed and visualized to identify possible conformational states. Comparison of structure annotation between such groups may generate hypotheses that help explaining their structural variability. For instance, the analysis of 3,514 groups of homologous structures, which showed more than 50% sequence identity, revealed that more than 2,500 groups exhibit notable structural diversity. For instance, in 101 groups most structural diversity can be explained by the annotation of the structure determination method, either X-ray crystallography or NMR (nuclear magnetic resonance). However, such structural differences are generally not biologically relevant. In other cases, structural diversity in some groups can be explained by presence or absence of a ligand (e.g. methotrexate in dihydrofolate reductase).

The third part extends homology modeling to multimer modeling using a combination of derived structural restraints and protein-protein docking. As a biological important application the modeling of ring assemblies of AAA+ proteins is presented. The derivation of structural restraints from known ring structures forms the basis of the multimer modeling. Subsequently, they are applied in the docking of the monomers. This leads to oligomer models that can better explain biological function of AAA+ proteins. For the apoptosome, we propose a reorientation of domains in the ring consistent with the derived structural restraints.

Contents

1	General introduction	14
1.1	Proteins	14
1.2	Evolution	15
1.3	Homology modeling	17
1.4	Multimer modeling	19
1.5	Overview	20
2	MolTalk	22
2.1	Introduction	22
2.2	Implementation	22
2.2.1	Hierarchical object representation of macromolecular structures	24
2.2.2	Structure manipulation	25
2.2.3	Sequence and structural alignment	25
2.2.4	Coordinate hashing	26
2.3	Comparison of toolkits	27
2.3.1	PDBlib	27
2.3.2	BALL	27
2.3.3	CCP4 mmdb	27
2.3.4	MMTK	28
2.3.5	pymmtk	28
2.3.6	MBT	28
2.3.7	Benchmark	28
2.4	Summary and Outlook	31
3	Relations of macromolecules	32
3.1	Introduction	32
3.2	MTDB	32
3.2.1	Overview of relational databases for protein structures	34
3.3	MBSIS	35

3.3.1	Implemented relational operators	37
3.3.2	Implemented feature filters	37
3.3.3	Query designer	39
3.4	Summary and Outlook	40
4	iMolTalk	42
4.1	Introduction	42
4.2	Implementation	43
4.2.1	Toolchains and cases	43
4.2.2	Toolchain editor	45
4.2.3	Structural objects and databases	47
4.2.4	Clipboard and report generator	47
4.2.5	Visualization	48
4.3	Structural analyses	48
4.3.1	Residue contact finder	49
4.3.2	Protein-protein interface description	49
4.3.3	Distance map	49
4.3.4	Structural alignment and differential distance map	50
4.3.5	Miscellaneous toolchains and integrated third-party analyses	50
4.4	Other web resources	54
4.5	Summary and Outlook	54
5	Template selection	56
5.1	Introduction	56
5.2	Remodeling of aspartate aminotransferases	57
5.3	PDBChainSaw	59
5.3.1	PDBfused - compression of redundancy	60
5.3.2	Template search in PDBChainSaw	61
5.4	PDBalert	64
5.5	Protopolis	65
5.5.1	Structural comparison	65
5.5.2	Clustering	65
5.5.3	Superimposition	67
5.5.4	Annotation	67
5.5.5	Clustering trees	68
5.5.6	Detection of conformational states in p97 AAA+ structures	69
5.6	Summary and Outlook	72

6	Modeling of AAA+ ring structures	75
6.1	Introduction	75
6.2	Structural restraints	77
6.2.1	Position and orientation of monomers	77
6.2.2	Relative position of ATPase and C-domain	79
6.2.3	Nucleotide coordination	80
6.2.4	Oligomer interfaces	83
6.3	Optimization of ring structures	83
6.3.1	Monte-Carlo sampling	85
6.3.2	Web-interface	87
6.3.3	Benchmark	88
6.4	Modeling cases	92
6.4.1	ClpB	92
6.4.2	Apaf-1	94
6.4.3	MalT	96
6.5	Summary and Outlook	96
	Bibliography	99
A	Protein structures by EC families	112
B	AAA+ ring modeling data	115
C	Curriculum Vitae	122

List of Figures

1.1	Homology between proteins.	16
1.2	Homology modeling protocol.	17
2.1	Number of structures deposited in the PDB.	23
2.2	Class diagram of MolTalk.	24
3.1	MTDB database layout and class diagram.	33
3.2	MBSIS query designer.	40
4.1	Communication schema.	43
4.2	Toolchains made up from cases.	44
4.3	Configuration of the iMolTalk server.	45
4.4	Toolchain editor.	46
4.5	Object specific menus.	47
4.6	Visualization of a residue selection.	48
4.7	Distance maps of β -trefoil proteins and their comparison.	51
4.8	Differential distance map reveals domain motion in aminotransferases.	52
5.1	Yearly depositions to the PDB.	56
5.2	Remodeling of aspartate aminotransferases in open and closed form.	57
5.3	Domain motion in aspartate aminotransferases.	58
5.4	PDBChainSaw database schema.	59
5.5	Alignment improvements by PDBChainSaw.	60
5.6	PDBalert update procedure.	64
5.7	Clustering tree of p97 structures.	66
5.8	Annotation of branch 0L in the clustering tree of p97 structures.	67
5.9	Differential annotation view.	68
5.10	Cumulative number of trees versus top-level split distances.	70
5.11	Verification of p97 structure clustering.	71

6.1	Gallery of AAA+ ring structures.	76
6.2	Topology of a AAA+ NBD monomer.	77
6.3	Orientation of monomers in AAA+ ring structures.	78
6.4	MBSIS filter to automatically define P1-3 in AAA+ structures from structural and sequence restraints.	79
6.5	Nucleotide coordinating residues highlighted in the alignment of AAA+ structures.	81
6.6	Nucleotide coordinating residues.	82
6.7	Handedness of AAA+ ring complexes.	83
6.8	Residues at the inter-monomer interfaces mapped to the structural alignment of ring forming AAA+ structures.	84
6.9	Ring structure optimization protocol.	85
6.10	Potential for distance restraint evaluation.	86
6.11	Energy graphs as shown in the web-interface.	87
6.12	Table of results of the ring modeling.	88
6.13	Benchmark of ring structure remodeling from single monomers.	89
6.14	Evaluation of parameters in the automated ring modeling.	91
6.15	Hexamer model of ClpB tandem AAA+ domains.	93
6.16	Model of the apoptosome with the AAA+ cassette placed in its center.	95
6.17	Alignment between MalT and Apaf-1.	96

List of Tables

2.1	Comparison of programming libraries.	30
3.1	Feature filters implemented in MBSIS.	38
3.2	Feature filters (selectors) implemented in MBSIS.	39
4.1	Parametrization of H-bond donors and acceptors.	50
5.1	Compression of sequence redundancy in the PDB.	60
5.2	List of the 25 most populated groups of homologous protein structures in PDBChainSaw.	62
5.3	Number of trees with information gain 1.0 per attribute.	68
A.1	Protein structures by EC families.	112
B.1	Key residues in AAA+ structures I.	115
B.2	Key residues in AAA+ structures II.	116
B.3	Reconstructed AAA+ ring structures.	117
B.4	Non-ring forming, experimentally determined AAA+ structures.	118
B.5	Distances describing the relative orientation of AAA+ nucleotide binding domain vs. C-domain.	119
B.6	Angular parameters describing the relative orientation of AAA+ nucleotide binding domain vs. C-domain.	120
B.7	Benchmark of the ring modeling pipeline: remodeling of p97.	120
B.8	Benchmark of the ring modeling pipeline: remodeling of NSF.	121

Chapter 1

General introduction

The goal of this work was to develop computational methods to improve the analysis of protein structures and their modeling by homology in order to increase our understanding of proteins. Therefore, in the first section I will introduce proteins in general, followed by a description of their evolution and the sequence-to-structure-to-function paradigm. On this basis, the technique of homology modeling is presented and results are summarized. At the end, multimer modeling of protein complexes is introduced and its application to a biological relevant case is demonstrated.

1.1 Proteins

Cellular function in living organisms depends on biochemical reactions, which proteins (enzymes) facilitate. Other proteins, for example, build structures to maintain the shape of cells or to provide routes along which macromolecules may be transported. The information for a protein is encoded in DNA and maintained as a gene in the genome of an organism. Transcription of this DNA region to RNA yields a copy, which serves as a template for the synthesis of the protein. At the ribosome the RNA is translated into a polypeptide chain. The genetic code assigns one amino acid to three consecutive positions on the RNA. At any position in DNA four different types of nucleotides may occur (ACGT). In proteins, there exist 20 different types of amino acids (ACDEFGHIKLMNPQRSTVWY) and they are distinguished by their sidechain atom groups. The mainchain heavy atoms are the same for all amino acids and when they are covalently bonded via the peptide bond between the carboxy and amino groups of adjacent amino acids they form the backbone of the protein. While a protein is being synthesized by the ribosome, its continuously produced polypeptide chain starts entering a complex process, termed folding, to find a distinct three-dimensional structure (Anfinsen, 1973; Anfinsen and Scheraga, 1975). It is largely unknown how the polypeptide chain folds descending along the gradient of a free energy funnel to find its native conformation, even in different environments and, in some cases, without help from outside (Dinner et al., 2000). It is assumed today that the major driving force of this process is the burying of non-polar amino acids in the core of the protein to shield them from surrounding solvent. Moreover, formation of

H-bond stabilized secondary structures (α -helices, β -sheets or turns) also contributes to this process (Dill, 1999). Nevertheless, proteins do find their distinct three-dimensional structure (fold) and it is predicted that the number of different folds is limited to around 1,000, supported by the observation that even unrelated proteins may adopt similar folds (Orengo et al., 1994). A more recent study predicts the number of folds to be at least 10,000 (Coulson and Moulton, 2002). The key argument is that most folds only span a few sequence families, thus the sequence space around each fold is limited. This suggests that protein folds either have evolved relatively recently and they did not have time to explore far into sequence space, or each fold is unique and tolerates only a limited variability in sequence. The latter also suggests that new folds probably did not arise from existing ones by sequential modifications, but rather by more dramatic and still unknown mechanisms.

1.2 Evolution

During the course of evolution populations evolve. They do this genetically by accumulating mutations in their genes, which could provide them with new or modified biological function. In return this might prove advantageous to an organism and help it to better adapt in a changing environment. Following, the new alleles might become frequent in the population. Such a mechanism can lead to speciation, i.e. the creation of a new species. The corresponding genes in both species are said to be orthologous. Another mechanism by which new genes can arise is by duplication within the same organism. These two paralogous genes can then diverge independently. In both cases, genes that are descendants from a common ancestor are homologous.

Three properties are key to proteins: sequence, structure and function (Figure 1.1). Comparison of these properties can define homology between proteins. Two proteins can be assumed to be homologous if comparison of their sequences is statistically significant. However, the converse generally does not hold. Distant homology can be assumed for proteins which commonly share structural and functional similarity (Murzin, 1998). The comparison of two sequences results in their pairwise alignment (Smith and Waterman, 1981; Needleman and Wunsch, 1970). All aligned amino acid pairs in the alignment are scored using substitution matrices (Henikoff and Henikoff, 1992) and gapped pairs are penalized. The sum of these scores is the total score S of the alignment. If one compares a single sequence against a database of sequences, a number of pairwise alignments are computed and the question is how one can select statistical significant hits. Using an extreme value distribution, the expectation score of a hit is $E = K m n e^{-\lambda S}$, where K and λ are estimated parameters and m and n are the lengths of the compared sequences. This score can be interpreted as the frequency of finding an alignment with score S just by chance. The probability of finding another alignment with at least the same score is $P(x \geq S) = 1 - e^{-E}$ (Altschul et al., 1990; Altschul and Gish, 1996). An important property of this statistics is its dependency on the length of the compared sequences. Further, it can be shown that the size of the database also is influencing this type of statistics. In the remainder of this thesis it is assumed that the expectation value E is small ($E < 0.001$), indicating a significant alignment, and only the percentage of identical amino acids in the

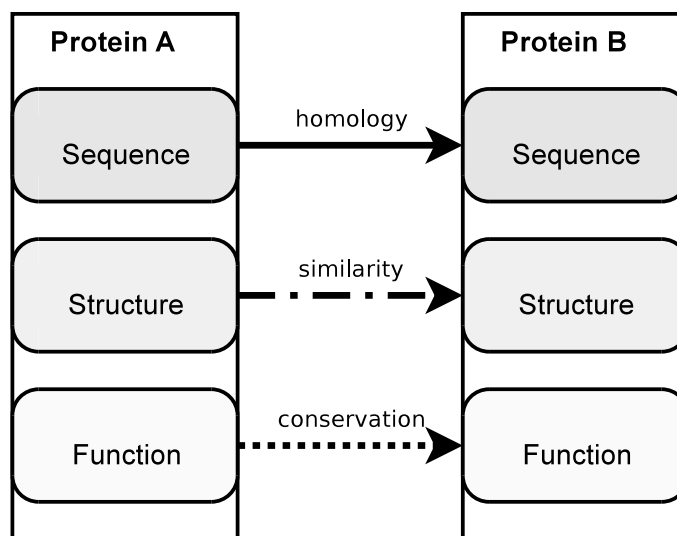


Figure 1.1: Homology between proteins.

The sequence-to-structure-to-function paradigm builds on the observation that sequence homology between proteins implies structural similarity, which manifests in conservation of residues involved in biochemical function. The thickness of the arrows indicates how established the relationship is.

alignment is considered to indicate the degree of homology between two proteins.

The determination of high-resolution structures of hemoglobin (Perutz et al., 1960) and myoglobin (Kendrew et al., 1960) mark the onset of structural biology. It became apparent that proteins can exhibit similar 3D structures even if their sequences are not identical (Perutz et al., 1965), thus structure is more conserved than sequence. Moreover, a sufficiently high degree of sequence conservation implies structural similarity (Chothia and Lesk, 1986). Together with the observation that catalytic sites are non-locally encoded in protein sequences, but are highly conserved, led to the sequence-to-structure-to-function paradigm. Because most proteins can fold autonomously, folding restraints must be encoded in their sequences, thus structure is mainly determined by sequence. And because only spatially close residues can form biochemically active catalytic centers, structure determines protein function. For studying protein function, it is therefore of interest not only to know a protein's sequence but also its structure.

A number of complete genomes have been sequenced (Venter et al., 2001). Together with data from expressed genes, this allows us to look at drafts of the complete proteome of different organisms. Most importantly, this information can be searched comparatively for genes specific to a single organism, thus representing proteins that can be targeted selectively (e.g. aiding the design of novel drugs to control microorganisms related to human diseases). However, biological function is not known for many proteins, yet their sequences and, to a lesser extent, their structures are available to bioinformatics analyses (Baker and Sali, 2001). There is tremendous interest to describe proteins of unknown function and the most supportive tool is homology. Therefore, if one can outline a homo-

logy between two proteins, one may also be tempted to transfer annotated knowledge between them. Homology modeling, described in the next section, can provide in some cases three-dimensional model structures, which can be annotated by such knowledge.

1.3 Homology modeling

Homology modeling is an established procedure to infer three-dimensional model structures of a protein sequence based on homology to experimentally determined structures (Greer, 1981; Guex et al., 1999). Starting with only the model sequence, four steps in the protocol (Figure 1.2) lead to a model structure: 1) template selection, 2) target to template alignment, 3) structure computation and 4) model quality assessment.

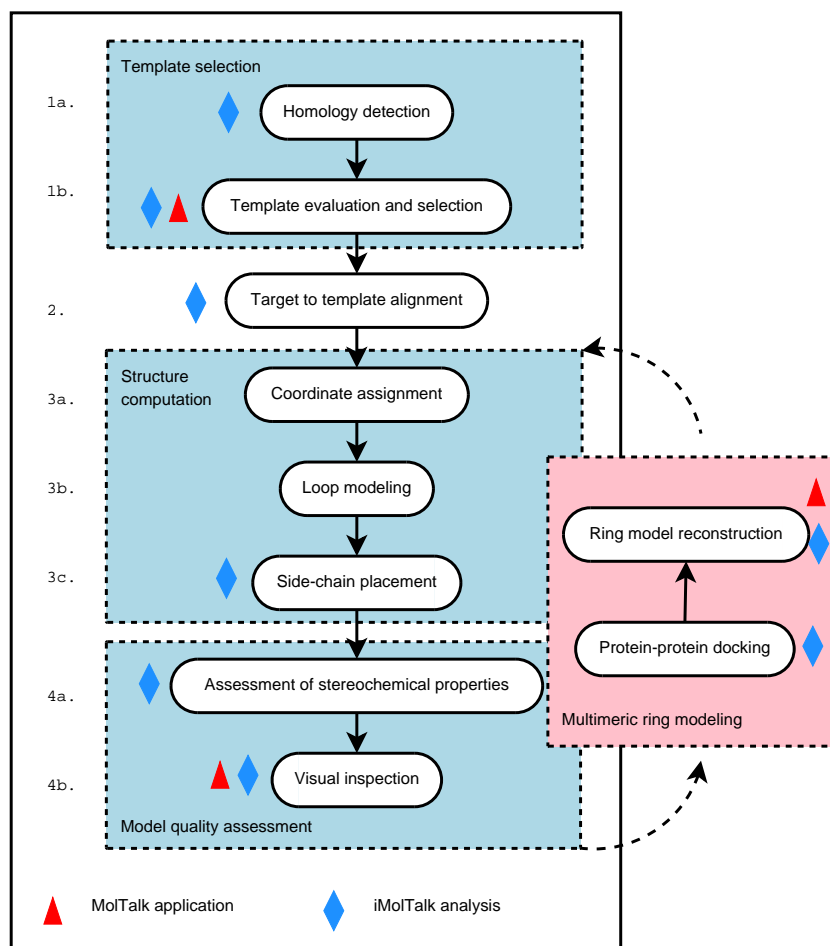


Figure 1.2: Homology modeling protocol.

1) Template selection

Template selection can be divided into two sub-problems: 1a) homology detection and 1b) evaluation and selection of suitable template(s).

Homology detection is mainly done using sensitive sequence-to-profile comparison methods, like PSI-BLAST (Altschul et al., 1997), or profile-profile comparisons, i.e. hhsearch (Soding, 2005), against the database of protein sequences with known structures. To support such searches, we developed *PDBChainSaw* (section 5.3) to thoroughly reconstruct these sequences by taking into account non-resolved or modified residues. This method was implemented in the structural bioinformatics toolkit *MolTalk* (Diemand and Scheib, 2004b), described in chapter 2, and was integrated into the SNP modeling and analysis pipeline ModSNP (Yip et al., 2004).

In most cases, the best scoring homolog(s) are selected and used in the subsequent modeling without further evaluation of their suitability. Because homology detection methods only rely on sequence properties and ignore conformational changes, a novel method for comparative structural analysis, *Protopolis* (section 5.5), was developed to infer conformations in homologous structures and to evaluate their suitability. First, *Protopolis* structurally compares the set of homologous chains and clusters them according to structural similarity. Then, structural superpositions of members of subgroups in these clusters may reveal distinct conformations. Furthermore, the differential analysis of structural annotation may lead to hypotheses that could explain the clustering into subgroups. As an example, I will discuss the findings for aspartate aminotransferases (AAT) and the structures of the AAA+ protein p97 (chapter 5). Template selection is also supported by *PDBAlert* (section 5.4), a software agent, which finds homologous hits to user-supplied sequences in the newly released structures of the Protein Data Bank (PDB) and reports putative templates by email. It is not restricted to sequence comparison, but can also search with structural models that are stored in the relational database *MTDB* (section 3.2). These applications are integrated into the structure analysis web-server *iMolTalk* (chapter 4).

2) Target to template alignment

To proceed in the modeling protocol, an optimal sequence alignment between the model and the template(s) needs to be computed. Target to template alignment is a difficult task (Levitt, 1997; Karplus et al., 1999; Koretke et al., 1999, 2001) and remains, apart from template selection, the main source of errors in homology modeling. Recently, a protocol has been established to computationally explore alignment alternatives (John and Sali, 2003).

3) Structure computation

Structure computation can be done either fully automated (Sali and Blundell, 1993; Guex and Peitsch, 1997) or manually using interactive structure viewers (e.g. SwissPDB-Viewer). 3a) Coordinates of model residues are computed from aligned positions in the

template(s), guided by the alignment from the previous step. 3b) Unaligned regions have to be remodeled, especially loop regions. Generally, loops are more variable than the core, i.e. indels (insertion/deletion) occur frequently (Fiser et al., 2000). 3c) Side-chain orientations are modeled to optimize interactions and remove clashes (Lovell et al., 2000).

4) Model quality assessment

Model quality assessment has similar aspects as assessment of three-dimensional structures that were fitted to experimental data, i.e. in X-ray crystallography or NMR spectroscopy (Kleywegt and Jones, 1996; Hooft et al., 1996; Kleywegt, 2000). 4a) Evaluation methods take into account stereo-chemical properties (Laskowski et al., 1993) or atom naming and packing (Vriend, 1990). 4b) Moreover, visual inspection of the computed models is necessary to validate them against biological knowledge that was not included in the computation. This step requires profound knowledge of structural biology and special computer hard- and software. The interactive structure analysis web-server *iMolTalk* (Diemand and Scheib, 2004a) addresses these needs and provides an user-friendly interface to predefined structural analyses (chapter 4).

1.4 Multimer modeling

The established protocol of homology modeling is tailored towards modeling of monomeric structures. Nevertheless, modeling of protein complexes is of increasing importance, because many proteins function only in homo- or hetero-oligomeric assemblies. In other words, to help understanding biological function of these proteins, their complex structures need to be determined. However, such experimental determination often is hampered by non-diffracting crystals or crystallization in non-physiological arrangements. Moreover, such complexes are in general too large to be studied by NMR spectroscopy. Using homology modeling in general, the oligomeric state of a protein can be predicted only if the template structures themselves were solved in the same oligomeric state. In absence of such complex templates, protein-protein docking may be applied, which searches for relative rotations and translations of two molecules and evaluates the quality of spatial interaction between them (Comeau and Camacho, 2005; Schneidman-Duhovny et al., 2005). However, to fully search the six free parameters in protein-protein docking (three rotations around x, y and z axes and three translations along the same axes) leads to the problem of large computational complexity. Moreover, the discrimination between desirable and unwanted solutions using scoring functions is a challenging problem. One solution to this problem is to restrain the degrees of translational and rotational freedom within biologically meaningful values. As a result, computational complexity is greatly reduced. Moreover, the quality of resulting models can be evaluated using the applied restraints.

For many members of the AAA+ protein family (Lupas and Martin, 2002; Frickey and Lupas, 2004; Neuwald et al., 1999), which provide mechanic work fueled by ATP hydrolysis, only electron microscopy images of their mostly hexameric complexes are available.

Therefore, we developed a semi-automated modeling procedure (chapter 6) for constructing ring-shaped complexes from monomeric structures of AAA+ proteins (Diemand and Lupas, 2006). This procedure is based on a number of constraints derived from known crystal structures and uses a combination of Monte-Carlo sampling and protein docking that iteratively leads to favorable model structures. Our analyses show that the position of the core ATPase and C-domain is preserved within a narrow range in the extended AAA+ cassette and that both provide essential interactions to nucleotide binding. Applying this method to a number of AAA+ proteins showed substantial improvement in subunit interactions as compared to modeling by simple superimposition, and this yields new insight into the oligomeric structure of these proteins and might better explain biological function. For the structure of the apoptosome, we proposed a reorientation of domains in the ring, compatible with the derived structural restraints.

1.5 Overview

This thesis consists of three parts. In the first part the computational environment is introduced and described, followed by the second part, which addresses template selection in homology modeling in more detail. The third part presents the application of the first two parts to the biological important modeling of AAA+ ring structures.

Computational Environment

Chapter 2

MolTalk

A programming library for structural bioinformatics

2.1 Introduction

With the growing size of the macromolecular structure repository Protein Data Bank¹ (PDB), a need for computational analyses in structural biology is apparent and the new field of structural bioinformatics addresses this (Figure 2.1). The number of structural chains with more than 20 amino acids has exceeded 80,000 in August 2006. As an example, homology search with the sequence of human cyclin-dependent kinase 2 (CDK2) against the derived sequences from coordinates of the complete PDB yielded 726 significant hits. Among them, 159 share more than 89% sequence identity, meaning that nine out of ten amino acids are not altered. Different experimental conditions were tried to study these proteins and in many cases only the co-crystallization of small chemical compounds was tested, yet structurally this might have an influence on the proteins's conformations.

To facilitate access to such data and to provide an environment for structural computations, MolTalk and supportive applications, e.g. PDBChainSaw (section 5.3), MTDB (section 3.2) and MBSIS (section 3.3) have been developed. Here, I describe the implementation of MolTalk and compare it to other programming libraries that serve similar tasks. MolTalk was shown to be very fast in interpreting PDB-formatted structure files and its memory requirement was medium.

2.2 Implementation

MolTalk is implemented in the computer language Objective-C as a programming library and in the following is referred to as *libmoltalk*. It uses the rich set of data structures

¹<http://www.rcsb.org/pdb>

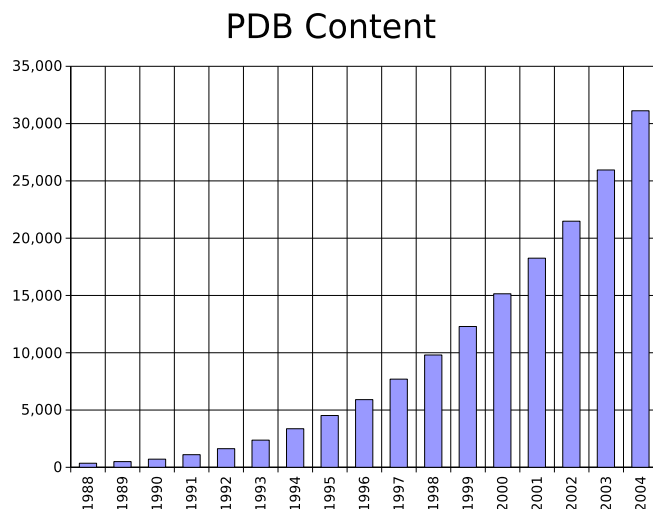


Figure 2.1: Number of structures deposited in the PDB.

Currently, the Protein Data Bank (PDB) contains more than 35,000 structures. Exponential growth has been observed over the last 10 years. As structures can be put on hold for up to 12 months, only data before 2005 are shown.

(among them lists, sets, dictionaries) defined in the framework GNUstep², a reimplementation of the OpenStep specification. Objective-C is a super-language of C. It provides Smalltalk-like message calling, while the added syntax remains minimal. This makes Objective-C advantageous to the more commonly used C++ language. In contrast to C++, the memory address of a method call is not defined at compile and link time, but it is evaluated as the message is passed from the sending object to the recipient. Nevertheless, caching of evaluated addresses guarantees minimal overhead. This dynamic translation of messages to object methods renders Objective-C ideal for implementing interpreted languages. The GNUstep project integrates StepTalk³, a scripting language framework and an interpreter for the Smalltalk language. To this interpreter, I have added access to the objects provided by *libmoltalk* and extended the set of mathematical operators. Furthermore, I optimized the interpreter with regards to execution speed and ported it to Windows and MacOSX. To differentiate it from the programming library *libmoltalk*, the interpreter is referred to as *MolTalk*. Combining Objective-C and Smalltalk has the advantage that the two languages inter-operate without any stub code bridging between them, i.e. new methods added to the Objective-C library *libmoltalk* are immediately available from the scripting language *MolTalk*. While the interpreted language is only responsible for program execution, i.e. message passing between objects, time consuming algorithms are implemented in C and compiled to native code.

Technical documentation of the implemented classes is available from the MolTalk⁴

²<http://www.gnustep.org>

³<http://www.gnustep.org/experience/StepTalk.html>

⁴<http://www.moltalk.org>

homepage, as well as an extensive tutorial for learning how to program with *libmoltalk* and scripting in *MolTalk*. The project was licensed under GPL (GNU General Public License) and the source code is available from bioinformatics.org⁵.

2.2.1 Hierarchical object representation of macromolecular structures

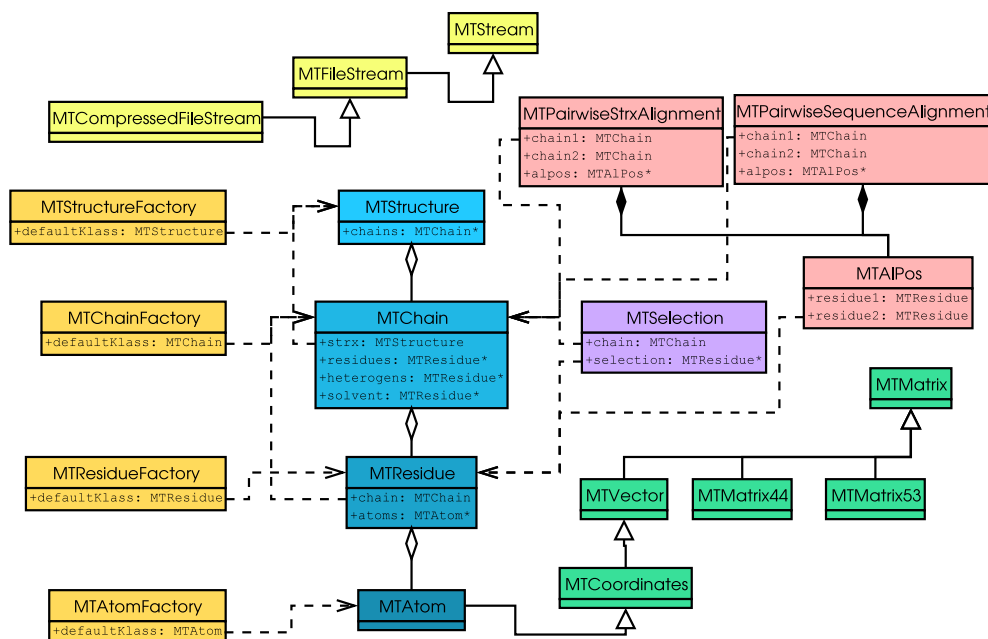


Figure 2.2: Class diagram of MolTalk.

Central to MolTalk is its representation of macromolecular structures using a hierarchy of instances of structural classes (*MTStructure*, *MTChain*, *MTResidue* and *MTAtom*). Object factories hide actual implementation details and provide a general interface for object creation to utility classes (i.e. file readers). Connections starting with triangles indicate inheritance between classes and those starting with a rhomb indicate composition. Dashed lines indicate associated classes via references.

Macromolecular structures are hierarchically represented by the four classes *MTStructure*, *MTChain*, *MTResidue* and *MTAtom* (Figure 2.2). An object of class *MTStructure* contains one or more lists of structural chains (*MTChain*) that are sequences of atom groups (*MTResidue*). Such residues maintain lists of instances of the class *MTAtom*. Object creation is facilitated by factories, specialized classes that instantiate a type of object on behalf of other utility classes (i.e. structure file readers). By exchanging factory implementations, programs can switch the type of created structural classes without actually knowing them. This technique is used in many implementations, where derived structural classes contain additional fields and respond to more messages. As an example, the implementation of the mapping of structural objects to a relational database, MTDB

⁵http://www.bioinformatics.org/groups/?group_id=307

(section 3.2), re-implements the structural classes but also provides its own factories to create them. Once these factories are registered with the system, loading and interpreting a structure from a PDB-formatted file will create instances of the derived structural classes. The file interpreter is not aware of this, and no change in this utility class is needed to obtain the new functionality.

Basic vector algebra is implemented in *MTMatrix* and its derived classes. Reading and writing to streams, of which files are a special case, is provided by *MTStream* and subclasses. Special utility classes are *MTPairwiseSequenceAlignment* and *MTPairwiseStrx-Alignment*. The first class computes local (Smith-Waterman) and global (Needleman-Wunsch) pairwise sequence alignments, which can subsequently drive the structural alignment using least-squares fitting in the latter.

2.2.2 Structure manipulation

Loading structure files is provided by the class *MTStructureFactory*, which returns the top-level entry point *MTStructure* to access the hierarchy of structural objects. At every level, the hierarchy may be inquired for associated objects using enumerators, or objects may be accessed through their identifiers. Atoms are identified by their names, residues by their numbers (plus an insertion code, eventually), and models by their number. The PDB format defines chain identifiers as single characters (non-control character or a space). In general, such a character is in the range of A to Z, but some structures use numbers or lower case characters as well. However, the structure of a virus capsid, 1gav, contains so many chains in a single structure file that the authors had to choose other characters as well (among them '_', 'l'). Therefore, in order to be most general, we address a chain using the ASCII code of its chain identifier (e.g. the space character or '=32, 'A'=65, 'Z'=90).

The hierarchy of structural objects can be manipulated by attaching or releasing references to chains, residues and atoms in their higher-level containers. As an example, residue R307 in chain A might be moved to a different chain simply by referring to it from the new chain B and releasing its reference in the previous. As soon as the program traverses the hierarchy again, for example on writing the structure to a PDB-formatted file, it will follow the reference to this residue from chain B only.

