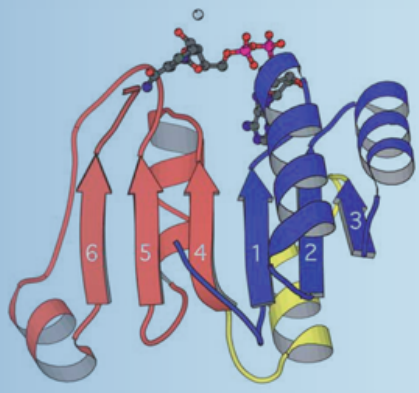
Methods in Molecular Biology 932

Springer Protocols



Alexander E. Kister *Editor*

Protein Supersecondary Structures

Methods and Protocols

Humana Press

METHODS IN MOLECULAR BIOLOGY™

Series Editor
John M. Walker
School of Life Sciences
University of Hertfordshire
Hatfield, Hertfordshire, AL10 9AB, UK

For further volumes:
http://www.springer.com/series/7651

Protein Supersecondary Structures

Edited by

Alexander E. Kister

Rutgers University, Piscataway, NJ, USA

Humana Press

Editor
Alexander E. Kister
Rutgers University,
Piscataway, NJ, USA

ISSN 1064-3745 ISSN 1940-6029 (electronic)
ISBN 978-1-62703-064-9 ISBN 978-1-62703-065-6 (eBook)
DOI 10.1007/978-1-62703-065-6
Springer New York Heidelberg Dordrecht London

Library of Congress Control Number: 2012944373

Printed on acid-free paper

Humana Press is a brand of Springer
Springer is part of Springer Science+Business Media (www.springer.com)

Preface

Structural and functional properties of proteins are convenient to consider in terms of structural hierarchy. Linderstrøm-Lang suggested three levels of protein organization in 1952 [1]. *Primary structure* is the amino acid sequence in a polypeptide chain. *Tertiary structure* is the three-dimensional structure of a protein represented by the atomic coordinates of its residues. Definition of *secondary structure* requires a short preface about how polypeptide chains fold to form a 3D structure.

Folded polypeptide chain may be divided into "regular" and "irregular" parts. Regular regions form helix-like fragments, which are defined by protein backbone torsion angles with specified periodic values. Different torsion angle values correspond to two main patterns of hydrogen bonds. Hydrogen bonding along the spiral axis results in formation of α-helix, 3_{10}-helix, or π-helix. Hydrogen bonding directed orthogonally to the spiral axis yields beta-strand configuration. The orthogonally oriented hydrogen bonds are insufficient to stabilize individual beta strands. Rather they provide stability through contacts between two or more beta strands. A set of hydrogen-bonded beta strands is referred to as a "beta sheet." The regular parts are connected to each other by the irregular parts, called "turns" and "loops." The regular and irregular regions can be regarded as the structural units of a protein structure. *Secondary structure* refers to the sequence of regular and irregular peptide fragments (alpha helices, beta strands, loops, etc.) that comprise 3D protein structure and to localization of these fragments in amino acid sequence.

Examination of protein structures revealed a number of structurally conserved arrangements of strands and helices. The finding of structural invariants is, perhaps, not unexpected, since it is difficult to imagine that an evolutionary process would yield countless number of disparate architectural motives. The famous mathematician, Professor Israel Gelfand, who studied protein structure for many years, drew an analogy between a construction company engaged in mass building of houses and synthesis of proteins within a cell (also a form of mass production). Construction company will not survive in a competitive market if it does not use standard designs. Different combinations of standard designs allow for construction of buildings with various features and functions quickly and efficiently. It appears that a similar principle is at work with regard to protein production as well.

Knowledge of conserved combinations of secondary structure elements is very important for understanding of the general principles of protein folding and function. Therefore, Michael Rossmann introduced in 1973 a new level in protein structure classification—*supersecondary structure* (SSS), which emphasizes the conserved combinations of secondary structure elements [2]. In the first chapter of this volume, Rossmann outlines the main considerations that led him to the formulation of the new concept. An example of a supersecondary structure, a combination of beta strands and alpha helices known as "Rossmann fold," appears on the cover of this volume.

Hierarchically, SSS is intermediate between secondary and tertiary structure. SSS may be used to describe either part of a protein domain or the whole protein. For example, combination of several secondary structure elements that make up "Zinc finger" consists of two beta strands with an alpha helix [3], while "Greek Key" [4] and "interlock" [5] are

composed of two distinct arrangements of four beta strands. SSS that describes all strands and helices in a protein is a schematic representation of a protein structure and is often referred to as "protein motif." It is important to note that proteins with the same SSS may have different functions, confirming the principle of economy in protein architecture.

SSS helps to understand the relationship between primary and tertiary structure of proteins. It was shown, for example, that proteins with an identical SSS share a unique set of crucial residues even when they belong to different families and have very little sequence homology [6]. This finding implies that protein motif may be determined by residue content at a few critical positions, whose localization can be deduced from the protein's SSS. Thus, the relation between protein sequence and structure is reciprocal: not only amino acid sequence determines protein structure, as was pointed out by Anfinsen over 50 years ago, but knowledge of supersecondary structure yields information about primary sequence as well.

Concept of SSS has been very fruitful for structural biology of proteins. The aim of this volume is to illustrate the usefulness of the study of SSS in different areas of protein research. The volume comprises 18 reviews written by experts in the field on different aspects of SSS. The reviews can be broadly organized around four main topics: SSS Representation; SSS Prediction; SSS and Protein Folding; and Other Applications of SSS Concept to Protein Biology.

Part I: Representation of Supersecondary Structure

There are several techniques for representing SSS. Koch and her colleagues use graphs to visualize SSS (Chapter 2). The vertices of the graphs stand for α-helices and β-strands, while the edges illustrate spatial relationships between secondary structure elements. Graphic representations of different proteins are collected in Protein Topology Graph Library database, a useful supplement to PDB and PDBsum that enriches our understanding of protein SSS.

In Chapter 3, Thirup and his colleagues suggest to supplement SSS definition with information about lengths of secondary structural elements and hydrogen bonds involved in SSS formation. The authors illustrate the advantages of the expanded definition for classification of beta-propeller proteins. The beta-propeller proteins have a fascinating architecture and their proper structural classification is especially important since they have been implicated in the pathogenesis of Alzheimer's disease, Huntington's disease, and many other conditions. Better understanding of these protein structures could prove very important for protein engineering and rational drug design.

Konagurthu and Lesk suggest representing interactions of helices and strands in SSS as a square symmetric matrix or tableau (Chapter 4). The tableau representation of SSS is shown to be sensitive and effective for identifying even distantly related proteins. The authors further formulate and address an important question in the field of protein structure prediction: "How little do we need

to know to specify a protein fold?" In other words: if SSS of only a part of a protein domain is known, is this sufficient to identify SSS of the whole structure? It is clear that prediction of the whole structure based on the knowledge of its part is only possible if we understand the rules that dictate arrangement of secondary structural elements in protein structures.

Part II: Supersecondary Structure Prediction

In Chapter 5, Chen and Kurgan present a very detailed review of current tools of secondary and supersecondary structure prediction. The authors provide helpful commentary and practical advice for the users of these prediction tools.

In Chapter 6, by Martin and his colleagues survey prediction tools based on machine learning. A massive amount of data on protein sequences and inability at present to find reasonable correlation between amino acid sequence and structures make the method of machine learning very popular in molecular biology.

In Chapter 7, Rockovsky considers one of the most intriguing problems in the study of SSS: deducing protein architecture from amino acid sequence. The main idea underlying his approach is the use of Fourier analysis to estimate characteristics of a protein sequence by taking into account properties of its constituent residues. Each residue in a sequence is characterized by a number of physical and structural properties such as hydrophobicity, preference to be located in helix or loop, etc. Amino acid sequence may therefore be represented as a series of strings, where each string characterizes a particular property of the residues in a sequence. Since amino acid sequences are presented in a series of numerical strings, they can be Fourier-transformed. The author provides detailed information about the use of software for carrying out Fourier analysis of a protein sequence. Determining total property of a protein from the properties of its constituent residues is an original and promising approach to structure prediction.

One of the major obstacles to predicting protein structure from amino acid sequence based on energy considerations is the huge number of possible structures to be evaluated from which optimal candidate is to be selected. Crivelli and Max discuss how to sharply reduce the number of starting possibilities by taking advantage of SSS considerations (Chapter 8). Molecular manipulation tool "BuildBeta" can quickly generate all possible arrangements of beta strands into beta sheets. These hypothetical SSS can be used as an input into free energy minimization algorithm to output optimal (lowest free energy) SSS. Successful application of this combinatorial approach to a number of most difficult targets in Critical Assessment of Techniques for Protein Structure Prediction (CASP)

indicates that this method can be used successfully to construct reasonable models of large proteins.

Protein structures may be represented as being composed of different supersecondary building units. Fernandez-Fuentes and Fiser in Chapter 9 consider the simplest such units composed of two strands or helices. They classified and carefully analyzed the frequency of occurrences of all variants of these basic SSS units in different types of protein folds. Itemization of all standard SSS building units opens new possibilities to structure modeling and design.

Knowledge of the interactions between secondary structure elements that stabilize SSS is invaluable for structure prediction. Sobolev, Edelman, and their colleagues developed an automatic tool for detailed analysis of all contacts between residues, secondary structure elements, and SSS (Chapter 10). This publically available software is a very useful tool for furthering our understanding of structure formation. The authors showed that application of the software to wide variety of sandwich-like proteins revealed that about half of both intra- and inter-domain interactions are conserved.

Part III: Supersecondary Structure and Protein Folding

A crucially important area of molecular biology concerns the study of how polypeptide chain folds into a stable, functional, three-dimensional structure. Understanding of this phenomenon may open the doors to protein engineering and rational drug design for the many "protein misfolding diseases." The mechanism of folding is presently not completely understood with respect to its driving forces and relative role of individual amino acids. Most of chapters in this volume, to a greater or lesser extent, relate to the folding problem.

The concept of SSS has proved to be extremely useful for investigation of protein folding. At first approximation, the folding process may be represented as formation of SSS from several secondary structure elements. This SSS serves as a core and mostly defines motif of the entire structure. The folding process is completed when other strands and helices assemble around the core forming a protein domain. Efimov uses this approach to model folding of SSS for several proteins, and to deduce the rules that determine how strands and helices are then added to core SSS structure to form the complete domain (Chapter 11).

Molecular dynamic simulation is a widely used method for analysis of structural alteration during protein folding. However, application of this method to large proteins with vast conformational space places such a high demand on computer resources as to make it practically impossible. The concept of SSS may be used to simplify dynamic simulations for large proteins. In Chapter 12,

Gerstman and Chapagain describe how computational folding simulations can be used for analysis of kinetics of SSS formation. Importantly, this analysis can also disclose the effect of residue substitution on SSS stability.

Folding of small- and medium-size proteins is usually described as a two-state kinetic process. In this model ensembles of states are divided by free-energy barriers. Disadvantage of the model is that a mechanism of folding cannot be determined experimentally [7]. Consequently, results of folding tests cannot be used in computational folding calculations. Another folding model, the so-called downhill folding, describes protein folding without free energy barrier. This model was suggested by Muñoz and his colleagues and is considered in Chapter 13. Advantage of downhill folding is that the intermediate structures between the denatured and native states may be potentially detectable by experiment. Nuclear magnetic resonance spectroscopy method was used for thermodynamic analysis of folding mechanisms of small proteins. Experiments showed this approach to be perfectly suited for analysis of microsecond folding kinetic processes. Authors describe in detail publically available Web applications for atom-by-atom analysis in folding process.

Another original approach to understand the principles of protein SSS formation is through analysis of SSS formation of nonprotein molecules, "foldamers." Foldamer is a polymer, but not a polypeptide, whose tertiary structure consists of elements analogous to elements of protein secondary structure—helices, sheets, and loops. Hence the structure of a foldamer can be described in terms of protein SSS. The validity of the analogy between proteins and foldamers is reinforced by the observation that these nonprotein molecules can have typical protein functions and properties such as catalysis, presence of specific binding sites, which direct flow of electrons, and others. This indirect approach to a protein folding promises to be very effective because it allows one to focus on principles, which are general not only for proteins but for nonprotein SSS formation as well. In Chapter 14, Hu and Chen consider design, chemical synthesis, and structural studies of SSS of aromatic oligoamide foldamers.

Part IV: Other Applications of Supersecondary Structure to Protein Biology

Failure of proteins to fold correctly may result in protein malfunction and disease. If misfolded proteins escape degradation they tend to form stable aggregates, which can injure and kill cells. Many different disorders, including Alzheimer's and Parkinson's disease and amyloid cardiomyopathy, are linked to aggregation of misfolded proteins in brain, heart, and other tissues. Many of these diseases are associated with proteins that are normally soluble, but under certain circum-

stances change their structure and form amyloid insoluble beta sheet aggregates. Understanding the pathological process of amyloidogenesis will be crucial for developing rational therapies for numerous diseases. In Chapter 15, Sabaté and Ventura review the most widely used methods for identification of amyloid SSS.

This volume includes two chapters on current developments in membrane protein research. Transmembrane proteins constitute more than 20 % of the genome of most organisms. Membrane proteins are of great biological and medical significance, but three-dimensional structures are known for only about 1 % of them due to the fact that they are unstable outside of lipid membrane and are difficult to crystallize. For this reason development of computational approaches for structural prediction of these important proteins is a priority.

In contrast to globular proteins, membrane proteins are turned "inside out": their hydrophobic regions are exposed to the outside, rather than being on the inside, in order to make contact with lipid bilayer. Presumably, the assembly process involves first a formation of strands and helices, and then insertion of secondary structure elements across membrane. The interaction of alpha helices and strands is a key factor that drives formation of SSS of membrane proteins. An effective method for measurement of helices' interactions in membrane is described in detail in Chapter 16 by Schneider and his colleagues. They set down a complete protocol for estimation of the energies of helix–helix interaction in *E. coli* membrane proteins.

An original method for ab initio SSS prediction of membrane proteins by using a graph model is presented by Tran and his collaborators in Chapter 17. Authors specify details of graph model construction and free energy estimation. This approach was used to predict the supersecondary structure of transmembrane β-barrel proteins and accurately discriminate them from other proteins.

Chapter 18 describes the relationship between interaction interfaces and supersecondary structures. For this analysis Kinjo and Nakamura utilize a database of interface structures, which includes all protein binding interfaces calculated from PDB entries. Comprehensive analysis of SSS found in interaction interface proves to be very useful for characterization of protein–ligand, protein–protein, and protein–nucleic acid interactions.