2.2.3 Sequence and structural alignment

The class *MTPairwiseSequenceAlignment* implements both local (Smith and Waterman, 1981) and global (Needleman and Wunsch, 1970) alignment algorithms to align two amino acid sequences. The implementation follows the improved alignment algorithms with affine gap penalty (Gotoh, 1982, 1999). The result is an ordered list of instances of *MTAlPos* relating two residues at each position in the alignment, or in the case of indels (insertion or deletion) only one amino acid. To compute the best scoring sequence alignment, Blosum exchange matrices (Henikoff and Henikoff, 1993) were included with the programming library (blosum45, blosum62, blosum80). Using least-squares fitting (LSQ) of distances between aligned pairs in the sequence alignment of two structural

chains, *MTPairwiseStrxAlignment* computes the transformation of the second onto the first chain. The implementation follows a quaternion based method (Kearsley, 1989), which led to an eigenvalue problem. In MolTalk, diagonalization of the symmetric 4x4 matrix to find the four eigenvectors and eigenvalues is performed using the Jacobi algorithm (Schwarz, 1997), which iteratively applies Givens rotations to eliminate off-diagonal elements. The convergence criteria was set to $\sum x_{ij} \approx 0$ where $i \neq j$ (default: 10^{-10}). In comparison to the implementation of the QR-algorithm in the GNU scientific library⁶ (GSL), this implementation of the Jacobi method is 11% faster (significance evaluated using t-test, data not shown) in the case of 4x4 matrices. In contrast, for higher-dimensional matrices the QR-algorithm outperforms the simple Jacobi algorithm. Therefore, GSL can optionally be linked to *libmoltalk* for special applications.

Once an initial superimposition is available, the structural alignment of two structures can be iteratively optimized (Cohen, 1997; Petitjean, 1998). First, the pairwise structural alignment is derived. It aligns all residues, which show small $C_\alpha - C_\alpha$ distances below a given threshold (default 6 Å). As alternative pairings may occur, but sequence order must be preserved, dynamic programming is applied with a scoring function that takes into account pairwise distances to find the optimal alignment. Either global or local alignments can be derived using the class *MTPairwiseStrxAlignment*. Second, these aligned pairs of residues are resubmitted to a least-squares fit as described above resulting in a transformation that superimposes the two structures more closely. This process continues until the number of aligned residues in the derived structural alignment stagnates.

2.2.4 Coordinate hashing

Information about protein structures is optimally organized in PDB files if the backbone chain from one residue to its neighbors is followed. Such a search is directed by the peptide bond between residues. To determine spatially close residues, all atoms in a protein need to be investigated. The time complexity of such a search grows linearly with the length of the sequence $O(n)$. However, the number of contacts of every residue already has quadratic complexity $O(n^2)$. Further increase in complexity can be envisaged, if more than one protein chain is involved in the search for contacts. To circumvent this time consuming contact search, the procedure of geometric hashing of three-dimensional coordinates (Nussinov and Wolfson, 1991) was added. Initially, all residues have to be entered once into a hash table. Their coordinates are encoded as the key, which indicates a bin in space, and their identity is used to label them (complexity $O(n)$). Subsequently, search positions are similarly encoded and the hash is queried for spatially close residues (worst case complexity $O(\log n)$, but constant in general). As a consequence, an exhaustive list of a contacts of every residue can be solved in linear time complexity (worst case: $O(n * \log n)$), which is significantly lower than the quadratic complexity for a complete search (Weiss, 1995).

⁶<http://www.gnu.org/software/gsl>

2.3 Comparison of toolkits

In this section I first describe other toolkits, which target structural computations, and then present a benchmark and discuss its results.

2.3.1 PDBlib

PDBlib⁷ (Chang et al., 1994) version 2.2 was released on 1998-11-06. The type of licensing is undefined, but the source code contains a copyright statement by “The Trustees of Columbia University in the City of New York”. Because, the C++ source code is not compatible with today’s GCC compiler⁸, the library had to be compiled with the outdated version 2.8.1. This might have had implications on execution speed as newer compilers contain substantially improved optimizers.

2.3.2 BALL

The biochemical algorithms library⁹ aims at providing a framework for software prototyping in computational molecular biology, especially focusing on protein docking and drug design (Kohlbacher and Lenhof, 2000). It implements molecular mechanics and simulation algorithms, thus provides a complete environment for force field development and deployment. Recently, the group also presented their interactive viewer and modeling program Ballview (Moll et al., 2006), which is based on BALL. In newer versions, this library also contains a bridge to Python and thus allows for scripting. To efficiently search for spatial contacts, BALL provides a hashing grid over three-dimensional data, which returns contacts to other close atoms when queried with atom positions.

2.3.3 CCP4 mmdb

The Collaborative Computational Project number 4 (CCP4, 1994) aims at developing software for X-ray crystallography. Recently, they published the description of their common programming library¹⁰ for coordinate-related programs (Krissinel et al., 2004). This library includes a fast PDB parser (see Table 2.1) and also provides access to formatted files in XML and mmCIF. The code is written in a style similar to C, not following object-oriented design at large. Although, organizing the data in table-like data structures improves speed of computation, it goes along with a loss of elegance and extensibility in the future. Besides its speed, a strength of this programming library is certainly the elaborated mechanism of object selection in macromolecular structures. This library can also be accessed from scripting languages such as Python. The integrated “bricking” algorithm allows generating a coordinate hash, which returns spatially close atoms when subsequently queried with three-dimensional positions. The SSM (secondary-structure

⁷<http://www.sdsc.edu/pb/pdblib/pdblib.html>

⁸<http://gcc.gnu.org>

⁹<http://www.ball-project.org>

¹⁰<http://www.ebi.ac.uk/~keb/cldoc/>

matching) method (Krissinel and Henrick, 2004) is implemented using this library and made available on the EBI web-server¹¹.

2.3.4 MMTK

The Molecular Modeling Toolkit¹² (Hinsen, 2000) provides a complete set of algorithms for molecular mechanics, simulation and normal mode analysis. The library is primarily written in Python, and computationally intensive routines were implemented in C. A first application is the DomainFinder (Hinsen et al., 1999) that identifies dynamic domains in proteins from normal mode analysis.

2.3.5 pymmtk

Another toolkit written in the interpreted programming language Python is mmLib¹³ (Painter and Merritt, 2004). It provides an elaborated mmCIF interpreter that also interprets generic mmCIF dictionaries. CIF is a dictionary based file format, which has been adopted by the PDB consortium to replace the current PDB format. An example application of this toolkit is the visual mmCIF editor to view and manipulate CIF formatted files. Moreover, this library was the basis for the application TLSViewer, which visualizes anisotropic displacement parameters of macromolecules (Merritt, 1999).

2.3.6 MBT

The Molecular Biology Toolkit (Moreland et al., 2005) is being developed at the San Diego Supercomputer Center¹⁴, which also hosts the Protein Data Bank (PDB). The framework is written in Java; thus, it depends on the installation of a Java virtual machine for the interpretation and execution of programs. In the presented benchmarks here, it is much slower than natively compiled code (Table 2.1). On the other hand, using the programming language Java has the advantage that such applets can potentially be integrated into web pages. MBT is used for visualization of protein-ligand interactions in the application Ligand Explorer that is available from the PDB site.

2.3.7 Benchmark

The implementation of MolTalk is compared to the programming libraries presented in Table 2.1. The fact that they all are built on object-oriented technology reflects the hierarchical organization of information about macromolecular structures. In order to compare these libraries, a number of key features were analyzed qualitatively as well as the memory requirements and the time needed to load a structure file and traverse its derived object hierarchy. The structures used for this benchmark were crambin (1crn, single

¹¹<http://www.ebi.ac.uk/msd-srv/ssm/>

¹²<http://dirac.cnrs-orleans.fr/MMTK>

¹³<http://pymmlib.sourceforge.net>

¹⁴<http://mbt.sdsc.edu>

chain, 46 amino acids, 327 heavy atoms) and the ATP-dependent protease HslU/HslV complex (1kyi, 12 chains, 5,916 amino acids and 12 heterogeneous groups, 45,756 heavy atoms). Computing time was measured using the UNIX command “time”, and only user time was noted (100 repetitions). Space complexity was noted as the maximally allocated memory reported by the program “valgrind” for the compiled programs. For Java bytecode-interpreted programs, the Java runtime¹⁵ was directly interrogated after explicitly invoking the garbage collector. Additionally, the memory requirement of the process (RSS, resident set size) was queried from the kernel. The test system was running Linux 2.6.9 in native 64-bit mode on an Athlon 64 (FX-55, 2600 MHz).

From documentation and source code the following features were qualitatively evaluated for each toolkit:

- * **Scripting** Whether the toolkit provides an interface to a scripting language.
- * **Compilation** Whether programs can be compiled to native code and linked to the toolkit.
- * **Object Factories** Whether the toolkit creates objects using factory classes that simplify the implementation of derived structural classes.
- * **Coordinate hashing** Whether the toolkit speeds up the search of atomic contacts using similar techniques as geometric hashing.
- * **Superimposition** Whether the toolkit can compute superimpositions of two structural chains using least-squares fitting.
- * **Structural alignment** Whether the toolkit employs alignment algorithms to derive structural alignments based on dynamic programming.
- * **Visualization** Whether the toolkit allows for preparation of a molecular scene and its visualization.

¹⁵java.lang.Runtime.getRuntime().totalMemory()

Table 2.1: Comparison of programming libraries.

	MolTalk	PDBlib	BALL	CCP4 mmdb	MMTK	pymmtk	MBT
Version	3.0	2.2	1.1.1	1.09	2.4.4	0.9.8	1.1.2
License	GPL	undefined	GPL/LGPL	CCP4/LGPL	LGPL	Artistic	restricted
Reference	Diemand and Scheib, 2004b	Chang et al., 1994	Kohlbacher and Lenhof, 2000	Krissinel et al., 2004	Hinsen, 2000	Painter and Merritt, 2004	Moreland et al., 2005
Programming language	Objective-C	C++	C++	C++	Python	Python	Java
Primary application	iMolTalk	-	Ballview	SSM	Domain Finder	TLSView	Ligand Explorer
Scripting?	yes	no	yes	yes	yes	yes	yes
Compilation?	yes	yes	yes	yes	no	no	no
Object factories?	yes	no	no	no	no	no	yes
Coordinate hashing?	yes	no	yes	yes	no	yes	no
Superimposition?	yes	no	yes	yes	yes	yes	no
Structural alignment?	yes	no	no	yes	no	no	no
Visualization?	yes	no	yes	no	yes	yes	yes
Valgrind allocated, Icm (bytes)	1,468,461	11,767,767	1,127,098	175,504	n/a	n/a	8,077,312
Kernel RSS (bytes)	5,599,232	5,791,744	6,234,112	1,662,976	7,884,800	13,713,408	84,967,424
Valgrind allocated, Ikyi (bytes)	27,947,308	293,883,734	59,164,394	48,123,477	n/a	n/a	25,190,400
Kernel RSS (bytes)	36,782,080	12,406,784	38,424,576	20,971,520	77,144,064	162,844,672	117,850,112
Loading of Icm (seconds)	0.01	0.08	0.02	0.01	0.15	0.3	3.0
Loading of Ikyi (seconds)	0.41	7.8	3.8	0.17	8.39	6.14	9.4

2.4 Summary and Outlook

MolTalk is a complete toolkit targeting structural bioinformatics applications. Its object-oriented design, especially the use of object factories, guarantees a high degree of extensibility. Its speed of computation and the integrated scripting language target it for deployment in distributed computing environments. The simple installation with minimal dependencies on external libraries further supports this.

In comparison to other toolkits, MolTalk shows fast interpretation of PDB-formatted files and medium memory footprint. This efficiency represents a prerequisite to run highly parallel applications such as MBSIS (section 3.3). Our benchmark included both compiled and interpreted toolkits, which were developed targeting specific applications. Therefore, only the time to load a structure and the required memory footprint were compared and other features of these toolkits were evaluated only qualitatively. Moreover, several interactive structure viewers also include scripting capabilities. SPDBV (Guex and Peitsch, 1997), VMD (Humphrey et al., 1996) and Rasmol (Sayle and Milner-White, 1995) implement basic macro languages, which serve the sole purpose to imitate user input from text files instead of via the graphical user interface. These programs were not included in the benchmark as the setup and rendering of the graphical scene takes considerable time.

Algorithms for molecular mechanics simulations are not yet included in MolTalk, but could be provided by the toolkits BALL or MMTK. The technique of delegating object creation to specific factories would reduce the extent of code needed to bridge them to MolTalk. Therefore, it is envisaged to add such support to the target toolkit first.

Chapter 3

Relations of macromolecules

Data handling and querying of structural features in macromolecular structures

3.1 Introduction

Storage and retrieval of massive information has been revolutionized by relational database systems (Codd, 1970). Since their introduction, information technology has been developed to manage data from diverse sources. In this chapter, I introduce MTDB, a mapping of MolTalk's object hierarchy to a relational database, followed by MBSIS, which is a query language based on relational algebra to interrogate relations between sequence and structure of macromolecules.

3.2 MTDB

A relational database of macromolecular structures

Loading and saving structural models from and to file-based repositories, such as the mirrored PDB, is convenient for structural analyses using structure viewers. In a distributed computing environment, where many independent jobs request read and write operations in parallel, the technical implementation of the repository through a distributed file system (NFS) reaches its limits. For example, complex file locking has to be performed in a read/write scenario and single data record level locking is not possible. However, relational databases were specifically designed to handle multiple requests in parallel and provide efficient mechanisms of data locking.

Whereas PDBChainSaw (section 5.3) derives and stores meta data, MTDB directly maps MolTalk's data abstraction of macromolecules to a relational model (Figure 3.1a). The

simplest application of this mapping would be to load a structure from a PDB file and to store it in MTDB. Conversely, it is also possible to instantiate objects from the relational database and to write the object hierarchy to a local PDB-formatted file. Internally, MolTalk and the database server communicate using the structured query language (SQL), a *de-facto* standard for relational databases. As a consequence, the database backend can be chosen freely.

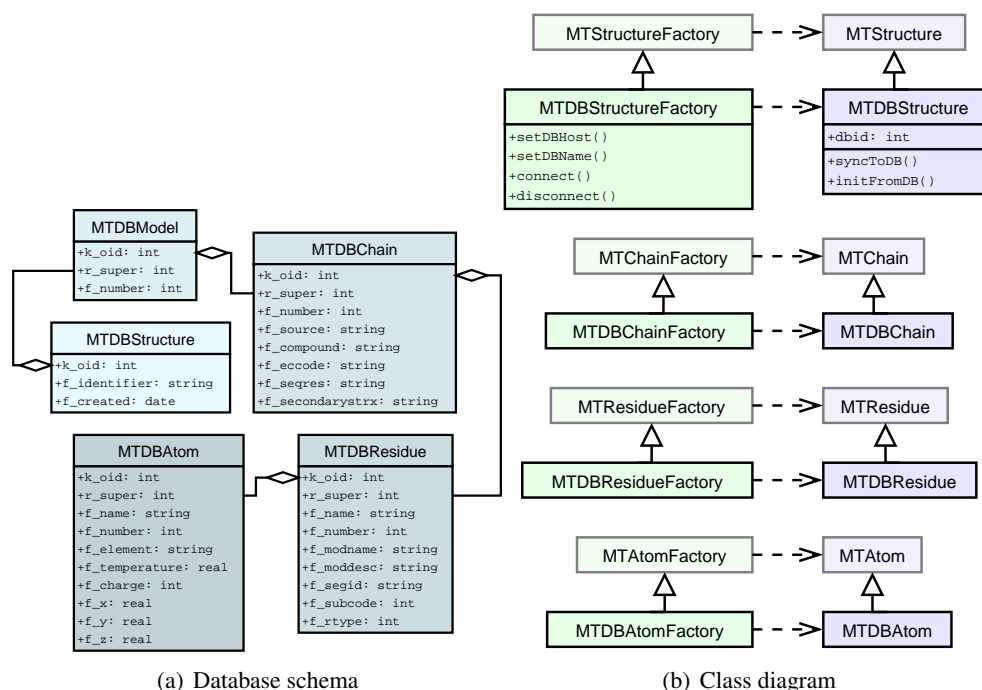


Figure 3.1: MTDB database layout and class diagram.

MTDB provides derived classes of all factories and structural object classes in MolTalk (Figure 3.1b). A connection to the database server can be initiated using the class *MTDBStructureFactory*. Subsequent read and write operations are performed through this communication channel. All structural object classes respond to the message *syncToDB*., which updates their data representation in the relational database, assigns their unique identifier (*k_oid*), and relates them to their higher-level containers (residue for atom, chain for residue or structure/model for chain). The container classes also respond to the message *initFromDB*: to set their own values and create all their subordinate object instances from the database. The class *MTDBResidue* allows the deferred loading of atoms. The first access to atom data through an instance of *MTDBResidue* triggers the signal to load atom instances. This mechanism prevents loading and instantiation of most of the objects in a structure that are never accessed. As an example, suppose a protein structure consisting of four identical chains. If only one chain is studied in an analysis, atom data for residues in the other three chains are neither loaded nor instantiated. Solvent atoms are another important application of late loading. They are often present in high-resolution protein structures, but structural analyses hardly include them in their

computations. Therefore, they may safely be ignored. Using late loading, they are not instantiated until they are accessed.

To evaluate the time and memory requirements of late loading vs. explicit loading of atom data, the structure of hisactophilin (1hcd, NMR, 118 residues, 1,821 atoms of which 864 are hydrogens) was instantiated from MTDB. Late loading allocated 3,989,035 bytes, whereas explicit loading of all atoms required 8,224,369 bytes (as indicated by *valgrind*); thus, late loading saved 49% memory space. The time to load and instantiate this structure was 3.3 seconds for explicit loading and only 0.9 seconds for late loading (averaged over 31 repetitions). Therefore, late loading saved 72% of time. Object instantiation of atoms is the most expensive task in loading a structure representation from MTDB. In the future, it is anticipated that improving the program code for structure loading from MTDB yields even more efficient data handling.

To test the installation of MTDB in a parallel read/write scenario, a selection of 35,412 PDB structures consisting of at least 20 amino acids was loaded into MTDB within 28.5 hours (on an eight processor (750 MHz) Sun V880). The database used approximately 70 GB on disk and contained 35,412 structures and models, 119,180 chains, 19,541,649 residues and 155,336,729 atoms. Furthermore, 460,355 records, among them title, release date and resolution, were derived from the selected PDB files. Only the first models and the first residue conformations were read into memory, then the created object hierarchy was synchronized with the database.

3.2.1 Overview of relational databases for protein structures

In 1989, the first relational database of protein structures (BIPED, based on ORACLE) was published (Islam and Sternberg, 1989). Not only did it store the data of macromolecular structures but also inferred additional information such as secondary structure, hydrogen bonds, disulphide bridges, close contacts. For each residue, close contacts, amino acid type and secondary structure assignment were determined and stored for each of the 15 flanking up- and downstream residues. This data organization introduced numerous redundancies, but on the other hand enabled the authors to run simple queries (without sub-queries) within reasonable response time. As an example, Thornton and colleagues successfully queried BIPED to analyze β -turns in protein structures (Wilmot and Thornton, 1990) by applying only two filter criteria: C_{α} - C_{α} distance between residue i and $i+3$ must be less than 7 Å and the central residues ($i+1$, $i+2$) may not be in a helical conformation. For this analysis, they extracted precomputed mainchain dihedral angles directly from the database. However, the data redundancy in BIPED violates Codd's normalization rule (Codd, 1970). Consequently, the additional space requirements for storage are enormous and would render loading of the complete content of today's PDB difficult.

Data normalization has been addressed in the SESAM project (Huysmans et al., 1991), which provides a relational database (based on SYBASE) integrating both sequence and structure information. A key benefit of the system was the validation of PDB structures by relating to additional tables (e.g. residue topologies, chemical properties of atoms and parameters for conformational energy calculations).

The European Bioinformatics Institute (EBI) maintains MSD, a mapping of PDB data to

a relational database and data cleaning procedure (Boutselakis et al., 2003). The National Center for Biotechnology Information (NCBI) developed MMDB (Chen et al., 2003), a database of PDB data translated into ASN.1 format following detailed encoding rules. ASN.1 encoded structures are stored in single files, but they could probably also be mapped to a relational model. Both systems could potentially be used to store 3D models generated by homology modeling. Such massive data is available from the Sali group, who provides precomputed homology models for a number of protein sequence sources (complete genomes, Swiss-Prot/TrEMBL) through their web-service ModBase (Sanchez and Sali, 1999). Currently, the number of homology models computed and judged trustful after automated validation exceeds 3 million. This huge number of homology models, each accompanied by additional files (i.e. alignment, logs), in ModBase leads to questions about how to manage and analyze them. Clearly, file systems are quite inefficient for flat hierarchies, where a lot of files reside in a single directory. On the other hand, relational databases use efficient indexing that scales well with the size of the database. Furthermore, they also provide technical solutions, such as distributed and redundant servers, to increase security and performance.

In summary, BIPED and SESAM are data centric databases where data and query logic are co-localized. They allow for complex queries, but the central data storage renders any attempt to parallelization of queries difficult. MSD provides additional data consistency checks, but builds on a complex database schema and MMDB is a repository of single ASN.1 files. Therefore, we designed and implemented MTDB for storage and retrieval of macromolecular structures optimized toward applications in a distributed computing environment. The underlying relational database facilitates this objective, but is not used for handling complex, i.e. CPU intensive queries. Separation of storage and query handling helped to overcome the limitations of data centric databases.

3.3 MBSIS

A spatial information system for molecular biology

Finding structural motifs in the huge amount of structural information present in the Protein Data Bank (PDB) is a challenging task in bioinformatics. Currently, only static models with little flexibility can be searched for. Some implementations such as DALI (Holm and Sander, 1993), CE (Shindyalov and Bourne, 1998) and VAST (Gibrat et al., 1996) compare a 3D model to the database of known structures. Their sensitivity relies on least squares (LSQ) fitting and calculation of the root mean square deviation (RMSD) of matching regions. SPASM (Kleywegt and Jones (1997); Kleywegt (1999)) is more flexible. It searches with a 3D model and allows for flexibility with regards to the conservation of residue types and whether side chains or only C_{α} atoms are considered. Additional restraints can be placed on the sequence distances between residues in both the search model and the found motifs to further exclude unwanted hits. To predict function of proteins from low-resolution homology models, FFF (fuzzy functional form) describes active

sites using geometry, residue identity and conformation (Fetrow and Skolnick, 1998). In this algorithm, fuzzy means the satisfaction of distance restraints with mean and variance.

$$pdf(x, \mu, \sigma) = \frac{1}{\sigma\sqrt{2\pi}} e^{-\frac{(x-\mu)^2}{2\sigma^2}} \quad (3.1)$$

On the other hand, knowledge-driven approaches were demonstrated by the relational databases BIPED and SESAM (section 3.2), which use SQL to find structural motifs. In the object-oriented database system P/FDM (Gray et al., 1990) the logic programming language Prolog is used to query the database. This allows for arbitrary complex programs as queries. In the program PACADE (Satou et al., 1993) queries may also be expressed in Prolog at the level of super-secondary structure elements, such as hairpins and their assembly, to compare abstract topologies of protein structures.

As a combination of both the search algorithms and the knowledge-bases, MBSIS (molecular biology spatial information system) is a relational system of sequence and structural features of macromolecules. Comparable systems, termed GIS (geographical information system), exist in natural sciences where they help to manage thousands of objects and their interactions in three-dimensional space. MBSIS evaluates features in macromolecular structures using predefined filters implemented in MolTalk. For ease of use, such a query can be programmed through a web-interface (section 3.3.3). The resulting list of filters and the order of their application is written to a MolTalk script file, and submitted to a distributed computing environment for execution. Following, results can be inspected on-line through the same user web-interface.

Filters accept as input a set (unordered list of data tuples) of objects (e.g. residues, atoms) and compute the result set by applying their internal logic to every data tuple in the input. The computed feature value is compared to a preset normal distribution $N(\mu, \sigma)$ and only values within one standard deviation (σ) from the expected mean (μ) lead to generation of an output data tuple, labeled with the computed feature value and the log-score of the evaluated probability density function (eq. 3.1). Conveniently, relational algebra operators can be applied to the resulting sets to combine different searches at a higher level. The sum of the log-scores represents the overall score of the query.

Two qualities distinguish MBSIS from the before mentioned relational databases: First, knowledge about protein structures does not need to be precomputed and stored in a database system, and second, the expressive power of the system is not limited to the schema and the database content, but can easily incorporate new feature filters. The deductive databases P/FDM and PACADE only allow for strict filters, where an answer is either true or false, similarly to the evaluation of Prolog terms. In FFF, the relaxed restraints are used to introduce flexibility in selecting solutions, but are not used to score them. In contrast, results evaluate to a probabilistic value in MBSIS, allowing for a finer grade of decision making. This allows for querying with weak restraints and the evaluated values of mean and standard deviation can be re-applied in subsequent queries to confine the search. Moreover, it seems to be difficult to parallelize database centric computations, whereas the distributed, independent queries in MBSIS practically scale with the number of available processors. Changes to the underlying algorithm invalidate precomputed data, thus they would need to be re-computed, and the current size of the PDB would

require reasonable storage capacities. However, today's computing power allows us to circumvent this obstacle and compute knowledge on the fly.

3.3.1 Implemented relational operators

The system performs relational algebra on sets. The union operator ($A+B$) combines all data tuples from two sets (complexity $O(n)$), which have the same number of columns with the same names. The difference ($A-B$) operator selects only data tuples in A , which are not in B ($(a \in A) \wedge (a \notin B)$, complexity $O(n^2)$). Again, the two sets must have the same number of columns with the same headings. The projection ($\Pi_{T_1, \dots} (A)$) operator selects only named columns (T_1, \dots) from a set. The most powerful operator is the natural join between two sets: $A \bowtie B = \sigma_{\Theta}(A \times B)$, where $A \times B$ is the cartesian product between the two input sets and σ_{Θ} is the selection of data tuples based on commonly named columns. This operator allows to add columns of another set based on comparison of data between matching columns. For example, set A contains columns $\{T_{A,0}, T_{A,1}, T_X\}$ and set B contains columns $\{T_{B,0}, T_X, T_{B,1}\}$. The algorithm first determines the matching columns, in this case $\{T_X\}$, which will be used in the pairwise comparison. The combined output set thus contains columns $\{T_{A,0}, T_{A,1}, T_X, T_{B,0}, T_{B,1}\}$. For every data tuple $a \in A$ and $b \in B$, this operator compares the values of the commonly named column T_X and in case of equality creates a new data tuple from a and b in the output set. The complexity of this operator is $O(n^2)$.

3.3.2 Implemented feature filters

Currently, MBSIS implements 26 feature filters (Tables 3.1 and 3.2), which can be grouped into two classes. The filters in the first group produce new data tuples from the input set and, if required, from a combination with the second input set. The second group provides filters, which act on an input set and select all data tuples that pass the filter criteria with parameters μ and σ .

Table 3.1: Feature filters implemented in MBSIS.

name	input 1	input 2	output	complexity
Structure Enumeration	strxsrc, strxid		MTStructure	n
Chain Enumeration	MTStructure		MTChain	n
Chain Selection	MTStructure	chainid	MTChain	n
Chain Contact Finder	MTChain	MAtom, MResidue	MAtom, MResidue, MAtom2, MResidue2	$n * \log(n)$
Residue Enumeration	MTChain		MResidue	n
Residue Range Enumeration	MTChain	from, to	MResidue	n
Residue Distance Product	MResidue	MResidue	MResidue, MResidue2	n^2
Residue Secondary Structure Annotation	MTChain		MResidue, sse	n
Atom Enumeration	MResidue		MAtom, MResidue	n
Atom Named Enumeration	MResidue	atomname	MAtom, MResidue	n
Disulphide	MResidue	MResidue	MAtom, MResidue, MAtom2, MResidue2	n^2
Salt-bridge	MResidue	MResidue	MAtom, MResidue, MAtom2, MResidue2	n^2
HBond Selection	MAtom, MResidue	MAtom, MResidue	MAtom, MResidue, MAtom2, MResidue2	n^2
HBond Finder	MResidue	MResidue	MAtom, MResidue, MAtom2, MResidue2	n^2

Table 3.2: Feature filters (selectors) implemented in MBSIS.

name	input 1	input 2	output	complexity
Residue Distance Selection	MTResidue, MTResidue2		MTResidue, MTResidue2	n
Atom Selection	MAtom, MTResidue	atomname	MAtom, MTResidue	n
Residue2Chain	MTResidue		MTResidue, MTChain	n
Chain2Strx	MTChain		MTChain, MTStructure	n
Annotate Structure	MTStructure		MTStructure, ...	n
Annotate Chain	MTChain		MTChain, ...	n
Annotate Residue	MTResidue		MTResidue, ...	n
Annotate Atom	MAtom, MTResidue		MAtom, MTResidue, ...	n
Atom Contacts	MAtom, MTResidue	MAtom, MTResidue	MAtom, MTResidue, MAtom2, MTResidue2	n^2
Residue Phi Selection	MTResidue		MTResidue	n
Residue Psi Selection	MTResidue		MTResidue	n
Residue Omega Selection	MTResidue		MTResidue	n

3.3.3 Query designer

A query in MBSIS is an extended MolTalk script, which can be executed by the MolTalk interpreter. A web-based query designer was implemented that helps to design more complex queries (Figure 3.2). At its center, the query designer maintains a list of filters to be applied to input sets. The computed output hits are available to subsequent filters. From the same user interface, the query can be launched on a single structural chain indicated by its code, or submitted to the distributed computing system to be executed on each structure in predefined selections. Such selections were defined for structures from the PDB with resolution better than 1 Å or chains containing more than 20 amino acids. These selections can be updated using SQL queries from PDBChainSaw (section 5.3) once the underlying PDB database is mirrored.

home program execute analyse

MBSISinterchainssbonds definitions file ready. last modif: 2005/5/13 16:42:13

	input	filter	parameter	output	μ/σ	Z thr.
	0 [type, strxid]	1 Structure Enumeration [type, strxid]->[strx]		2 allstrx [strx]		
	2 allstrx [strx]	1 Chain Enumeration [strx]->[chain]		3 allchains [chain]		
	3 allchains [chain]	1 Chain Selection [chain]->[chain]	1 [chainid]	4 selectedchain [chain]		
+X / ▲ ▼	3 allchains [chain]	1 Residue Enumeration [chain]->[residue]		101 resall [residue]		
+X / ▲ ▼	4 selectedchain [chain]	1 Residue Enumeration [chain]->[residue]		103 resselected [residue]		
+X / ▲ ▼	101 resall [residue]	1 Difference of Relations	103 resselected	104 resother		
+X / ▲ ▼	103 resselected [residue]	Disulphide [residue]->[residue, atom, residue2, atom2]	104 resother	102 ssbonds [residue, atom, residue2, atom2]	2.00/0.10	1
+X / ▲ ▼	102 ssbonds [residue, atom, residue2, atom2]	Residue2Chain [residue]->[residue, chain]		107 [residue, chain]		
+X / ▲ ▼	107 [residue, chain]	1 Annotate Chain [chain]->[chain, name, ...]		108 [chain, name, ...]		
+X / ▲ ▼	108 [chain, name, ...]	1 Join Relations	107	113		
+X / ▲ ▼		1 New Relation		110 [select]		
+X / ▲ ▼	113	1 Project Relation	110 [select]	109		
+X / ▲ ▼		1 New Relation		115 [from, to]		
+X / ▲ ▼	109	1 Rename Attributes	115 [from, to]	116 [from, to]		
+X / ▲ ▼		1 New Relation		116 [from, to]		
+X / ▲ ▼	102 ssbonds [residue, atom, residue2, atom2]	1 Rename Attributes	116 [from, to]			
+X / ▲ ▼	102 ssbonds [residue, atom, residue2, atom2]	Residue2Chain [residue]->[residue, chain]		118 [residue, chain]		
+X / ▲ ▼	118 [residue, chain]	1 Annotate Chain [chain]->[chain, name, ...]		119 [chain, name, ...]		
+X / ▲ ▼	118 [residue, chain]	1 Join Relations	119	112		
+X / ▲ ▼	112	1 Project Relation	110 [select]	121		
+X / ▲ ▼		1 New Relation		122 [from, to]		

Figure 3.2: MBSIS query designer.

The query consists of a number of filters, which are applied in a specific order of evaluation. Every line represents the generation of an output set from the application of a filter to its input set(s). On the left are controls to add, remove, and edit filters, or change their order in the query. This example shows the query to search for disulphide bonds (distance criterion: 2 ± 0.1 Å) between different chains in a structure.

3.4 Summary and Outlook

MTDB has been derived from MolTalk by reimplementing object factories and structural classes. Loading of structural data can be accomplished in reasonable time and the space requirements are modest. Properties which allowed to load the complete PDB into a relational database. Mapping of the structural object hierarchy in MolTalk to a relational database enables users to store structural models within the iMolTalk web-server (chapter 4). Furthermore, these models are then available to the application PDBalert (section 5.4) as search models. Further normalization needs to be implemented to minimize the size of the database at the expense of compromising simplicity and probably speed of queries.

MBSIS is a versatile system to express probabilistic models of structure-to-sequence relations. It computes sets of structural objects that satisfy predefined filters (i.e. restraints). These sets can then be combined at a higher level using relational algebra. In contrast to

previous systems, which are limited to single queries, MBSIS has the advantage of computing queries in parallel. The web-interface supports both the programming of complex queries as well as the execution on a distributed computing environment and analysis of the results. Currently, several MBSIS queries are implemented in the iMolTalk web-server. Moreover, the definition of the central β -sheet in AAA+ proteins (section 6.2.1) is based on MBSIS queries.

Chapter 4

iMolTalk

The protein structure analysis web-server

4.1 Introduction

Online-services are frequently used to build 3D models and scientists integrate results from analyzing such models in the planning of their experiments (Guex and Peitsch, 1997; Guex et al., 1999). However, models of protein structures are difficult to analyze and interpret for non-experts. Besides lack of supporting computer soft- and hardware, experience in structural biology is still restricted to experts. Nevertheless, there is a limited number of structural analyses that users request most frequently. Some of these analyses were standardized and implemented in our interactive web-server, iMolTalk, whose design builds on five pillars: applicability, streamlined analyses, object-orientation, navigation, memorization (Diemand and Scheib, 2004a).

Applicability The application is implemented as a web-service. Thus, no local installation of software is required and hardware requirements are kept minimal for the user. Computation is delegated to the server, which is accessed through a web-interface. Thus, database internals and program installations are completely hidden from the user.

Streamlined The user is guided through defined analyses. At every step of the analysis the user communicates with the server and provides necessary input parameters. Then, the server calls the underlying algorithm and presents the computed result. Through text messages the user is informed of necessary actions.

Object-oriented Structural objects, such as structure, chain or residue, are identified in computed results and direct actions may be performed on them via pop-up menus; thus, they may be submitted to further analyses.

Navigation The user may navigate to a previous step in the analysis to change parameters and rerun the calculation.

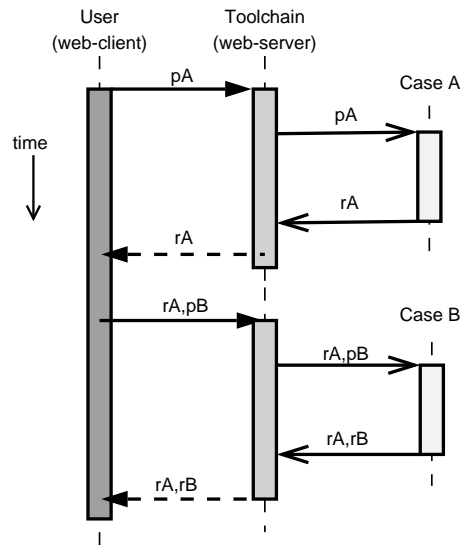


Figure 4.1: Communication schema.

The user communicates with the selected toolchain through sequential requests sent to the implemented cases on the server. The parameters (p) provided by the user are carried on throughout the analysis (r).

Memorization Structural objects and analyses can be managed on a graphical clipboard or organized using the report generator. Facilitated by the object-oriented nature of the system, this allows users to store results and launch corresponding actions on structural objects later.

4.2 Implementation

4.2.1 Toolchains and cases

The user communicates with the web-server through his web-client or Internet browser. On the web-server, analyses are abstracted as so-called *toolchains*: defined sequences of communication between the client and the server for input parameter gathering and presentation of computed results. At every step of a selected analysis (toolchain), a number of input parameters must be submitted to complete the computation and to advance to the next step. The server contains the logic of the analysis; however, it only knows about the current state of the client because of the disconnected communication between them. Therefore, all state variables are carried on in the analysis and sent back and forth (Figure 4.1).

Single steps in a toolchain are termed *cases* and are implemented on the server as program code to manage and verify input as well as to dynamically render the output. Along with a case are defined the parameters that are requested from the user and help messages. Cases are related to a specific position in one or several toolchains and every user

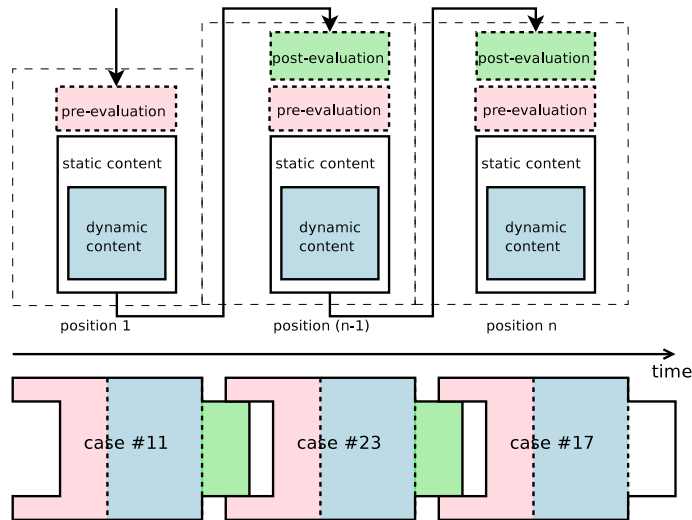


Figure 4.2: Toolchains made up from cases.

In this example the user runs an analysis consisting of three steps. For each user request, the server identifies the corresponding case and evaluates it (upper subgraph). It first executes its pre-evaluation code (pink) followed by dynamic content rendering (blue). The post-evaluation code (green) is evaluated in the following transaction. Static content is maintained in template files.

request must indicate toolchain and position for the server to determine the current case and its program code, which is executed in three stages (Figure 4.2). First, unless it is the first case in a toolchain, the server locates and executes the post-evaluation code of the preceding case. Second, the pre-evaluation code of the current case is executed and evaluated. Third, unless an error was detected, the dynamic content code is executed to render the dynamic output and presented to the user. Program code was written in the logic programming language Prolog (Clocksin and Mellish, 1994; Covington et al., 1997) and dynamically loaded on request as precompiled bytecode to minimize startup time and memory consumption. Special toolchains help managing, writing and compiling the code on-line (section 4.2.2).

At any step in an analysis, the server is aware of the parameters of preceding cases. This allows the user to navigate to the previous input step for changing parameters and then rerun the analysis.

A single case can be evaluated when all its necessary input parameters are available. Thus, the output of the evaluation of a case is determined by these parameters. After a case has verified the input parameters by its pre-evaluation code, the result is then computed and its output rendered by the dynamic content code. Therefore, we can think of the case as a function, which relates the set of input parameters to the computed output of a case. This deterministic mapping can be expressed as an address for the output, encoding the input. In the world wide web (WWW) resources are addressed using URLs (uniform resource locator). Such an address consists of three parts. First, the protocol of the negotiated communication. Second, the name or numeric address of the Internet

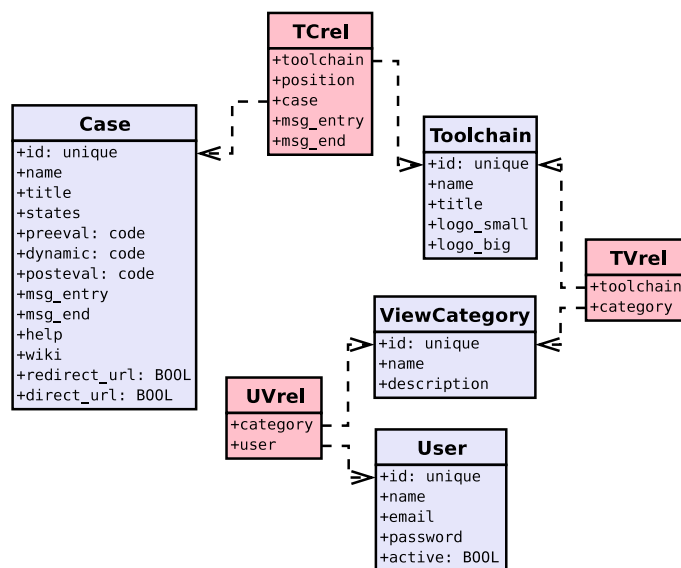


Figure 4.3: Configuration of the iMolTalk server.

Toolchains, cases, view categories and users are managed in a relational database. The relation TCrel assigns a case to a specific position in a toolchain. Users may access view categories defined by the relation UVrel and through TVrel also toolchains.


server. Third, a unique identifier, which the server can interpret to find the requested resource. In iMolTalk, the protocol is http (hypertext transport protocol) and the identifier is constructed as a file path from the name of the toolchain followed by the position of the case and all the parameters in order of their declaration. As an example, the second case in the toolchain *pdb_information* displays derived information from a structure, in this case 1crn from the PDB. The constructed URL¹ contains all the necessary information for the server to execute the query. The rewriting of analyses as URLs is the basis for efficient caching of the evaluated output.

4.2.2 Toolchain editor

The iMolTalk server is configured using a relational database (Figure 4.3). It can be managed online through special toolchains (Figure 4.4). The order of cases in a toolchain can be defined in the *toolchaineditor*. The *caseeditor* allows editing of the programming code of the three stages pre-evaluation, dynamic content generation, and post-evaluation. After saving the code and its compilation into fast loading bytecode, the changed program of a case immediately becomes effective on the server. The *vieweditor* maintains groups of toolchains, which may be related to users. Information about registered users can be managed in the *useditor*.

¹http://i.moltalk.org/pdb_information/2/PDB/1CRN

Toolchain Editor



Toolchain:

Start toolchain: **interface**

Name:

Title:

Logo small:

Logo big:

active:

remove	position	case	states	message entry	message end
x	1	001 Select Structure by Identifier	strxsrc, strxid	<input type="button" value="set"/>	<input type="button" value="set"/>
x	2	025 Select First and Second Chain	chaincode1, chaincode2	<input type="button" value="set"/>	<input type="button" value="set"/>
x	3	026 Protein-Protein Interface	dtreshold	<input type="button" value="set"/>	<input type="button" value="set"/>

message:

append case:

PDBAlert
 Protopolis
 analyses
 basic
 management

Figure 4.4: Toolchain editor.

In this example, the toolchain "interface" for interface detection is shown in the toolchain editor. At the top, title and logo image location can be entered. In the middle, cases are arranged in the toolchain. At the bottom, the toolchain can be related to view categories.

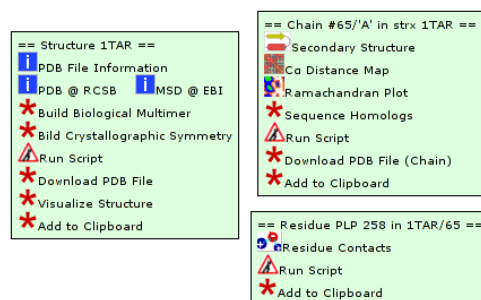


Figure 4.5: Object specific menus.

Objects of type structure, chain or residue have their corresponding action menus, which allow the direct access to specific functions.

4.2.3 Structural objects and databases

The server recognizes structural objects of type structure, chain and residue. Actions, which require such objects as input, are directly available from object specific pop-up menus (Figure 4.5). Additionally, the special objects “residue selection” and “transformation matrix” can be managed on the clipboard.

iMolTalk is a front-end to the structure database PDB (Berman et al., 2000) and homology models from ModBase (Sanchez and Sali, 1999). Structural models are accessible in both databases via their identifiers. Access to a ModBase model triggers its download from the original site² to the server in the background. Furthermore, iMolTalk integrates access to uploaded files and local databases such as the database of AAA+ models and reconstructed ring structures (chapter 6).

Authenticated users also have the possibility to persistently store models in MTDB (section 3.2). At a later time, these models can be checked out from the database and subjected to further analyses.

4.2.4 Clipboard and report generator

A first connection to the server initiates the creation of a personal clipboard associated with the user’s browser through *cookies*. Structural objects may then be placed on the interactive clipboard for later reuse. Object specific menus allow users to directly launch actions on such objects. Furthermore, drag and drop actions between object representations of chains allow to directly initiate the computation of their structural alignment or interface detection. Rewriting the call to an analysis as an URL is the basis for memorizing analyses on the clipboard and in the report generator.

Analyses stored on the clipboard can be arranged in the report generator. Its dynamically rendered web-page can be downloaded as a file to the user’s computer for later reuse or can be shared with other users.

²<http://modbase.compbio.ucsf.edu/>

Authenticated users have their last used clipboard reinstalled.

4.2.5 Visualization

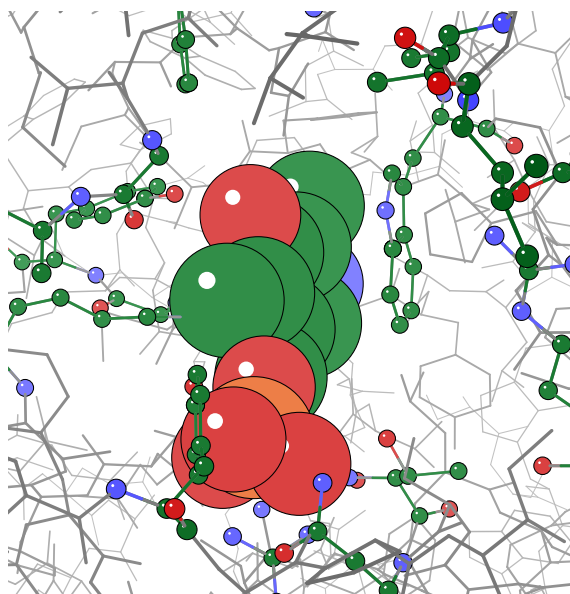


Figure 4.6: Visualization of a residue selection.

Residues in the annotated site ACT1 (1tar, aspartate aminotransferase) are highlighted in ball-and-stick representation and displayed in the kinemage viewer KiNG. The covalently bound co-factor pyridoxal-5'-phosphate to lysine 258 (from left) is nicely stacking against tryptophan 140 (on the right).

All structures and generated models can be downloaded to the user's computer for further manipulation and visualization in interactive molecule viewers. However, this procedure is time consuming and requires properly installed software as well as training. To provide visual overview about a structure, or to highlight certain local aspects, the server generates a kinemage (Richardson and Richardson, 1992) from PDB formatted files. After download, they are presented in a Java-based viewer (KiNG) within the browser's window. Figure 4.6 shows an example of a visualization of an annotated SITE record in a PDB file. The structure is read into memory and the SITE record is interpreted and translated into a *MTSelection* object. A MolTalk script sets up the desired object properties (color and rendering style) and renders the structure to a kinemage file, which is then interpreted and displayed in the Java viewer.

4.3 Structural analyses

The current version 3.1 implements 75 toolchains and 100 cases. Some of them provide administrative tools (e.g. *toolchaineditor*) or user interfaces to the integrated applications

PDBalert (section 5.4) and Protopolis (section 5.5). The other toolchains are visible to the user from analyses, modeling, and search menus. The first menu groups all toolchains, which provide structural analyses (e.g. general information derivation or contact finder). The modeling menu lists toolchains that are used to modify structures or prepare alignments between structure and sequence. And the search category mainly provides a front-end to the database PDBChainSaw (section 5.3) and the application Protopolis (section 5.5).

4.3.1 Residue contact finder

Contacts made by a selected residue, either amino or nucleic acid or hetero group, are searched in the structure. The hashing of coordinates speeds up the search (section 2.2.4). Pairwise atomic distances below a threshold (default 3.4 Å) are further analyzed and annotated for their type of bonding. H-bonds are detected from heavy atoms using the parametrization shown in Table 4.1 (Stickle et al., 1992). Additionally, this includes geometrical restraints to describe the maximal distance between the electronegative heavy atoms, angles at the acceptor and donor atoms as well as the planarity of H-bonds involving at least one sp^2 group (Baker and Hubbard, 1984). Salt-bridges are inferred between charged side-chains in proteins, the N-terminal amino group and the C-terminal carboxy group of the backbone. Furthermore, the phosphate backbone in nucleic acids is also recognized as being negatively charged. Inference of partial charges is not accomplished, but charges in the PDB file are taken into account. The latter is important in describing hetero groups.

4.3.2 Protein-protein interface description

Structural interfaces between two chains are identified and the pairwise contacts below a distance cutoff (default 3.4 Å) are then analyzed using the same bonding type inference as described in the previous section. The selected interface residues are available for visualization or mapping onto the sequence.

4.3.3 Distance map

Internal contacts are mostly preserved within folds and their visualization using distance maps (Phillips, 1970; Richardson, 1981) may help to relate two structures even in cases where sequence homology is very low. The example in figure 4.7 highlights the detection of structural domains, which predominantly exhibit more contacts within each domain (Rossmann and Liljas, 1974). In the large structure (1abrB, abrin-A) we observe a tandem repeat of distinct structural domains, which show many internal contacts, but only a few between the domains. The smaller structure (1hcd, hisactophilin) can be superimposed onto either domain, but a homology based on sequence comparison is hard to detect (Habazettl et al., 1992). Nevertheless, proteins of the β -trefoil fold (Murzin et al., 1992) may have evolved from a common ancestor (Ponting and Russell, 2000).

Table 4.1: Parametrization of H-bond donors and acceptors.

On the left are shown the H-bond participating atoms of peptides and on the right those of nucleic acids. Serine, threonine and tyrosine, each contain terminal groups, which may act as donors or acceptors. The protonation state of histidine was assumed to be neutral.

<i>residue</i>	<i>atom</i>	<i>donor/acceptor</i>	<i>residue</i>	<i>atom</i>	<i>donor/acceptor</i>
all	O	A	A,U,T,C,G	O1P	A
except Pro	N	D		O2P	A
Tyr	OH	D/A		O6	A
Ser	OG	D/A		O4	A
Thr	OG1	D/A		O2	A
Asn	ND2	D		O5*	A
	OD1	A		O4*	A
Gln	NE2	D		O3*	A
	OE1	A		N4	D
Asp	OD1	A		N6	D
	OD2	A		N7	D
Glu	OE1	A		N2	D
	OE2	A	A	N1	A
Cys	SG	A	G	N1	D
Met	SD	A	U,T	N3	D
Trp	NE1	D	A,C,G	N3	A
Arg	NH1	D			
	NH2	D			
	NE	D			
His	ND1	D			
	NE2	A			
Lys	NZ	D			

4.3.4 Structural alignment and differential distance map

The derivation of structural alignments follows the procedure of computing an initial sequence alignment and determining the transformation using least-squares fitting (section 2.2.3). Alternatively, the integrated program MAMMOTH (Ortiz et al., 2002) computes a superimposition independent of the sequences of the two structures to compare. The extracted transformation matrix of either method is then used to superimpose the two protein structures and to derive their pairwise structural alignment based on dynamic programming (the scoring function includes $C\alpha - C\alpha$ distances). From this structural alignment a differential distance map (Nishikawa and Ooi, 1974) is computed to highlight regions with local movement relative to the rest of the protein (Figure 4.8).

4.3.5 Miscellaneous toolchains and integrated third-party analyses

General information extraction A MolTalk script loads a structure file into memory and outputs the hierarchy of structure, models and chains. The output first lists the structural information, which is parsed from the PDB file. Then, for each model in the structure, all chains are listed with their sequences and the count of atom

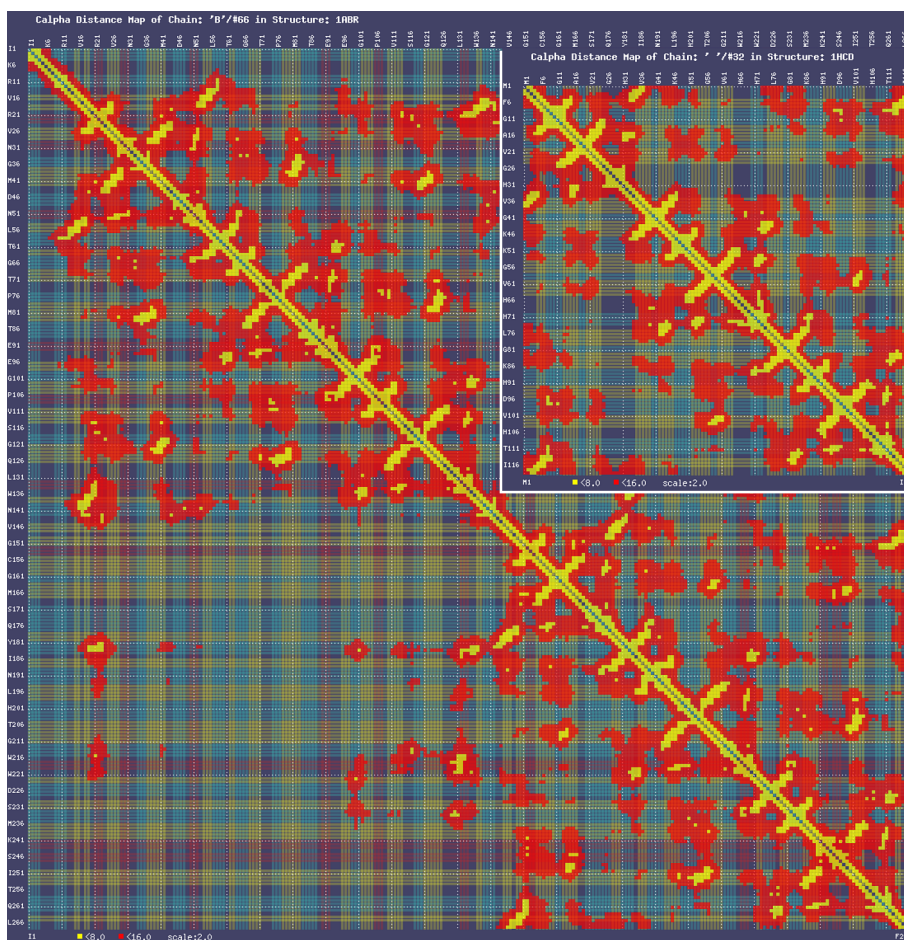


Figure 4.7: Distance maps of β -trefoil proteins and their comparison.

Structural domains can be identified by comparing their internal contacts. As an example, in the β -trefoil fold, 1abrB (background graph) and 1hcd (small overlaid graph, upper right corner) show a similar pattern of anti-parallel β -strands, notably the first and last are in contact. These structures can be superimposed well (RMSD 2.9 Å over 112 aligned residues), though the derived structural alignment only shows a sequence identity of 7%.

groups, either amino or nucleic acids, hetero groups or solvent. If the file contains SITE annotations, links are provided for their immediate visualization.

Search in PDBChainSaw The database PDBChainSaw (section 5.3) may be queried through its user-interface in the iMolTalk server. The form allows for entering query strings in the fields and to select their sorting order. Results are presented in a tabular view. Structure identifiers and chain codes are recognized and provide links to the object specific action menus (Figure 4.5) for direct access.

Ramachandran plot The Φ/Ψ -dihedral angle plot (Ramachandran et al., 1963) follows the IUPAC convention with angles ranging from -180° to $+180^\circ$ and centered on

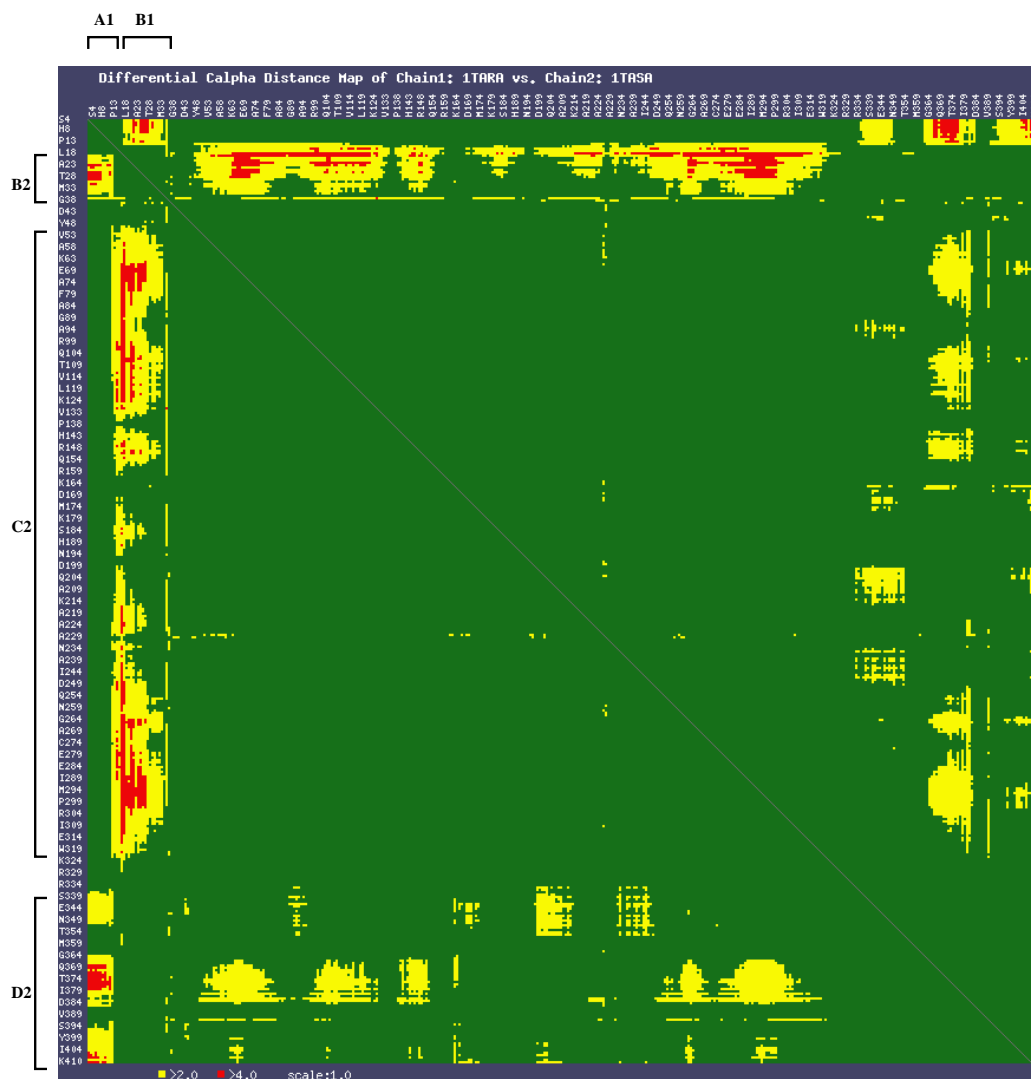


Figure 4.8: Differential distance map reveals domain motion in aminotransferases. The underlying structural alignment between the two sequence identical aminotransferases (1tar and 1tas) shows an RMSD of 1.5 Å over 400 aligned residues. The coloring indicates motion with values below half the threshold (2 Å, yellow) or above (4 Å, red). Several blocks can be distinguished, which move relative to each other: A1 against B2 and D2, B1 against C2.

the origin (Kendrew, 1970). The plot is divided into four regions: core, allowed, generous and disallowed (Morris et al., 1992).

Secondary structure assignment The program STRIDE (Frishman and Argos, 1995) is used to consistently annotate structures with the derived secondary structure assignment. The sequence is colored according to the type of secondary structure element (helix, strand and turn).

Sequence to structure alignment The alignment of the SEQRES sequence from the PDB file onto a structure quickly identifies the extent of resolved residues. Furthermore, this toolchain can also align any homologous sequences, either globally or locally. Such an alignment may serve as an initial alignment for homology modeling.

Side-chain remodeling The program SCWRL (Canutescu et al., 2003) can be used to remodel side-chain orientations in protein models. Furthermore, the program can also be used to introduce point mutations and remodel the environment of the changed amino acid.

Channel analysis Pores in protein structures can be analyzed using the program HOLE (Smart et al., 1993). The output lists the residues that form the pore and their corresponding surface area. Graphs of the pore radius vs. pore coordinates indicate the shape of the pore. The pore, abstracted using pseudo-atoms, can also be visualized.

Quality checks WHATCHECK (Hooft et al., 1996) is a versatile utility to thoroughly check a protein structure. Among the integrated analyses are computation of the Ramachandran plot as well as verification of atom nomenclature and stereo-chemical properties of residues. Identified objects in the textual report are translated into active links, which provide object-specific menus. Moreover, the web-server integrates the statistical potential Anolea (Melo and Feytmans, 1997) to detect residues with badly placed side-chains. These are turned into selections, which may be reused in subsequent analyses (i.e. side-chain remodeling).

4.4 Other web resources

PROCHECK (Laskowski et al., 1993) is available as a standalone application and as a web-service³. This is probably the most cited structural analysis tool. The program checks the stereo-chemical quality of protein structures and returns graphical plots in Postscript. Most importantly, the program computes the Ramachandran plot to help detecting non-canonical conformations of the backbone. But, this software is restricted to only this type of analysis.

The application Ligand Explorer on the RCSB site⁴ was developed using the Java-based toolkit MBT (section 2.3.6). It is restricted to detect and display contacts and distances between a ligand and a protein.

WHAT IF is a complete software package for interactive structure modeling (Vriend, 1990). Additionally, methods of the software have been wrapped within a web-server⁵. The user may indicate the code of a PDB structure or upload a model file as input to most of the wrapped methods. Then, the web-server starts WHAT IF in the background and returns its textual output. Though, this server implements a number of analyses, they are not interconnected. In contrast to iMolTalk, this means that objects are not recognized on the server, and users cannot roll back an analysis and rerun it.

4.5 Summary and Outlook

The current implementation of iMolTalk provides three entry points: structural analyses, modeling support and database searches. This covers some of the most recurrent analyses. Moreover, it serves as a front-end to PDB, ModBase and other structure databases. The database PDBChainSaw, which contains meta information derived from PDB files, is fully searchable through the iMolTalk web-server. The server performs actions on structural objects, such as structure, chain and residue, and also residue selection and transformation matrices. This object-based architecture allows one to store the results (objects) of predefined analyses on a graphical clipboard. Later, they may be submitted to further analyses. Furthermore, the iMolTalk server is used as a framework for storing and publishing structural models, either from experimental data or homology modeling. Its novel integration of editing and visualization capabilities, as well as its database support are unique. The applications Protopolis (section 5.5) and PDBAlert (section 5.4) are tightly integrated into the iMolTalk server and underline the extensibility of the server's functionality. Addition of more toolchains geared toward molecular and homology modeling is envisaged.

³<http://www.biochem.ucl.ac.uk/~roman/procheck/procheck.html>

⁴<http://www.rcsb.org/>

⁵<http://swift.cmbi.kun.nl/WIWWWI>

Template Selection in Homology Modeling

Chapter 5

Template selection

5.1 Introduction

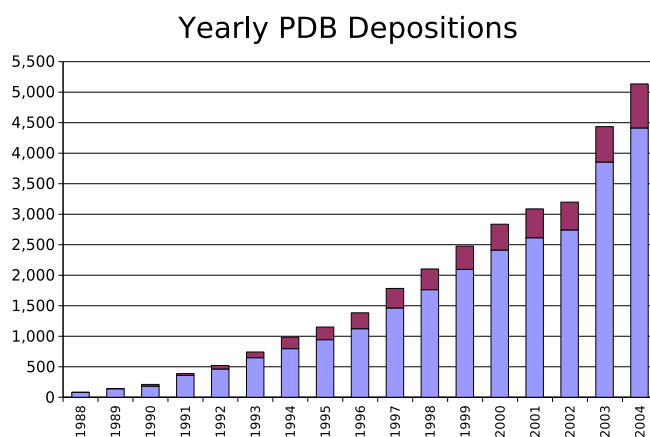


Figure 5.1: Yearly depositions to the PDB.

Entries have been split by the main structure determination techniques X-ray crystallography (blue) and NMR spectroscopy (red).

An increasing number of structures are deposited in the Protein Data Bank (PDB) every year (Figure 5.1). This is mainly due to world-wide structural genomics initiatives, which aim at finding every protein fold that exists in nature. In 2001, it was estimated that 16,000 new structures would then allow modeling the majority of all proteins (Vitkup et al., 2001). At the end of 2000, the PDB counted ~15,000 structures. Five years later, its content has more than doubled and increased to ~35,000. From a modeling perspective, it is increasingly difficult to navigate in such a vast space. Therefore, we developed PDBChainSaw (section 5.3) to extract information from PDB files and reconstruct their sequences from coordinates. Every week, approximately 100 structures are released by the PDB. To be alerted, when a new template of interest is among these newly deposited structures, PDBAlert (section 5.4) maintains a database of user-supplied sequences and

searches them against the weekly released PDB structures to find new hits. On the basis of PDBChainSaw, sequence redundancy in PDB, at the level of 50% sequence identity, can be estimated to more than eightfold. To analyze groups of homologous structures and evaluate their suitability in homology modeling tasks, I developed Protopolis (section 5.5), which compares and clusters structures in these groups and, through the overlay of annotation, helps to generate hypotheses that explain the division into subgroups. First, we present and discuss a motivating example of a remodeling of aminotransferases, which exist in two distinct conformations (open and closed). Then, the results from analyzing 3,514 trees of homologous structural chains are discussed. Interestingly, two third of these trees reveal some structural diversity and in half of the cases annotation can help to discriminate between the first two groups. At the end, Protopolis is used to discriminate between conformations of the AAA+ protein p97. Structures in distinct conformational states form subgroups and the most discriminative annotations between them are resolution and the observed spacegroup from crystallization.