This volume seeks to highlight some of the major advances in the many fast-growing areas of supersecondary structure research. It is hoped that in many cases the knowledge of SSS may be sufficient to answer various relevant questions in protein biology, which heretofore necessitated detailed information about residues' atomic coordinates. If experiment confirms the validity of this assertion, studies into the relationship between protein sequence, structure, and function could proceed much more quickly and efficiently.

References

1. Linderstrøm-Lang K (1952) Proteins and enzymes, lane memorial lectures. Stanford University Press, Stanford, CA, p 54
2. Rao S, Rossmann M (1973) Comparison of super-secondary structures in proteins. J Mol Biol 76(2):241–256
3. Miller J, McLachlan AD, Klug A (1985) Repetitive zinc-binding domains in the protein transcription factor IIIA from Xenopus oocytes. EMBO J 4(6):1609–1614
4. Richardson JS (1977). β-sheet topology and the relatedness of proteins. Nature 268(5620):495–500
5. Kister, AE, Finkelstein AV, Gelfand IM (2002) Common features in structures and sequences of sandwich-like proteins. Proc Natl Acad Sci USA 99(22):14137–14141
6. Kister AE, Gelfand I (2009) Finding of residues crucial for supersecondary structure formation. Proc Natl Acad Sci USA 106(45):18996–19000
7. Fersht A, Matouschek A, Serrano L (1992) The folding of an enzyme: I. Theory of protein engineering analysis of stability and pathway of protein folding. J Mol Biol 224(3):771–782

Contents

Contributors

Eva de Alba • *Centro de Investigaciones Biológicas, Consejo Superior de Investigaciones Científicas (CSIC), Madrid, Spain*

Prem P. Chapagain • *Department of Physics, Florida International University, Miami, FL, USA*

Philippe Chassignet • *Laboratory of Computer Science, Ecole Polytechnique, Palaiseau Cedex, France*

Chuan-Feng Chen • *Beijing National Laboratory for Molecular Sciences, CAS Key Laboratory of Molecular Recognition and Function, Institute of Chemistry, Chinese Academy of Sciences, Beijing, China*

Ke Chen • *Department of Electrical and Computer Engineering, University of Alberta, Edmonton, AB, Canada*

Silvia Crivelli • *Department of Computer Science, University of California, Davis, CA, USA*

Florian Cymer • *Department of Pharmacy and Biochemistry, Johannes Gutenberg-University Mainz, Mainz, Germany*

Marvin Edelman • *Department of Plant Sciences, Weizmann Institute of Science, Rehovot, Israel*

Alexander V. Efimov • *Institute of Protein Research, Russian Academy of Sciences, Moscow Region, Russia*

Narcis Fernandez-Fuente • *Institute of Biological, Environmental and Rural Sciences (IBERS) Aberystwyth University Aberystwyth, Ceredigion, SY23 3DA UK*

Andras Fiser • *Department of Systems and Computational Biology, Albert Einstein College of Medicine, Bronx, NY, USA; Department of Biochemistry, Albert Einstein College of Medicine, Bronx, NY, USA*

Bernard S. Gerstman • *Department of Physics, Florida International University, Miami, FL, USA*

Vikas Gupta • *CARB Centre, Department of Molecular Biology, Aarhus University, Aarhus, Denmark*

Hui Kian Ho • *Department of Computer Science and Software Engineering, University of Melbourne, National ICT Australia, Parkville, VIC, Australia*

Hai-Yu Hu • *Beijing National Laboratory for Molecular Sciences, CAS Key Laboratory of Molecular Recognition and Function, Institute of Chemistry, Chinese Academy of Sciences, Beijing, China*

Akira R. Kinjo • *Institute for Protein Research, Osaka University, Osaka, Japan*

Ina Koch *Molecular Bioinformatics Group, Institute of Computer Science, Johann Wolfgang Goethe-University, Frankfurt am Main, Germany*

Arun S. Konagurthu • *Clayton School of Information Technology, Monash University, Clayton, VIC, Australia*

Annika Kreuchwig • *Structural Bioinformatics Group, Leibniz-Institut fuer Molekulare Pharmakologie, Berlin, Germany*
Lukasz Kurgan • *Department of Electrical and Computer Engineering, University of Alberta, Edmonton, AB, Canada*
Arthur M. Lesk • *Department of Biochemistry and Molecular Biology, Huck Institute of Genomics, Proteomics, and Bioinformatics, Pennsylvania State University, University Park, PA, USA*
Shawn Martin • *Department of Computer Science, University of Otago, Dunedin, New Zealand*
Nelson Max • *Department of Computer Science, University of California, Davis, CA, USA*
Patrick May • *Bioinformatics Group, Luxembourg Centre for Systems Biomedicine, University of Luxembourg, Esch-Belval, Luxembourg*
Victor Muñoz • *Centro de Investigaciones Biológicas, Consejo Superior de Investigaciones Científicas (CSIC), Madrid, Spain; Department of Chemistry and Biochemistry, University of Maryland, College Park, MD, USA*
Haruki Nakamura • *Institute for Protein Research, Osaka University, Osaka, Japan*
Vladimir Potapov • *Department of Biology, MIT, Cambridge, MA, USA*
Esben M. Quistgaard • *MIND Centre, Department of Molecular Biology, Aarhus University, Aarhus, Denmark;* • *Department of Medical Biochemistry and Biophysics, Karolinska Institute, Stockholm, Sweden*
Shalom Rackovsky • *Department of Pharmacology and Systems Therapeutics, Mount Sinai School of Medicine, New York University, New York, NY, USA; Department of Chemistry and Chemical Biology, Cornell University, Ithaca, NY, USA*
Kotagiri Ramamohanarao • *Department of Computer Science and Software Engineering, University of Melbourne, Melbourne, VIC, Australia*
Michael G. Rossmann • *Department of Biological Sciences, Purdue University, West Lafayette, IN, USA*
Raimon Sabaté • *Institut de Biotecnologia i de Biomedicina and Departament de Bioquímica i Biologia Molecular, Universitat Autònoma de Barcelona, Bellaterra (Barcelona), Spain*
Mourad Sadqi • *Centro de Investigaciones Biológicas, Consejo Superior de Investigaciones Científicas (CSIC), Madrid, Spain*
Charles R. Sanders • *Department of Biochemistry, School of Medicine, Vanderbilt University, Nashville, TN, USA*
Lorenzo Sborgi • *Centro de Investigaciones Biológicas, Consejo Superior de Investigaciones Científicas (CSIC), Madrid, Spain*
Dirk Schneider • *Department of Pharmacy and Biochemistry, Johannes Gutenberg-University Mainz, Mainz, Germany*
Vladimir Sobolev • *Department of Plant Sciences, Weizmann Institute of Science, Rehovot, Israel*
Jean-Marc Steyaert • *Laboratory of Computer Science, Ecole Polytechnique, Palaiseau Cedex, France*
Søren Thirup • *Department of Molecular Biology, MIND Centre, CARB Centre, Aarhus University, Aarhus, Denmark*

Van Du T. Tran • *Laboratory of Computer Science, Ecole Polytechnique, Palaiseau Cedex, France*

Salvador Ventura • *Institut de Biotecnologia i de Biomedicina and Departament de Bioquímica i Biologia Molecular, Universitat Autònoma de Barcelona, Bellaterra (Barcelona), Spain*

Abhinav Verma • *Centro de Investigaciones Biológicas, Consejo Superior de Investigaciones Científicas (CSIC), Madrid, Spain*

Lei Zhang • *Department of Computer Science and Software Engineering, University of Melbourne, National ICT Australia, Melbourne, Victoria, Australia*

Chapter 1

Super-secondary Structure: A Historical Perspective

Michael G. Rossmann

Abstract

The history of the concept of protein folds is discussed, starting with the original concept of super-secondary structure. This has led to the recognition of a fairly small number of distinct folds defining individual domains within larger proteins. Each fold can usually be associated with a specific function. Thus the active site of an enzyme is likely to be at the boundary between domains, each contributing a simple function to a more complex process.

Key words: Super-secondary structure, Rossmann fold

Little was known about protein structure in 1973, although fundamentals had been established by Pauling, Linderstrøm-Lang, and others. Pauling had predicted the α-helix as well as parallel and antiparallel β-sheets (1, 2) based on fiber diffraction results of Astbury (3) and crystal structures of amino acids. Linderstrøm-Lang had suggested that proteins had primary, secondary, and tertiary structure (4, 5). Perutz had verified the probable presence of α-helices in hemoglobin. In addition the first few protein structures had been determined. Of particular note were the structures of myoglobin (6) and hemoglobin (7) that had shown the evolutionary conservation of an oxygen binding fold, as well as a few enzymes such as hen egg white lysozyme (8), ribonuclease (9, 10), various serine proteases (11–13), and some dehydrogenases (14, 15). Further evidence for the conserved nature of tertiary structure had become evident in the study of serine proteases, dehydrogenases, repeating domains in a calcium-binding structure found in carp muscle (16), and in some antibody components (17, 18).

New names were starting to become current in the early 1970s such as protein domain and protein fold. Although there is still no complete agreement as to the exact definition of these names, it is probably generally accepted that a protein domain is a unit of structure with a specific function (e.g., binding oxygen such as in

Alexander E. Kister (ed.), *Protein Supersecondary Structures*, Methods in Molecular Biology, vol. 932,
DOI 10.1007/978-1-62703-065-6_1,

the globins or binding NAD such as in the dehydrogenases) that is found in diverse proteins, often in combination with other, spatially separated, domains within the same polypeptide. Because of the similarity of structure and function, it is reasonable to assume that similar domains might have had a common evolutionary origin, even if there is little memory of a common primordial amino acid sequence. A protein fold is usually used exclusively to indicate a similarity of structure without necessarily having a common function. Thus, protein folds may or may not have had a common evolutionary origin. Clearly secondary structural elements such as α-helices and β-sheets taken on their own are not likely to have a traceable common origin. It would not be unreasonable to consider that specific combinations of secondary structure would be the result of folding preferences caused by sequence properties such as hiding hydrophobic residues in the interior. I, therefore, introduced the concept of "super-secondary structure" in 1973 (19) to represent small combinations of secondary structure that might be easy folding units. In particular, I observed that the NAD-binding domain in dehydrogenases could be considered as the repeat of two similar units related by an approximate twofold axis and that the same unit existed in the structure of *Desulfovibrio vulgaris* flavodoxin (Fig. 1). These super-secondary structural elements would have a combination of secondary structural elements common to many diverse proteins with no apparent common function, but yet some low level of sequence similarity to help maintain the three-dimensional structure.

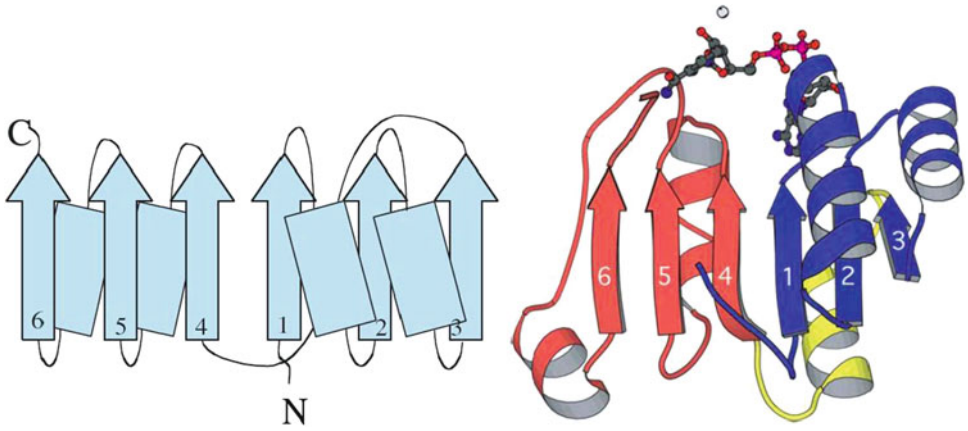


Fig. 1. *Left*: Schematic drawing of the topology called the Rossmann fold. *Right*: The NAD-binding domain of *Sulfolobus solfataricus* alcohol dehydrogenase, a typical Rossmann fold (order of strands 654123). The first βαβαβ motif of the domain is numbered 123 and the second 654, and the connection reaches from 4 to between 1 and 2. The nucleotide is bound at the C-terminal ends of the β-strands at the center of the sheet. The phosphates of the NAD molecule are located at the N-termini of two helices where their binding is favored by the helical dipoles (PDB: 1R37). Reprinted from "Textbook of Structural Biology" (p. 29), by A. Liljas, L. Liljas, J. Piskur, G. Lindblom, P. Nissen and M. Kjeldgaard, 2009, Singapore: World Scientific Publishing Co. Pte. Ltd. Copyright 2009 by World Scientific Publishing Co. Pte. Ltd. Reprinted with permission.

In the early days of protein crystallography, there was a great deal of skepticism for claims of structural similarity among diverse protein segments. Therefore, it was necessary to establish quantitative measurements of similarity. Hence a large part of the original paper on super-secondary structure (20) is concerned with the mathematical procedure of comparing three-dimensional structure and establishing criteria for specifying structural similarity. One of these criteria, which remains useful today, is the percentage of amino acids that could be sequentially superimposed (say within 3.8 Å, the distance between C_α atoms) between two proteins. It is a reversal of history that today there is usually some skepticism for claims of a newly discovered fold, whereas in earlier times it was often difficult to persuade others to see similarity of structure. The developments from these early searches for an understanding of protein structure have led to comparison programs such as DALI (21) and fold libraries such as SCOP (22) and CATH (23).