5.2 Remodeling of aspartate aminotransferases

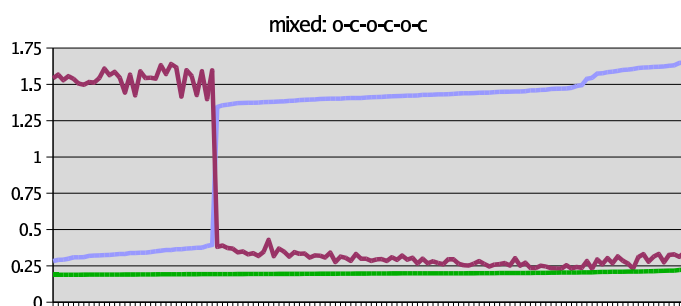


Figure 5.2: Remodeling of aspartate aminotransferases in open and closed form.

The modeling was based on six sequence identical templates, three in open and three in closed form. The Y-axis indicates the calculated RMSD value of the comparison against open (7aat, red line) and closed form (1ivr, blue line). The 125 homology models (X-axis) were sorted for their RMSD values against the open form. The green line indicates the level of diversity if only closed form templates were used in the modeling.

As an example re-modeling of aspartate aminotransferases (AAT) is presented. The active form of this protein is a homodimer. Each monomer is around 400 amino acids long and exists in two distinct conformations: open (apo) and closed (holo, substrate bound) form (Hohenester and Jansonius, 1994; McPhalen et al., 1992). This protein is part of a family of enzymes, which use the coenzyme pyridoxal 5'-phosphate (PLP, a form of vitamin B_6) to catalyze the transfer of amino groups and play a major role in amino acid metabolism (Jansonius, 1998). The coenzyme is covalently linked to the active site lysine. Upon binding of the amino acid substrate, it is released from the lysine and forms an external aldimine (Schiff base, $R-N=C<$) with the substrate (Figure 4.6). Two isoforms of this enzyme exist, one cytosolic and one in mitochondria. In clinic, test for AAT activity

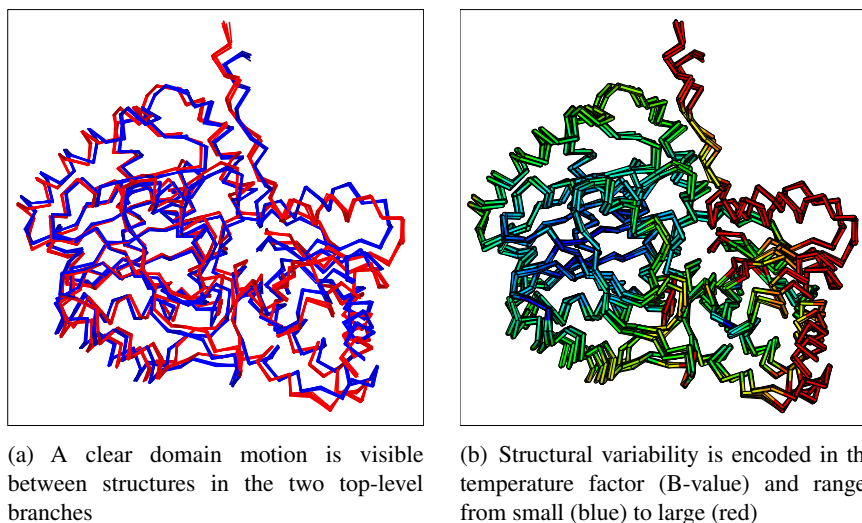


Figure 5.3: Domain motion in aspartate aminotransferases.

in blood samples indicates diseases or physical damage of liver.

Here, we clustered the 23 available structural chains of the mitochondrial isoform. In each group, from the resulting tree, three representative structures were chosen: open (1oxo, 1tar, 7aat) and closed form (1ivr, 1tas, 1tat). Superimposition of the two groups revealed discrete domain motion (Figure 5.3). All of these structures are identical at the level of sequence, so aligning them was trivial. The alignment was given to MODELLER (Sali and Blundell, 1993) and 125 models were computed. The high number of computed models accounts for the probabilistic modeling, that is, every model is computed starting from a randomly chosen start model. Then, the models were compared back to the experimentally determined structures of the open (7aat) and closed form (1ivr). The calculated RMSD values are shown in Figure 5.2. Clearly, the mixed modeling, using both open and closed forms, has a preference towards the open form (93:32 or 74% : 26%). Even worse, the diversity of the models is larger than if only one conformation was used. Modeling using only closed form templates resulted in RMSD values $0.2 \pm 0.01\text{\AA}$ against closed (1ivr) and $1.58 \pm 0.01\text{\AA}$ against open (7aat) form structures. The 32 closed form models from the mixed modeling experiment had a RMSD of $0.34 \pm 0.03\text{\AA}$ against 1ivr and $1.54 \pm 0.06\text{\AA}$ against 7aat. The other 93 open form models showed RMSD values of $1.46 \pm 0.09\text{\AA}$ and $0.3 \pm 0.04\text{\AA}$ against closed (1ivr) and open (7aat) form, respectively.

The large RMSD values correspond to a mix of both conformations in the resulting models, a modeling artifact that is without biological meaning. This is just one example where evaluation of homologous structures leads to identification of conformational states and helps to select appropriate templates for a specific modeling task. As the PDB continues to grow and its redundancy in terms of sequence homology increases, modeling tasks will hit more and more populated families of homologous structures. To address this issue, I developed Protopolis (section 5.5), which is tailored towards automation of such analyses, as has been demonstrated for the modeling of aminotransferases.

5.3 PDBChainSaw

Derivation of structural features

Extracting and deriving knowledge from PDB files is a non-trivial procedure and a standardized procedure to do so does not exist to date. Additional needs, such as integration of information derived from someone’s own homology models, called for a flexible solution. I therefore implemented PDBChainSaw in *MolTalk* to parse PDB files and extract a defined set of features to be stored in a relational database (Figure 5.4). When reconstructing the amino acid sequence from coordinates three problems arose. First, structures may contain chemically modified residues (post-translational), which would need to be back-translated into the standard amino acid code based on the PDB file annotation. Second, parts of the structure may not be resolved as the experimental data did not justify modeling of side-chains and backbone atoms. Information about such missing residues should also be included in the coordinate derived sequence. And third, the information in the SEQRES records, which indicates the sequence of the underlying construct (gene) must be validated against the present coordinates. In most cases, this sequence information is missing for model structures and is therefore unavailable.

PDBChainSaw reconstructs the amino acid sequence of each chain in a protein structure from its coordinates and also takes into account the information given in the records for modified residues (MODRES records). As an example, it determines a threonine at position 160 in 1qmzA (cyclin-dependent kinase 2), which is post-translationally phosphorylated and thus named “TPO” instead of “THR”.

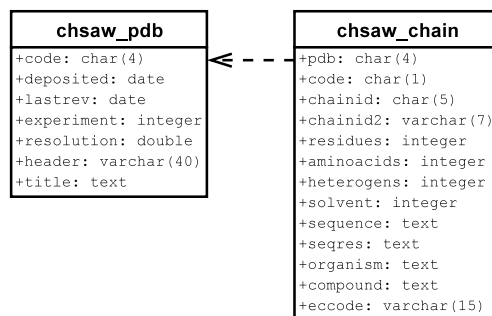


Figure 5.4: PDBChainSaw database schema.

Numerous structure files lack residues, which were not resolved during structure determination, but are part of the native protein as can be verified from the SEQRES record, if present. Such amino acids are responsible for breaks in the main-chain of a structure and may introduce artifacts in the sequence that do not represent “real” insertions or deletions due to biological variation. Because such missing residues penalize Blast scores as well as the expectation value if aligned against “native” sequences, PDBChainSaw introduces neutral residues “X” in the structure-derived sequences, one for each missing amino

```

>sp|P36542|ATPG_HUMAN (ATP5C1)ATP synthase gamma chain,
mitochondrial precursor (EC 3.6.3.14).[Homo sapiens]
Score = 103 bits (256), Expect = 2e-22
Identities = 55/64 (85%), Positives = 58/64 (90%)
Query: 31 LCGAIHSSVAKQTTSEGSARMTAMDNASKNASEMIDKLTTLFNRTROAVITKELIEIISGAAAL 94
L I+ S+ + TTSEGSARMTAMDNASKNASEMIDKLTTLFNRTROAVITKELIEIISGAAAL
Sbjct: 234 LANIYYSLKESSTTSEGSARMTAMDNASKNASEMIDKLTTLFNRTROAVITKELIEIISGAAAL 297
Score = 62.0 bits (149), Expect = 4e-10
Identities = 43/92 (46%), Positives = 44/92 (47%), Gaps = 46/92 (50%)
Query: 1 ATLKDITRRLKSIKNIQIKTKSMKIVAAK----- 30
ATLKDITRRLKSIKNIQIKTKSMKIVAAK
Sbjct: 26 ATLKDITRRLKSIKNIQIKTKSMKIVAAKYARAERELKPARIVGLGSLALYEKADIKGP 85
Query: 31 -----LCGAIHSSVAKQTTSE 46
LCGAIHSS+AKQ SE
Sbjct: 86 EDKHKHLLIGVSSDRGLCGAIHSSIAKQMKSE 117
(a) ambiguous

>sp|P36542|ATPG_HUMAN (ATP5C1)ATP synthase gamma chain,
mitochondrial precursor (EC 3.6.3.14).[Homo sapiens]
Score = 98.6 bits (244), Expect = 2e-20
Identities = 52/52 (100%), Positives = 52/52 (100%)
Query: 188 TTSEGSARMTAMDNASKNASEMIDKLTTLFNRTROAVITKELIEIISGAAAL 239
TTSEGSARMTAMDNASKNASEMIDKLTTLFNRTROAVITKELIEIISGAAAL
Sbjct: 246 TTSEGSARMTAMDNASKNASEMIDKLTTLFNRTROAVITKELIEIISGAAAL 297
Score = 55.5 bits (132), Expect = 2e-07
Identities = 37/88 (42%), Positives = 38/88 (43%)
Query: 1 ATLKDITRRLKSIKNIQIKTKSMKIVXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXX 60
ATLKDITRRLKSIKNIQIKTKSMKIV
Sbjct: 26 ATLKDITRRLKSIKNIQIKTKSMKIVAAKYARAERELKPARIVGLGSLALYEKADIKGP 85
Query: 61 XXXXXXXXXXXXXXXXXXXXLCGAIHSSVAKQ 88
LCGAIHSS+AKQ
Sbjct: 86 EDKHKHLLIGVSSDRGLCGAIHSSIAKQ 113
(b) correct

```

Figure 5.5: Alignment improvements by PDBChainSaw.

Alignment of human ATPase gamma subunit onto the solved structure of its bovine homolog. Ambiguous alignment (left) due to unresolved residues. On the right side, the alignment is correct despite the lower bit score.

acid. The relevance of these considerations is demonstrated for the BLASTP alignment of the human mitochondrial ATPase gamma subunit (ATPG_HUMAN) to the homologous bovine protein structure (1ohhG) in Figure 5.5. Although aligning with the unmodified sequence yields a higher score compared to the sequence with filled main-chain breaks, the alignment is ambiguous and simply wrong. In the case of the first alignment, the dynamic programming algorithm extended the first alignment at its N-terminal end and thereby increased the Blast score considerably, which would suggest the wrong alignment as the solution of choice.

The ensemble of extracted sequences from structure files builds the basis of a weekly all-against-all sequence homology comparison using BLASTP (Altschul et al., 1990). The procedure runs in parallel and feeds a relational database with the found hits (str1, str2, E-value, percent identity of alignment) with an E-value threshold of 1.0. The substitution matrix used is “Blosum45” (Henikoff and Henikoff, 1992) to better perform on lower sequence identity. The time needed for computation is less than 2 hours on a 40 CPU cluster.

5.3.1 PDBfused - compression of redundancy

Table 5.1: Compression of sequence redundancy in the PDB.

A total number of 79,954 chains (as of August 2006) have been iteratively fused at the indicated level of sequence identity.

name	seq. id.	iterations	remainders	compression level	redundancy
fused90	90%	17,800	17,595	78%	4.5x
fused80	80%	13,176	13,060	84%	6.1x
fused70	70%	11,893	11,786	85%	6.8x
fused60	60%	10,912	10,761	87%	7.4x
fused50	50%	9,669	9,498	88%	8.4x

To determine the list of non-redundant structural chains in the PDB at different levels k

(percentage sequence identity), the database of BLASTP hits is queried for all chains, which have a homologous hit with at least $k\%$ identity and $E\text{-value} \leq 0.001$. Iteratively, the algorithm then fuses these hits to the query and continues with the next un-visited chain unless only queried chains remain. The algorithm guarantees that shorter sequences will be fused to a longer one by starting with the longest sequence first. As of August 2006, the PDB contained 79,954 chains with at least 20 amino acids and the number of corresponding Blast hits was 16,173,461. In the case of 50% sequence identity, the fusing algorithm stopped after 9,669 iterations and only 9,498 chains remained (Table 5.1). The biggest group was made by 1qgc52 (immunoglobulin), which hits 1,739 chains with at least 50% sequence identity (Table 5.2). Other groups of immunoglobulins could probably also be merged, on the basis of structural similarity, with this group yielding an even larger cluster. The results in Table 5.1 highlight the fact that the Protein Data Bank (PDB) is highly redundant, at least at the level of sequence. This is in accordance with previous works that attempted to reduce database redundancy and to provide best-possible selections of non-redundant structural chains (Hobohm et al., 1992). Such attempts proved successful in improving the speed of sequence homology detection as this method does not benefit from redundant sequence data in the database. However, structural variability is not detectable at the level of sequence; thus, such valuable information is potentially lost when the database is optimized with respect to sequence redundancy only. To relate sequence conservation to structural diversity (i.e. conformational changes), I developed Protopolis (section 5.5), which structurally compares and clusters homologous proteins. Moreover, the software agent PDBalert (section 5.4), which periodically searches the PDB for new sequence homologs and structural similar proteins, also helps to navigate in the increasing redundancy of PDB.

5.3.2 Template search in PDBChainSaw

The database PDBChainSaw may be queried through its user-interface in the iMolTalk server (chapter 4). The form allows for entering query strings in information fields (title, header, compound, source organism, EC code) or numeric ranges (resolution, amino acids, nucleic acids, heterogeneous or solvent groups), and to select their sorting order.

An elaborated query is the report of lists and counts of PDB structures annotated with EC codes. Enzymes were classified by the Enzyme Commission (EC, IUBMB - International Union of Biochemistry and Molecular Biology (1992)) for their biochemical function and cofactors involved into groups, which are labeled with a distinct code. This EC code consists of 4 numbers separated by a dot, comparable to IPv4 network addresses. The first number indicates the top-level biochemical group the enzyme belongs to. The following levels specify the chemical group that is modified by the enzyme or distinct between cofactors and donors/acceptors. All structures in PDBChainSaw were grouped by their EC codes and the reports per top-level group are listed in appendix A. This analysis indicates that for almost all second level groups in the EC hierarchy there are experimentally determined protein structures available for structural analyses or homology modeling. Because the EC classification is based on the biochemical reaction only, it is possible that members of the same group do not share any detectable sequence homology, which could hinder homology based analyses.

Table 5.2: List of the 25 most populated groups of homologous protein structures in PDBChainSaw.

The list contains all structural chains that have at least 200 homologs with 50% sequence identity.

	representative structure			members	common name	length
	PDB code	chain letter	chain number			
1	1qgc	4	52	1,725	Immunoglobulin Fab	438
2	1lsg		32	527	Lysozyme	144
3	1iga	A	65	490	Immunoglobulin Fab	475
4	1hvc		32	471	HIV-1 protease	203
5	1jtn	A	65	464	Lysozyme	177
6	1dxt	B	66	409	Hemoglobin (β -chain)	147
7	1c7d	A	65	405	Hemoglobin (α -chain)	284
8	1bit		32	400	Trypsin (Salmo salar)	222
9	1ezx	C	67	373	Trypsin (Bos taurus)	140
10	1y0l	A	65	366	Catalytic antibody Fab (light chain)	216
11	2f9l	A	65	357	Trypsin (P. leptodactylus)	237
12	1ypz	B	66	349	MHC I β 2-microglobulin	102
13	2bck	A	65	309	MHC I HLA	286
14	1we3	F	70	302	groEL chaperonin	529
15	1bjq	B	66	300	Lectin	253
16	1a5i	A	65	280	Serine protease	265
17	2bc3	A	65	270	Streptavidin	145
18	1uvu	L	76	255	Thrombin	24
19	1ltr	F	70	250	Enterotoxin	109
20	2f2p	A	65	239	Calmodulin	169
21	1ai0	B	66	234	Insulin (β -chain)	30
22	101m		32	233	Myoglobin	154
23	1y6o	A	65	232	Phospholipase A2	131
24	1a7f	A	65	232	Insulin (β -chain)	21
25	1bzw	A	65	229	Lectin	232

The query is available as a distinct toolchain in the iMolTalk server¹ and reports the summary report on the basis of the actual PDBChainSaw database.

¹http://i.moltalk.org/chainsaw_by_eccode/1/Ec1/Ec2/-/-/

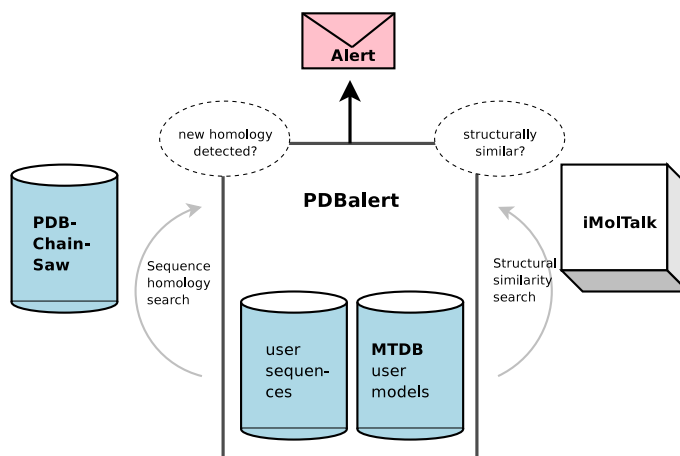


Figure 5.6: PDBalert update procedure.

The list of user-supplied sequences is searched for homology against PDBChainSaw. New and significant hits are reported to the user. The user will also receive an alerting email with details about the structural similarity of 3D models stored in MTDB to the recently released PDB structures.

5.4 PDBalert

Users who are particularly interested in a family of proteins, need to check the PDB periodically for newly found homologous structures. As this procedure proved to be time consuming and error-prone, we developed a novel software agent, *PDBalert*, which searches every week for putative templates in the newly released structures from PDB (Figure 5.6). The user interface of *PDBalert* is integrated into the iMolTalk server (chapter 4) and available to authenticated users. *PDBalert* maintains a database of user-supplied sequences that are compared against PDBChainSaw after the weekly update of the PDB for finding homologous sequences. The list of significant hits is then compared to the one of the previous run and each newly found structural chain triggers an alert email to the user. Additionally, *PDBalert* can also search for structural similarities between user-stored 3D models in MTDB (section 3.2) and newly released PDB structures. For this, the structure comparison toolchain in the iMolTalk server (chapter 4) is called. Computed structural alignments with at least 40 aligned residues with at most 7 Å RMSD are reported. Currently, the procedure relies on BLASTP for sequence comparison, but it is envisaged that more sensitive methods, such as PSI-BLAST, will be included in the future.

5.5 Protopolis

Browsing through multiple conformations of protein structures

The process of homology modeling starts with a protein sequence, for which homologous structures need to be identified using sequence comparison techniques. A crucial step is the evaluation and the selection of templates, as has been demonstrated for re-modeling of aspartate aminotransferases (section 5.2). In cases where a lot of homologous structures are available (i.e. more than 20) this process becomes laborious. For instance, the redundancy of sequence information in PDB, at the level of 50% sequence identity, is more than eightfold (Table 5.1). Hence, I have selected from PDBChainSaw 3,514 groups of homologous structures that contain at least 5 members. To computationally support analyses of these populated groups of homologous structures, I precomputed an all-against-all structural comparison within each group and clustered them subsequently based on the calculated structural similarity. The resulting trees can be inspected through the iMolTalk server (chapter 4) and multiple structural superimpositions can be computed to compare structures in different branches. Information gain of structural annotation, derived from PDB files, is taken into account when the branching of the trees is analyzed. Annotations with maximal information gain might propose hypotheses to explain structural diversity among the selected homologous structures.

5.5.1 Structural comparison

For every group, the $n(n-1)/2$ structural comparisons of its members define a distance matrix, which can then be subjected to clustering. The pairwise distance (eq. 5.1) between two structural chains is defined as the weighted sum of fractions of aligned residues relative to the shorter protein chain (l_{min}). First, an optimized superimposition is computed (see section 2.2.3), in which aligned residue pairs are counted (r_d) using seven distance criteria (1, 1.5, 2, 2.5, 3, 3.5 or 4 Å).

$$d_{AB} = 1.0 - \frac{\sum r_d}{7 l_{min}} \quad (5.1)$$

This measurement is zero for identical conformations and one for completely unaligned structures.

5.5.2 Clustering

The $n \times n$ distance matrix is clustered using UPGMA (unweighted pair group method with arithmetic mean) (Sokal and Michener, 1958). The clustering algorithm iteratively combines the two closest row vectors, replacing them by their mean vector, thus generating a binary tree in bottom-up manner. The first split of the rooted tree maximally separates two groups (Figure 5.7).

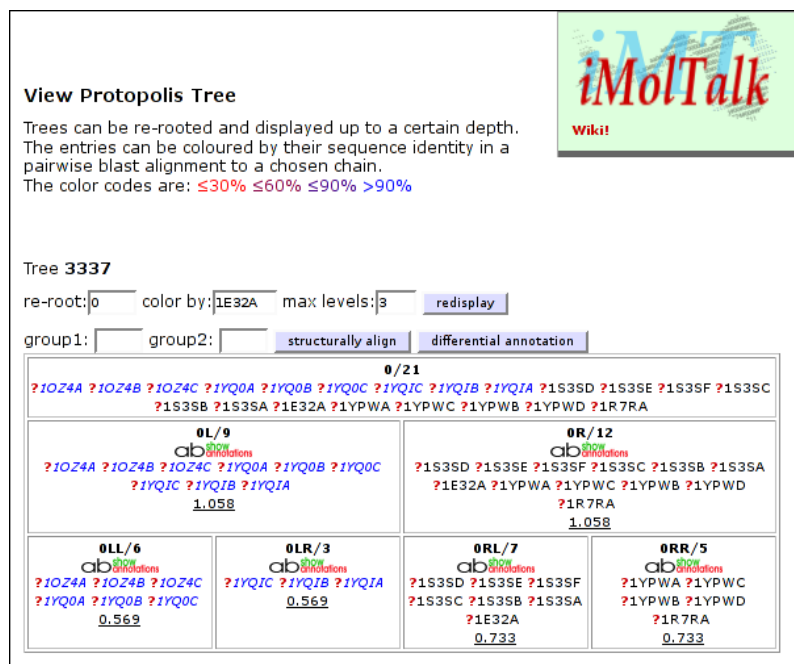


Figure 5.7: Clustering tree of p97 structures.

The structural chains in this tree share 100% sequence identity, but the structural comparison splits the tree into at least four branches. Chains in the left branch (0L) are highlighted in blue and all share the same annotation "resolution(low)" and "spacegroup('I 2 2 2')". This represents a possible explanation for the structural diversity as all other chains in the tree are of better resolution and crystallized in another form. Two selected branches in this view can be superimposed or their annotations inspected. The structural chains may also be colored according to sequence identity in the alignment to a homologous chain. Moreover, the tree can be re-rooted and the displayed depth of the tree can be adjusted. The numbers at the end of the boxes indicate the distances of their centroids in the last split.

resolution information gain: 100.0%	
■ resolution(low) @ [1.00, 0.00]	■ resolution(high) @ [0.00, 0.58]
	■ resolution(medium) @ [0.00, 0.42]
spacegroup information gain: 100.0%	
■ spacegroup(I 2 2 2) @ [1.00, 0.00]	■ spacegroup(P 65) @ [0.00, 0.50]
	■ spacegroup(P 3) @ [0.00, 0.33]
	■ spacegroup(P 6 2 2) @ [0.00, 0.17]

Figure 5.8: Annotation of branch 0L in the clustering tree of p97 structures.

Branch 0L may be explained by the annotations "resolution(low)" and "spacegroup('I 2 2 2')", branch 0RL by "resolution(high)".

5.5.3 Superimposition

Two groups in the clustering tree may be selected and the ten first structural chains of each group superimposed and written to a PDB-formatted file. The superimposition can be visualized and colored either by groups (Figure 5.3a) or by the RMSD values of the structural alignment of the first structures from the two groups (Figure 5.3b).

5.5.4 Annotation

Structural chains are annotated for their attributes: spacegroup, EC code, source organism and keywords, interpreted from PDB files using MolTalk and stored in a relational database. A qualitative attribute for the resolution has values: high, better than 3 Å; medium, between 3 and 4 Å; low, worse than 4 Å. Furthermore, heterogeneous groups in contact with the chain are detected and added as attributes *hetgroup* and *hetname*. In subsequent analyses, this database is queried and the frequency of every attribute-value pair in each branch is calculated.

The annotation report shows the attributes in a selected branch and their frequencies within the branch (f_A^B) versus their frequencies in all other branches in the tree ($f_A^{\bar{B}}$) and sorts them according to the difference $f_A^B - f_A^{\bar{B}}$ (Figure 5.8). Moreover, the information gain $I_G(A)$ of an attribute A is calculated for a selected branch B (Shannon and Weaver, 1949; Russel and Norvig, 1995; Poole et al., 1998):

$$I_G(A) = 1 - \sum_v f_{A,v} * I_E(A, v) \quad (5.2)$$

where $f_{A,v}$ is the frequency of annotations with attributes A and value v in the tree, the entropy is written as $I_E(A, v) = I_E(f_{A,v}^B, f_{A,v}^{\bar{B}}) = -f_{A,v}^B \log_2 f_{A,v}^B - f_{A,v}^{\bar{B}} \log_2 f_{A,v}^{\bar{B}}$.

The information gain in bits (eq. 5.2) indicates the importance of the attribute to explain the difference between the selected branch versus all other branches. In decision tree learning, examples are split into subgroups by selecting the criteria with the highest information gain (Quinlan, 1986). In contrast, trees in Protopolis were derived from structural similarity. Accordingly, the splitting is compared to the available annotation for the hypothesis generation and attributes that explain a split with more than 0.75 bit

Table 5.3: Number of trees with information gain 1.0 per attribute.

Trees showing structural diversity (top-level distance ≥ 1.0) have been analyzed for discriminative attributes.

	$I_G = 1.0$	
eccode	125	17%
resolution	122	16%
spacegroup	262	35%
hetgrp	354	47%
hetname	365	49%
pdbkw	204	27%
srcorg	217	29%

are highlighted. The first occurring annotations are the most discriminative ones, i.e. they occur more often in the selected branch than in all other. This might lead to the generation of hypotheses that explain the splitting into groups in the tree.

Similarly, in a second type of report two selected branches may be compared, on the basis of the computed frequencies, by the difference of $f_i^{B_1} - f_i^{B_2}$ (Figure 5.9). This is valid under closed world assumption, thus pretending complete knowledge, and by neglecting possible erroneous annotation.

a:0L	b:0RL
<input type="checkbox"/> spacegroup(1 2 2 2) @ [1.00, 0.00]	<input type="checkbox"/> resolution(high) @ [1.00, 0.00]
<input type="checkbox"/> resolution(low) @ [1.00, 0.00]	<input type="checkbox"/> spacegroup(P 65) @ [0.86, 0.00]
<input type="checkbox"/> pdbkw(AAA) @ [1.00, 0.00]	<input type="checkbox"/> pdbkw(UBX DOMAIN) @ [0.86, 0.00]
<input type="checkbox"/> pdbkw(STRUC) @ [1.00, 0.14]	<input type="checkbox"/> pdbkw(SPECTR PROTEIN QUINASE) @

Figure 5.9: Differential annotation view.

Annotation in the clustering tree of p97 structures is compared between branches 0L and 0RL. Attribute-value pairs are annotated with their frequency of occurrence in the first versus second branch. Sort order of the annotation is given by the difference of the two frequencies.

5.5.5 Clustering trees

From PDBfused (50% sequence identity level), 3,514 structural chains were selected that are related to five or more homologs. For each selected chain, the set of homologous chains was structurally compared and clustered according to structural similarity (eq. 5.1). Among the 3,514 trees, 935 trees showed less than 0.1 distance of the top level split, indicating very little structural diversity, and 121 trees showed at least 3.0 distance (Figure 5.10). Among the 748 trees, which show a top-level split distance larger than 1.0, *hetgrp* and *hetname* are the most discriminative attributes (Table 5.3).

From analyzing the trees that have at least one attribute with information gain 1.0 in Table 5.3, a global picture can be drawn:

The EC code often reflects the degree of homology between structural chains. As an example, EC codes 3.4.21.59 and 3.4.21.4 split tree #131 (1ltoA, serine protease) into

subgroups. The first group contains 43 structures of the human protein, whereas 98 orthologous chains from other higher eukaryotes are found in the second group.

Further analyzing the 122 trees, whose first split shows 1 bit information gain by the attribute *resolution*, reveals that in 101 trees the attribute-value pair *resolution(NMR)* is most discriminative and can serve as an explanation of the structural diversity. This is a strong indication that high-resolution structures from X-ray crystallography are distinct from solution structures solved by NMR.

The trees that split by *resolution(NMR)* also show 1 bit information gain for the attribute *spacegroup* as they are annotated with the default spacegroup P1. For the other cases, it is not clear whether the preference for a particular spacegroup of a crystal depends on the protein's conformation, or whether the conformation is a crystallization artifact.

Discriminative annotations of heterogeneous atom groups (*hetgrp* and *hetname*) in trees indicate conformational changes that can be explained by the presence or absence of such ligands, but this has to be verified individually. As an example, the ligand methotrexate inhibits dihydrofolate reductase (DHFR). Tree #304 shows the single chain (1rg7) that is co-crystallized with methotrexate apart from the group of 61 other chains. The only conformational change is indeed visible in the Met20 loop which adopts a closed conformation (Sawaya and Kraut, 1997).

Keywords (*pdbkw*) parsed from PDB files might indicate structures that were solved by the same experimentalists; thus, they were annotated using the same keywords.

Trees that split by the attribute *srcorg* reveal structural diversity that is related to sequence conservation. As an example, tree #585 (1m79B) splits into one group of structures from *Candida albicans* and a second group of structures from *Plasmodium falciparum* and *P. vivax*. The structures in the second group have an insertion of around 70 amino acids in a loop, whereas the core is structurally well aligned. This insertion might be the cause of the clustering into distinct subgroups.

5.5.6 Detection of conformational states in p97 AAA+ structures

AAA+ proteins use the chemical energy from ATP hydrolysis to provide mechanical work through conformational changes (Davies et al., 2005; Pye et al., 2006). The protein p97 contains two AAA+ ATPase domains, D1 and D2, and forms double rings of hexamers. A more thorough introduction of this interesting protein family follows in the multimer modeling part (chapter 6).

A number of crystal structures in different states have been determined to observe this protein at work and understand the transformation of energy from hydrolysis to mechanical work (Figure 5.7). The two structures, 1oz4 and 1yq0, in group OLL of the tree represent an activated state with bound ADP and AlF_3 in D2. A re-refinement of data from the structure 1yq0 to higher resolution resulted in a new structural model 1oz4 that clearly shows the presence of the AlF_3 moiety. Group OLR contains the structure 1yqi and represents the ADP bound state. Group ORL contains structures of only the D1 AAA+ cassettes (1s3s and 1e32). Group ORR contains the structures 1r7r and 1ypw. The first is in apo form, the latter represents the ATP bound state (ANP, an ATP analoga was co-crystallized).

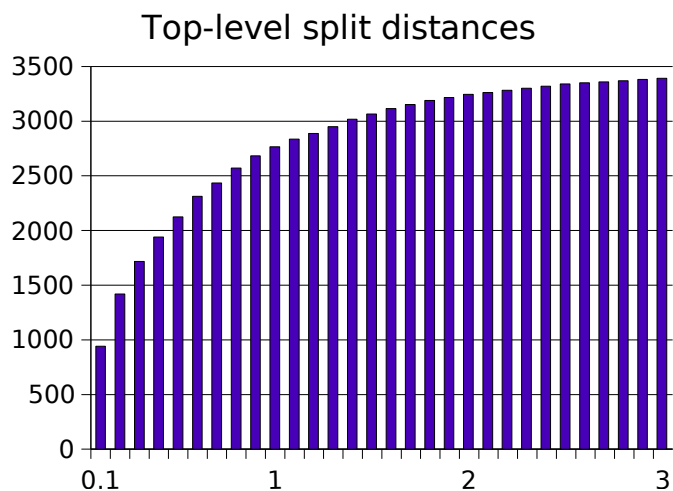


Figure 5.10: Cumulative number of trees versus top-level split distances.

935 trees show less than 0.1 distance in the first split. The remainder 2,579 trees show a variety of structural diversity: 748 trees have at least 1.0 distance, 268 show distances larger than 2.0 and 121 trees have distances of at least 3.0. The measurement of distance is in arbitrary units computed from comparison of dissimilarity vectors.

The proposed hydrolysis cycle goes from the ATP state (1ypw, group ORR) to the activated state (1yq0, group OLL) and then the ADP state (1yqi, group OLR) (DeLaBarre and Brunger, 2005). A large change can be observed while turning to the activated state and a smaller change between this state and the following ADP state. Then, a large change occurs again while exchanging ADP for ATP. In between, the apo structure (1r7r, group ORR) does not show much difference to the ATP state (1ypw, group ORR), probably indicating little change.

However, this interpretation does not cover group ORL, which consists of the best resolved p97 structures (1s3s and 1e32). A characteristic of these structures is that they only include the N-domain and the D1 AAA+ cassette (Dreveny et al., 2004; Zhang et al., 2000). The D1 domain has been shown to be responsible for hexamerization whereas the D2 domain only forms a compact hexameric ring structure in presence of nucleotides (Wang et al., 2003b,a). It is therefore no surprise that D2 is not resolving well probably due to its motility.

But, which state do these structures represent? In Protopolis, the comparison of structures maximizes the local overlap of similar structures. Shorter structures are not penalized *per se* and could result in the same score for the overlapping region as longer ones. This means that the D1 domain of structures in group ORL is distinct from the one in the other groups. As an exception, 1r7r groups with 1ypw but is structurally very similar to 1e32 (RMSD 0.2 Å over 436 amino acids, D1 only). However, its additional N-terminal residues and the C_{α} trace of its D2 domain make it more compatible with 1ypw. As shown in Figure 5.11, the dissimilarity vector of 1r7rA is closer to those of chains in ORR than in ORL.

5.6 Summary and Outlook

PDBChainSaw extracts information from PDB structures and reconstructs their sequences from coordinates (section 5.3). This database is fully searchable through the iMolTalk server to find templates at the meta-level of structural information and forms the basis of many analyses in this work. Moreover, the reconstructed sequences are used in homology searches (i.e. BLASTP and PSI-BLAST). PDBChainSaw is not restricted to PDB files only, but may also contain structural information from other sources. This makes PDBChainSaw a central access point for structural data of any kind.

The example of remodeling of aspartate aminotransferases has indicated a problem in the template selection step of homology modeling that has not been addressed before. When mixing templates of the two known distinct conformations of this protein in the remodeling, a larger RMSD value was observed when the models were compared to the experimentally determined structure of each conformational state than if only one conformational state was used as a template. This result suggests that blindly selecting templates in homology modeling may result in models that do not reflect true biological conformations. The novel method Protopolis (section 5.5) is tailored towards the identification of conformational states and helps selecting the most appropriate templates for a homology modeling task. Protopolis has been integrated into the iMolTalk web-server along with PDBalert, which sends email to registered users if the weekly released PDB files contain putative templates.

Structure similarity search algorithms, such as DALI (Holm and Sander, 1993), CE (Shindyalov and Bourne, 1998) and VAST (Gibrat et al., 1996), are limited to return one-dimensional lists of structural similar proteins in the PDB. In Protopolis, high-dimensional clustering of structural similarity is very sensitive and allows to compare groups of homologous structures and to generate clustering trees. Results from PDBfused (section 5.3.1) indicate a more than eightfold redundancy in PDB at the level of 50% sequence identity. This redundancy at the sequence level can represent a welcome diversity at the structural level and provide interesting insights into conformational states and protein function. In the future, we will observe even more sequence redundancy in the PDB. This is mainly due to technical advances, which lead to remaking of a number of protein structures in different environments and co-crystallized ligands, and structural genomics initiatives which so far elucidated more structural similar proteins than novel folds.

The analysis of 3,514 clustering trees in Protopolis revealed that structural annotation can indeed explain, based on maximal information gain, the top-level split in approximately half of the cases. Observed structural diversity between homologous crystal and solution structures can be explained in 101 trees. This indicates that there still exists a gap in resolution between models solved by the two techniques. Moreover, a number of splits can be explained by annotation of bound ligands. This calls for further analyses in each case to determine the conformational change and to assess the ligand's influence. In the future, an improved decision making procedure needs to be implemented to better discriminate between true conformational changes or structural diversity that is due to changes at the level of sequence.

The clustering of p97 protein structures showed how Protopolis can help to identify con-

formational states in homologous protein structures. It was able to recapitulate the different states in the nucleotide hydrolysis cycle, but also raised the question how the high-resolution structures, which resolved only the first ATPase domain, are related to the other structures in the tree.

Protopolis has been successfully applied to a number of modeling cases and its integration into the iMolTalk web-server make it available to a large scientific community. In the future, Protopolis can include more annotations and the threshold of sequence identity could be lowered to include more structures in the clustering trees. Results from this work suggest that Protopolis addresses a necessary step in homology modeling that needs to be integrated into today's structure prediction pipelines.

Multimer Modeling

Chapter 6

Modeling of AAA+ ring structures

6.1 Introduction

AAA+ protein domains (ATPase associated with diverse function) act as general molecular motors driven by the free energy from ATP hydrolysis (Lupas and Martin, 2002). They form oligomeric ring structures, a prerequisite for their function in physiological state. This is highlighted by the fact that in most AAA+ families the bound nucleotide in the active site of one monomer is coordinated by an arginine, termed Arg finger, of a neighbor monomer (Ogura et al., 2004). Mutation of this residue reduces ATP hydrolysis drastically.

Since their first description (Erdmann et al., 1991), AAA+ proteins were found in all kingdoms of life. Most contain a single nucleotide binding domain (NBD) but some show tandem domains and form double rings. In the extreme case of midasin (Swiss-Prot: MDN1_HUMAN) six are found in sequence. The AAA+ domain is formed by a canonical NTPase domain with characteristic Walker A and B sequence motifs (Walker et al., 1982) and a helical domain at its C-terminus. In the first, a major structural feature is the central, five-stranded parallel β -sheet where 5-1-4-3-2 is the order of the strands (Figure 6.2). They are connected by loops and helices such that three helices flank each side of the β -sheet. Some of these loops extend toward the center of the ring and contain family-specific sequences to shape the central pore ($\beta 2 - \beta 3$, $\beta 3 - \beta 4$). Strand $\beta 1$ ends in the Walker A P-loop, a highly conserved structure in NTPases, which coordinates the bound nucleotide and positions the catalytic lysine in place. At the end of $\beta 3$, we identify the Walker B (DEAD box) acidic residues, which coordinate a cation in the active site. After $\beta 5$, the polypeptide chain continues in the helical domain (C-domain), which itself is involved in nucleotide coordination. Before this strand, a conserved region, termed second region of homology (SRH) (Swaffield et al., 1992), is common to all AAA+ proteins. A number of AAA+ protein structures were solved by X-ray crystallography, but only a few show the expected ring structures (Figure 6.1 and Table B.3), whereas the remainder crystallized in non-ring shaped forms even though electron microscopy (EM) studies have indicated hexameric or heptameric ring structures (Table B.4). Therefore, it is of interest to model such ring structures from available high-resolution monomers. The analysis of structural restraints on AAA+ rings is set as the rational basis of the modeling. Models are

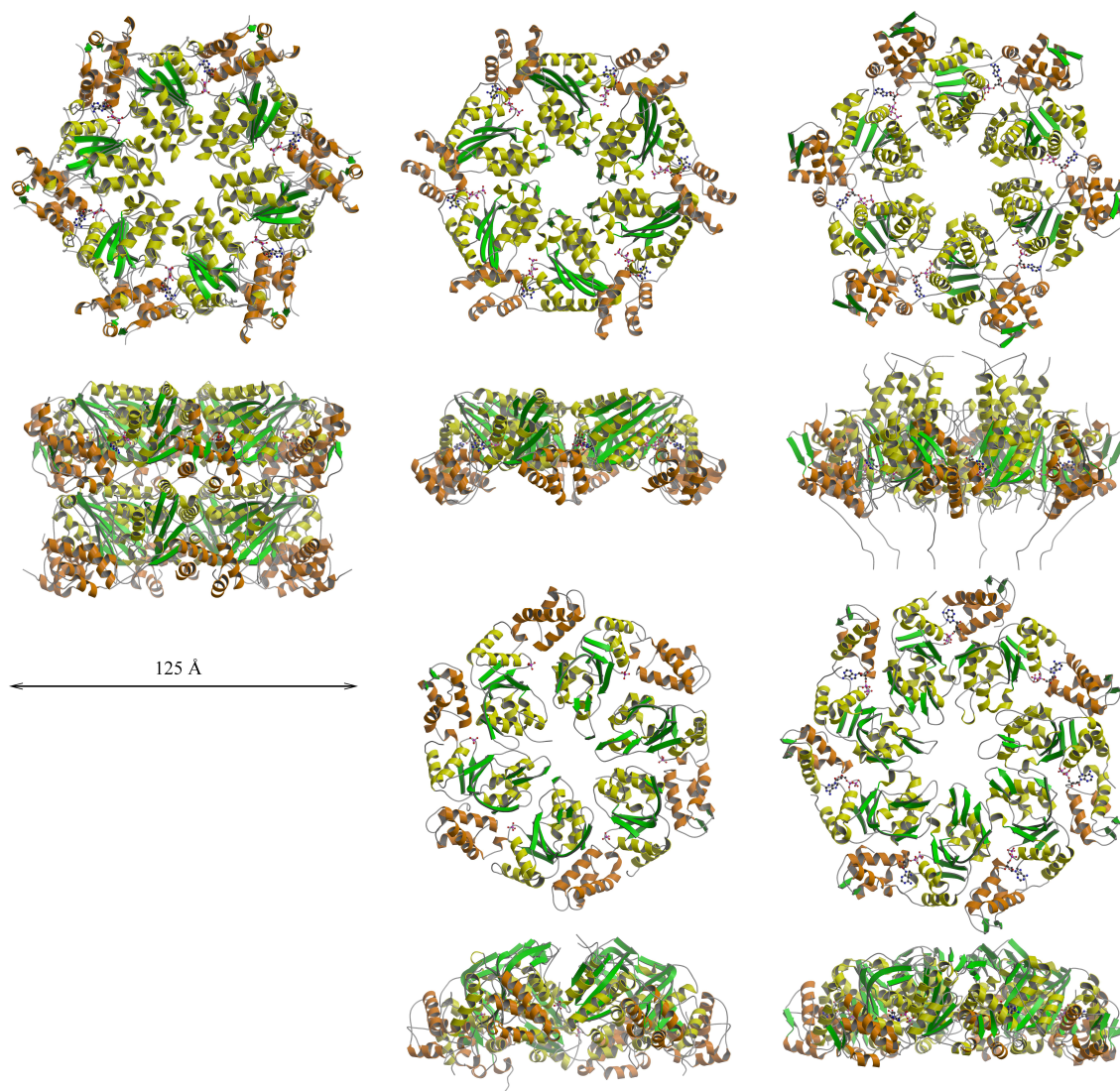


Figure 6.1: Gallery of AAA+ ring structures.

From top left are shown in the upper row: p97 (tandem D1 and D2 AAA+ domains, 1r7r), NSF (1nsf), HslU (1kyi) and in the lower row: ZraR (1ojl), NtrC1 (1ny6). The coloring is green for β -strands and yellow for α -helices (orange in the C-domain). The homologous proteins ZraR and NtrC1 crystallized in hexameric and heptameric form.

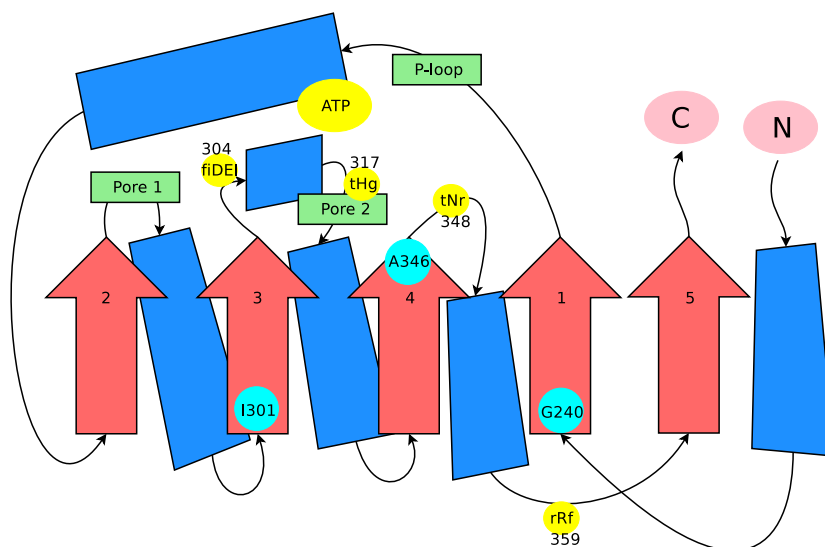


Figure 6.2: Topology of a AAA+ NBD monomer.

The order of the strands (red arrows) in the β -sheet is 5-1-4-3-2. The Walker A P-loop is located after β_1 , the Walker B acidic residues are the C-terminal part of β_3 . Between β_2 and β_3 we identify the pore 1 region, and between β_3 and β_4 the pore 2 region, which carries the prominent histidine 317 in the AAA protein p97 (1e32). Before β_5 , the SRH region contains the Arg finger (R359) pointing into the active site of a neighbor monomer in the ring structure. Residues highlighted at the N-terminal end of β_3 and β_1 as well as A346 in β_4 indicate the positions which abstract the ideal plane of the β -sheet.

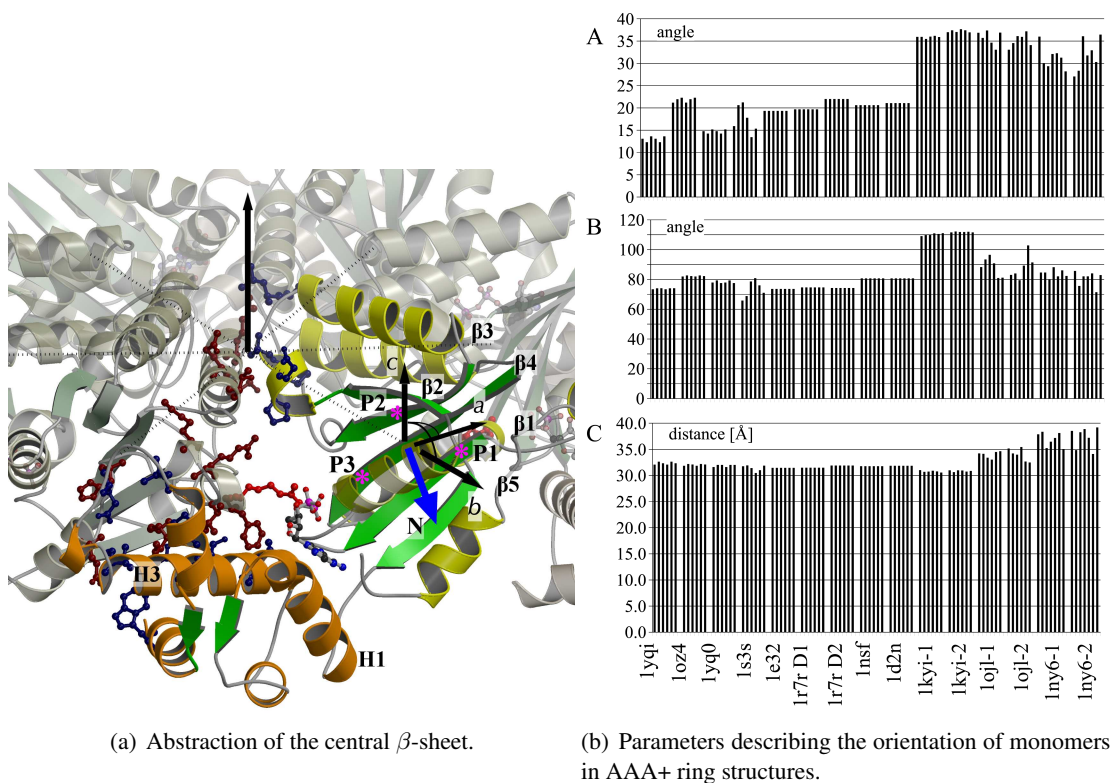
further optimized using an established protein-protein docking protocol. For three different AAA+ proteins we report the modeled ring structures and discuss possible biological implications.

6.2 Structural restraints

Through manual analyses and using the iMolTalk web-server presented in chapter 4, a number of structural restraints from ring structures of AAA+ proteins (Table B.3) were derived. These restraints were then applied in the modeling and optimization of rings (section 6.3).

6.2.1 Position and orientation of monomers

At the center of the NTPase, the β -sheet is highly conserved. It positions the pore-shaping loops and the Arg finger at its N-terminal end and the residues essential for nucleotide binding at its C-terminal end and is thus highly constrained in its position within the ring. To be able to compute reproducible structural alignments of AAA+ proteins we abstract the orientation and position of the central β -sheet by selecting three residues (denoted

(a) Abstraction of the central β -sheet.

(b) Parameters describing the orientation of monomers in AAA+ ring structures.

Figure 6.3: Orientation of monomers in AAA+ ring structures.

In the left subgraph, the three points P1-3 at defined positions in the central β -sheet define a plane, which abstracts the location of the monomer within the ring complex (p97, 1e32). Strands are colored green and helices yellow (orange in the C-domain). Residues at the interface between monomers are shown in ball-and-stick representation and colored blue and red to discriminate between monomers. In the right subgraph, the angle around axis c (**A**) and axis a (**B**) indicate sub-family specific orientations. **C** Many structures show similar distances from their central β -sheet to the central axis.

P1, P2 and P3) that are easily identified in all AAA+ proteins, one at the N-terminus of $\beta 1$ (P1), one at the center of $\beta 3$ (P2), and the last at the C-terminus of $\beta 4$ (P3). These three residues define a plane which is parametrized by its normal vector $((P1-P2) \times (P1-P3))$ and its center point (mean position of P1, P2 and P3) (Figure 6.3). The unique transformation, which describes the rotation and translation of (P1,P2,P3) of one structure to match (P1,P2,P3) of a second structure, is used to compute the superimposition of the two structures. Using this transformation in the generation of starting structures for the oligomer modeling procedure (section 6.3), the problem of non-deterministic structural alignments is circumvented.

The subunit positions in AAA+ ring structures, determined using this parametrization, are shown in the right subgraph of Figure 6.3.

To reliably detect P1, P2 and P3 in a AAA+ monomer structure, a MBSIS query has been developed using the relational filters as described in section 3.3 (Figure 6.4). Parametrization of the spatial restraints has been done for D1 and D2 domains independently (data not shown).

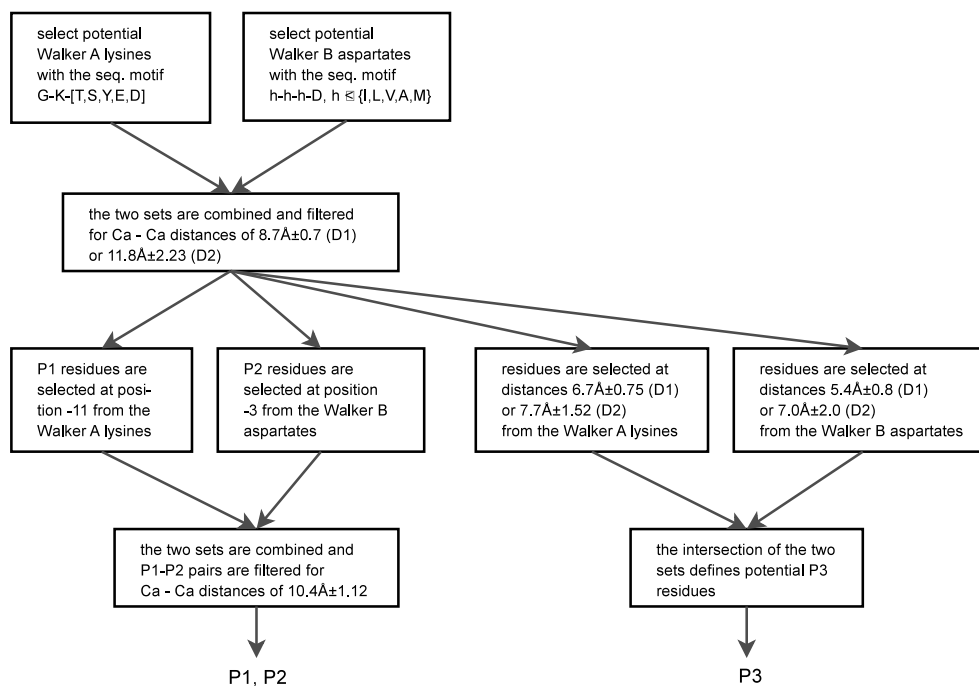


Figure 6.4: MBSIS filter to automatically define P1-3 in AAA+ structures from structural and sequence restraints.

6.2.2 Relative position of ATPase and C-domain

In modeling ring complexes from monomers the issue arises whether the core ATPase domain and the C-domain can rotate freely around the hinge region which connects them.

The ATPase and C-domains in AAA+ proteins are connected by a short linker after $\beta 5$. For the analysis of the relative orientations between the two domains we designated a residue in this linker as a hinge, around which free rotation is possible. In the multiple structural alignment, assembled from pairwise structural alignments and manually curated to align functional homologous regions, $\beta 5$ of some AAA+ proteins contains an acidic residue at its C-terminal end (Figure 6.5). We defined the hinge residue three positions C-terminal to the acidic residue, or, in cases where it is missing, the residue at the same position (Table B.1). Rotation around the hinge is computed as the change of the angle described by the C_α atoms of the Walker A lysine, the hinge residue and sensor-2 in helix H3. A second hinge angle was similarly defined but included the conserved hydrophobic residue in helix H1 of the C-domain in place of sensor-2.

From our analysis little variation was found as judged from the distances and angles between characteristic residues. The dihedral angle describing the relative rotation between the two domains had a mean value of $64.6^\circ \pm 5.1$ (Table B.6). Only NSF-D2 had a dihedral angle of 50° , indicating a remarkable rotation compared to the other AAA+ proteins.

6.2.3 Nucleotide coordination

The nucleotide binding pocket in AAA+ proteins is located in a cleft between ATPase and C-domain. We identified residues contacting the bound nucleotide within a distance of 3.8 Å and mapped these to the structural alignment of a number of AAA+ proteins (Figure 6.5). The Arg finger is only visible in the p97 ring structure and similar to FtsH is found upstream to the canonical position d). In the ATPase domain, most contacts are made by residues N-terminal to $\beta 1$ and by the Walker A motif. In the C-domain contacts are made mainly by helix H3 in the sensor-2 region and by a conserved hydrophobic residue (denoted e), in the second turn of helix H1. Contacts are also observed from the residue (denoted f) in the third turn of helix H1, frequently aromatic, in the structures of p97, FtsH, ClpA-D2, ZraR, NtrC1 and Apaf-1. In ClpA/B-D1, this residue is found one turn further. HslU and ClpX show an insertion at this position in H1 and in HslU the aromatic residue is compensated by a histidine in the second turn of helix H3. The side-chain of this residue is oriented toward the ribose moiety of the bound nucleotide (Figure 6.6). Our findings confirm and extend the results of Botos et al. (2004).

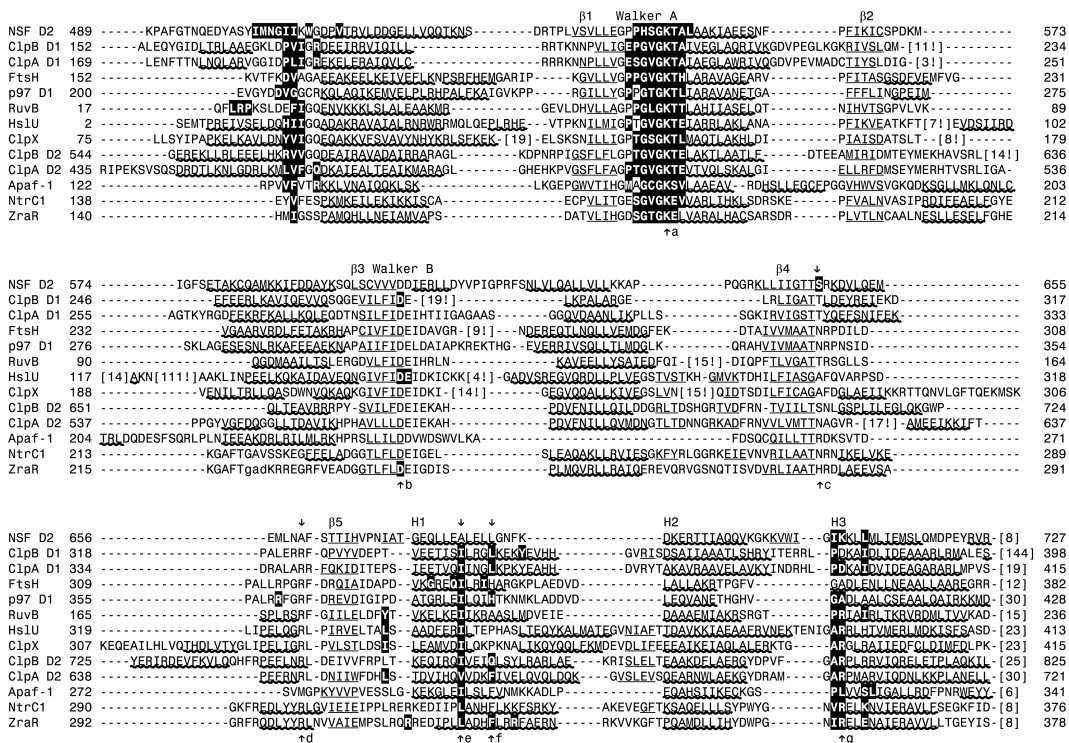


Figure 6.5: Nucleotide coordinating residues highlighted in the alignment of AAA+ structures.

Residues shown in inverse colouring indicate contacts (distance threshold 3.8 Å) to bound nucleotides. The columns highlighted by arrows are: a) Walker A lysine, b) Walker B aspartate, c) sensor-1, d) Arg finger, e) conserved hydrophobic position, f) aromatic position, g) sensor-2.

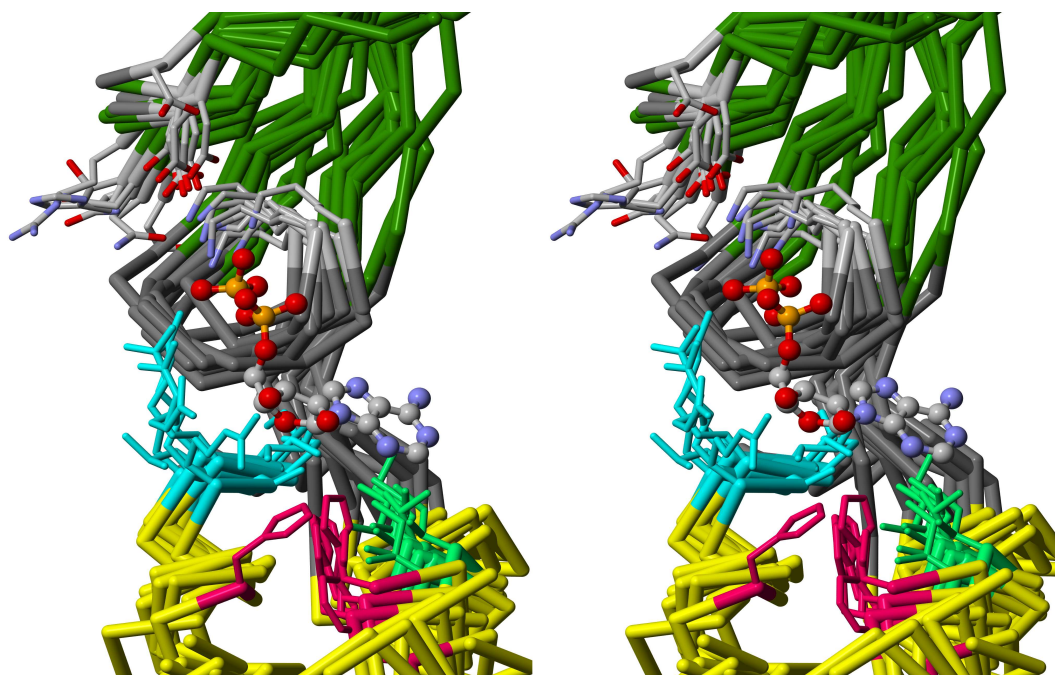


Figure 6.6: Nucleotide coordinating residues.

The superimposition of AAA+ structures is shown in parallel stereo view. At the top from right to left are shown strands β_5 , β_1 , β_4 and colored green. Strand β_5 continues into the C-domain whose helices are shown in yellow. The bound nucleotide is mainly coordinated by the Walker A lysine (CPK colouring) in the NBD and sensor-2 (cyan) in helix H3, conserved hydrophobic (light green) and aromatic (magenta) positions in helix H1 of the C-domain. In HslU, the aromatic residue is found in helix H3.

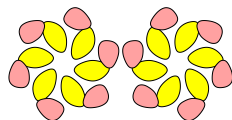


Figure 6.7: Handedness of AAA+ ring complexes.

The left-handed (left subgraph) or right-handed (right subgraph) orientation of the ring are defined with the axis of rotation, perpendicular to the image plane, pointing toward the viewer. Monomers are schematically drawn as NTPase (yellow) and C-domain (red).

6.2.4 Oligomer interfaces

Inter-monomer contacts were analyzed for conserved positions in experimentally determined AAA+ ring complexes. In some cases, we had to reconstruct ring coordinates using the crystallographic transformations given in the structure file (Table B.3). We selected four hexameric structures, NSF (1d2n), p97 (1e32), ZraR (1ojl) and HslU (1g41), and identified residues at monomer interfaces with a distance threshold of 3.6 Å. The location of these residues is shown both mapped to the structural alignment and schematically in Figure 6.8. Most contacts are van der Waals interactions that are not conserved in the superfamily, but consistently occur between the same structural elements. The main specific contact is made by the second acidic residue in the Walker B motif (at the C-terminal end of $\beta 3$), which forms a salt-bridge with a residue in the loop region before $\beta 5$ (ZraR and p97) or $\beta 4$ (NSF).

Using the definition of the central axis (section 6.2.1) one may define two types of handedness of the rotation around this axis (Figure 6.7). All AAA+ ring structures studied so far show the right-handed orientation, which further supports the conservation of the inter-monomer interfaces.

6.3 Optimization of ring structures

Rebuilding oligomers by superimposition of monomers onto a template oligomer results in a poor fit of the subunits, as judged by solvated gaps and van der Waals overlaps of backbone atoms between monomers. In our present implementation we superimpose an input monomer onto a template ring structure using the transformation as described in section 6.3. This starting structure is then optimized for the packing between monomers. We implemented an iterative procedure (Figure 6.9) which generates random orientations of the monomer, rebuilds oligomeric ring structures and uses the protein docking program RosettaDock (Gray et al., 2003) to remodel the monomer interfaces by sampling side-chain rotamers while keeping the backbone fixed (Wang et al., 2005). To score the oligomer model, we used a combination of the RosettaDock score, the score of the inter-monomer distance restraints and the difference of solvent accessible surface area (dSASA) in docked versus undocked states. The iterative procedure continues for a defined number of iterations or until convergence of the total energy is reached. The last accepted parameter set and the corresponding ring model represents the optimal solution

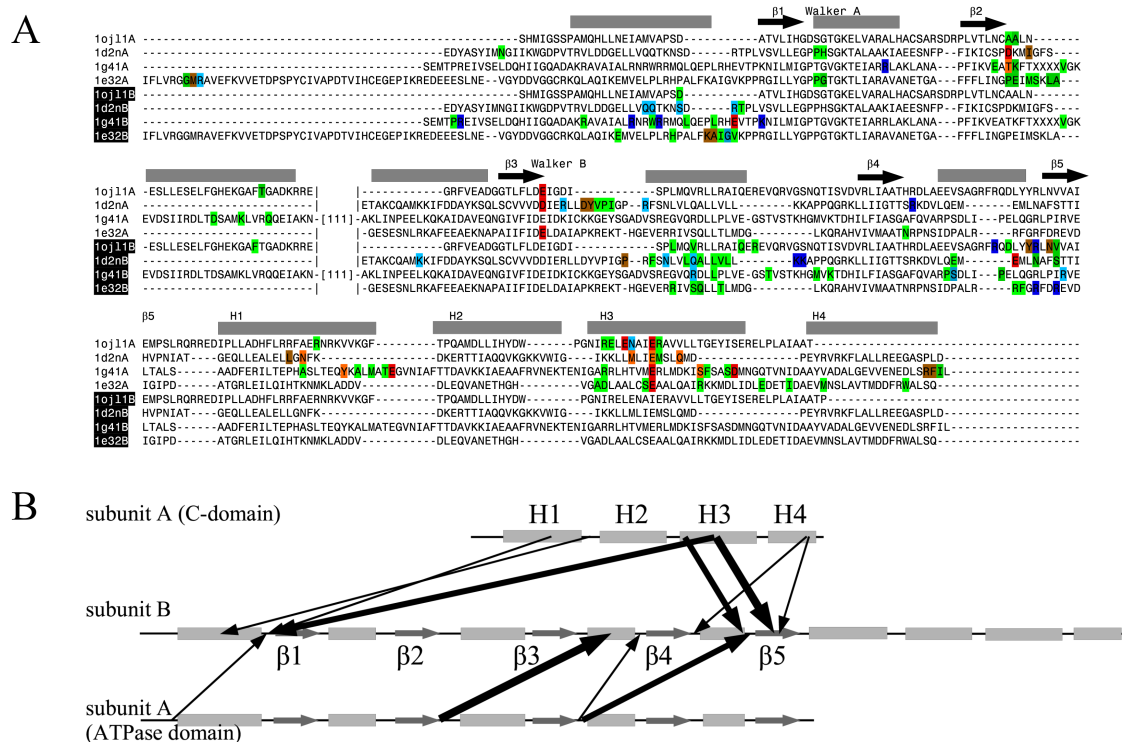


Figure 6.8: Residues at the inter-monomer interfaces mapped to the structural alignment of ring forming AAA+ structures.