References

1. Pauling L, Corey RB, Branson HR (1951) The structure of proteins: two hydrogen-bonded helical configurations of the polypeptide chain. Proc Natl Acad Sci USA 37:205–211
2. Pauling L, Corey RB (1951) Configurations of polypeptide chains with favored orientations around single bonds: two new pleated sheets. Proc Natl Acad Sci USA 37:729–740
3. Astbury WT (1933) Fundamentals of fiber structure. Oxford University Press, London
4. Linderstrøm-Lang K (1952) Proteins and enzymes, lane memorial lectures. Stanford University Press, Stanford, CA, p 54
5. Huggins ML (1943) The structure of fibrous proteins. Chem Rev 32:195–218
6. Kendrew JC, Bodo G, Dintzis HM et al (1958) A three-dimensional model of the myoglobin molecule obtained by X-ray analysis. Nature 181:662–666
7. Perutz MF, Rossmann MG, Cullis AF et al (1960) Structure of haemoglobin. A three-dimensional Fourier synthesis at 5.5-Å resolution, obtained by X-ray analysis. Nature 185:416–422
8. Blake CC, Koenig DF, Mair GA et al (1965) Structure of hen egg-white lysozyme. A three-dimensional Fourier synthesis at 2 Å resolution. Nature 206:757–761
9. Kartha G, Bello J, Harker D (1967) Tertiary structure of ribonuclease. Nature 213:862–865
10. Wyckoff HW, Hardman KD, Allewell NM et al (1967) The structure of ribonuclease-S at 3.5 Å resolution. J Biol Chem 242:3984–3988
11. Matthews BW, Sigler PB, Henderson R et al (1967) Three-dimensional structure of tosyl-α-chymotrypsin. Nature 214:652–656
12. Stroud RM, Kay LM, Dickerson RE (1972) The crystal and molecular structure of DIP-inhibited bovine trypsin at 2.7 Å resolution. Cold Spring Harb Symp Quant Biol 36: 125–140
13. Watson HC, Shotton DM, Cox JM et al (1970) Three-dimensional Fourier synthesis of tosyl-elastase at 3.5 Å resolution. Nature 225: 806–811
14. Adams MJ, Ford GC, Koekoek R et al (1970) Structure of lactate dehydrogenase at 2.8 Å resolution. Nature 227:1098–1103
15. Hill E, Tsernoglou D, Webb L et al (1972) Polypeptide confirmation of cytoplasmic malate dehydrogenase from an electron density map at 3.0 Å resolution. J Mol Biol 72:577–589
16. Nockolds CE, Kretsinger RH, Coffee CJ et al (1972) Structure of a calcium-binding carp myogen. Proc Natl Acad Sci USA 69: 581–584
17. Edelman GM, Cunningham BA, Gall WE et al (1969) The covalent structure of an entire γG immunoglobulin molecule. Proc Natl Acad Sci USA 63:78–85
18. Poljak RJ, Amzel LM, Avey HP et al (1972) Structure of Fab′ New at 6 Å resolution. Nat New Biol 235:137–140
19. Rao ST, Rossmann MG (1973) Comparison of super-secondary structures in proteins. J Mol Biol 76:241–256

20. Bella J, Kolatkar PR, Marlor CW et al (1999) The structure of the two amino-terminal domains of human intercellular adhesion molecule-1 suggests how it functions as a rhinovirus receptor. Virus Res 62:107–117
21. Holm L, Sander C (1995) Dali: a network tool for protein structure comparison. Trends Biochem Sci 20:478–480
22. Murzin AG, Brenner SE, Hubbard T et al (1995) SCOP: a structural classification of proteins database for the investigation of sequences and structures. J Mol Biol 247: 536–540
23. Orengo CA, Michie AD, Jones S et al (1997) CATH—a hierarchic classification of protein domain structures. Structure 5:1093–1108

Part I

Representation of Supersecondary Structure

Chapter 2

Hierarchical Representation of Supersecondary Structures Using a Graph-Theoretical Approach

Ina Koch, Annika Kreuchwig, and Patrick May

Abstract

The unique representation of proteins becomes more and more important with the growing number of known protein structure data. Graph-theory provides many methods not only for the description but also for comparison and classification of protein structures. Here, we describe a graph-theoretical modeling approach of the protein supersecondary structure. The resulting linear notations are intuitive and can be used to find common substructures very fast and easily. We illustrate the necessary definitions by biological examples and discuss the representation of various supersecondary structure motifs.

Key words: Graph-theory, Supersecondary structure, Protein graph, Folding graph, Adjacent notation, Reduced notation, Key notation, Sequence notation, Greek key, Four-helix bundle, Globin fold, Up-and-down barrel, Immunoglobulin fold, β-Propeller, Jelly roll, Rossman fold, TIM barrel, Ubiquitin roll, αβ-Plaits

1. Introduction

With the new high-throughput facilities, among other fields, proteomics, and thus protein structure elucidation results in a growing amount of new structural data, usually stored in the Protein Data Bank—PDB (1, 2). Nowadays the PDB contains about 82,000 proteins structures. Therefore, structure comparison and classification cannot be done manually anymore. Computational processing requires a unique description of the protein structure. In particular, the abstraction from the atomic to the secondary and supersecondary structure level is advantageous because of two reasons. First, the complexity will decrease, because we do not consider thousands of atoms anymore, but only up to hundred

Alexander E. Kister (ed.), *Protein Supersecondary Structures*, Methods in Molecular Biology, vol. 932,
DOI 10.1007/978-1-62703-065-6_2,

secondary structure elements (SSEs). Second, structural motifs can occur in different dimensions, thus they may not be identified, working at the atom level.

We characterize structural motifs or the fold of a protein by the spatial arrangement of SSEs. We define the topology of a protein as the relationship between the sequential arrangement of SSEs and their spatial organization.

Secondary structure classification databases have been developed since about 20 years. The most widespread are SCOP (3) and CATH (4), which also classify protein structures at secondary structure level, each using different classification criteria. For example, SCOP requires the same order of SSEs in the sequence in equal structural motifs. Both databases are derived by semi-automatic processes and manual curation of high-resolution 3D structures using different heuristics, which are each advantageous for different cases. But in both databases, the description of protein topology is mathematically not uniquely defined. We developed a unique description of protein supersecondary structure based on graph-theory. This description enables for an easy and fast search for topologies and similarities between proteins. We stored the graph-theoretic description for each protein of the PDB in our database—the Protein Topology Graph Library, PTGL (5). With its strong mathematical definitions PTGL represents a useful extension of SCOP and CATH and can give new insights into protein structure topology. PTGL can be used for any kind of theoretical protein structure analysis, protein structure prediction, and protein function prediction.

In this chapter, we describe how we model the protein supersecondary structure as a graph, the resulting linear notations, and illustrate examples of some typical supersecondary structure motifs. We start in the first Subheading 2 with the databases we used. Then in Subheading 3, we explain the definitions and linear notations, their visualization, and their use for the description of supersecondary structure motifs. In Subheading 4, we summarize and discuss the concepts and use of PTGL, giving details and the potential of the approach.

2. Materials

As this is a bioinformatics approach, our materials are databases and special software tools. In the following subsections we shortly describe the databases we used. All these resources are publicly available.

2.1. Protein Data Base

The Protein Data Bank (PDB, http://www.rcsb.org) (1, 2) at Research Collaboratory for Structural Bioinformatics (RCSB) is the main collection of 3D-structures of proteins, nucleic acids, and other biological macromolecules determined with X-ray crystallography, nuclear magnetic resonance (NMR) spectroscopy, or low-temperature electron microscopy (cryo-EM). By June 2011 the PDB contained 73,951 entries that can be classified into 68,467 protein structures with their prosthetic groups, such as cofactors, substrates, inhibitors, or other ligands including, nucleic acids, 2,261 oligonucleotide or nucleic acid structures, 3,184 protein/nucleic acid complexes, and 39 other biomolecules, including carbohydrate structures. In the PDB, data exhibit a high amount of redundancy on sequential as well as on structural level (see also Fig. 1), because often there is a special, mainly pharmaceutical-driven, interest in certain proteins. Thus, often the same protein has been investigated under different experimental conditions or with different ligands. The exponential growth of numbers of structures and growth of unique folds is depicted in Fig. 1a–c.

The diagrams in Fig. 1 illustrate that the number of new folds is rather small in comparison to the rapidly growing amount of new protein structures. The existing classification methods do not mathematically, uniquely describe protein structure topologies. Thus, new and better automatic classification approaches became necessary.

2.2. Define Secondary Structures of Proteins

To uniquely define supersecondary structures in proteins we first need to assign the SSEs to each amino acid residue. For solving this task, several approaches have been developed, such as STRIDE (6), which uses H-Bond patterns, DEFINE (7), which applies C_α-distances, P-Curve (8), which is based on mathematical analysis of protein curvature, Segno (9), which defines SSEs on a number of geometric parameters for backbone atoms, and STICK (10), which considers SSEs as linear line segments, independently of any external SSE definition.

The DSSP-algorithm (Define Secondary Structure of Proteins) (11) is one of the most commonly used programs to define SSEs in protein structures. It is mainly based on the computation of regular hydrogen bonding patterns of the backbone's N- and C-atoms. The algorithm distinguishes between eight different secondary structure states. The hydrogen bonds are described by an electrostatic model. Three different helix types (states *I, H, G*) can be defined according to at least two consecutive amino acid residues with the hydrogen bond patterns H-bond (i, $i+n$) and H-bond (i, $i+n$) for $n=(3–5)$ (3 for *I-helix*, also called π-*helix*, 4 for *H-helix* known as α-*helix*, 5 for *G-helix* also named 3_{10}-*helix*) with i, denoting the residue number from N- to C-terminus. A β-*strand* or β-*sheet* residue (state *E* derived from *extended* structure elements) is defined as either having two hydrogen bonds in a sheet, or being

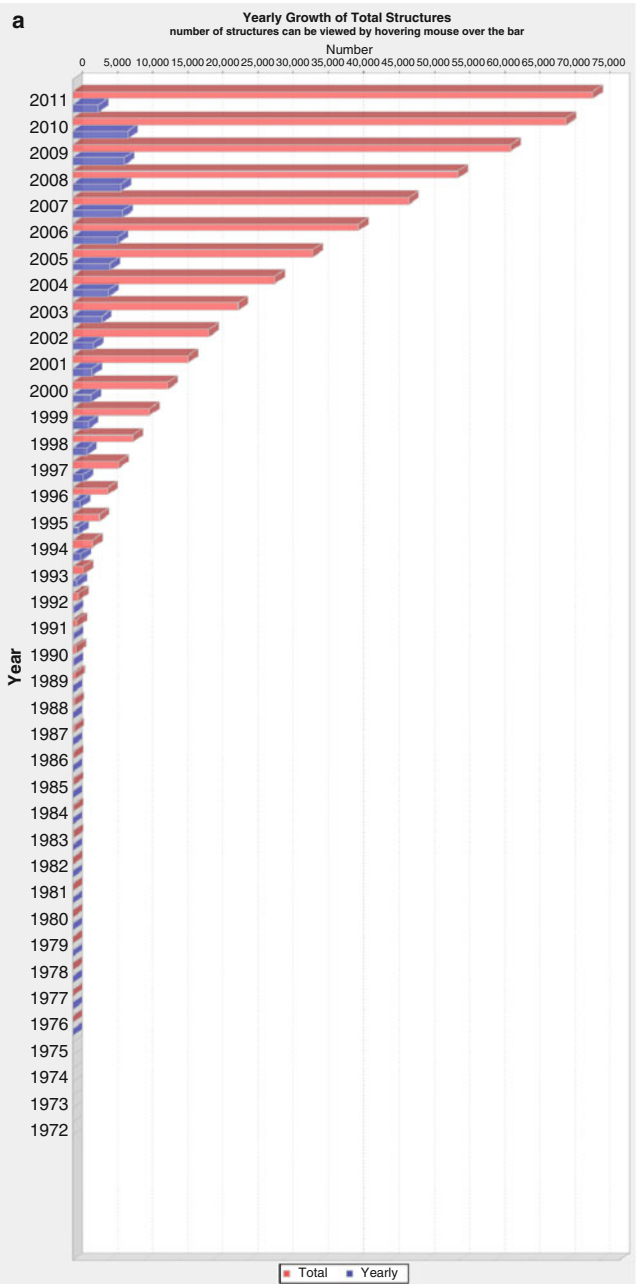


Fig. 1. (**a**) Number of structures in the PDB per year. (http://www.pdb.org/pdb/statistics/contentGrowthChart.do?content=total&seqid=100). (**b**) Growth of unique folds per year as defined by SCOP (v1.75). (http://www.pdb.org/pdb/statistics/contentGrowthChart.do?content=fold-scop). (**c**) Growth of unique folds (topologies) per year as defined by CATH (v3.4.0). (http://www.pdb.org/pdb/statistics/contentGrowthChart.do?content=fold-cath)

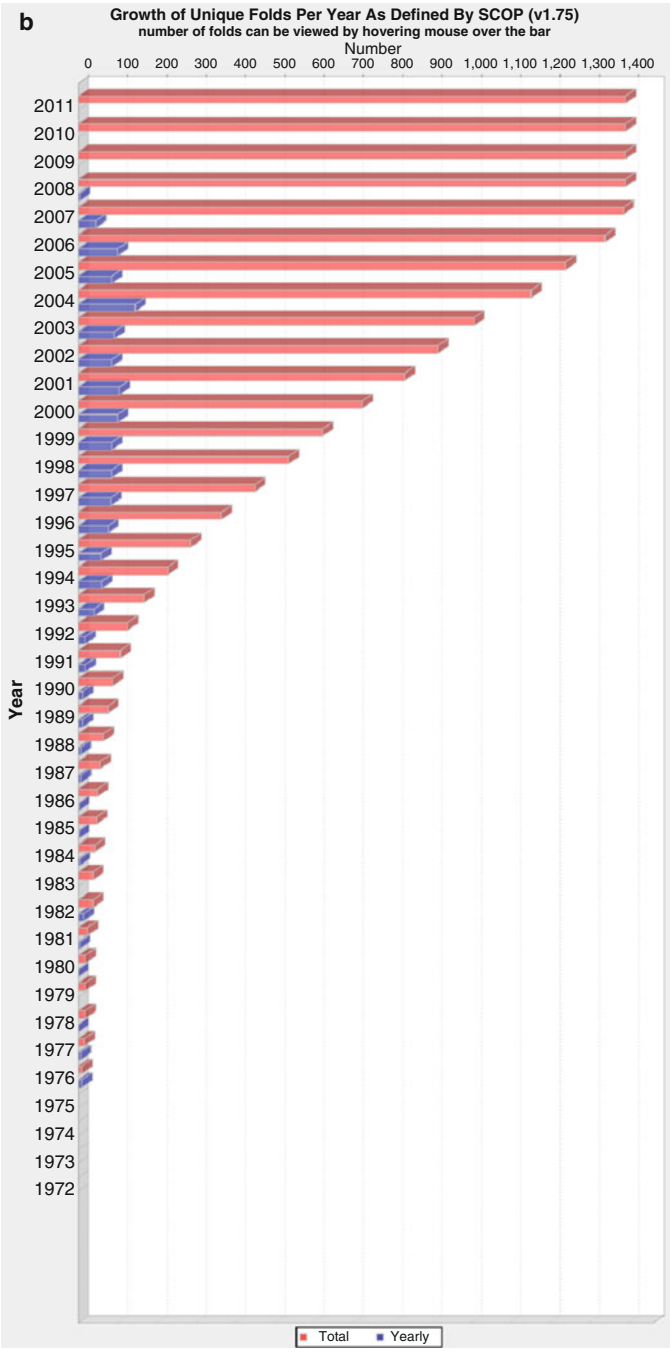


Fig. 1. (continued) Fig. 1a shows the exponentially growing number of known protein structures stored in the PDB. Figure 1b, c illustrates in both topology databases, SCOP and CATH, that the number of different folds is limited. Despite the huge amount of new protein structures the number of new protein folds only slowly grows. Additionally, it is indicated that the annotation process in SCOP as well as in CATH is very slow, because it is not fully automated such as in PTGL.