A, Contacts at the subunit interfaces in the experimentally determined ring structures p97 (1e32), NSF (1d2n), HslU (1g41) and ZraR (1o1l) are color coded for charge interactions (red, blue), H-bond donors (sidechains: cyan, backbone: petrol) and H-bond acceptors (sidechains: orange, backbone: brown). Residues that may act as H-bond donors or acceptors are colored magenta. Van der Waals contacts are colored green. Interface residues were identified using a distance threshold of 3.6 Å. **B**, Schematic representation of the main contacts. The thickness of the arrows is drawn according to the number of contacts observed in the four structures. For clarity, contacts between ATPase domains are shown separately from contacts between ATPase and C-domain.

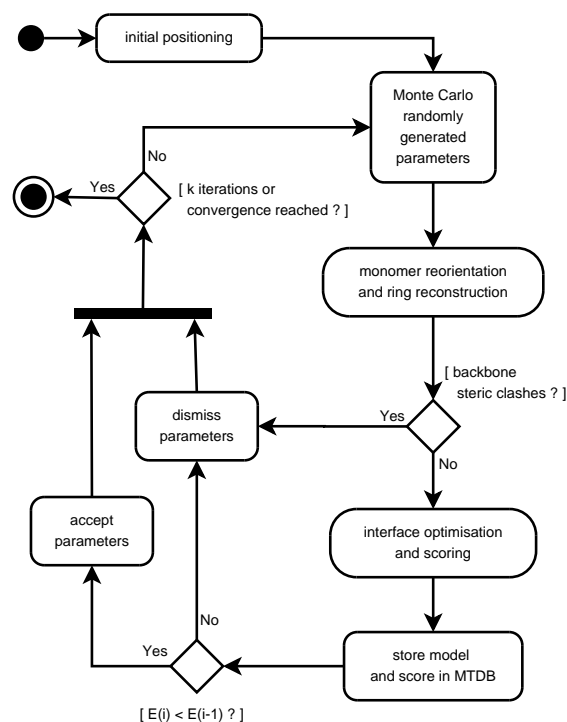


Figure 6.9: Ring structure optimization protocol.

The schema shows the iterative algorithm based on Monte-Carlo sampling to position monomers in the ring structure and evaluating its score using the protein-protein docking program Rosetta-Dock.

found by the algorithm, but all intermediary computed models are stored together with their scores in a relational database (based on MTDB, section 3.2). The user may thus follow the docking runs via our web-server, inspect plots of scores and parameters, and analyze the model structures on-line or download them in PDB format to a local computer.

6.3.1 Monte-Carlo sampling

A Monte-Carlo (MC) sampling algorithm (Metropolis et al., 1953) was implemented to explore possible monomer orientations and translations relative to the central axis. The user sets the number of iterations, the permissive range of translation and rotations, and the magnitude of their variation (step-sizes). These step-sizes are scaled by random numbers between -0.5 and 0.5 and added to the current parameter set at each iteration to generate the new parameters used for orientation of the monomer.

The protein docking procedure consists of the following steps (Figure 6.9):

1. The input monomer is reoriented using the parameters generated by the MC algorithm.

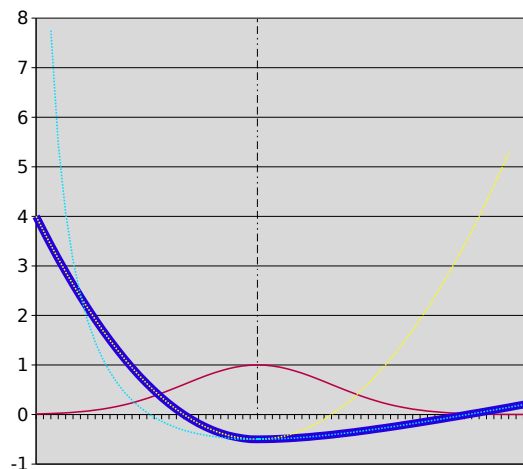


Figure 6.10: Potential for distance restraint evaluation.

The evaluated distance x is compared to a normal distribution $N(\mu, \sigma)$ (red). The potential E_r (blue line) is based on the lognormal distribution. Formula 6.1 is designed to penalize short distances if $x < \mu$ (yellow curve) and to relax restraints for distances that are far off the desired mean μ (cyan curve).

2. The oligomer is rebuilt using transformation matrices and tested for van der Waals overlaps of backbone atoms between monomers. In case of severe overlaps the proposed orientation is dismissed.
3. Two monomers (denoted A and B) are extracted from the oligomer and their interface is rebuilt and scored by RosettaDock.
4. As RosettaDock independently optimizes side-chain orientations in the two monomers, but the oligomeric structure is rebuilt using a single monomer, we transfer side-chain orientations of residues within 5 Å of the interface from monomer B to monomer A.
5. The final oligomer model is rebuilt from copies of monomer A rotated around the central axis.

The MC energy is computed as a weighted linear combination of the resulting RosettaDock score (E_{RD}), the evaluated distance restraints (E_r , eq. 6.1 and Figure 6.10), and the change of solvent accessible surface area (E_{dA}). The weights are set by default to $0.8 * E_{RD}$, $1.0 * E_r$, and $0.05 * E_{dA}$, but this can be changed by the user.

$$E_r = \begin{cases} 0.5 \cdot \left(\frac{x-\mu}{\sigma}\right)^2 - 1 & \text{if } x < \mu \\ 4.5 \cdot \left(\frac{\log(x)-\log(\mu)}{\sigma}\right)^2 - 1 & \text{otherwise} \end{cases} \quad (6.1)$$

Including the difference in solvent accessible surface area into the total energy function helps minimizing solvated gaps between monomers, whereas the RosettaDock score

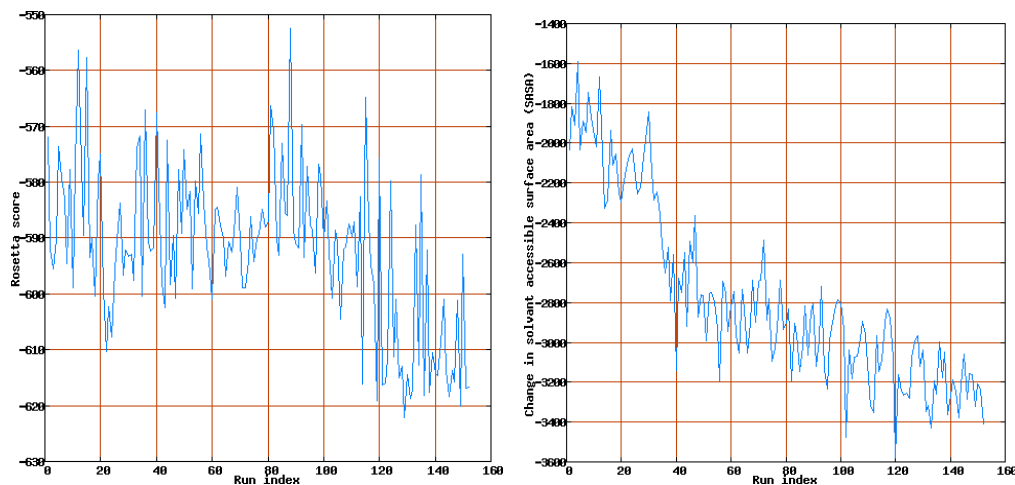


Figure 6.11: Energy graphs as shown in the web-interface.

The decrease of energy terms in the target function is shown in the left subgraph for the Rosetta-Dock score, and more clearly, in the right subgraph for the change in solvent accessible surface area (dSASA). In this example, the algorithm stopped after 250 iterations. 152 ring models were computed, 98 were dismissed due to backbone van der Waals overlaps.

mimics a binding energy. The potential to evaluate distance restraints (eq. 6.1) follows the lognormal distribution and was designed to relax restraints for values far off the desired optimum (Habeck et al., 2006, 2005). The new parameters are accepted if the MC energy decreases ($\Delta E = E_i - E_{i-1} \leq 0$). If the energy increases ($\Delta E > 0$), the algorithm accepts the new parameters only with a probability of $e^{-\beta\Delta E}$. The constant factor β can be set by the user (default value 1.0). Optionally, the main MC sampling is followed by a second sampling with linearly decreasing step-sizes to confine the search to the best solution found so far.

The calculated slope of the last 41 valid energies is used as a convergence criterion. When this slope becomes zero or positive the sampling is aborted. However, this does not guarantee to have found the global minimum, but just a local one to which the algorithm has converged.

6.3.2 Web-interface

The web-interface to the AAA+ oligomer modeling pipeline is implemented within the iMolTalk server (chapter 4). Steps of user interaction are:

1. To upload a monomer to be modeled
2. To search for abstraction of the central β -sheet
3. To choose an initial orientation from a template AAA+ structure
4. To test for van der Waals overlaps of backbone atoms in a primarily modeled ring structure

Num ↑	nmer	Score RosettaDock	Score weighted	Xrot	Yrot	Zrot	Tr	ASASA
36726	6	-616.7	-663.9	+6.3	-3.6	+1.1	-1.1	-3409.42
36717	6	-617.0	-655.4	+6.1	-3.5	+1.0	-1.0	-3236.99
36716	6	failed		+6.3	-3.6	+1.1	-1.1	0
36705	6	-592.8	-634.6	+6.2	-3.5	+1.1	-1.0	-3207.08
36689	6	-620.0	-662.2	+6.2	-3.5	+1.0	-1.1	-3322.58
36673	6	-601.1	-639.0	+6.2	-3.5	+0.9	-1.0	-3161.61
36659	6	-615.8	-650.6	+6.1	-3.8	+0.9	-1.1	-3158.47
36645	6	-613.7	-655.2	+6.2	-3.5	+1.1	-1.0	-3285.92
36644	6	failed		+6.0	-3.7	+1.0	-1.1	0
36632	6	-618.4	-647.6	+6.0	-3.8	+0.9	-1.0	-3058.6
36631	6	failed		+5.9	-3.9	+1.1	-1.0	0
36620	6	-613.8	-651.4	+6.1	-3.9	+0.8	-1.0	-3207.52
36600	6	-600.9	-649.7	+6.3	-3.9	+0.8	-1.2	-3378.9

Figure 6.12: Table of results of the ring modeling.

5. To submit the chosen initial position to the Monte-Carlo/Metropolis optimization protocol
6. To inspect modeling results on-line

The user may follow the course of the modeling and inspect plots of the energy terms in the target function (Figure 6.11) or select models to analyze from the table of results (Figure 6.12).

6.3.3 Benchmark

Two AAA+ proteins, NSF and p97, show symmetric ring structures. Monomers have been extracted from both structures, positioned on the other and optimized using the modeling procedure outlined above. The RMSD (root mean squared deviation) over the course of the modeling decreased gradually (Figure 6.13C). In both cases, a solution was found with RMSD values below 1 Å to the known ring assembly.

Remodeling of p97

The monomer of p97 (1e32) has been initially positioned with parameters derived from the ring structure of NSF (1nsf). An additional translation of 5 Å away from the central axis was necessary as the initial model showed severe van der Waals overlaps between monomers. The four runs with seed 387 did not find a valid orientation of the monomer without van der Waals overlaps within the strict limitation of 500 iterations and thus, failed to produce a model structure. The best model (seed 543 and weight E_{dA} 5%) showed an RMSD of only 0.2 Å over 1554 aligned residues to the solved ring structure (Table B.7). For this particular test case, half of the final models showed an RMSD of less than 1 Å. The other half showed an RMSD of more than 2 Å. The difference between these two populations may be explained by the ability of the MC algorithm to find a positive angle of rotation around the X axis and a translation toward the central axis to optimally close the gap between monomers. Difficulties probably arose from the hook-like structure in the C-domain which embraces the neighboring monomer and often invalidates alternative positioning of the monomers due to steric clashes. Such

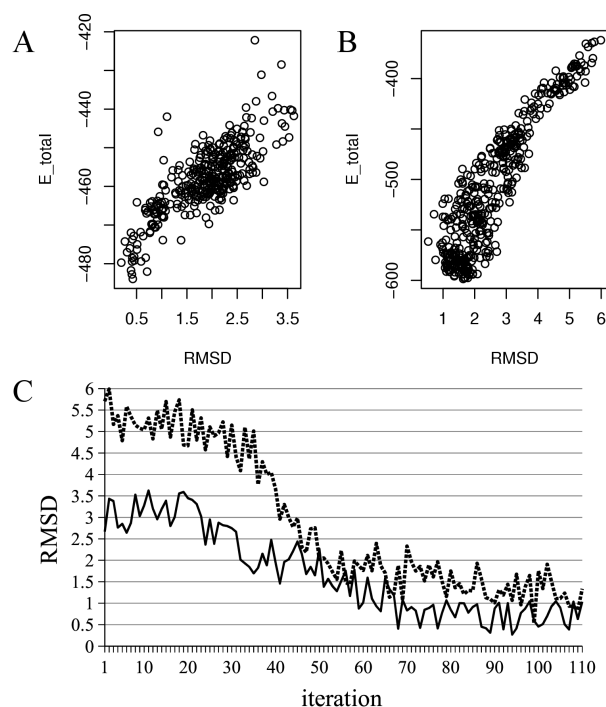


Figure 6.13: Benchmark of ring structure remodeling from single monomers.

A, The monomer from p97 (1e32) has been initially positioned on NSF and then subjected to the Monte-Carlo sampling method for optimization. **B**, Similarly, NSF (1nsf) has been positioned on p97. **C**, The RMSD values in both cases gradually decrease and the algorithm converges on distances below 1 Å (full line: p97, dotted line: NSF).

intermediate models are often needed to overcome energy barriers to find a deeper well and thus a better solution. The energy plot in Figure 6.13A indicates a good correlation between RMSD and energy values, thus low energy models show RMSD values below 1 Å. The failure of the method using random generator seed 387 highlights the dependence on the input parameters and the trade-off between computation time and convergence. To overcome this technical limitation we offer the submission of parallel modeling jobs with different seeds on our web-server.

Remodeling of NSF

Similarly, the monomer of NSF (1nsf) has been oriented with parameters derived from p97 (1e32). An additional translation of 5 Å away from the central axis increased the probability of finding a first valid docking between monomers without van der Waals overlaps. All runs succeeded in finding acceptable results and the MC algorithm in most cases converged in less than 100 iterations (Table B.8). The best model (seed 101 and weight 0 for E_{dA}) showed an RMSD of 0.5 Å over 1482 aligned residues to the solved ring structure. This solution can be discriminated from the other models by a negative angle of rotation around the Y axis and its larger angle of rotation around the X axis. Furthermore, the best model showed a positive translation away from the central axis.

These results indicate that the underlying scoring function cannot discriminate between the best solution and alternative positioning of monomers in NSF, at least at this fine resolution. The plot of energy vs. RMSD values in Figure 6.13B indicates a target RMSD value of 1.5 to 2 Å for the lowest scores.

Evaluation of parameters

The automated docking of monomers into a ring structure depends on a number of parameters. To evaluate the influence of the weighted energy terms in the scoring function, the benchmark results have been analyzed for each component separately (Figure 6.14). The RosettaDock energy alone does not correlate well with low RMSD values. Nevertheless, both E_{dA} and E_r correlate with low RMSD values and if they are included in the combined score, low energies indeed indicate good models with low RMSD values.

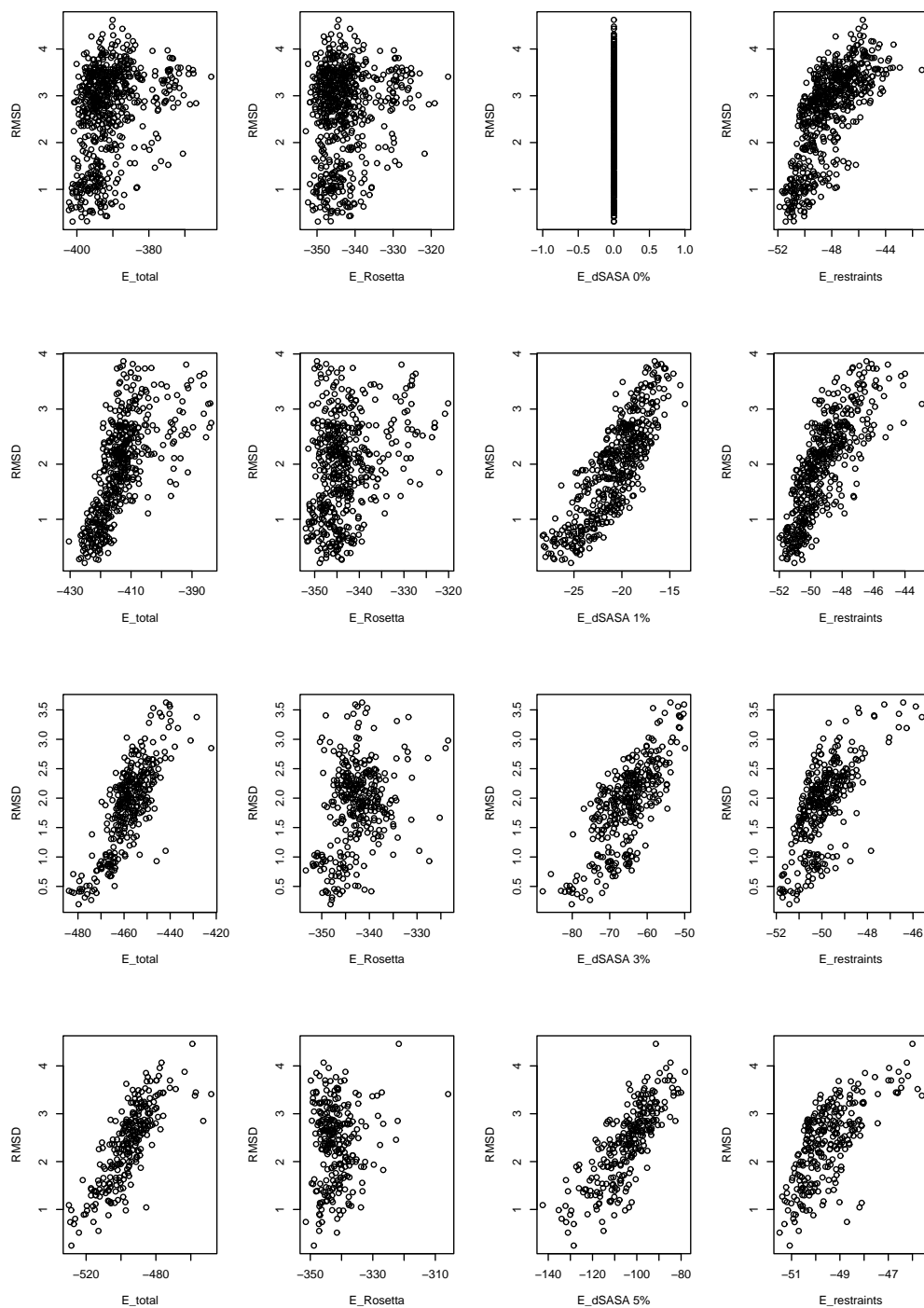


Figure 6.14: Evaluation of parameters in the automated ring modeling.

A remodeling of p97 ring structures was run with different weights (0, 0.01, 0.03, 0.05) for the change of solvent accessible surface area (E_{dA}). From left to right, the total energy is a weighted linear combination of the RosettaDock score, E_{dA} and the evaluated distance restraints. The inclusion of E_{dA} and E_r in the energy function indeed leads to a better correlation between low energies and low RMSD values.

6.4 Modeling cases

6.4.1 ClpB

Members of the Clp/Hsp100 family can interact with proteases to degrade misfolded or aggregated proteins. ClpB is remarkable in that it can rescue proteins from aggregated state (Mogk and Bukau, 2004; Lee et al., 2003b). ClpB contains a tandem AAA+ domain and is known to form hexameric double rings (Akoev et al., 2004). The two AAA+ domains (D1 and D2) show significant differences in their sequences. The sequence identity in the pairwise alignment of their nucleotide binding domains only showed 10% sequence identity. The first domain (D1) contains a 180 amino acid long insertion of a coiled coil in its C-domain. Furthermore, ATP binding in D1 is a prerequisite for oligomerization, and the rate of ATP hydrolysis is higher in the second NTPase (Watanabe et al., 2002). The non-ring forming crystal structures of ClpB (1qvr, Lee et al. (2003b)) and of its homolog ClpA (1ksf, Guo et al. (2002)) have been solved containing both AAA+ domains. EM reconstructions have indicated ring forms in both cases. Our model of a hexameric oligomer of ClpB may explain the different function of both domains and that the tandem domains are offset by one position in the double ring, contradictory to previous models where the AAA+ domains stack on each other.

For the modeling, the monomeric structure of ClpB has been split into D1 (1qvrB, 4-543) and D2 (1qvrB, 544-850). Both rings were independently optimized using the proposed modeling pipeline (section 6.3) and then re-assembled into a double doughnut structure (Fig. 6.15). The D1 monomer was initially positioned on p97 (1e32) and the selected ring model showed a RosettaDock score of -609 and a change in solvent accessible surface area (dSASA) of -4028 \AA^2 . The D2 monomer was initially positioned on p97 (1e32), rotated by 3° around Z-axis and additionally translated 2 \AA away from the central axis. The selected ring model showed a RosettaDock score of -370 and dSASA of -1894 \AA^2 . When the two rings were in register, the C-terminus of D1 was considerably closer to the N-terminus of the next D2 subunit in the lower ring than to its cognate one. Such a staggered structure for the domain arrangement in the two rings has been proposed previously for ClpA Guo et al. (2002). This observation suggests that rings offset by one subunit are a general feature of the Clp/Hsp100 family.

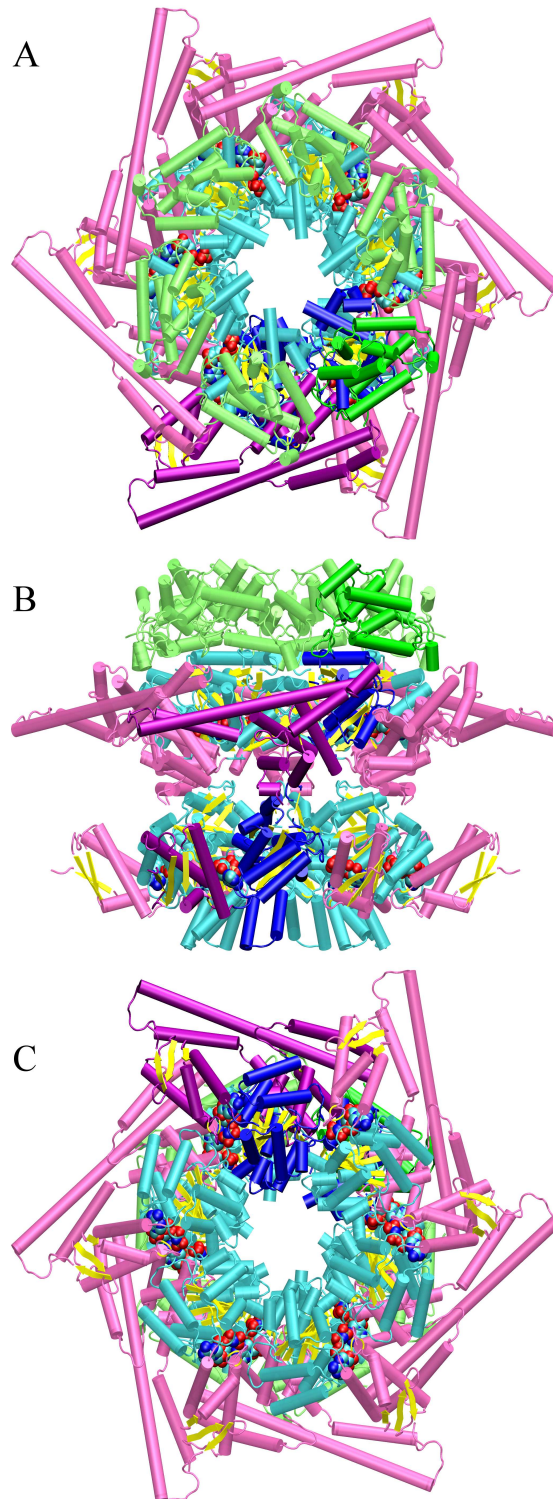


Figure 6.15: Hexamer model of ClpB tandem AAA+ domains.

The structure is shown in top view (A), side view (B) and bottom view (C). The coloring shows the N-domain in green, helices in the NTPases in cyan and in the C-domain in magenta, β -sheets in yellow. One chain is shown in bold colors to highlight the staggered assembly of D1 and D2 rings.

6.4.2 Apaf-1

Apaf-1 molecules form the apoptosome, which activates caspases, a necessary step in the initiation of apoptosis. The N-terminal CARD domain is responsible for the recruitment of pro-caspases. C-terminal to this domain, Apaf-1 contains an extended AAA+ ATPase domain, followed by a winged-helix domain, an alpha-alpha solenoid domain and two WD40-propellers. The N-terminal part, including the extended AAA+ domain, was crystallized in monomeric form (1z6t, Riedl et al. (2005)). A model for the arrangement of Apaf-1 subunits in the apoptosome was recently proposed (Acehan et al., 2002; Yu et al., 2005) by fitting the domains of the protein independently into an electron microscopic 3D reconstruction of the particle. The model placed the N-terminal CARD domain innermost to form the central ring and located the ATPase and the C-domain into an outer ring. The other domains of Apaf-1 were fitted into the spokes projecting from the rings.

We have modeled the hexameric and heptameric forms of Apaf-1 from the coordinates of 1z6t in multiple repetitions. The hexameric models show a mean difference in solvent accessible surface area of $2778 \text{ \AA}^2 (\pm 203)$ and a mean RosettaDock score of $-335.3 (\pm 2.6)$. The heptameric models had a mean change in solvent accessible surface area of $1625 \text{ \AA}^2 (\pm 233)$ and mean score $-363 (\pm 35.3)$, thus yielding less favorable results than the hexameric ones.

Superimposition of the Apaf-1 structure with other AAA+ proteins showed a small difference in the position of ATPase and C-domain relative to each other. We therefore tested whether we could improve the model structures by introducing certain flexibility at the hinge between ATPase and C-domain (section 6.2.2). The best hexamer model showed a score of -368 and a change in solvent accessible surface area of 1862 \AA^2 for a rotation around the first hinge of -0.8° ; the best heptamer model showed a score of -400 and a change in solvent accessible surface area of 2398 \AA^2 for a rotation of $+1.3^\circ$ around the second hinge. Thus, a minor amount of flexibility in domain orientation improved the heptamer model remarkably.

For comparison, automated docking of the extended AAA+ domain of Apaf-1 (1z6t, residues 105-348) using ClusPro (Comeau et al., 2004) returned 10 models if the ring stoichiometry was set to six and only one model if it was set to seven. None of the hexameric models were ring-shaped and in the heptameric model the ring was not closed.

Our heptameric model of Apaf-1 (6.16) is not compatible with the reconstruction of the apoptosome. Comparison of the model with the 3D density map indicates that the core ATPase domains form the inner ring. The outer ring is formed in alternation by the C-domains and the CARD domains. We note that in the reconstruction proposed by Yu et al. (2005), several important restraints described in section 6.2 are violated. First, there is no contact between the core ATPase domains, which form the main surface of interaction in all AAA+ proteins. Second, the position of the C-domains is far outside all observed values both relative to their cognate ATPase domains and relative to the ring, thus precluding the formation of a functional nucleotide binding site.

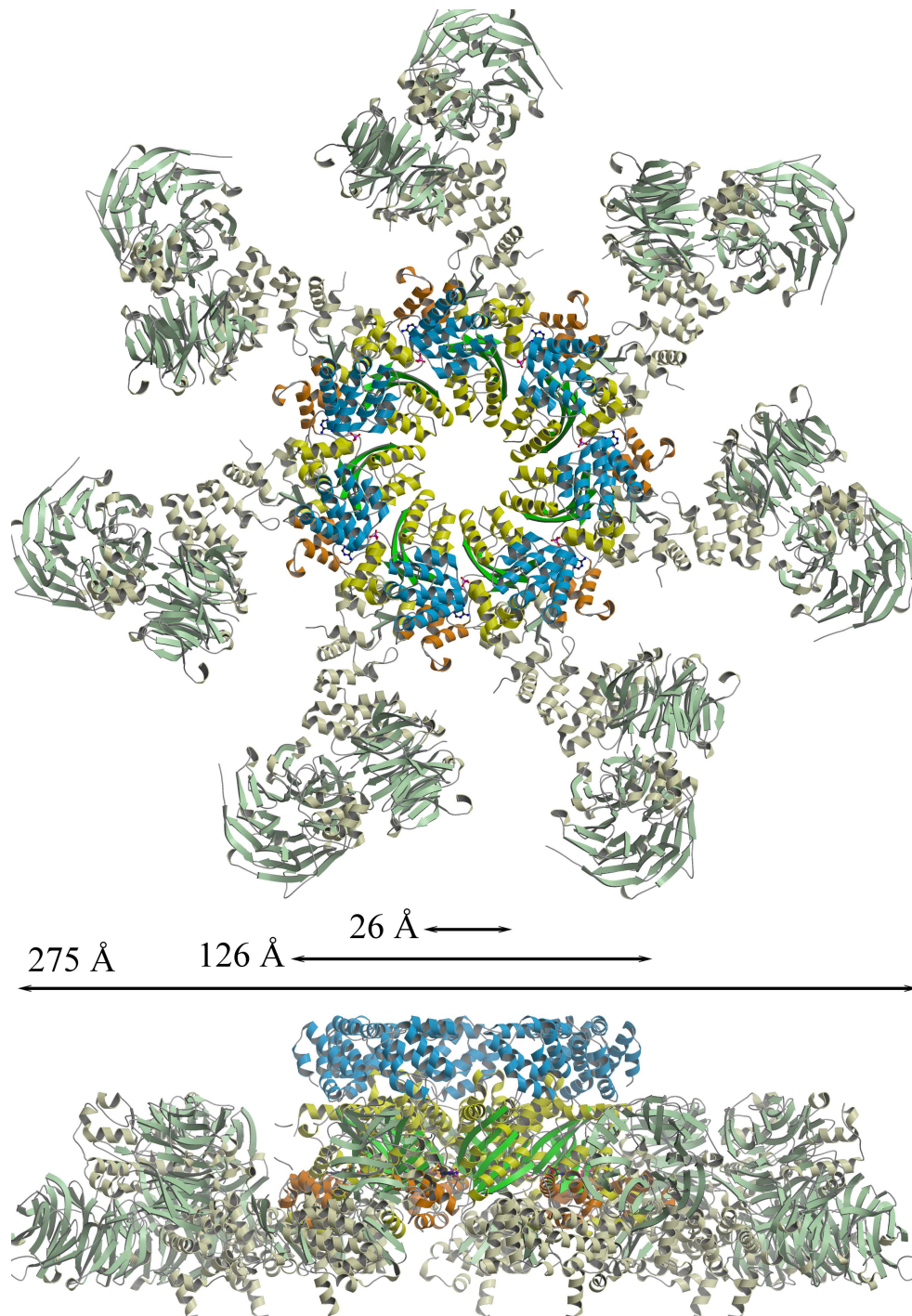


Figure 6.16: Model of the apoptosome with the AAA+ cassette placed in its center. The heptameric ring model is shown from top (the upper subgraph) and in side view below. The center, which is formed by the AAA+ ring, is highlighted (green, β -sheet; yellow, NTPase helices; orange, C-domains) and placed in the model by Yu et al. (2005). The CARD domain (blue), forming a crown-like structure, is now positioned above the ring.

```

Ma1T      4  -----PSKLSRPVRL-DHTVVBRELLAKLSGANN-FRLALITSPAGYKTTLLSQWAAGKN-----DIGWYSLD 65
Apaf-1   105 GITSYVRTVLCCEGGVQRPVVFVTRKLVNATQQKLSKLGEPGWVTIHGMAGCGKSVLAFAVARDHSLLEGCEPGGVHWVS 186

Ma1T      66  EGDNQDEEASVYLIAVQQATNGHCAICETMAQKROYASLTSLEAQLFELAEWHSPLYLVIDDYHLITNPVHESMBEETBH 148
Apaf-1   187  VGKDKSGLLVKKLQNLCTRL-----DODESFSORLPLNIEEAKDRLRLMLRKHPRSLLILDVV-----WDSVWLKAED 255

Ma1T      149  OPENLILVVLARNLPOLGIANLRVRDQLLEIGSQOLAETHQAKQFEEDCRLSSPTEAAESSRICDDVSGWATALQIALSARQNT 233
Apaf-1   256  S--QCQILLITRDKS--VTDVSMGPKYVYVPESSLGKEKLEILSLEV-NMKKADLPEQAHSTIKECKGSPLVVSLIGALLRDFP 335

```

Figure 6.17: Alignment between MalT and Apaf-1.

Arrows indicate the conserved hydrophobic position and the aromatic residue in helix H1 of the C-domain.

6.4.3 MalT

The available structure of Apaf-1 (1z6t) served as a template in modeling the AAA+ domain of the prokaryotic transcription factor MalT (Richet and Raibaud, 1989). The domain organization of this protein is from N- to C-terminus: AAA+ domain, unknown domain, TPR repeats (Steggborn et al., 2001) and HTH DNA-binding domain. A distant but clear homology between the AAA+ domains of these proteins was detected using the sensitive alignment program hhsearch (Soding, 2005) showing a pairwise sequence identity of only 13%. Nevertheless, the initial alignment showed good agreement in the conserved regions of the NTPase and was further improved using the restraints derived in this work. The conserved hydrophobic position in the first helix of the C-domain was identified and allowed to align $\beta 5$ and the C-domain (Figure 6.17). The monomer of MalT was built using MODELLER (Sali and Blundell, 1993) based on the alignment to the Apaf-1 structure. The hexameric ring model had a RosettaDock score of -165 and a change in solvent accessible surface area of 1777 \AA^2 , values which suggest a model of medium quality despite the extremely low sequence identity of target to template.

6.5 Summary and Outlook

We have developed a semi-automated modeling procedure for constructing ring-shaped complexes from the monomeric structures of AAA+ proteins. This procedure is based on a number of constraints derived from known crystal structures. The position of the core ATPase and C-domain is preserved within a narrow range and both contribute to nucleotide binding. Application of this method to a number of AAA+ proteins shows substantial improvement in subunit interactions relative to modeling by superimposition. Analyses of such models yielded new insights into the oligomeric organization of AAA+ proteins. The proposed three-dimensional structure of the apoptosome was optimized with respect to the derived structural restraints. A reorientation of the CARD domain and the AAA+ NBD in the center were satisfactory and provide a new model of the apoptosome. Further development of this method will require a more thorough exploration of possible orientations between core ATPase and C-domain, within the allowed boundaries, and more robust scoring functions for the fit of subunit interfaces.

It is envisaged that this method can readily be applied to other ring forming protein families. Furthermore, variants of this novel modeling protocol can also be developed to target specific oligomer modeling of selected protein families.

Bibliography

- Acehan, D., Jiang, X., Morgan, D., Heuser, J., Wang, X., and Akey, C. (2002). Three-dimensional structure of the apoptosome: implications for assembly, procaspase-9 binding, and activation. *Mol Cell*, 9(2):423–32.
- Akoev, V., Gogol, E., Barnett, M., and Zolkiewski, M. (2004). Nucleotide-induced switch in oligomerization of the AAA+ ATPase ClpB. *Protein Sci*, 13(3):567–74.
- Altschul, S. and Gish, W. (1996). Local alignment statistics. *Methods Enzymol*, 266:460–80.
- Altschul, S., Gish, W., Miller, W., Myers, E., and Lipman, D. (1990). Basic local alignment search tool. *J Mol Biol*, 215(3):403–10.
- Altschul, S., Madden, T., Schaffer, A., Zhang, J., Zhang, Z., Miller, W., and Lipman, D. (1997). Gapped BLAST and PSI-BLAST: a new generation of protein database search programs. *Nucleic Acids Res*, 25(17):3389–402.
- Anfinsen, C. (1973). Principles that govern the folding of protein chains. *Science*, 181(96):223–30.
- Anfinsen, C. and Scheraga, H. (1975). Experimental and theoretical aspects of protein folding. *Adv Protein Chem*, 29:205–300.
- Baker, D. and Sali, A. (2001). Protein structure prediction and structural genomics. *Science*, 294(5540):93–6.
- Baker, E. and Hubbard, R. (1984). Hydrogen bonding in globular proteins. *Prog Biophys Mol Biol*, 44(2):97–179.
- Berman, H., Westbrook, J., Feng, Z., Gilliland, G., Bhat, T., Weissig, H., Shindyalov, I., and Bourne, P. (2000). The Protein Data Bank. *Nucleic Acids Res*, 28(1):235–42.
- Bochtler, M., Hartmann, C., Song, H., Bourenkov, G., Bartunik, H., and Huber, R. (2000). The structures of HsIU and the ATP-dependent protease HsIU-HsIV. *Nature*, 403(6771):800–5.
- Botos, I., Melnikov, E., Cherry, S., Khalatova, A., Rasulova, F., Tropea, J., Maurizi, M., Rotanova, T., Gustchina, A., and Wlodawer, A. (2004). Crystal structure of the AAA+ alpha domain of E. coli Lon protease at 1.9A resolution. *J Struct Biol*, 146(1-2):113–22.

- Boutselakis, H., Dimitropoulos, D., Fillon, J., Golovin, A., Henrick, K., Hussain, A., Ionides, J., John, M., Keller, P., Krissinel, E., McNeil, P., Naim, A., Newman, R., Oldfield, T., Pineda, J., Rachedi, A., Copeland, J., Sitnov, A., Sobhany, S., Suarez-Uruena, A., Swaminathan, J., Tagari, M., Tate, J., Tromm, S., Velankar, S., and Vranken, W. (2003). E-MSD: the European Bioinformatics Institute Macromolecular Structure Database. *Nucleic Acids Res*, 31(1):458–62.
- Canutescu, A., Shelenkov, A., and Dunbrack, R. (2003). A graph-theory algorithm for rapid protein side-chain prediction. *Protein Sci*, 12(9):2001–14.
- CCP4 (1994). The CCP4 suite: programs for protein crystallography. *Acta Crystallogr D Biol Crystallogr*, 50(Pt 5):760–3.
- Chang, W., Shindyalov, I., Pu, C., and Bourne, P. (1994). Design and application of PDBlib, a C++ macromolecular class library. *Comput Appl Biosci*, 10(6):575–86.
- Chen, J., Anderson, J. B., DeWeese-Scott, C., Fedorova, N. D., Geer, L. Y., He, S., Hurwitz, D. I., Jackson, J. D., Jacobs, A. R., Lanczycki, C. J., Liebert, C. A., Liu, C., Madej, T., Marchler-Bauer, A., Marchler, G. H., Mazumder, R., Nikolskaya, A. N., Rao, B. S., Panchenko, A. R., Shoemaker, B. A., Simonyan, V., Song, J. S., Thiessen, P. A., Vasudevan, S., Wang, Y., Yamashita, R. A., Yin, J. J., and Bryant, S. H. (2003). MMDB: Entrez’s 3D-structure database. *Nucleic Acids Res*, 31(1):474–7.
- Chothia, C. and Lesk, A. (1986). The relation between the divergence of sequence and structure in proteins. *EMBO J*, 5(4):823–6.
- Clocksin, W. and Mellish, C. (1994). *Programming in Prolog*. Springer-Verlag, 4 edition.
- Codd, E. F. (1970). A relational model of data for large shared data banks. *Comm ACM*, 13(6):377–87.
- Cohen, G. H. (1997). ALIGN: a program to superimpose protein coordinates, accounting for insertions and deletions. *J Appl Cryst*, 30:1160–1.
- Comeau, S. and Camacho, C. (2005). Predicting oligomeric assemblies: N-mers a primer. *J Struct Biol*, 150(3):233–44.
- Comeau, S., Gatchell, D., Vajda, S., and Camacho, C. (2004). ClusPro: an automated docking and discrimination method for the prediction of protein complexes.p. *Bioinformatics*, 20(1):45–50.
- Coulson, A. and Moulton, J. (2002). A unfold, mesofold, and superfold model of protein fold use. *Proteins*, 46(1):61–71.
- Covington, M., Nute, D., and Vellino, A. (1997). *Prolog programming in depth*. Prentice-Hall.
- Davies, J., Tsuruta, H., May, A., and Weis, W. (2005). Conformational Changes of p97 during Nucleotide Hydrolysis Determined by Small-Angle X-Ray Scattering. *Structure*, 13(2):183–95.

- DeLaBarre, B. and Brunger, A. (2003). Complete structure of p97/valosin-containing protein reveals communication between nucleotide domains. *Nat Struct Biol*, 10(10):856–63.
- DeLaBarre, B. and Brunger, A. (2005). Nucleotide dependent motion and mechanism of action of p97/VCP. *J Mol Biol*, 347(2):437–52.
- Diemand, A. and Lupas, A. (2006). Modeling AAA+ ring complexes from monomeric structures. *J Struct Biol*, 156(1):230–43.
- Diemand, A. and Scheib, H. (2004a). iMolTalk: an interactive, internet-based protein structure analysis server. *Nucleic Acids Res*, 32(Web Server issue):W512–6.
- Diemand, A. and Scheib, H. (2004b). Moltalk - a programming library for protein structures and structure analysis. *BMC Bioinformatics*, 5(1):39.
- Dill, K. (1999). Polymer principles and protein folding. *Protein Sci*, 8(6):1166–80.
- Dinner, A., Sali, A., Smith, L., Dobson, C., and Karplus, M. (2000). Understanding protein folding via free-energy surfaces from theory and experiment. *Trends Biochem Sci*, 25(7):331–9.
- Dreveny, I., Kondo, H., Uchiyama, K., Shaw, A., Zhang, X., and Freemont, P. (2004). Structural basis of the interaction between the AAA ATPase p97/VCP and its adaptor protein p47. *EMBO J*, 23(5):1030–9.
- Erdmann, R., Wiebel, F., Flessau, A., Rytka, J., Beyer, A., Frohlich, K., and Kunau, W. (1991). Pas1, a yeast gene required for peroxisome biogenesis, encodes a member of a novel family of putative atpases. *Cell*, 64(3):499–510.
- Erzberger, J., Pirruccello, M., and Berger, J. (2002). The structure of bacterial DnaA: implications for general mechanisms underlying DNA replication initiation. *EMBO J*, 21(18):4763–73.
- Fetrow, J. and Skolnick, J. (1998). Method for prediction of protein function from sequence using the sequence-to-structure-to-function paradigm with application to glutaredoxins/thioredoxins and T1 ribonucleases. *J Mol Biol*, 281(5):949–68.
- Fiser, A., Do, R., and Sali, A. (2000). Modeling of loops in protein structures. *Protein Sci*, 9(9):1753–73.
- Fodje, M., Hansson, A., Hansson, M., Olsen, J., Gough, S., Willows, R., and Al-Karadaghi, S. (2001). Interplay between an AAA module and an integrin I domain may regulate the function of magnesium chelatase. *J Mol Biol*, 311(1):111–22.
- Frickey, T. and Lupas, A. (2004). Phylogenetic analysis of AAA proteins. *J Struct Biol*, 146(1-2):2–10.
- Frishman, D. and Argos, P. (1995). Knowledge-based protein secondary structure assignment. *Proteins*, 23(4):566–79.

- Gibrat, J., Madej, T., and Bryant, S. (1996). Surprising similarities in structure comparison. *Curr Opin Struct Biol*, 6(3):377–85.
- Gotoh, O. (1982). An improved algorithm for matching biological sequences. *J Mol Biol*, 162(3):705–8.
- Gotoh, O. (1999). Multiple sequence alignment: algorithms and applications. *Adv Biophys*, 36:159–206.
- Gray, J., Moughon, S., Wang, C., Schueler-Furman, O., Kuhlman, B., Rohl, C., and Baker, D. (2003). Protein-protein docking with simultaneous optimization of rigid-body displacement and side-chain conformations. *J Mol Biol*, 331(1):281–99.
- Gray, P., Paton, N., Kemp, G., and Fothergill, J. (1990). An object-oriented database for protein structure analysis. *Protein Eng*, 3(4):235–43.
- Greer, J. (1981). Comparative model-building of the mammalian serine proteases. *J Mol Biol*, 153(4):1027–42.
- Guex, N., Diemand, A., and Peitsch, M. (1999). Protein modelling for all. *Trends Biochem Sci*, 24(9):364–7.
- Guex, N. and Peitsch, M. (1997). SWISS-MODEL and the Swiss-PdbViewer: an environment for comparative protein modeling. *Electrophoresis*, 18(15):2714–23.
- Guo, F., Maurizi, M., Esser, L., and Xia, D. (2002). Crystal structure of ClpA, an Hsp100 chaperone and regulator of ClpAP protease. *J Biol Chem*, 277(48):46743–52.
- Habazettl, J., Gondol, D., Wiltscheck, R., Otlewski, J., Schleicher, M., and Holak, T. A. (1992). Structure of hisactophilin is similar to interleukin-1 beta and fibroblast growth factor. *Nature*, 359(6398):855–8.
- Habeck, M., Nilges, M., and Rieping, W. (2005). Replica-exchange monte carlo scheme for bayesian data analysis. *Physical Review Letters*, 94(1):18105.
- Habeck, M., Rieping, W., and Nilges, M. (2006). Weighting of experimental evidence in macromolecular structure determination. *Proc Natl Acad Sci U S A*, 103(6):1756–61.
- Henikoff, S. and Henikoff, J. (1992). Amino acid substitution matrices from protein blocks. *Proc Natl Acad Sci U S A*, 89(22):10915–9.
- Henikoff, S. and Henikoff, J. (1993). Performance evaluation of amino acid substitution matrices. *Proteins*, 17(1):49–61.
- Hinsen, K. (2000). The molecular modeling toolkit: A new approach to molecular simulations. *J. Comp. Chem.*, 21(2):79–85.
- Hinsen, K., Thomas, A., and Field, M. (1999). Analysis of domain motions in large proteins. *Proteins*, 34(3):369–382.

- Hobohm, U., Scharf, M., Schneider, R., and Sander, C. (1992). Selection of representative protein data sets. *Protein Sci*, 1(3):409–17.
- Hohenester, E. and Jansonius, J. (1994). Crystalline mitochondrial aspartate aminotransferase exists in only two conformations. *J Mol Biol*, 236(4):963–8.
- Holm, L. and Sander, C. (1993). Protein structure comparison by alignment of distance matrices. *J Mol Biol*, 233(1):123–38.
- Hooft, R., Vriend, G., Sander, C., and Abola, E. (1996). Errors in protein structures. *Nature*, 381(6580):272.
- Humphrey, W., Dalke, A., and Schulten, K. (1996). VMD – Visual Molecular Dynamics. *Journal of Molecular Graphics*, 14:33–38.
- Huysmans, M., Richelle, J., and Wodak, S. J. (1991). SESAM: a relational database for structure and sequence of macromolecules. *Proteins*, 11(1):59–76.
- Huyton, T., Pye, V., Briggs, L., Flynn, T., Beuron, F., Kondo, H., Ma, J., Zhang, X., and Freemont, P. (2003). The crystal structure of murine p97/VCP at 3.6Å. *J Struct Biol*, 144(3):337–48.
- Islam, S. and Sternberg, M. (1989). A relational database of protein structures designed for flexible enquiries about conformation. *Protein Eng*, 2(6):431–42.
- IUBMB - International Union of Biochemistry and Molecular Biology (1992). *Enzyme Nomenclature*. Academic Press.
- Jansonius, J. (1998). Structure, evolution and action of vitamin B6-dependent enzymes. *Curr Opin Struct Biol*, 8(6):759–69.
- John, B. and Sali, A. (2003). Comparative protein structure modeling by iterative alignment, model building and model assessment. *Nucleic Acids Res*, 31(14):3982–92.
- Karplus, K., Barrett, C., Cline, M., Diekhans, M., Grate, L., and Hughey, R. (1999). Predicting protein structure using only sequence information. *Proteins*, Suppl 3:121–5.
- Kearsley, S. K. (1989). On the orthogonal transformation used for structural comparison. *Acta Cryst A*, A(45):208–210.
- Kendrew, J. (1970). IUPAC-IUB Commission on Biochemical Nomenclature. Abbreviations and symbols for the description of the conformation of polypeptide chains. *J Mol Biol*, 52(1):1–17.
- Kendrew, J., Dickerson, R., Strandberg, B., Hart, R., Davies, D., Phillips, D., and Shore, V. (1960). Structure of Myoglobin. *Nature*, 185:422–427.
- Kim, D. and Kim, K. (2003). Crystal structure of ClpX molecular chaperone from *Helicobacter pylori*. *J Biol Chem*, 278(50):50664–70.

- Kleywegt, G. (1999). Recognition of spatial motifs in protein structures. *J Mol Biol*, 285(4):1887–97.
- Kleywegt, G. (2000). Validation of protein crystal structures. *Acta Crystallogr D Biol Crystallogr*, 56 (Pt 3):249–65.
- Kleywegt, G. and Jones, T. (1996). Phi/psi-chology: Ramachandran revisited. *Structure*, 4(12):1395–400.
- Kleywegt, G. J. and Jones, A. T. (1997). Detecting Folding Motifs and Similarities in Protein Structures. *Meth Enzymol*, 277:525–45.
- Kohlbacher, O. and Lenhof, H. (2000). Ball–rapid software prototyping in computational molecular biology. biochemicals algorithms library. *Bioinformatics*, 16(9):815–24.
- Koretke, K., Russell, R., Copley, R., and Lupas, A. (1999). Fold recognition using sequence and secondary structure information. *Proteins*, Suppl 3:141–8.
- Koretke, K., Russell, R., and Lupas, A. (2001). Fold recognition from sequence comparisons. *Proteins*, 45 Suppl 5:68–75.
- Krissinel, E. and Henrick, K. (2004). Secondary-structure matching (SSM), a new tool for fast protein structure alignment in three dimensions. *Acta Crystallogr D Biol Crystallogr*, 60(Pt 12 Pt 1):2256–68.
- Krissinel, E., Winn, M., Ballard, C., Ashton, A., Patel, P., Potterton, E., McNicholas, S., Cowtan, K., and Emsley, P. (2004). The new CCP4 Coordinate Library as a toolkit for the design of coordinate-related applications in protein crystallography. *Acta Crystallogr D Biol Crystallogr*, 60(Pt 12 Pt 1):2250–5.
- Krzywda, S., Brzozowski, A., Verma, C., Karata, K., Ogura, T., and Wilkinson, A. (2002). The crystal structure of the AAA domain of the ATP-dependent protease FtsH of *Escherichia coli* at 1.5 Å resolution. *Structure*, 10(8):1073–83.
- Kwon, A., Kessler, B., Overkleeft, H., and McKay, D. (2003). Structure and reactivity of an asymmetric complex between HslV and I-domain deleted HslU, a prokaryotic homolog of the eukaryotic proteasome. *J Mol Biol*, 330(2):185–95.
- Laskowski, R., MacArthur, M., Moss, D., and Thornton, J. (1993). Procheck: a program to check the stereochemical quality of protein structures. *J. Appl. Cryst.*, 26:283–91.
- Lee, S., De La Torre, A., Yan, D., Kustu, S., Nixon, B., and Wemmer, D. (2003a). Regulation of the transcriptional activator NtrC1: structural studies of the regulatory and AAA+ ATPase domains. *Genes Dev*, 17(20):2552–63.
- Lee, S., Sowa, M., Watanabe, Y., Sigler, P., Chiu, W., Yoshida, M., and Tsai, F. (2003b). The structure of ClpB: a molecular chaperone that rescues proteins from an aggregated state. *Cell*, 115(2):229–40.

- Lenzen, C., Steinmann, D., Whiteheart, S., and Weis, W. (1998). Crystal structure of the hexamerization domain of N-ethylmaleimide-sensitive fusion protein. *Cell*, 94(4):525–36.
- Levitt, M. (1997). Competitive assessment of protein fold recognition and alignment accuracy. *Proteins*, Suppl 1:92–104.
- Liu, J., Smith, C., DeRyckere, D., DeAngelis, K., Martin, G., and Berger, J. (2000). Structure and function of Cdc6/Cdc18: implications for origin recognition and checkpoint control. *Mol Cell*, 6(3):637–48.
- Lovell, S., Word, J., Richardson, J., and Richardson, D. (2000). The penultimate rotamer library. *Proteins*, 40(3):389–408.
- Lupas, A. and Martin, J. (2002). AAA proteins. *Curr Opin Struct Biol*, 12(6):746–53.
- McPhalen, C., Vincent, M., Picot, D., Jansonius, J., Lesk, A. M., and Chothia, C. (1992). Domain closure in mitochondrial aspartate aminotransferase. *J Mol Biol*, 227(1):197–213.
- Melo, F. and Feytmans, E. (1997). Novel knowledge-based mean force potential at atomic level. *J Mol Biol*, 267(1):207–22.
- Merritt, E. A. (1999). Expanding the model: anisotropic displacement parameters in protein structure refinement. *Acta Crystallogr D Biol Crystallogr*, 55(Pt 6):1109–17.
- Metropolis, N., Rosenbluth, A., Rosenbluth, M., Teller, A., and Teller, E. (1953). Equation of state calculations by fast computing machines. *J Chem Phys*, 21(6):1087–92.
- Mogk, A. and Bukau, B. (2004). Molecular chaperones: structure of a protein disaggregase. *Curr Biol*, 14(2):R78–80.
- Moll, A., Hildebrandt, A., Lenhof, H. P., and Kohlbacher, O. (2006). BALLView: a tool for research and education in molecular modeling. *Bioinformatics*, 22(3):365–6.
- Moreland, J., Gramada, A., Buzko, O., Zhang, Q., and Bourne, P. (2005). The molecular biology toolkit (mbt): a modular platform for developing molecular visualization applications. *BMC Bioinformatics*, 6(1):21.
- Morris, A., MacArthur, M., Hutchinson, E., and Thornton, J. (1992). Stereochemical quality of protein structure coordinates. *Proteins*, 12(4):345–64.
- Murzin, A. (1998). How far divergent evolution goes in proteins. *Curr Opin Struct Biol*, 8(3):380–7.
- Murzin, A. G., Lesk, A. M., and Chothia, C. (1992). beta-Trefoil fold. Patterns of structure and sequence in the Kunitz inhibitors interleukins-1 beta and 1 alpha and fibroblast growth factors. *J Mol Biol*, 223(2):531–43.
- Needleman, S. and Wunsch, C. (1970). A general method applicable to the search for similarities in the amino acid sequence of two proteins. *J Mol Biol*, 48(3):443–53.

- Neuwald, A., Aravind, L., Spouge, J., and Koonin, E. (1999). AAA+: A class of chaperone-like ATPases associated with the assembly, operation, and disassembly of protein complexes. *Genome Res*, 9(1):27–43.
- Nishikawa, K. and Ooi, T. (1974). Comparison of homologous tertiary structures of proteins. *J Theor Biol*, 43(2):351–74.
- Niwa, H., Tsuchiya, D., Makyio, H., Yoshida, M., and Morikawa, K. (2002). Hexameric ring structure of the ATPase domain of the membrane-integrated metalloprotease FtsH from *Thermus thermophilus* HB8. *Structure*, 10(10):1415–23.
- Nussinov, R. and Wolfson, H. (1991). Efficient detection of three-dimensional structural motifs in biological macromolecules by computer vision techniques. *Proc Natl Acad Sci U S A*, 88(23):10495–9.
- Ogura, T., Whiteheart, S., and Wilkinson, A. (2004). Conserved arginine residues implicated in ATP hydrolysis, nucleotide-sensing, and inter-subunit interactions in AAA and AAA+ ATPases. *J Struct Biol*, 146(1-2):106–12.
- Orengo, C., Jones, D., and Thornton, J. (1994). Protein superfamilies and domain superfolds. *Nature*, 372(6507):631–4.
- Ortiz, A., Strauss, C., and Olmea, O. (2002). MAMMOTH (matching molecular models obtained from theory): an automated method for model comparison. *Protein Sci*, 11(11):2606–21.
- Painter, J. and Merritt, E. (2004). mmLib python toolkit for manipulating annotated structural models of biological macromolecules. *J Appl Cryst*, 37(1):174–8.
- Perutz, M., Kendrew, J., and Watson, H. (1965). Structure and Function of Haemoglobin - II. Some relations between polypeptide chain configuration and amino acid sequence. *J Mol Biol*, 13:669–678.
- Perutz, M., Rossmann, M., Cullis, A., Muirhead, H., Will, G., and North, A. (1960). Structure of Haemoglobin. *Nature*, 185:416–422.
- Petitjean, M. (1998). Interactive Maximal Common 3D Substructure Searching with the Combined SDM/RMS Algorithm. *Computers Chem*, 22(6):463–465.
- Phillips, D. (1970). The development of crystallographic enzymology. *Biochem Soc Symp*, 30:11–28.
- Ponting, C. P. and Russell, R. B. (2000). Identification of distant homologues of fibroblast growth factors suggests a common ancestor for all beta-trefoil proteins. *J Mol Biol*, 302(5):1041–7.
- Poole, D., Mackworth, A., and Goebel, R. (1998). *Computational intelligence: a logical approach*. Oxford University Press New York.

- Putnam, C., Clancy, S., Tsuruta, H., Gonzalez, S., Wetmur, J., and Tainer, J. (2001). Structure and mechanism of the RuvB Holliday junction branch migration motor. *J Mol Biol*, 311(2):297–310.
- Pye, V., Dreveny, I., Briggs, L., Sands, C., Beuron, F., Zhang, X., and Freemont, P. (2006). Going through the motions: The atpase cycle of p97. *J Struct Biol*, in Press.
- Quinlan, J. (1986). Induction of decision trees. *Machine Learning*, 1(1):81–106.
- Ramachandran, G., Ramakrishnan, C., and Sasisekharan, V. (1963). Stereochemistry of polypeptide chain configurations. *J Mol Biol*, 7:95–9.
- Richardson, D. and Richardson, J. (1992). The kinemage: a tool for scientific communication. *Protein Sci*, 1(1):3–9.
- Richardson, J. (1981). The anatomy and taxonomy of protein structure. *Adv Protein Chem*, 34:167–339.
- Richet, E. and Raibaud, O. (1989). MalT, the regulatory protein of the Escherichia coli maltose system, is an ATP-dependent transcriptional activator. *EMBO J*, 8(3):981–7.
- Riedl, S., Li, W., Chao, Y., Schwarzenbacher, R., and Shi, Y. (2005). Structure of the apoptotic protease-activating factor 1 bound to ADP. *Nature*, 434(7035):926–33.
- Rossmann, M. and Liljas, A. (1974). Letter: Recognition of structural domains in globular proteins. *J Mol Biol*, 85(1):177–81.
- Russel, S. J. and Norvig, P. (1995). *Artificial Intelligence A Modern Approach*. Prentice-Hall.
- Sali, A. and Blundell, T. (1993). Comparative protein modelling by satisfaction of spatial restraints. *J Mol Biol*, 234(3):779–815.
- Sallai, L. and Tucker, P. (2005). Crystal structure of the central and C-terminal domain of the sigma(54)-activator ZraR. *J Struct Biol*, 151(2):160–70.
- Sanchez, R. and Sali, A. (1999). ModBase: a database of comparative protein structure models. *Bioinformatics*, 15(12):1060–1.
- Satou, K., Furuichi, E., Takiguchi, K., Takagi, T., and Kuhara, S. (1993). A deductive database system pacade for analyzing 3-d and secondary structures of protein. *Comput Appl Biosci*, 9(3):259–65.
- Sawaya, M. and Kraut, J. (1997). Loop and subdomain movements in the mechanism of Escherichia coli dihydrofolate reductase: crystallographic evidence. *Biochemistry*, 36(3):586–603.
- Sayle, R. and Milner-White, E. (1995). RASMOL: biomolecular graphics for all. *Trends Biochem Sci*, 20(9):374.

- Schneidman-Duhovny, D., Inbar, Y., Nussinov, R., and Wolfson, H. (2005). Geometry-based flexible and symmetric protein docking. *Proteins*, 60(2):224–31.
- Schwarz, H. R. (1997). *Numerische Mathematik*. Teubner, 4. edition.
- Scott, A., Chung, H., Gonciarz-Swiatek, M., Hill, G., Whitby, F., Gaspar, J., Holton, J., Viswanathan, R., Ghaffarian, S., Hill, C., and Sundquist, W. (2005). Structural and mechanistic studies of VPS4 proteins. *EMBO J*, 24(20):3658–69.
- Shannon, C. and Weaver, W. (1949). *The Mathematical Theory of Information*. Urbana: University of Illinois Press, 97.
- Shindyalov, I. and Bourne, P. (1998). Protein structure alignment by incremental combinatorial extension (CE) of the optimal path. *Protein Eng*, 11(9):739–47.
- Singleton, M., Morales, R., Grainge, I., Cook, N., Isupov, M., and Wigley, D. (2004). Conformational changes induced by nucleotide binding in Cdc6/ORC from *Aeropyrum pernix*. *J Mol Biol*, 343(3):547–57.
- Smart, O., Goodfellow, J., and Wallace, B. (1993). The pore dimensions of gramicidin A. *Biophys J*, 65(6):2455–60.
- Smith, T. and Waterman, M. (1981). Identification of common molecular subsequences. *J Mol Biol*, 147(1):195–7.
- Soding, J. (2005). Protein homology detection by HMM-HMM comparison. *Bioinformatics*, 21(7):951–60.
- Sokal, R. and Michener, C. (1958). A statistical method for evaluating systematic relationships. *Univ Kansas Sci Bull*, 28:1409–38.
- Song, H., Hartmann, C., Ramachandran, R., Bochtler, M., Behrendt, R., Moroder, L., and Huber, R. (2000). Mutational studies on HslU and its docking mode with HslV. *Proc Natl Acad Sci U S A*, 97(26):14103–8.
- Sousa, M., Kessler, B., Overkleeft, H., and McKay, D. (2002). Crystal structure of HslUV complexed with a vinyl sulfone inhibitor: corroboration of a proposed mechanism of allosteric activation of HslV by HslU. *J Mol Biol*, 318(3):779–85.
- Steggborn, C., Danot, O., Huber, R., and Clausen, T. (2001). Crystal structure of transcription factor MalT domain III: a novel helix repeat fold implicated in regulated oligomerization. *Structure*, 9(11):1051–60.
- Stickle, D., Presta, L., Dill, K., and Rose, G. (1992). Hydrogen bonding in globular proteins. *J Mol Biol*, 226(4):1143–59.
- Swaffield, J., Bromberg, J., and Johnston, S. (1992). Alterations in a yeast protein resembling HIV Tat-binding protein relieve requirement for an acidic activation domain in GAL4. *Nature*, 357(6380):698–700.