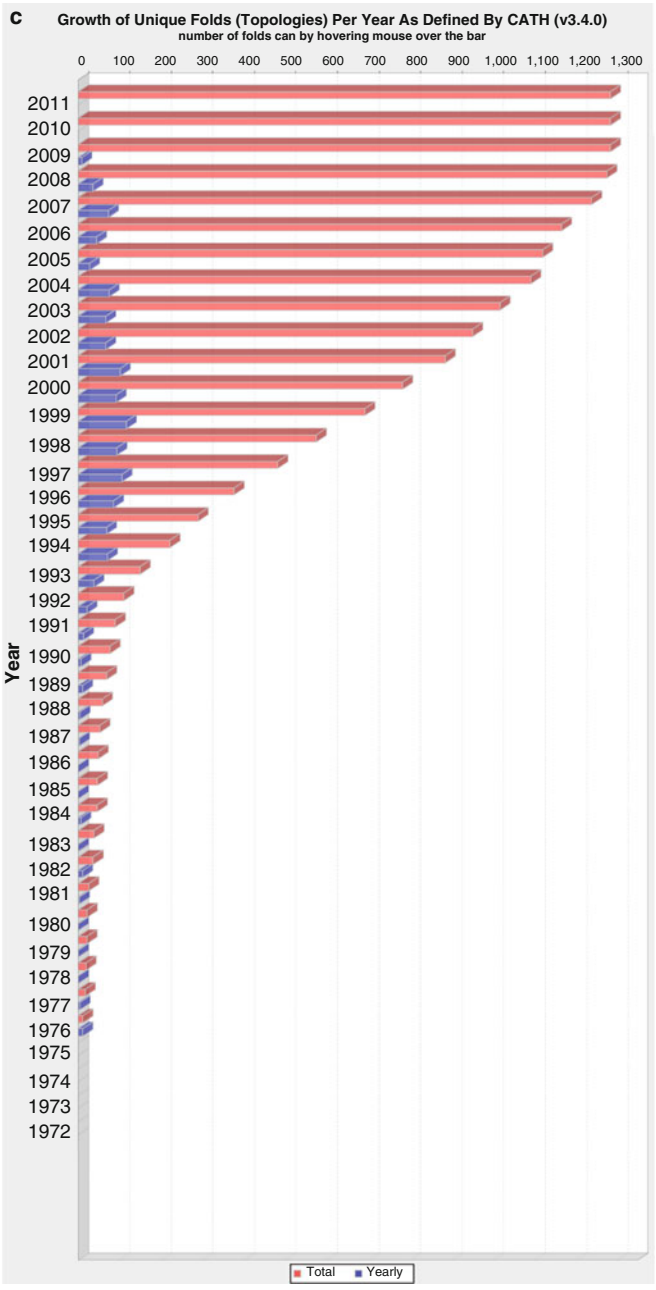


Fig. 1. (continued)

surrounded by two hydrogen bonds in a sheet. According to the direction from N- to C-terminus, the arrangement of neighbored β-strands can be classified to be *parallel* or *antiparallel*. A β-*bridge* (state *B*) is an isolated residue satisfying the hydrogen bonding property. *Turns* (state *T*) or β-*hairpins* describe sharp turns in the polypeptide chain defined by H-bonds $(i-1, i, i+1)$, $(i-1, i+2)$, $(i-2, i+1)$, or $(i-2, i, i+2)$. The state *S*, called *bend*, represents a wider turn and is not defined by H-bond patterns, but by the κ-*angle* between $(i-2, i, i+2)$. Residues with irregular secondary structures are assigned to nothing, indicating *loops* or *coil regions*.

In our approach we consider only the regular SSEs, i.e., we involve helices and strands, but not turns, bends, and coil regions. Because DSSP considers H-bonds only between backbone atoms, assigned SSEs can be too short, in particular if they are defined also by H-bonds between sidechain atoms. Thus, some additional rules have been applied such as the following: (1) a helix has to consist of at least three consecutive residues; (2) consecutive H-, I-, and G-helices are summarized to one H-helix; (3) a strand has to be formed by at least three consecutive residues; (4) if there is a gap of one residue between two strands or between two helices, additionally the corresponding phi and psi torsion angles (between C_α–N and C_α–CO, respectively) in the neighbored residues of the strand or helix are considered. Only, if the angles are nearly equal the gap will be filled by state *E* or *H*, respectively; (5) long helices and strands (>10 residues) are divided into two (sub)-SSEs if the backbone angles are suggesting two distinct SSEs.

2.3. Protein Topology Graph Library

The database Protein Topology Graph Library (PTGL) (5) is based on a graph-theoretical description of proteins which allows for a unique representation of super-SSEs and thus for a unique classification of protein topologies independently of the sequential order of SSEs. The PTGL stores the protein topologies for each PDB entry in a database and is available via a Web interface (http://ptgl.uni-frankfurt.de) allowing for different search modes. In this chapter, we refer to the PTGL, explaining the underlying mathematical concept.

3. Methods

Structural motifs, often also referred as supersecondary structures, are small substructures, in general, consisting of only few SSEs, and the spatial interactions between them. Specific structural motifs are seen repeatedly in many different protein structures. Most often they are integral elements of protein folds. Further, these motifs often have a functional significance, and in these cases, they

represent a minimal functional unit within a protein. Several motifs can be spatially neighbored, forming functional specific domains. The aim of our approach is to define supersecondary structure motifs in a unique manner to find new relations between protein topology and protein function. The methods we apply to analyze protein topology are based on graph-theory and are described in more detail in the next subsections, illustrating them using biological examples.

3.1. The Protein Graph

A graph $G=(V, E)$ consists of a set of vertices or nodes, V, and a set of edges or arcs, E. The vertices describe objects and the edges arbitrary relations between them. *Protein graphs* (*PGs*) describe the entire secondary structure topology of a single protein chain. Thus, the vertices describe α-helices and β-strands, and the edges spatial neighborhoods between them. According to the direction of the strands the edges can be labeled as parallel, antiparallel, or mixed in the remaining cases. Mathematically, we define a PG of a protein chain as follows:

Definition (Protein Graph, PG): Let R be the finite set of β-strands, e, and H the finite set of α-helices, h, of a protein chain. The protein graph, $PG=(V, E)$, is then defined as an undirected, labeled graph with the vertex set, $E=R\cup H$, and the edge set, $E\subseteq V^2$. Two vertices u and v are connected by an edge, i.e., $u, v\in E$, if there exists a spatial contact between u and v. Edges are labeled according to the direction of the spatial neighborhood with *"a"* for antiparallel, *"p"* for parallel, and *"m"* (mixed) otherwise.

These spatial contacts are defined by intersecting van der Waals radii, generalized to 2.0 Å, of the atoms of an SSE. In dependence of the type of the participating atoms of the SSEs (backbone–backbone, backbone–sidechain, or sidechain–sidechain), we define the contacts between

1. β-strands: if there are at least three backbone–backbone or at least three sidechain–sidechain intersections,
2. α-helices: if there are at least four backbone–sidechain or at least four sidechain–sidechain intersections, and
3. α-helices and β-strands: if there are at least two backbone–backbone and at least four backbone–sidechain intersections, or at least four sidechain–sidechain intersections.

If proteins exhibit several domains it could often appear that there are spatially neighbored, compact substructures which form independent folding units or folds, in particular, if the protein exhibits different functional units. We define these independent folds as *connected components*, if there is a path via edges from each vertex to each other vertex. The connected components of a PG are called *folding graphs* (*FGs*). They are named by an upper-case letter in alphabetical order (12, 13). For an example, see Fig. 2.

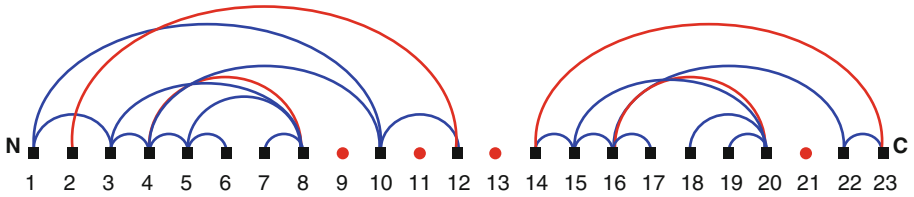


Fig. 2. Connected components represent FGs in the PG of the T cell antigen receptor, 1bec (14). The figure depicts the protein graph of the T cell antigen receptor (PDB identifier: 1bec, chain A). Helices are drawn as filled *red circles* and strands as *black filled squares*. Antiparallel edges are colored *blue*, parallel *red*, and mixed (not contained in this topology) *green*. The protein consists of two structural domains, and thus, the PG exhibits two FGs with more than two SSEs. Both FGs consist of strands only. The first one covers SSEs, 1 to 8, and SSEs 10 and 12; the second one contains the SSEs 14 to 20, and SSE 22 and 23. The four helices in the protein are not spatially neighbored to one of the domains.

FGs often describe domains and folding units. According to the type of the considered SSE, we can differentiate PGs and FGs into αβ-protein graphs, α-protein graphs, and β-protein graphs, and correspondingly αβ-folding graphs, α-folding graphs, and β-folding graphs, thus representing the αβ-, α-, or β- topology of a protein chain (12, 13).

3.2. The Linear Notations

Folding graphs describe closely, in space neighbored SSEs, thus defining the protein core of a functional domain. Because we are interested in the relationship between structure and function, we consider in the following section each functional domain separately, i.e., we consider FGs.

According to the maximal number of adjacent edges, we classify the graphs into *linear graphs*, exhibiting only vertices with at most two adjacent edges, and *bifurcated graphs*, which contain vertices with more than two adjacent edges. The different graph types are indicated by the different parentheses in the linear notation. We use "[ ]" for linear graphs, "{ }" for bifurcated graphs, and "()" for barrel structures.

3.2.1. Linear Graphs

For each linear FG, we can define four linear notations, (1) the *adjacent notation* (ADJ), (2) the *reduced notation* (RED), (3) the *sequence notation* (SEQ), and (4) the *key notation* (KEY). In the first three notations, SSEs are arranged in a line as they occur from N- to C-terminus. The vertex with only one adjacent edge, i.e., only one neighbored SSE, which is located nearest to the N-terminus, is our starting point. We write the SSE type ("*h*" for helix and "*e*" for strand), put a comma, and follow the edges, noting the sequential difference to the next vertex, i.e., the difference between corresponding SSE numbers, and the edge label ("*a*", "*p*," or "*m*"), the SSE type, and put a comma. For moves from C- to N-terminus, we additionally write "–". We proceed until we

have visited each edge. For α- or β-topologies we do not write the type of the SSE. In case of ADJ- and SEQ-type we consider all SSEs in the PG, even when describing an FG, whereas in the RED-type we consider only the SSEs of the same FG. In case of the SEQ-type, we represent only sequential neighborhoods, drawing the edges as black arcs.

The KEY-type is similar to the first topology notations for β-sheets introduced by J. Richardson (15, 16). We arrange SSEs as they occur in space, forming a nearly linear plane, which is only possible for linear graphs, because many supersecondary structure topologies cannot be arranged in a plane. Therefore, there is no notation for bifurcated graphs in Key-type. To derive the KEY-notation, we start with the SSE nearest to the N-terminus and follow the sequential edges, drawn as black arcs, and note the sequential distance of the corresponding SSEs. If the arcs cross the fold plane we additionally write an "*x*". We again note the SSE, proceeding until we reach the SSE nearest to the C-terminus. Here, we visualize helices by red cylinders and strands by black arrows. To illustrate the definitions Figs. 3, 4, 5, and 6 depict the four different notations of the same fold. Note that we use "[ ]"-parentheses to indicate the linear notation (17).

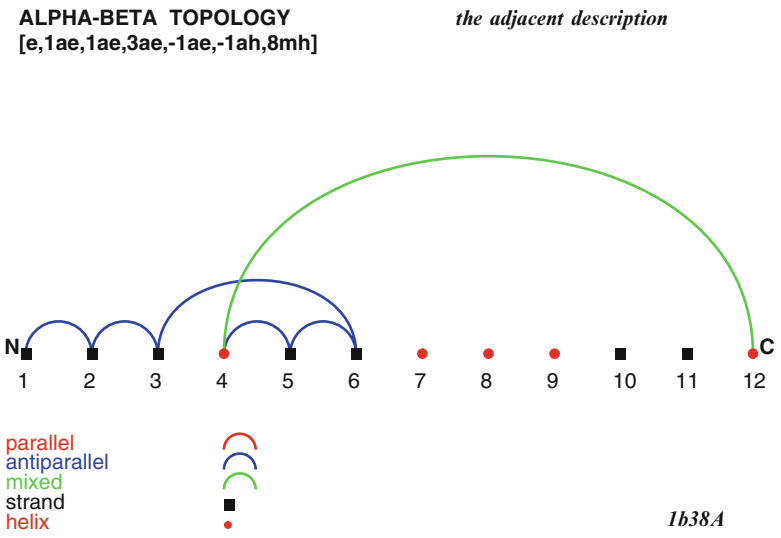


Fig. 3. The ADJ-notation of human cyclin-dependent kinase 2, 1b38A (18). The figure illustrates the FG in ADJ-representation and notation of 1b38, chain A, fold A. We arrange the SSEs as they occur in the sequence from N- to C-terminus. For deriving the linear notation, *[e, 1ae, 1ae, 3ae, –1ae, –1ah, 8mh]*, we start with SSE 1, which is nearest to the N-terminus. We follow the edge and note the difference of the SSE numbers (2 – 1 = 1), write the edge label, *a* (drawn as *blue arc*), and the SSE type, *e*, followed by a comma, until we have visited each edge. We additionally note a "–", if we move from C- to N-terminus.

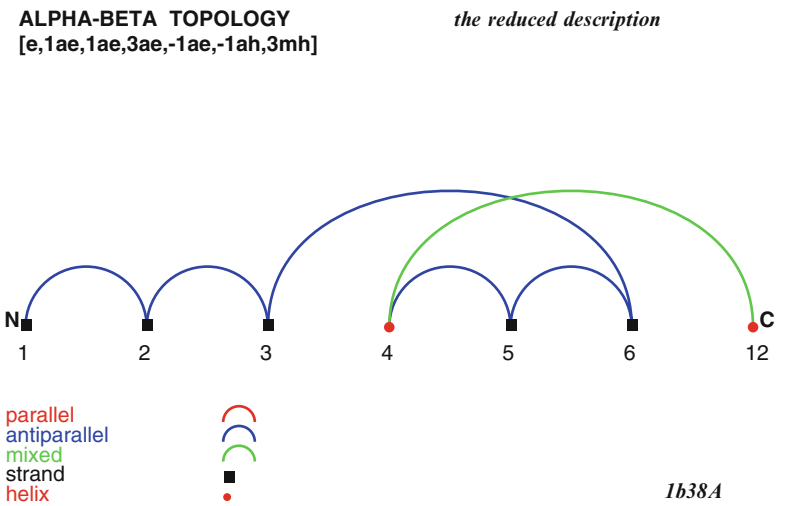


Fig. 4. The RED-notation of human cyclin-dependent kinase 2, 1b38A (18). The figure represents the FG in the RED-representation and notation of 1b38, chain A, fold A. In contrast to the ADJ-notation we consider only the SSEs, which belong to the same FG. We follow the same procedure as for the ADJ-notation and yield the linear notation, *[e, 1ae, 1ae, 3ae, 1ae, –1ah, 3mh]*, which differs in the last term, *3mh*, from the ADJ-notation.

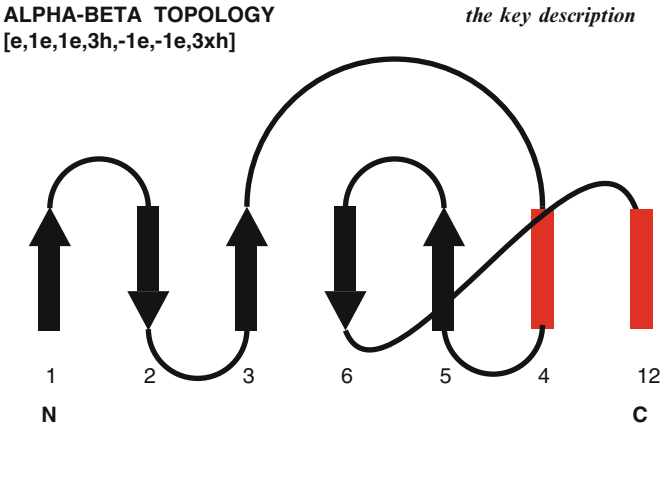


Fig. 5. The KEY-notation of human cyclin-dependent kinase 2, 1b38A (18). The figure depicts the FG in the KEY-representation and notation of 1b38, chain A, fold A. Now, we arrange the SSEs as they occur in space. We consider only the SSEs, which belong to the same FG. For deriving the linear notation, *[e, 1e, 1e, 3h, –1e, –1e, 3xh]*, we start with the SSE nearest to the N-terminus, here SSE 1. We follow the sequential black arcs and note the difference of the SSE numbers ($2-1=1$), write the SSE type, *e*, followed by a comma, until we have visited all edges. We additionally note a "–", if we move from C- to N-terminus. If we cross the plane of the fold such as from strand 6 to helix 12, we write an "x". Note that here, the SSE numbers correspond to the PG numbers as in the ADJ-notation, but for computing the difference we have to consider the FG numbers, neglecting SSEs of other folding graphs.

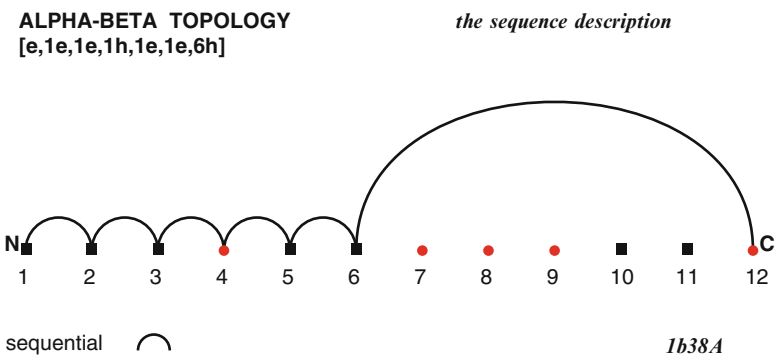


Fig. 6. The SEQ-notation of human cyclin-dependent kinase 2, 1b38A (18). The figure depicts the FG in the SEQ-representation and notation of 1b38, chain A, fold A. We arrange SSEs as in the ADJ- and RED-notation, considering all SSEs in the protein. As in the ADJ-notation we start with the first SSE nearest to the N-terminus, follow the arcs (drawn in *black*), and note the differences in the SSE numbers with the SSE type. Thus, we yield *[e, 1e, 1e, 1h, 1e, 1e, 6h]*.

3.2.2. Bifurcated Graphs

To represent the ADJ- and RED-notations of bifurcated graphs, we again start with the first SSE in the sequence with only one neighbor nearest to the N-terminus and note the SSE type. Then, we follow the edges as for nonbifurcated topologies noting the difference between corresponding SSE numbers and the edge type, until there is no unvisited edge to another SSE left. Now, we jump back or forward to the SSE nearest to the N-terminus, which exhibits one neighbor of still unvisited edges. We write, how many steps we jumped back or forward, i.e., the sequential difference, the SSE type of the target vertex, and add a "*z*" (derived from "*z*urück", meaning "back" in German). We continue this procedure until we have visited all edges. For examples for ADJ- and RED-notations see Figs. 7 and 8. Note that we use "{ }"-parentheses now (17).

3.3. Supersecondary Structure Motifs

Protein domains are often built from recurring simple combinations of SSEs with specific topological arrangements, called motifs or super-SSEs. These motifs assemble in various combinations into more complex motifs or whole domains. As in the SCOP database, motifs are classified according to their SSE composition into four different classes:

(1) α-helical motifs, containing α-helices only,
(2) β-sheet motifs, containing mainly antiparallel β-sheets,
(3) α/β motifs, containing β-strands with connecting helical segments, and
(4) α + β motifs, containing spatially separated helical and sheet regions.

Protein motifs can be uniquely defined using the graph-theoretical description of protein structures given as linear notations (Subheading 3.2) in combination with simple rules derived from SSE definition, contact definition, and number of SSEs. We define

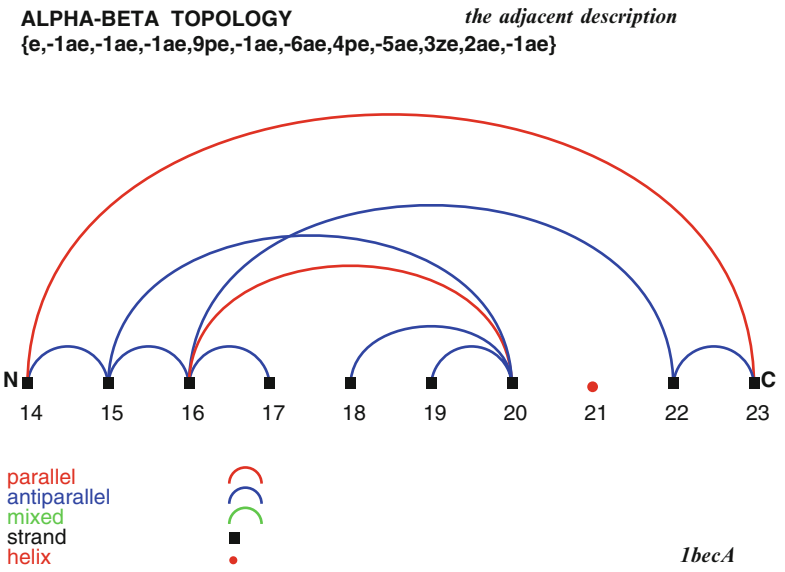


Fig. 7. A bifurcated FG in ADJ-notation in the T cell antigen receptor, 1becA (14). The figure illustrates the bifurcated FG of 1bec, chain A, fold E, in ADJ-notation. We proceed as in the nonbifurcated ADJ-notation. So, we start at SSE 17, note the SSE type, follow the edges, visiting the SSEs 16, 15, 14, 23, 22, 16, 20, and 15, and writing the difference between SSE numbers and the SSE type. We look for the next SSE (always nearest to the N-terminus) with unvisited edges, choosing first those with only one unvisited edge, and jump there, writing the difference between SSE numbers, a "z", and the SSE type, in our example, *3ze*. Then we start again until all edges are visited. We yield the notation *{e, –1ae, –1ae, –1ae, 9pe, –1ae, –6ae, 4pe, –5ae, 3ze, 2ae, –1ae}*.

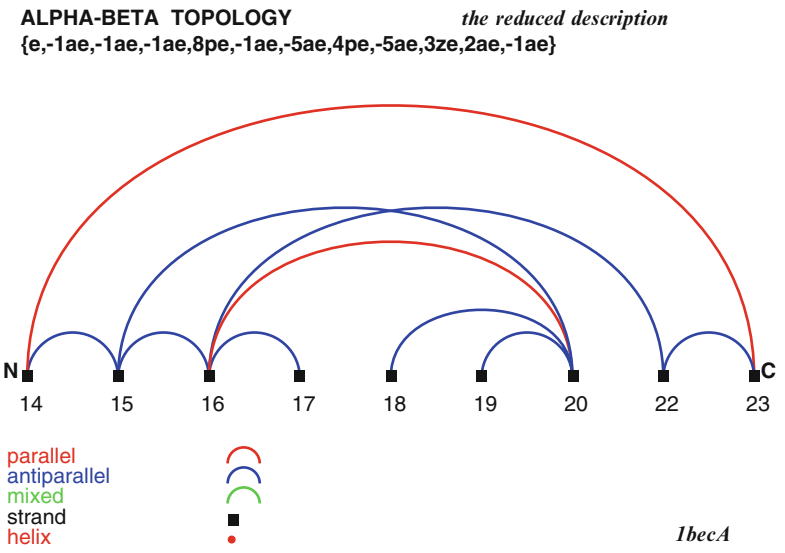


Fig. 8. A bifurcated FG in RED-notation in the T cell antigen receptor, 1becA (14). The figure depicts the bifurcated FG of 1bec, chain A, fold E, in RED-notation. The procedure is the same as for the ADJ-notation, but we neglect the SEEs that do not belong to the same FG. Then, we yield the notation *{e, –1ae, –1ae, –1ae, 8pe, –1ae, –5ae, 4pe, –5ae, 3ze, 2ae, –1ae}*. Differences in comparison to the ADJ-notation are in the 5th ("*8pe*", now) and 7th ("*–4ae*", now) term.

the following functions such that Boolean combinations are capable to describe general protein motifs (19):

(1) *Sheet(name, type, number of SSEs, topology)* is defined as a set of SSEs of type β-strand which are connected through a path within a folding graph, the number of SSEs, the type of contact between the SSEs (*p*arallel, *a*ntiparallel, or *m*ixed (*p*, *a*, or *m*)), and the β-topology of this set of SSEs described by a linear notation. Names consist of one upper-case letter and are alphabetically ordered from N- to C-terminus,
(2) *Helix(name, position)* defines a single SSE of type α-helix with a name and its position numbered within the motif from N- to C-terminus,
(3) *Helix_number(minHelix, maxHelix)* defines that a given motif contains at least *minHelix* helices and at most *maxHelix* helices,
(4) *Contact_helix(helixA, helixB, type)* defines that two helices, *helixA* and *helixB*, have a contact of the following types: *p*, *a*, or *m*, and
(5) *Contact_sheet(sheetA, sheetB, strandX, strandϒ)* defines that a contact between two sheets, *sheetA* and *sheetB*, is given by the contact between the two strands, *strandX* and *strandϒ*, with *X* as element of *A* and *ϒ* as element of *B*.

Let us start with the *Greek key* motif, which is a simple motif formed out of four antiparallel β-strands (see Fig. 9). The motif is not associated with any function, but recurrently occurs in more complex motifs or protein domains of various functions. Using the KEY notation, we can find four different topologies for this motif: "*–1, –1, 3*", "*–3, 1, 1*", "*1, 1, –3*", and "*3, –1, –1*", which describe four different geometrical arrangements of four antiparallel strands. The motif can then be described by the following four expressions:

sheet(A, a, 4, KEϒ (–1, –1, 3)) or *sheet(A, a, 4, KEϒ (–3, 1, 1))* or *sheet(A, a, 4, KEϒ (1, 1, –3))* or *sheet(A, a, 4, KEϒ (3, –1, –1))*.

To illustrate the terminology, we define some important supersecondary motifs for the four major supersecondary structure classes in the following.

3.3.1. α-Helical Motifs

(a) The *Four-Helix bundle* is a motif that consists of four α-helices arranged in a bundle. There are two types of the *Four-Helix-bundle* which differ in their connections between the helices. In the first type, the four helices are all arranged in antiparallel manner, and the second type has two pairs of parallel helices which have an antiparallel connection:

helix_number(4, 4) and *helix(A, 1)* and *helix(B, 2)*, and *helix(C, 3)* and *helix(D, 4)* and *contact_helix(A, B, a)* and *contact_helix(B, C, a)* and *contact_helix(C, D, a)*

or

helix_number(4, 4) and *helix(A, 1)* and *helix(B, 2)*, and *helix(C, 3)* and *helix(D, 4)* and *contact_helix(A, B, p)* and *contact_helix(B, C, a)* and *contact_helix(C, D,p)*.