- Trame, C. and McKay, D. (2001). Structure of Haemophilus influenzae HslU protein in crystals with one-dimensional disorder twinning. *Acta Crystallogr D Biol Crystallogr*, 57(Pt 8):1079–90.
- Venter, J., Adams, M., Myers, E., Li, P., Mural, R., Sutton, G., Smith, H., Yandell, M., Evans, C., Holt, R., Gocayne, J., Amanatides, P., Ballew, R., Huson, D., Wortman, J., Zhang, Q., Kodira, C., Zheng, X., Chen, L., Skupski, M., Subramanian, G., Thomas, P., Zhang, J., Gabor Miklos, G., Nelson, C., Broder, S., Clark, A., Nadeau, J., McKusick, V., Zinder, N., Levine, A., Roberts, R., Simon, M., Slayman, C., Hunkapiller, M., Bolanos, R., Delcher, A., Dew, I., Fasulo, D., Flanigan, M., Florea, L., Halpern, A., Hannenhalli, S., Kravitz, S., Levy, S., Mobarry, C., Reinert, K., Remington, K., Abu-Threideh, J., Beasley, E., Biddick, K., Bonazzi, V., Brandon, R., Cargill, M., Chandramouliswaran, I., Charlab, R., Chaturvedi, K., Deng, Z., Di Francesco, V., Dunn, P., Eilbeck, K., Evangelista, C., Gabrielian, A., Gan, W., Ge, W., Gong, F., Gu, Z., Guan, P., Heiman, T., Higgins, M., Ji, R., Ke, Z., Ketchum, K., Lai, Z., Lei, Y., Li, Z., Li, J., Liang, Y., Lin, X., Lu, F., Merkulov, G., Milshina, N., Moore, H., Naik, A., Narayan, V., Neelam, B., Nusskern, D., Rusch, D., Salzberg, S., Shao, W., Shue, B., Sun, J., Wang, Z., Wang, A., Wang, X., Wang, J., Wei, M., Wides, R., Xiao, C., Yan, C., Yao, A., Ye, J., Zhan, M., Zhang, W., Zhang, H., Zhao, Q., Zheng, L., Zhong, F., Zhong, W., Zhu, S., Zhao, S., Gilbert, D., Baumhueter, S., Spier, G., Carter, C., Cravchik, A., Woodage, T., Ali, F., An, H., Awe, A., Baldwin, D., Baden, H., Barnstead, M., Barrow, I., Beeson, K., Busam, D., Carver, A., Center, A., Cheng, M., Curry, L., Danaher, S., Davenport, L., Desilets, R., Dietz, S., Dodson, K., Doup, L., Ferriera, S., Garg, N., Gluecksmann, A., Hart, B., Haynes, J., Haynes, C., Heiner, C., Hladun, S., Hostin, D., Houck, J., Howland, T., Ibegwam, C., Johnson, J., Kalush, F., Kline, L., Koduru, S., Love, A., Mann, F., May, D., McCawley, S., McIntosh, T., McMullen, I., Moy, M., Moy, L., Murphy, B., Nelson, K., Pfannkoch, C., Pratts, E., Puri, V., Qureshi, H., Reardon, M., Rodriguez, R., Rogers, Y., Romblad, D., Ruhfel, B., Scott, R., Sitter, C., Smallwood, M., Stewart, E., Strong, R., Suh, E., Thomas, R., Tint, N., Tse, S., Vech, C., Wang, G., Wetter, J., Williams, S., Williams, M., Windsor, S., Winn-Deen, E., Wolfe, K., Zaveri, J., Zaveri, K., Abril, J., Guigo, R., Campbell, M., Sjolander, K., Karlak, B., Kejariwal, A., Mi, H., Lazareva, B., Hatton, T., Narechania, A., Diemer, K., Muruganujan, A., Guo, N., Sato, S., Bafna, V., Istrail, S., Lippert, R., Schwartz, R., Walenz, B., Yooshep, S., Allen, D., Basu, A., Baxendale, J., Blick, L., Caminha, M., Carnes-Stine, J., Caulk, P., Chiang, Y., Coyne, M., Dahlke, C., Mays, A., Dombroski, M., Donnelly, M., Ely, D., Esparham, S., Fosler, C., Gire, H., Glanowski, S., Glasser, K., Glodek, A., Gorokhov, M., Graham, K., Gropman, B., Harris, M., Heil, J., Henderson, S., Hoover, J., Jennings, D., Jordan, C., Jordan, J., Kasha, J., Kagan, L., Kraft, C., Levitsky, A., Lewis, M., Liu, X., Lopez, J., Ma, D., Majoros, W., McDaniel, J., Murphy, S., Newman, M., Nguyen, T., Nguyen, N., Nodell, M., Pan, S., Peck, J., Peterson, M., Rowe, W., Sanders, R., Scott, J., Simpson, M., Smith, T., Sprague, A., Stockwell, T., Turner, R., Venter, E., Wang, M., Wen, M., Wu, D., Wu, M., Xia, A., Zandieh, A., and Zhu, X. (2001). The sequence of the human genome. *Science*, 291(5507):1304–51.
- Vitkup, D., Melamud, E., Moulton, J., and Sander, C. (2001). Completeness in structural genomics. *Nat Struct Biol*, 8(6):559–66.

- Vriend, G. (1990). WHAT IF: a molecular modeling and drug design program. *J Mol Graph*, 8(1):52–6, 29.
- Walker, J., Saraste, M., Runswick, M., and Gay, N. (1982). Distantly related sequences in the alpha- and beta-subunits of atp synthase, myosin, kinases and other atp-requiring enzymes and a common nucleotide binding fold. *EMBO J*, 1(8):945–51.
- Wang, C., Schueler-Furman, O., and Baker, D. (2005). Improved side-chain modeling for protein-protein docking. *Protein Sci*, 14(5):1328–39.
- Wang, J., Song, J., Franklin, M., Kamtekar, S., Im, Y., Rho, S., Seong, I., Lee, C., Chung, C., and Eom, S. (2001a). Crystal structures of the HsIVU peptidase-ATPase complex reveal an ATP-dependent proteolysis mechanism. *Structure*, 9(2):177–84.
- Wang, J., Song, J., Seong, I., Franklin, M., Kamtekar, S., Eom, S., and Chung, C. (2001b). Nucleotide-dependent conformational changes in a protease-associated ATPase HsIU. *Structure*, 9(11):1107–16.
- Wang, Q., Song, C., and Li, C. (2003a). Hexamerization of p97-VCP is promoted by ATP binding to the D1 domain and required for ATPase and biological activities. *Biochem Biophys Res Commun*, 300(2):253–60.
- Wang, Q., Song, C., Yang, X., and Li, C. (2003b). D1 ring is stable and nucleotide-independent, whereas D2 ring undergoes major conformational changes during the ATPase cycle of p97-VCP. *J Biol Chem*, 278(35):32784–93.
- Watanabe, Y., Motohashi, K., and Yoshida, M. (2002). Roles of the two ATP binding sites of ClpB from *Thermus thermophilus*. *J Biol Chem*, 277(8):5804–9.
- Weiss, M. (1995). *Algorithms, data structures, and problem solving with C+*. Addison-Wesley.
- Wilmot, C. and Thornton, J. (1990). Beta-turns and their distortions: a proposed new nomenclature. *Protein Eng*, 3(6):479–93.
- Xia, D., Esser, L., Singh, S., Guo, F., and Maurizi, M. (2004). Crystallographic investigation of peptide binding sites in the N-domain of the ClpA chaperone. *J Struct Biol*, 146(1-2):166–79.
- Yamada, K., Kunishima, N., Mayanagi, K., Ohnishi, T., Nishino, T., Iwasaki, H., Shinagawa, H., and Morikawa, K. (2001). Crystal structure of the Holliday junction migration motor protein RuvB from *Thermus thermophilus* HB8. *Proc Natl Acad Sci U S A*, 98(4):1442–7.
- Yamada, K., Miyata, T., Tsuchiya, D., Oyama, T., Fujiwara, Y., Ohnishi, T., Iwasaki, H., Shinagawa, H., Ariyoshi, M., Mayanagi, K., and Morikawa, K. (2002). Crystal structure of the RuvA-RuvB complex: a structural basis for the Holliday junction migrating motor machinery. *Mol Cell*, 10(3):671–81.

- Yan, N., Chai, J., Lee, E., Gu, L., Liu, Q., He, J., Wu, J., Kokel, D., Li, H., Hao, Q., Xue, D., and Shi, Y. (2005). Structure of the ced-4-ced-9 complex provides insights into programmed cell death in *caenorhabditis elegans*. *Nature*, 437(7060):831–7.
- Yip, Y., Scheib, H., Diemand, A., Gattiker, A., Famiglietti, L., Gasteiger, E., and Bairoch, A. (2004). The Swiss-Prot variant page and the ModSNP database: a resource for sequence and structure information on human protein variants. *Hum Mutat*, 23(5):464–70.
- Yu, R., Hanson, P., Jahn, R., and Brunger, A. (1998). Structure of the ATP-dependent oligomerization domain of N-ethylmaleimide sensitive factor complexed with ATP. *Nat Struct Biol*, 5(9):803–11.
- Yu, X., Acehan, D., Menetret, J., Booth, C., Ludtke, S., Riedl, S., Shi, Y., Wang, X., and Akey, C. (2005). A structure of the human apoptosome at 12.8 Å resolution provides insights into this cell death platform. *Structure*, 13(11):1725–35.
- Zhang, X., Shaw, A., Bates, P., Newman, R., Gowen, B., Orlova, E., Gorman, M., Kondo, H., Dokurno, P., Lally, J., Leonard, G., Meyer, H., van Heel, M., and Freemont, P. (2000). Structure of the AAA ATPase p97. *Mol Cell*, 6(6):1473–84.

Appendix A

Protein structures by EC families

Table A.1: Protein structures by EC families.

EC		subgroups	structures	chains
1.	Oxidoreductases	348	3167	7537
1.1.	Acting on the CH-OH group of donors	95	703	1492
1.2.	Acting on the aldehyde or oxo group of donors	27	212	605
1.3.	Acting on the CH-CH group of donors	32	175	458
1.4.	Acting on the CH-NH(2) group of donors	21	160	422
1.5.	Acting on the CH-NH group of donors	22	209	362
1.6.	Acting on NADH or NADPH	23	148	321
1.7.	Acting on other nitrogenous compounds as donors	16	113	274
1.8.	Acting on a sulfur group of donors	15	76	173
1.9.	Acting on a heme group of donors	2	32	199
1.10.	Acting on diphenols and related substances as donors	5	47	294
1.11.	Acting on a peroxide as acceptor (peroxidases)	13	322	580
1.12.	Acting on hydrogen as donor	3	21	58
1.13.	Acting on single donors with incorporation of molecular oxygen	22	114	393
1.14.	Acting on paired donors, with incorporation or reduction of molecular oxygen	33	451	907
1.15.	Acting on superoxide as acceptor	2	156	441
1.16.	Oxidizing metal ions	3	4	5
1.17.	Acting on CH or CH(2) groups	3	77	140
1.18.	Acting on iron-sulfur proteins as donors	6	89	245
1.19.	Acting on reduced flavodoxin as donor	0	0	0

EC		subgroups	structures	chains
1.20.	Acting on phosphorus or arsenic in donors	1	12	12
1.21.	Acting on x-H and y-H to form an x-y bond	1	10	10
1.97.	Other oxidoreductases	2	10	77
2.	Transferases	354	4520	8958
2.1.	Transferring one-carbon groups	56	524	985
2.2.	Transferring aldehyde or ketone residues	3	34	78
2.3.	Acyltransferases	52	381	840
2.4.	Glycosyltransferases	57	715	1433
2.5.	Transferring alkyl or aryl groups, other than methyl groups	26	407	1032
2.6.	Transferring nitrogenous groups	19	208	387
2.7.	Transferring phosphorous-containing groups	128	2195	4088
2.8.	Transferring sulfur-containing groups	13	62	115
2.9.	Transferring selenium-containing groups	0	0	0
3.	Hydrolases	487	7599	12526
3.1.	Acting on ester bonds	112	1846	2989
3.2.	Glycosylases	74	2131	2908
3.3.	Acting on ether bonds	5	32	72
3.4.	Acting on peptide bonds (peptide hydrolases)	180	2534	4212
3.7.	Acting on carbon-carbon bonds	5	18	26
3.8.	Acting on halide bonds	5	41	51
3.9.	Acting on phosphorus-nitrogen bonds	1	1	4
3.10.	Acting on sulfur-nitrogen bonds	1	1	1
3.11.	Acting on carbon-phosphorus bonds	2	3	8
3.12.	Acting on sulfur-sulfur bonds	0	0	0
3.13.	Acting on carbon-sulfur bonds	1	4	6
4.	Lyases	163	1384	3397
4.1.	Carbon-carbon lyases	70	492	1464
4.2.	Carbon-oxygen lyases	56	695	1421
4.3.	Carbon-nitrogen lyases	13	56	154
4.4.	Carbon-sulfur lyases	12	44	92
4.5.	Carbon-halide lyases	1	2	24
4.6.	Phosphorus-oxygen lyases	7	72	212
4.99.	Other lyases	3	13	16
5.	Isomerases	85	829	1573
5.1.	Racemases and epimerases	25	135	285
5.2.	Cis-trans-isomerases	2	124	206
5.3.	Intramolecular oxidoreductases	26	325	609
5.4.	Intramolecular transferases (mutases)	20	144	298
5.5.	Intramolecular lyases	8	37	79
5.99.	Other isomerases	3	61	88

EC		subgroups	structures	chains
6.	Ligases	68	571	1103
6.1.	Forming carbon-oxygen bonds	21	196	308
6.2.	Forming carbon-sulfur bonds	5	21	46
6.3.	Forming carbon-nitrogen bonds	34	313	663
6.4.	Forming carbon-carbon bonds	3	21	55
6.5.	Forming phosphoric ester bonds	5	20	31
6.6.	Forming nitrogen-metal bonds	0	0	0

Appendix B

AAA+ ring modeling data

Table B.1: Key residues in AAA+ structures I.

Protein	p97	VPS4	FtsH	HslU	NSF	NSF	RuvB
Domain	D1				D2	D2	
PDB code	1e32	1xwi	1lv7	1kyi	1nsf	1d2n	1in4
bound nucleotide	ADP			ATP	ATP	ANP	ADP
Walker A lysine	K251	K180	K201	K63	K549	K557	K64
Walker B aspartate	D304	D234	K254	D257	D603	D611	D109
Sensor 1	N348	N279	N301	A310	S647	S655	T158
Arg-Finger	R359	R290	R312	R326	N659	N667	R170
Sensor 2	-	-	-	R394	K708	K716	R217
C-dom. hydrophob.	I380	M309	I333	I344	A679	A687	I188
aromatic	H384	H313	H337	H397(H3)	-	-	-
P1	G240	G169	G190	N52	S538	S546	H53
P2	I301	I231	I251	V254	V600	V608	L106
P3	A346	A277	A299	S308	T645	T653	A156
Hinge	I371	L300	L324	A335	N669	N677	F179

Table B.2: Key residues in AAA+ structures II.

Protein	ClpA	ClpA	ClpB	ClpB	ClpX	ZraR	NtrC1	Apaf-1	MalT
Domain	D1	D2	D1	D2					
PDB code	1r6b	1r6b	1qvr	1qvr	1um8	1ojl	1ny6	1z6t	model
bound nucleotide	ADP	ADP	ANP	ANP	ADP	ATP	ADP	ADP	
Walker A lysine	K220	K501	K204	K601	K155	K175	K173	K160	K39
Walker B aspartate	D285	D564	D270	D667	D214	D240	D238	D243	D158
Sensor 1	T323	N606	T307	N709	A280	H282	N280	R265	R157
Arg-Finger	R339	R643	R322	R747	R333	R301	R299	-	R168
Sensor 2	-	R702	-	R806	R396	R359	R357	-	T217
C-dom. hydrophob.	I357	V661	I340	I765	I351	L322	L320	I294	A188
aromatic	-	F665	-	-	-	F326	F324	F298	F191
P1	N209	S490	N193	S590	N144	T164	P162	W149	L31
P2	L282	L561	L267	I664	V211	L237	L235	L240	L122
P3	S321	T604	A305	T707	A278	A280	A278	T263	L155
Hinge	E348	H652	E331	P756	S342	S311	P309	S285	A188

Table B.3: Reconstructed AAA+ ring structures.

Subfam.	PDB	iMolTalk code	mer	date	res.	aa	nucl.	ref.
p97	1e32	1e32_hexamer	6	2000-06-05	2.9	438	ADP	Zhang et al., 2000
	1oz4	1oz4_hexamer	6	2003-04-07	4.7	698	ADP	DeLaBarre and Brunger, 2003
	1r7r	1r7r_hexamer	6	2003-10-22	3.6	683	ADP	Huyton et al., 2003
	1s3s	1s3s_hexamer	6	2004-01-14	2.9	436	ADP	Dreveny et al., 2004
	1yq0	1yq0_hexamer	6	2005-01-31	4.5	698	ADP	DeLaBarre and Brunger, 2005
	1yqi	1yqi_hexamer	6	2005-02-01	4.25	705	ADP	DeLaBarre and Brunger, 2005
NSF	1d2n	1d2n_hexamer	6	1998-06-30	1.75	246	ANP	Lenzen et al., 1998
	1nsf	1nsf_hexamer	6	1998-06-26	1.9	247	ATP	Yu et al., 1998
HslU	1e94	1e94_hexamer	6	2000-10-07	2.8	408	ANP	Song et al., 2000
	1do0	1do0_hexamer	6	1999-12-18	3	406	ATP	Bochtler et al., 2000
	1do2	1do21_hexamer	6	1999-12-18	4	407	ANP	Bochtler et al., 2000
		1do22_hexamer	6			407	ANP	
	1g3i	1g3i1_hexamer	6	2000-10-24	3.41	317	ATP	Trame and McKay, 2001
		1g3i2_hexamer	6			312	ATP	
	1g41	1g41_hexamer	6	2000-10-25	2.3	334	ADP	Trame and McKay, 2001
	1g4a	1g4a_hexamer	6	2000-10-26	3	356	DAT	Wang et al., 2001a
	1g4b	1g4b1_hexamer	6	2000-10-26	7	393		Wang et al., 2001a
		1g4b2_hexamer	6			393		
	1hqy	1hqy_hexamer	6	2000-12-20	2.8	408	ADP	Wang et al., 2001b
	1ht1	1ht11_hexamer	6	2000-12-27	2.8	408	ADP	Wang et al., 2001b
1ht12_hexamer		6			408	ADP		
1ht2	1ht21_hexamer	6	2000-12-27	2.8	408	ADP	Wang et al., 2001b	
	1ht22_hexamer	6			408	ADP		
1im2	1im2_hexamer	6	2001-05-09	2.8	346	ADP	Trame and McKay, 2001	
1kyi	1kyi1_hexamer	6	2002-02-04	3.1	321	ATP	Sousa et al., 2002	
	1kyi2_hexamer	6			317	ATP		
1ofh	1ofh_hexamer	6	2003-04-14	2.5	309	ADP	Kwon et al., 2003	
1ofi	1ofi1_hexamer	6	2003-04-14	3.2	299	ADP	Kwon et al., 2003	
	1ofi2_hexamer	6			295	ADP		
σ 54 ac-tivators	1ny6	1ny61_heptamer	7	2003-02-11	3.1	243	ADP	Lee et al., 2003a
		1ny62_heptamer	7			245	ADP	
	1ojl	1ojl1_hexamer	6	2003-07-10	3	251		Sallai and Tucker, 2005
		1ojl2_hexamer	6			247	ATP	

Table B.4: Non-ring forming, experimentally determined AAA+ structures.

Subfamily	PDB	date	res.	aa	nucl.	ref.
FtsH	1ixz	2002-07-10	2.2	238		Niwa et al., 2002
	1iy0	2002-07-10	2.95	240	ANP	Niwa et al., 2002
	1iy1	2002-07-10	2.8	234	ADP	Niwa et al., 2002
	1iy2	2002-07-10	3.2	245		Niwa et al., 2002
	1lv7	2002-05-26	1.5	251		Krzywda et al., 2002
VPS4	1xwi	2004-11-01	2.8	322		Scott et al., 2005
ClpA	1ksf	2002-01-12	2.6	714	ADP	Guo et al., 2002
	1r6b	2003-10-15	2.25	704	ADP	Xia et al., 2004
ClpB	1qvr	2003-08-28	3	803		Lee et al., 2003b
ClpX	1um8	2003-09-25	2.6	327	ADP	Kim and Kim, 2003
RuvB	1in4	2001-05-12	1.6	298	ADP	Putnam et al., 2001
	1in5	2001-05-12	2	301	ADP	Putnam et al., 2001
	1in6	2001-05-12	1.8	300	ADP	Putnam et al., 2001
	1in7	2001-05-12	1.9	298	ADP	Putnam et al., 2001
	1in8	2001-05-12	1.9	298	ADP	Putnam et al., 2001
	1j7k	2001-05-16	1.8	299	ATP	Putnam et al., 2001
	1ixr	2002-07-04	3.3	308	ANP	Yamada et al., 2002
	1ixs	2002-07-04	3.2	315	ANP	Yamada et al., 2002
	1hqc	2000-12-15	3.2	314	ADE	Yamada et al., 2001
Apaf-1	1z6t	2005-03-23	2.21	576	ADP	Riedl et al., 2005
CED-4	2a5y	2005-07-01	2.6	373	ATP	Yan et al., 2005
Cdc6	1fnn	2000-08-22	2	379	ADP	Liu et al., 2000
Orc2	1w5s	2004-08-09	2.4	390	ADP	Singleton et al., 2004
	1w5t	2004-08-09	2.4	394	ADP	Singleton et al., 2004
DnaA	1l8q	2002-03-21	2.7	321	ADP	Erzberger et al., 2002
Bchi	1g8p	2000-11-20	2.1	321		Fodje et al., 2001

Table B.5: Distances describing the relative orientation of AAA+ nucleotide binding domain vs. C-domain.

		WalkerA		h.phob.	Sensor2	distances [Å]						
str.	nucl.	Lysine	Hinge	(H1)	(H3)	<i>AB</i>	<i>AC</i>	<i>BC</i>	<i>AD</i>	<i>BD</i>		
		<i>A</i>	<i>C</i>	<i>B</i>	<i>D</i>							
Cdc48	1e32A	ADP	K251	I371	I380	A409	14.0	11.5	11.4	10.7	9.6	
HslU	1kyiA	ATP	K63	A335	I344	A393	14.5	11.0	13.1	11.3	8.2	
FtsH	1lv7A	(SO4)	K201	L324	I333	A362	15.2	10.9	11.2	10.8	9.7	
ClpA D1	1r6bX	ADP	K220	E348	I357	D396	14.5	10.2	12.2	11.0	10.3	
ClpA D2	1r6bX	ADP	K501	H652	V661	R702	14.2	11.2	13.1	11.5	10.8	
ClpB D1	1qvrA	ANP	K204	E331	I340	D379	13.7	9.8	12.2	10.8	10.5	
ClpB D2	1qvrA	ANP	K601	P756	I765	R806	14.1	11.5	13.3	11.7	10.9	
ClpX	1um8A	ADP	K155	S342	I351	R396	14.7	11.0	12.9	12.7	10.6	
RuvB	1in4A	ADP	K64	F179	I188	R217	15.7	12.1	13.3	11.2	9.8	
ZraR	1ojlE	ATP	K175	S311	L322	R359	15.4	10.6	13.0	12.4	11.3	
NtrC1	1ny6A	ADP	K173	P309	L320	R357	15.3	10.8	13.1	11.2	10.5	
Apaf-1	1z6tA	ADP	K160	S285	I294	L322	16.4	11.6	13.7	11.7	10.2	
VPS4	1xwiA	(SO4)	K180	L300	M309	A339	14.9	11.0	12.2	11.0	9.8	
							mean:	14.8	11.0	12.7	11.4	10.2
							stdev:	0.7	0.6	0.7	0.6	0.7
NSF	1nsf	ATP	K549	N669 ?	A679	K708	16.4	10.0	11.6	9.6	11.1	
	1d2nA	ANP	K557	N677 ?	A687	K716	16.3	10.1	11.5	9.7	11.1	
							mean:	16.3	10.0	11.6	9.6	11.1

Table B.6: Angular parameters describing the relative orientation of AAA+ nucleotide binding domain vs. C-domain.

str.	nucl.	WalkerA		h.phob.	Sensor2	angles [°]		dihedral	
		Lysine A	Hinge C	(H1) B	(H3) D	ACB	ACD	angle[°] ACBD	
Cdc48	1e32A	ADP	K251	I371	I380	A409	75.6	57.9	60.2
HslU	1kyiA	ATP	K63	A335	I344	A393	73.6	66.2	72.3
FtsH	1lv7A	(SO4)	K201	L324	I333	A362	86.7	62.0	57.4
ClpA D1	1r6bX	ADP	K220	E348	I357	D396	80.2	66.5	68.0
ClpA D2	1r6bX	ADP	K501	H652	V661	R702	71.0	56.7	62.1
ClpB D1	1qvrA	ANP	K204	E331	I340	D379	76.4	67.5	71.8
ClpB D2	1qvrA	ANP	K601	P756	I765	R806	69.1	58.6	65.9
ClpX	1um8A	ADP	K155	S342	I351	R396	75.4	62.9	66.6
RuvB	1in4A	ADP	K64	F179	I188	R217	76.2	57.1	57.3
ZraR	1ojlE	ATP	K175	S311	L322	R359	80.7	68.3	70.0
NtrC1	1ny6A	ADP	K173	P309	L320	R357	79.4	63.5	64.4
Apaf-1	1z6tA	ADP	K160	S285	I294	L322	80.6	62.1	60.4
VPS4	1xwiA	(SO4)	K180	L300	M309	A339	79.5	62.8	63.4
						mean:	77.3	62.5	64.6
						stdev:	4.4	3.8	4.9
NSF	1nsf	ATP	K549	N669 ?	A679	K708	98.0	59.5	49.7
	1d2nA	ANP	K557	N677 ?	A687	K716	98.0	59.6	50.0
						mean:	98.0	59.5	49.8

Table B.7: Benchmark of the ring modeling pipeline: remodeling of p97.

seed	wE_{dA}	iter.	conv.	score	rot_X	rot_Y	rot_Z	tr.	$\Delta SASA$	RMSD
101	0	215	70	-401	-0.5	8.3	2.3	1.2	-1862	2.2
	1	114	65	-430	4.3	6.8	0.3	-0.6	-2723	0.6
	3	73	n/a	-484	4.4	7.3	0.2	-0.6	-2767	0.4
	5	56	n/a	-506	-5.2	5.3	1.0	0.2	-2245	2.9
543	0	156	55	-402	3.3	8.6	0.3	0.0	-2264	0.6
	1	151	70	-427	4.2	8.1	-0.5	-0.7	-2706	0.3
	3	14	n/a	-470	-1.1	6.0	-0.1	0.4	-2426	0.8
	5	70	n/a	-528	4.0	8.3	-0.2	-0.4	-2570	0.2
790	0	233	52	-397	-6.1	5.2	2.2	1.1	-1825	3.5
	1	156	90	-418	-4.8	7.2	2.4	1.3	-1922	3.1
	3	10	n/a	-466	-3.0	6.4	0.5	0.0	-2372	2.2
	5	75	n/a	-511	-2.6	6.0	-0.2	0.8	-2355	2.4

Table B.8: Benchmark of the ring modeling pipeline: remodeling of NSF.

seed	wE_{dA}	iter.	conv.	score	rot_X	rot_Y	rot_Z	tr.	$\Delta SASA$	RMSD
101	0	277	95	-432	9.4	-2.2	1.3	0.3	-3079	0.5
	1	342	93	-473	5.9	3.2	1.0	-0.1	-3092	1.8
	3	284	93	-543	6.0	3.9	1.0	-0.3	-3166	1.9
	5	264	165	-601	6.7	2.3	0.9	-0.7	-3369	1.9
387	0	341	93	-439	6.1	1.5	1.6	0.3	-2836	1.3
	1	294	97	-471	6.0	3.3	1.4	0.0	-3104	1.7
	3	280	91	-532	6.3	3.4	1.2	-0.4	-3228	1.9
	5	268	110	-597	6.3	2.2	0.4	-0.4	-3194	1.8
543	0	361	94	-440	6.7	2.5	1.8	0.2	-2985	1.3
	1	327	102	-471	6.7	2.8	0.9	-0.2	-3173	1.6
	3	149	116	-537	6.2	4.2	0.8	-0.5	-3358	2.1
	5	243	96	-602	6.9	1.7	0.5	-0.7	-3431	1.9
790	0	395	n/a	-403	0.7	3.0	-5.9	1.7	-1628	3.8
	1	327	n/a	-468	6.8	1.3	1.5	0.1	-3034	1.2
	3	332	188	-529	6.0	4.1	0.2	-0.3	-3157	2.1
	5	211	148	-604	6.9	3.3	1.1	-0.2	-3291	1.7

Appendix C

Curriculum Vitae

Alexander Vasil Diemand

Born January 21st, 1971

Married, two children

Swiss citizen

Address: Brunnenstrasse 40, 72116 Mössingen

Telephone: +49 (0)7473 95 14 77

Email: axeld@moltalk.org

German: mother tongue

English and French: reading, writing and speaking fluently

Publications

Diemand, A., Lupas, A. (2006) "Modeling AAA+ ring complexes from monomeric structures" *J. Struct. Biol.*, in press.

Diemand, A., Scheib, H. (2004) "iMolTalk: an interactive, internet-based protein structure analysis server" *Nucl. Acid Res.*, 32, W512-516.

Yip, L., Scheib, H., Diemand, A., Gattiker, A., Famiglietti, L.M., Gasteiger, E., Bairoch, A. (2004) "The Swiss-Prot variant page and the ModSNP database: a resource for sequence and structure information on human variants" *Human Mutation*, 23, 464-470.

Diemand, A., Scheib, H. (2004) "MolTalk - a programming library for protein structures and structure analysis" *BMC Bioinformatics*, 5, 39.

Peitsch, M., Schwede, T., Diemand, A., Guex, N. (2003) "Protein homology modelling." in: *The Encyclopaedia of the Human Genome*, Cooper, D.N. and Lockyer, A., eds., Nature Publishing Group.

Schwede, T., Diemand, A., Guex, N., Peitsch, M. (2000) "Protein structure computing in the genomic area" *Res. Microbiol.*, 151:2, 107-112.

Guex, N., Diemand, A., Peitsch, M. (1999) "Protein Modelling for all" *Trends Biochem Sci.*, 24:9, 364-367.

Distinctions and prizes

- 2004 Web-award of the Molecular Graphics and Modeling Society, German section, for the iMolTalk server at the 18th Darmstaedter Modelling Workshop, Erlangen.
- 2000 Vontobel prize from ETH Zürich for best diploma thesis of the department of agronomy.

Conferences

- 2005 6th International Conference on AAA Proteins, Graz, Austria
Poster presentation: “Modeling of AAA+ ring structures”.
- 2004 18th Darmstaedter Molecular Modelling Workshop, Erlangen, Germany
Talk entitled: “iMolTalk - an interactive, internet-based Protein Structure Analysis Server”.
- 2004 University of Geneva, Switzerland
January: general presentation of iMolTalk; March: workshop on scripting capabilities of MolTalk through the iMolTalk server.
- 2003 Seminar presentation entitled “Cell-cycle control by kinase complexes” at group of Jean-Michel Claverie, CNRS, Marseille.
- 2002 CASP5 Critical Assessment on Structure Prediction, Assilomar CA, USA
Poster presentation: “Non-homology based search space extension in the template selection step of protein homology modeling”.
- 2001 9th International Conference on Intelligent Systems for Molecular Biology, Copenhagen
Poster presentation: “Attempt to improve template selection in protein homology modeling using logical feature descriptors”.
- 2000 10th International Conference on Inductive Logic Programming, London.

Academic work

- 2000 Diploma thesis in collaboration with Dr. Nicolas Guex, Glaxo Wellcome and Prof. Gerald Stranzinger, Federal Institute of Technology, Zürich “Exploring new ways of improving sequence alignments used for comparative protein modelling techniques”
Honored with the “Vontobel Price 2000” of the Federal Institute of Technology, Zürich.
- 1999 Semester project in collaboration with Dr. Nicolas Guex, Glaxo Wellcome and Prof. Christian Pellegrini, University of Geneva: Porting of Swiss PDB Viewer from the platform Mac to Unix (SGI, Linux) and implementation of the graphics in 2D and 3D using OpenGL.

1998 Semester project at the institute of “Nutztierwissenschaften”, Federal Institute of Technology, Zürich: Virtual breeding program written in C++.

Education

- 2000-01 Studies in informatics (predoctoral school) at the Ecole Polytechnique Fédérale, Lausanne
(Object-Oriented Technologies, Modern Approaches to Neural Network Theory, Foundations of Programming)
- 1998-99 Studies in informatics at the University of Geneva:
(Artificial Intelligence, Compilers and Interpreters, Bioinformatics)
- 1993-00 Studies in agronomy at the Federal Institute of Technology, Zürich
Specialization in biotechnology (10 semesters) to obtain a Master degree of science
- 1993 Maturité Fédérale Suisse, Bern
- 1989-93 Preparation for the Maturité Fédérale Suisse, by correspondence, with AKAD, Zürich
- 1987-90 Commercial apprenticeship with NCR (Suisse), Wallisellen

Personal experiences and knowledge

Informatics

Programming and system administration under Unix, Mac, Windows
Analysis and design of object-oriented software systems
Programming in logic, object-oriented and procedural computer languages
Knowledge of search algorithms, data structures and object design patterns
Database realization and interfacing with SQL

Others

Profound knowledge in biology, chemistry and physics as well as in economy and statistics
Experience in molecular biology laboratory work and techniques in molecular genetics

Work experience

- 2005- In the Department of Protein Evolution, Max-Planck-Institute, Tuebingen
- 2000-02 Pre-doctorate scientist in the Protein Bioinformatics Department, GlaxoSmithKline, Geneva
- 1998/99 Internship for 4 months at Glaxo Wellcome, Geneva
Protein homology modeling, programming in C with OpenGL
- 1998 Internship for 3 months at Genedata AG, Basel
Sequence databases, gene clustering, web interfaces, Java applets
- 1997-98 Independent consultant, realization of Internet projects: concept, programming, implementation
Installation and maintenance of a server dedicated to Internet services.
- 1990-93 POS Systems AG, Wallisellen
Programming of cash register systems (client-server architecture)

Interests

Strong interest in computer science, especially in computer languages and computational logic, and structural bioinformatics, especially in protein structure analysis and homology modeling, to better understand the interdependencies between evolution, biochemical function and structure.