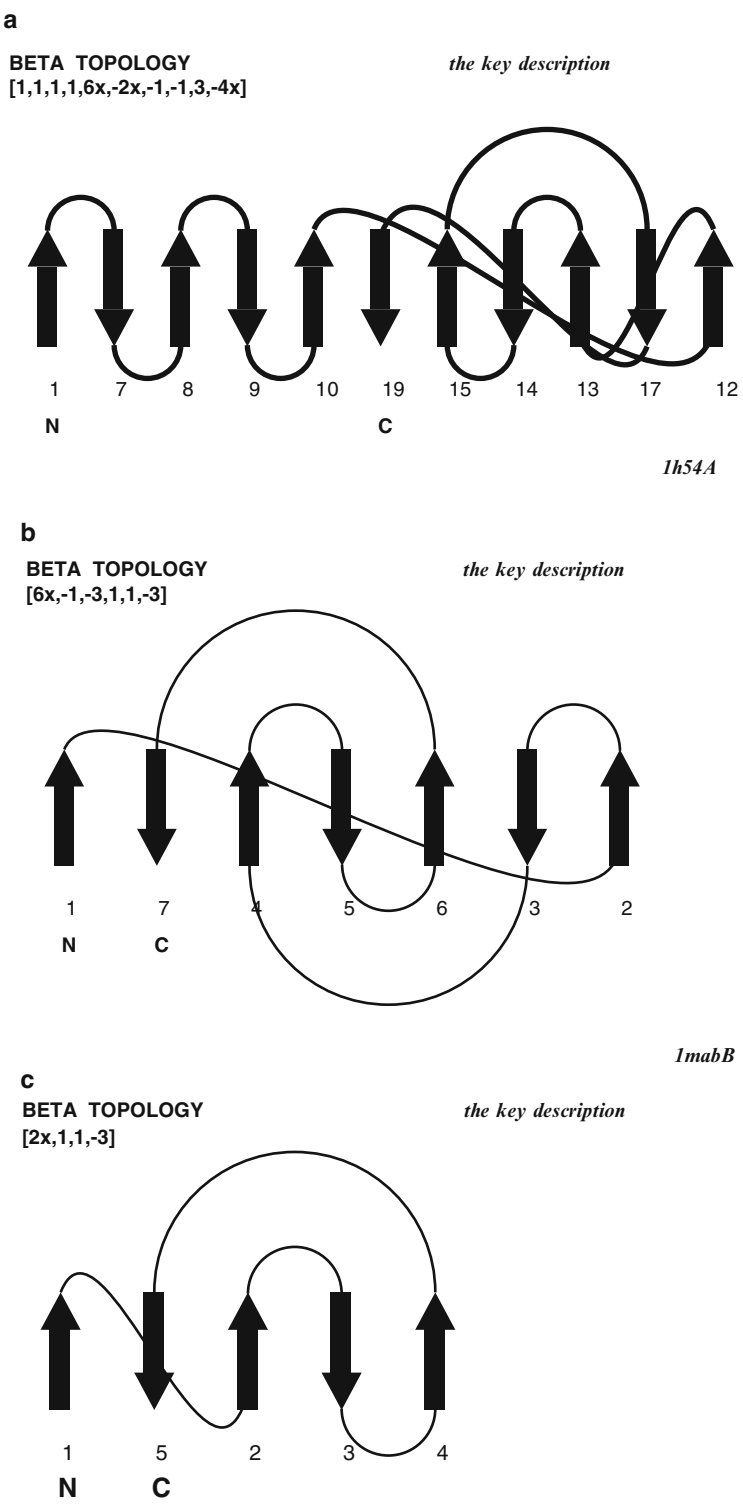


Fig. 9. (**a–c**) *Greek key* containing motifs in maltose phosphorylase from *Lactobacillus brevis*, 1h54 (20) and rat liver F1-ATPase, 1mab (21). Figure 9a–c illustrates the KEY-representations of four possible notations for the *Greek key* motif in β-graphs. Figure 9a contains the notation "*–1, –1, 3*" in 1h54, chain A, fold A, Fig. 9b the notations "*–3*, *1*, *1*" and "*1, 1, –3*" in 1mab, chain B, fold A, and Fig. 9c the notation "*1, 1, –3*" in 1mab, chain A, fold A.

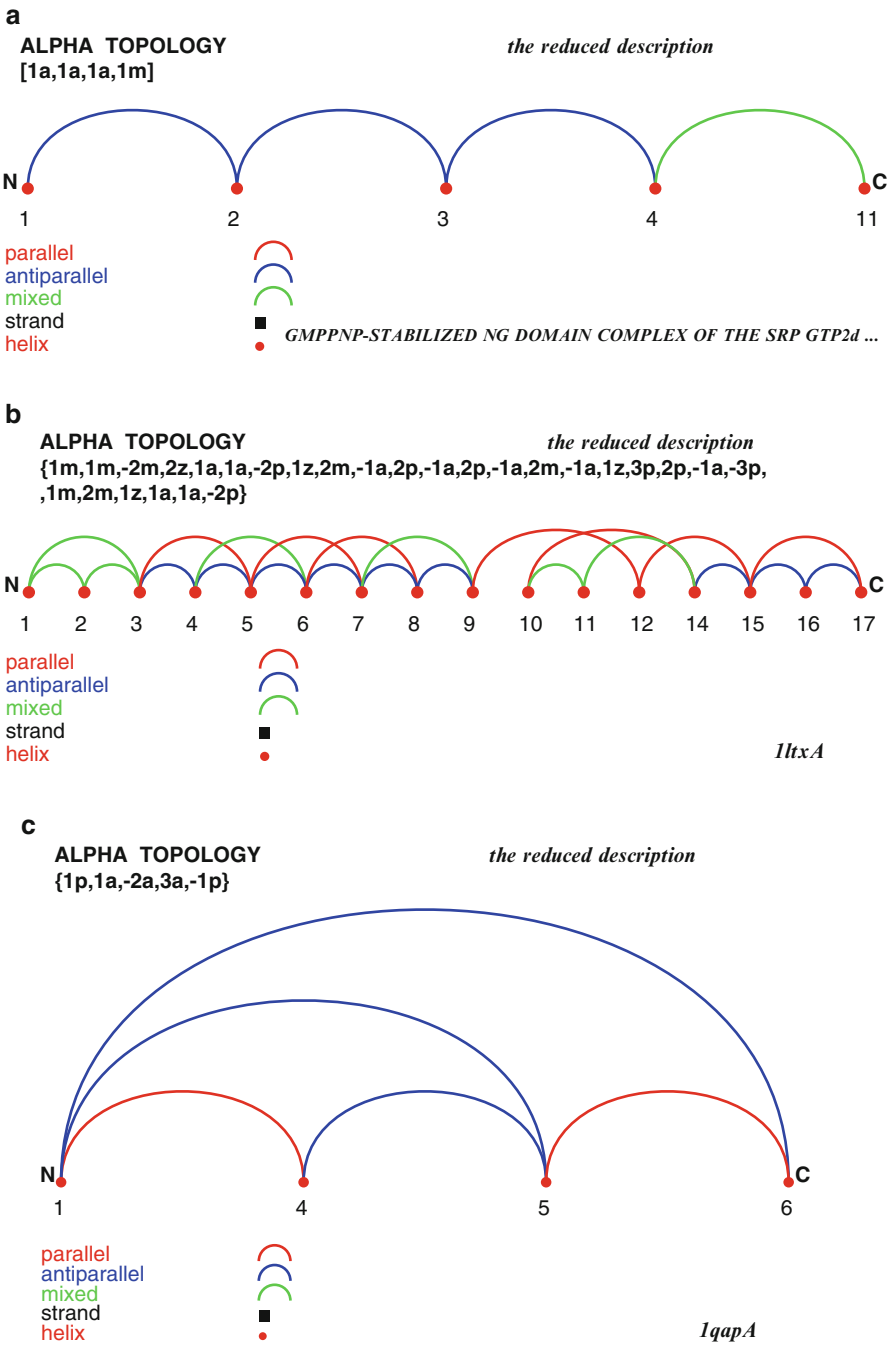


Fig. 10. (**a–c**) Examples for *Four-Helix bundles* containing motifs of type 1 in NG domain complex of the SRP GTPases Ffh and FtsY, 2j7p (22) and in Rab Escort Protein-1, 1ltx (23), and of motif 2 in quinolinic acid phosphoribosyltransferase, 1qap (24). Figure 10a depicts a *Four-Helix bundle* motif of type 1 in 2j7p, chain A, fold A, searching for "*1, 1, 1.*" Figure 10b represents also a type 1 motif, now in 1ltx, chain A, fold A, searching for "*–1, –1, –1.*" Figure 10c depicts a *Four-Helix-bundle* motif of type 2 in 1qap, chain A, fold A, searching for "*1a, 1p, 1a.*" All pictures show the RED-notation in α-graphs. Note that we do not write the type of SSEs in α-graphs (and β-graphs).

Figure 10 depicts the two types of *Four-Helix-bundles.*

(b) The *Globin fold* is composed of a bundle of eight α-helices, which are connected by short loop regions. The helices do not have a fixed arrangement, but the last two helices from the C-terminus exhibit an antiparallel arrangement:

helix_number(8, 8) and *helix(A, 1)* and *helix(B, 2)* and *helix(C, 3)* and *helix(D, 4)* and *helix(E, 5) and helix(F, 6)* and *helix(G, 7)* and *helix(H, 8)* and *contact_helix(G, H, a)*.

For an example see Fig. 11.

3.3.2. β-Sheet Motifs

(a) The *Up-and-Down barrel* is composed of a series of antiparallel β-strands. There are two major families of the Up-and-Down barrels, the ten-stranded and the eight-stranded sheet:

sheet(A, a, 8, RED(1a, 1a, 1a, 1a, 1a, 1a, 1a, –7a))

or

sheet(A, a, 10, RED(1a, 1a, 1a, 1a, 1a, 1a, 1a, 1a, 1a, –9a)).

Figure 12 depicts examples for both types of *Up-and-Down barrels.*

(b) The *Immunoglobin fold* is a two-layer sandwich. Usually, it consists of seven antiparallel β-strands, which are arranged within two β-sheets. The first sheet is composed of four and the second of three strands. Both are connected by a disulfide bond to build the sandwich structure:

sheet(A, a, 3, RED(1a, 1a)) and *sheet(B, a, 4, RED(–1a, 2a, 1a))*

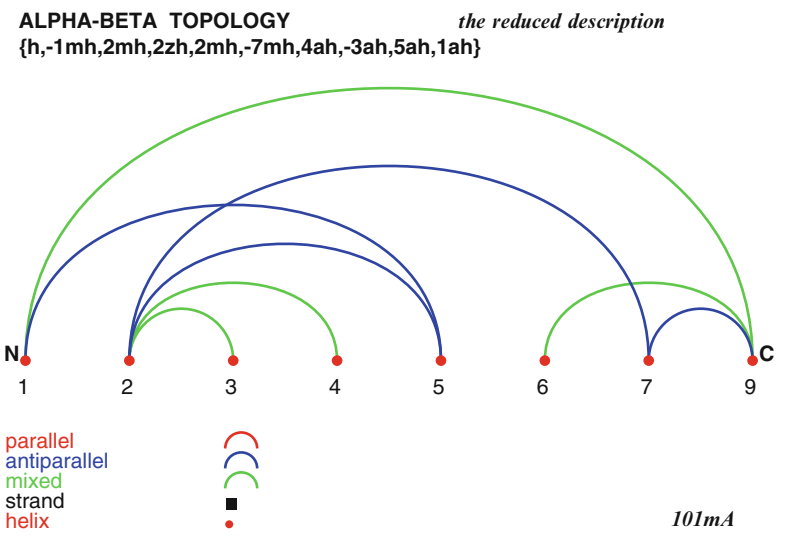


Fig. 11. A *Globin fold* in myoglobin, 101m (25). The figure shows a *Globin fold* in 101 m, chain A, fold A (25) as αβ-graph in RED-notation. Note that we jump from SSE 4 to SSE 6, because SSE 6 has only one unvisited edge nearest to the N-terminus.

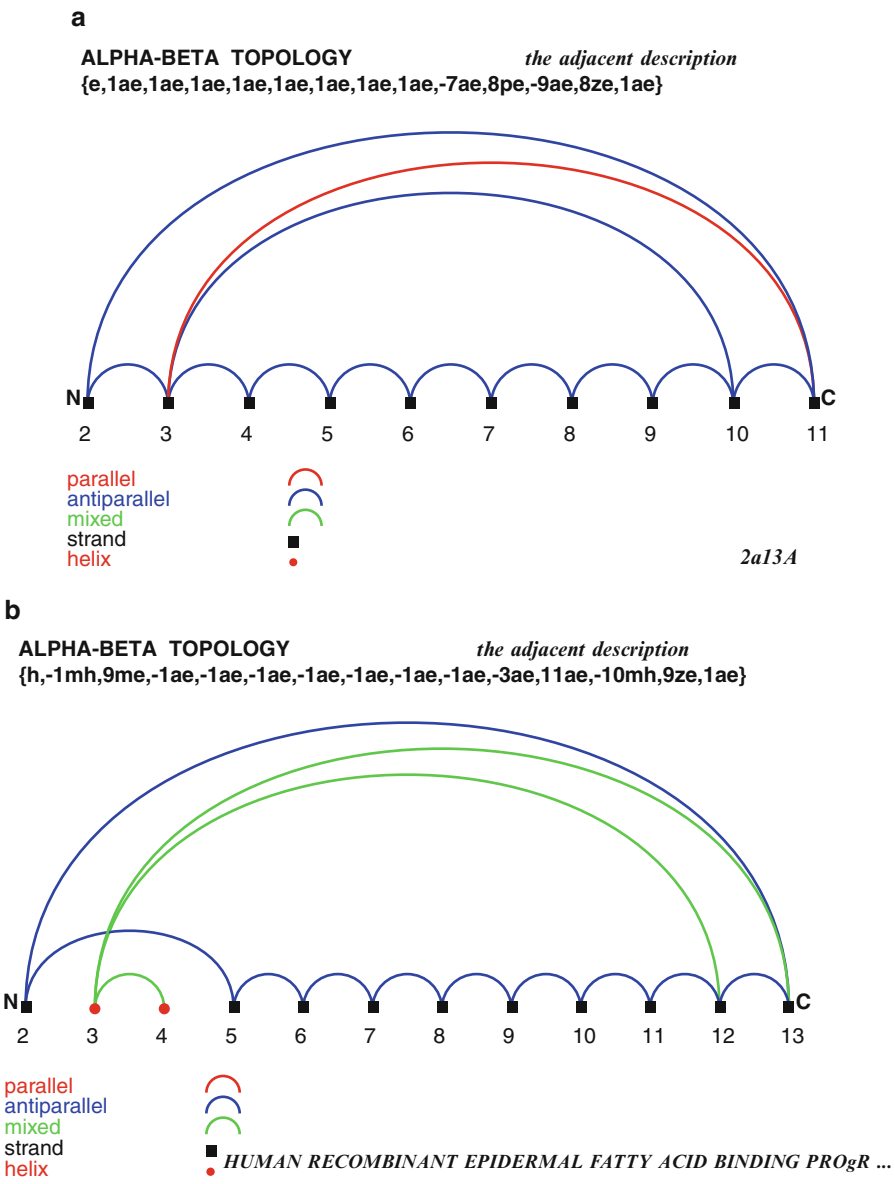


Fig. 12. (**a**) An *Up-and-Down barrel* containing motif of type 1 in the nitrophorin-like heme-binding protein from *Arabidopsis thaliana*, 2a13A (26). (**b**) An *Up-and-Down barrel* containing motif of type 2 in the *human* epidermal-type fatty acid binding protein, 1b56 (27). Figure 12a, b depicts two different types of *Up-and-Down barrels*. The first one, 2a13, chain A, fold B, contains eight strands, whereas the second one, 1b56, chain A, fold B, ten strands. The barrel structure is indicated by a large arc, which connects strands fare from each other in sequence. Note that the strands are neighbored in antiparallel manner and there are no helices in the sequence between participating strands, such as in TIM-barrel structures (*see* Subheading 3.3.3 and Fig. 17).

or

sheet(A, a, 5, RED(3a, –1a, –1a, 3a))

or

sheet(A, a, 5, RED(–3a, 1a, 1a, –3a)).

For an example see Fig. 13.

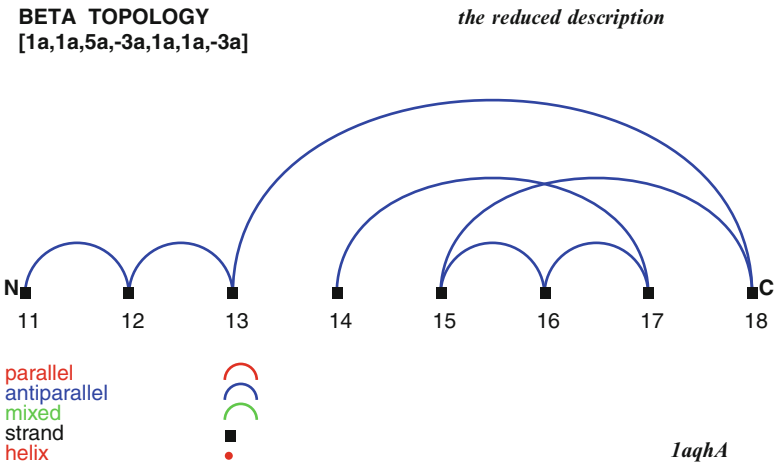


Fig. 13. An *Immunoglobin fold* in psychrophilic alpha-amylase from *Alteromonas haloplanctis*, 1aqh (28). The figure depicts an *Immunoglogulin fold* of type 3 in 1aqh, chain A, fold C, as β-graph in RED-notation. We find the motif, searching for the significant subtopology "*–3a, 1a, 1a, –3a*", which is explicitly contained in the notation.

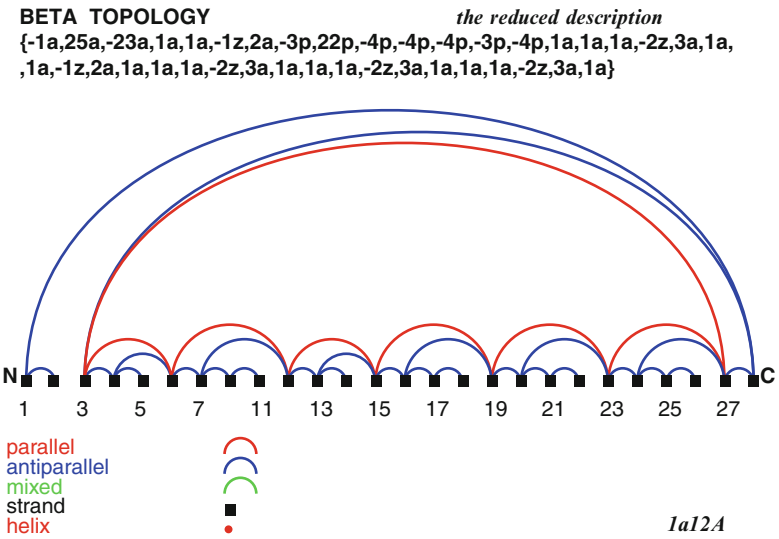


Fig. 14. A seven-bladed β-*Propeller* containing motif in the regulator of chromosome condensation, 1a12 (29). The figure depicts a seven-bladed β-*Propeller* containing motif in 1a12, chain A, fold A (29), represented as β-graph in RED-notation. We can easily identify the seven sheets each formed by four β-strands, and mainly neighbored in antiparallel manner.

(c) The β-*Propeller* motif consists of four to eight β-sheets, which are arranged around the center of the protein. Each sheet contains four antiparallel β-strands. One sheet defines one of the propeller blades. To build a four-bladed β-*Propeller* four β-sheets are grouped together:

sheet(A, a, 4, RED(1a, 1a, 1a)) and *sheet(B, a, p, RED(1a, 1a, 1a))* and *sheet(C, a, p, RED(1a, 1a, 1a))* and *sheet(D, a, p, RED(1a, 1a, 1a))* and *contact_sheet(A, B, 4, 1)* and *contact_sheet(B, C, 4, 1)* and *contact_sheet(C, D, 4, 1)*.

Fig. 14 gives an example for a seven-bladed β-*Propeller*.

(d) The *Jelly roll* motif has a barrel structure, which seems like a jelly roll. The barrel includes eight β-strands, which build a two-layer sandwich each holding four strands. Richardson (12) describes the Jelly roll motif as being a Greek key motif with an additional extra "swirl":

sheet(A, a, 4, RED(*3a, –1a, –1a, 3a, –5a, –1a, 7a*))

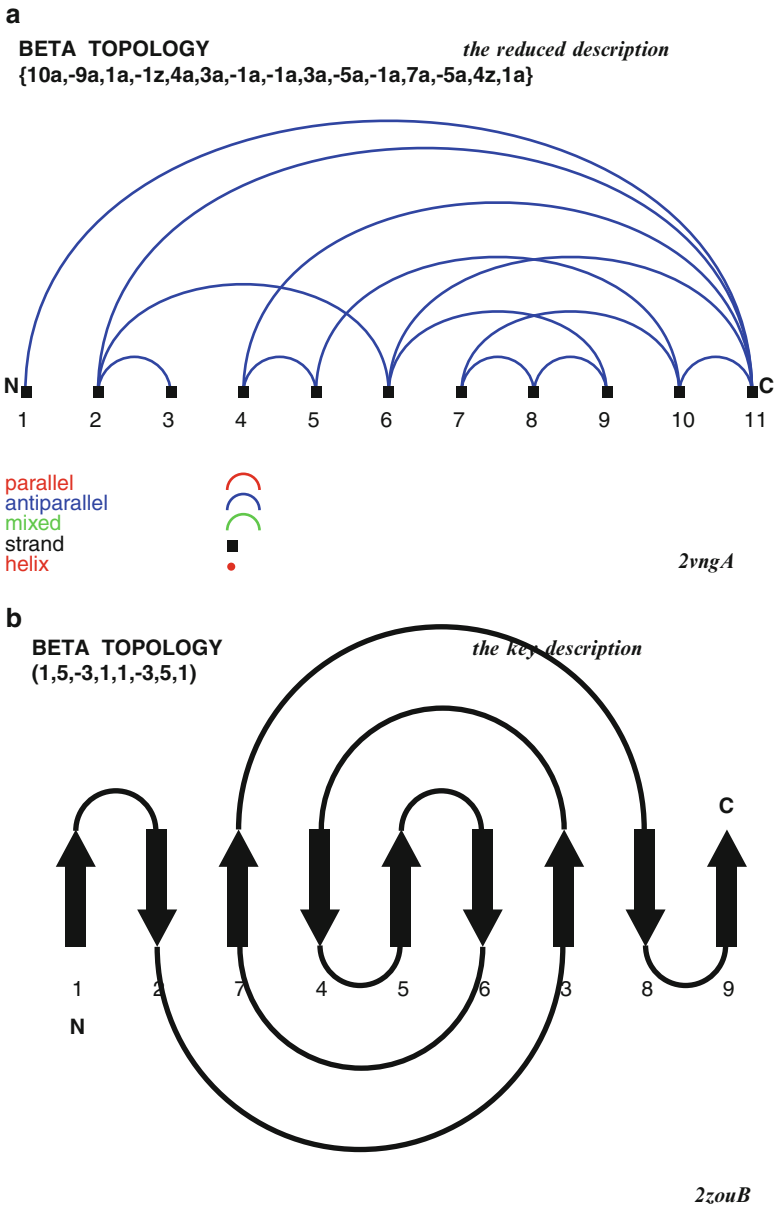


Fig. 15. (**a, b**) *Jelly roll* containing motifs in family 51 of carbohydrate-binding module, 2vng (30), and in the F-spondin reeler domain, 2zou (31). The figure illustrates two *Jelly roll* containing motifs as β-graphs, one in RED-notation of 2vng, chain A, fold A (30), in Fig. 15a, and one in KEY-notation of 2zou, chain B, fold B (31), in Fig. 15b. The first one contains the motif "*3a, –1a, –1a, 3a, –5a, –1a, 7a*" explicitly in the RED-notation. The second one, shown in Fig. 15b, depicts the typical *Jelly roll* topology in KEY-representation.

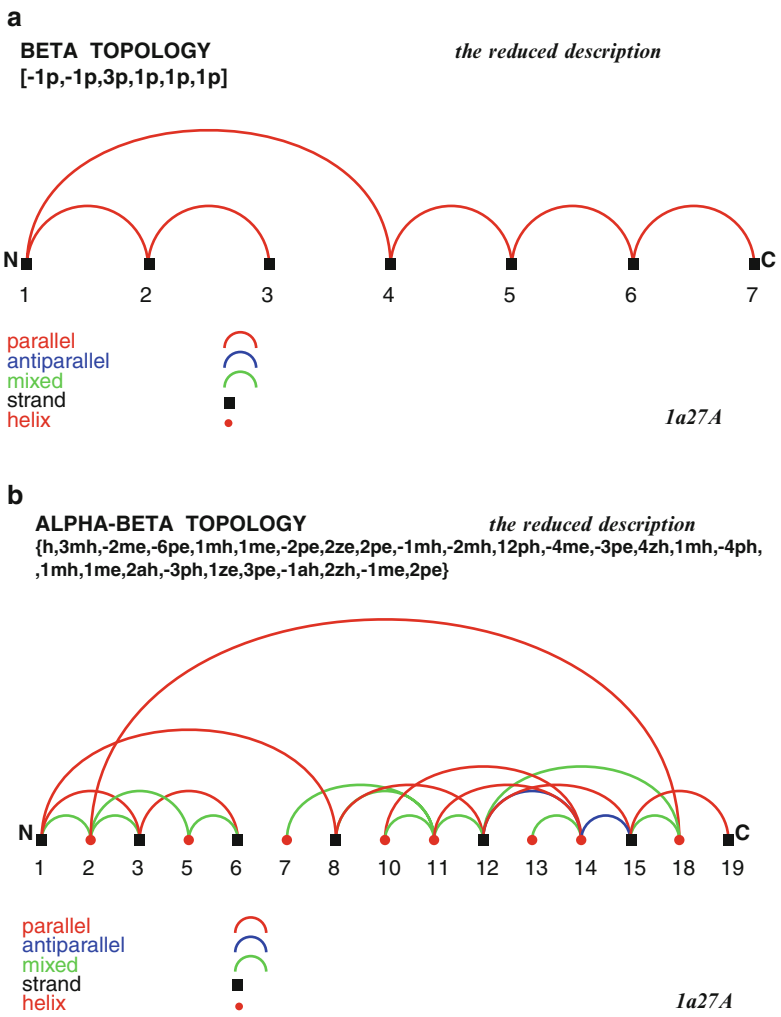


Fig. 16. (**a, b**) A *Rossman fold* containing motif in *human* type I 17Beta-hydroxysteroid dehydrogenase, 1a27 (32). The figure depicts the *Rossman fold* containing motif in 1a27, chain A, FG A (32), both in RED-notation. Figure 16a represents the αβ-graph, including the helices and the sheet. Figure 16b illustrates the β-graph in the RED-notation, which explicitly contains the subtopology "*–1p, –1p, 3p, 1p, 1p*".

Figure 15 illustrates two examples for *Jelly roll* motifs.

3.3.3. α/β Motifs

α/β motifs are mainly composed of small β-α-β-units with a helix neighbored to two parallel β-strands:

(a) The *Rossman fold* consists of β-α-β-units forming an open twisted parallel β-sheet surrounded by α-helices:

sheet(A, p, 6, RED(–1p, –1p, 3p, 1p, 1p)) and *helix_number(3, 4)* and *helix(B, 2)* and *helix(C, 4)* and *helix(D, 7)*.

For an example see Fig. 16.

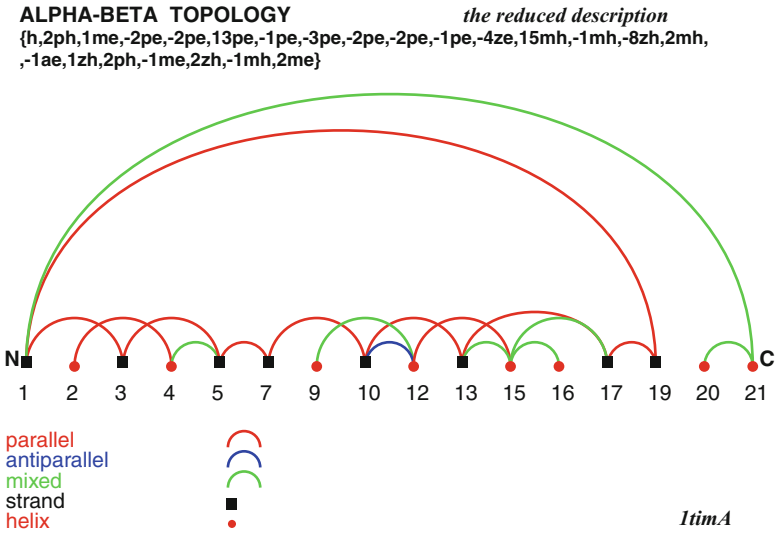


Fig. 17. The *TIM-barrel* containing motif in triose phosphate isomerase from *Chicken* muscle, 1tim (33). The figure illustrates a TIM-barrel containing motif in 1tim, chain A, fold A (33), as αβ-graph in RED-notation. In contrast to *Up-and-Down barrels* (Subheading 3.3.2 and Fig. 12) helices are located between strands in the sequence, which are neighbored in parallel manner.

(b) The *TIM-barrel* motif includes eight β-strands and eight α-helices, occurring alternately in the sequence. The β-strands define the inner barrel structure surrounded by α-helices outside the barrel:

sheet(A, p, 8, RED(1p, 1p, 1p, 1p, 1p, 1p, 1p)) and *helix_number(5, 9)* and *helix(B, 2)* and *helix(C, 4)* and *helix(D, 6)* and *helix(E, 8)* and *helix(F, 10)*.

Figure 17 depicts the classical *TIM-barrel* motif.

3.3.4. α + β *Motifs*

(a) The *Ubiquitin roll* motif is a composite motif of α-helices and a mixed β-sheet. According to the topology of the β-sheet we differentiate two types. Type one contains one helix, whereas the type 2 motif can involve one or two helices:

sheet(A, m, 4, RED(1a, –3p, 1a)) and *helix_number(1, 1)* and *helix(B, 3)*,

or

sheet(A, m, RED(–1a, 4p, –2a, 1a)) and *helix_number(1, 2)* and *helix(B, 3)* or/and *helix(B, 6)*

Figure 18 depicts two examples of *Ubiquitin roll* containing motifs.

(b) αβ-*Plaits* have four or five β-strands, which form an antiparallel β-sheet. In between the sheets two or more helices are arranged:

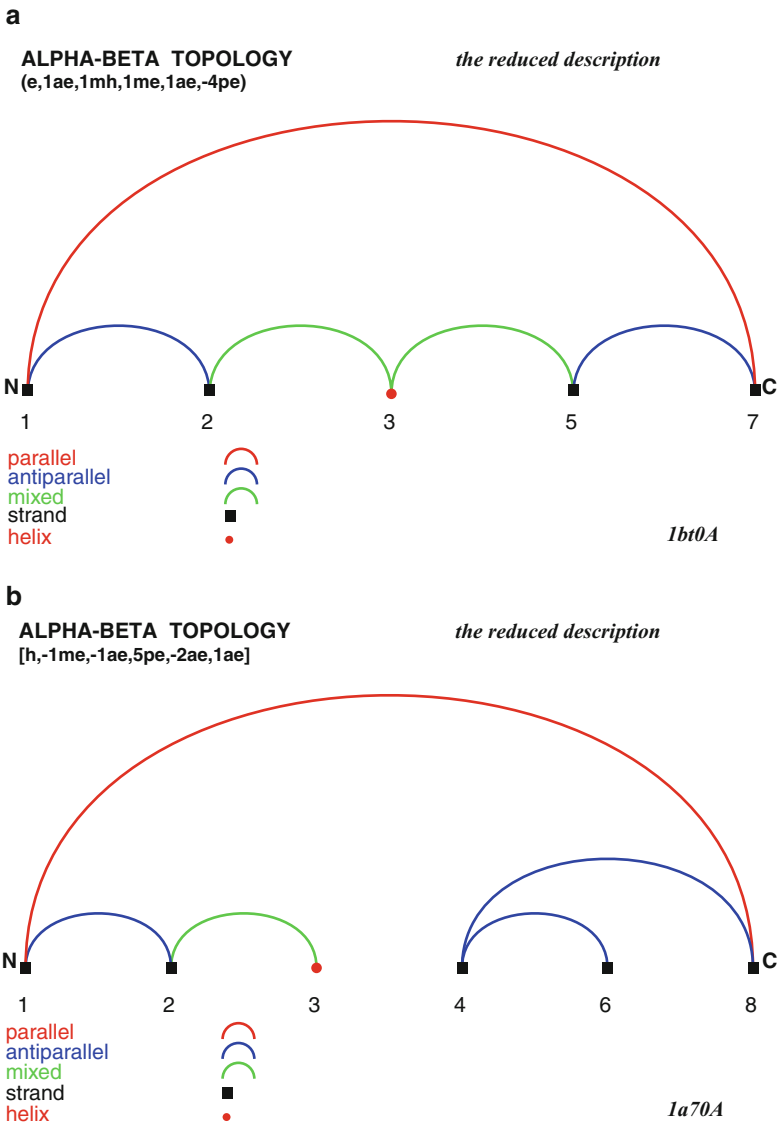


Fig. 18. (**a, b**) *Ubiquitin roll* containing motifs of type 1 in *Arabidopsis* rub1, 1bt0 (34), and of type 2 in ferredoxin I from *Spinacia oleracea*, 1a70 (35). Figure 18a depicts the type 1 motif, occurring in the 1bt0, chain A, fold A (34), in the αβ-graph in RED-notation. We can easily identify the significant β-sheet subtopology, "*1a, –3p, 1a*", here as "*–1a, 3p, –1a*", and the one helix at position 3. Figure 18b represents the type 2 motif, occurring in 1a70, chain A, fold A (35), in the αβ-graph in ADJ-notation. Here, we can see the β-sheet topology, "*–1a, 4p, –2a, 1a*", and the two helices at positions 3 and 6, which are typical for this motif.

sheet(A, a, 4, RED(1a, –2a, 3a)) and *helix_number(2, 2)* and *helix(B, 2)* and *helix(C, 5)*

or

sheet(A, m, 4, RED(1a, –2p, 3a)) and *helix_number(2, 2)* and *helix(B, 2)* and *helix(C, 5)*.

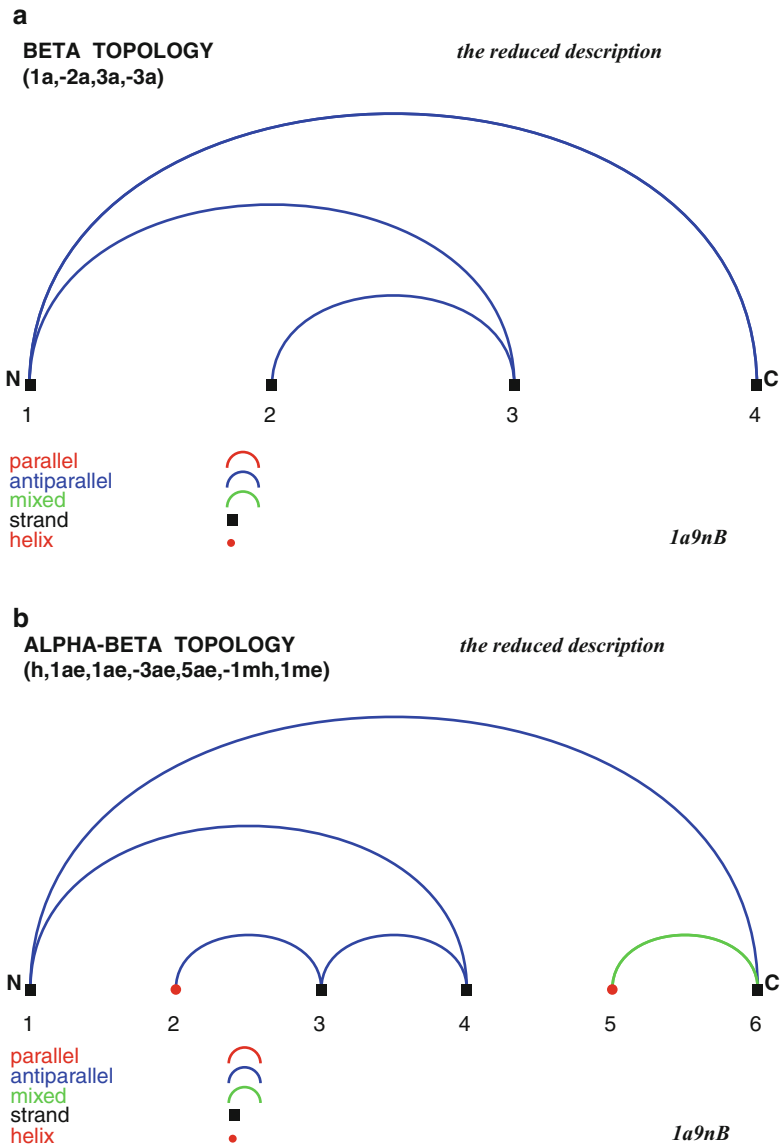


Fig. 19. (**a–c**) *αβ-Plaits* containing motifs of type 1 in the spliceosomal U2B"-U2A' protein complex, 1a9n *(36)*,, and type 2 in the ribosomal protein S6 from *Thermus thermophilus*, 1ris *(37)*. The figure b depicts the motif 1 in 1a9n, chain B, fold A, one as β-graph in RED-notation, exhibiting the subtopology "*1a, –2a, 3a*" (Fig. 19a), and one as αβ-graph in ADJ-notation, showing the positions of the helices in the topology (Fig. 19b). Figure 19c represents the αβ-topology in RED-notation, illustrating the helix positions and the typical β-sheet topology, "*1a, –2p, 3a*".

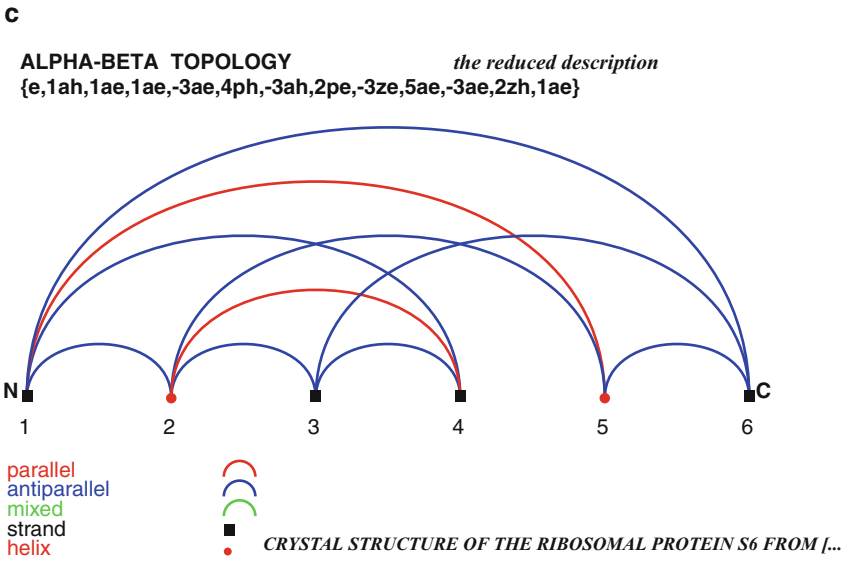


Fig. 19. (continued)

Figure 19 illustrates examples for the two types of αβ-*Plaits*.

4. Notes

1. First of all, decide at which level of abstraction you want to work, whether you want to consider topologies of helices *and* strands, i.e., αβ-protein graphs and αβ-folding graphs, or *only* helices, i.e., α-protein graphs and α-folding graphs, or *only* strands and sheets, i.e., β-protein graphs and β-folding graphs.
2. Then, decide whether you want to consider all SSEs, i.e., ADJ- or SEQ-notation, or only those which belong to the FG under consideration, i.e., RED- or KEY-notation.
3. Let us summarize the derivation of the notation for nonbifurcated FGs for ADJ and RED:
   (a) Start with the SSE with only one spatial neighbor nearest to the N-terminus. If you consider αβ-FGs, note the type of SSE, *e* or *h*.
   (b) Follow the edges and note the difference of SSE numbers. Indicate backward movement to the N-terminus by "–".
   (c) Add the edge label and the type of the SSE at the end of the edge, and put a comma.
   (d) Repeat (b) and (c) until all edges have been visited.

4. Let us summarize the derivation of the notation for bifurcated FGs for ADJ and RED:
   (a) Perform (a) to (c), as in Note 3, until the end of the path.
   (b) If there are still unvisited edges, go back or forwards to the SSE with only one neighbor nearest to the N-terminus, where an unvisited edge starts. Note the difference of SSE numbers, indicating backward movement to the N-terminus by "–", write a "z" and the type of the SSE, where the edge ends, and put a comma.
5. PTGL is an online database for supersecondary structure topologies that can be used for any kind of theoretical protein structure analysis and search for protein subtopologies and predefined motifs in different notations at different levels of abstraction. Searching for a special topology, all proteins, which contain the requested motif will be retrieved, and the corresponding two- and three-dimensional pictures with additional information about participating SSEs will be displayed. Additionally, a search for predefined motifs is available. The mathematically unique description enables exhaustive and comparative investigation of protein structure topologies.

Acknowledgement

We thank Thomas Steinke for many stimulating discussions and Norbert Dichter for technical support.

References

1. Bernstein FC, Koetzle TF, Williams GJ et al (1977) The Protein Data Bank: a computer-based archival file for macromolecular structures. J Mol Biol 112:535–542
2. Berman HM, Westbrook J, Feng Z et al (2000) The Protein Data Bank. Nucl Acids Res 28:235–242
3. Murzin AG, Brenner SE, Hubbard T et al (1995) SCOP: a structural classification of proteins database for the investigation of sequences and structures. J Mol Biol 247:536–540
4. Orengo CA, Michie AD, Jones DT et al (1997) CATH: a hierarchic classification of protein domain structures. Structure 5: 1093–1108
5. May P, Kreuchwig A, Steinke T et al (2010) PTGL: a database for secondary structure-based protein topologies. Nucl Acids Res 38:D326–330
6. Frishman D, Argos P (1995) Knowledge-based secondary structure assignment. Proteins 23:566–579
7. Richards FM, Kundrot CE (1988) Identification of structural motifs from protein coordinate data: secondary structure and first-level supersecondary structure. Proteins 3:71–84
8. Sklenar H, Etchebest C, Lavery R (1989) Describing protein structure: a general algorithm yielding complete helicoidal parameters and a unique overall axis. Proteins 6:46–60
9. Cubellis MV, Cailliez F, Lovell SC (2005) Secondary structure assignment that accurately reflects physical and evolutionary characteristics. BMC Bioinform 6(Suppl 4):S8
10. Taylor WR (2001) Defining linear segments in protein structure. J Mol Biol 310:1135–1150
11. Kabsch W, Sander C (1983) Dictionary of protein secondary structure: pattern recognition

of hydrogen-bonded and geometrical features. Biopolymers 22:2577–637

12. Koch I, Lengauer T (1997) Detection of distant structural similarities in a set of proteins using a fast graph-based method. In proceedings of 5th international conference on intelligent systems for molecular biology, p 167–178
13. Koch I, Lengauer T, Wanke E (1996) An algorithm for finding maximal common subtopologies in a set of protein structures. J Comp Biol 3:289–306
14. Bentley GA, Boulot G, Karjalainen K et al (1995) Crystal structure of the beta chain of a T cell antigen receptor. Science 267:1984–1987
15. Richardson JS (1981) The anatomy and taxonomy of protein structure. Adv Protein Chem 34:167–339
16. Richardson JS (1977) Beta-sheet topology and the relatedness of proteins. Nature 268: 495–500
17. Koch I (1998) Ein graphentheoretischer Ansatz zum paarweisen und multiplen Vergleich von Proteinstrukturen. (in German) Wissenschaft und Technik Verlag Berlin
18. Brown NR, Noble ME, Lawrie AM et al (1999) Effects of phosphorylation of threonine 160 on cyclin-dependent kinase 2 structure and activity. J Biol Chem 274:8746–8756
19. Kreuchwig A (2007) Development and comparing investigations of search patters for topological protein structure motifs using graph-theory (in German). Bachelor's Thesis at Free University Berlin
20. Egloff M-P, Uppenberg J, Haalck L et al (2001) Crystal structure of Maltose phosphorylase from Lactobacillus Brevis: unexpected evolutionary relationship with Glucoamylases. Structure 9:689–697
21. Bianchet MA, Hullihen J, Pedersen PL et al (1998) The 2.8-A structure of rat liver F1-ATPase: configuration of a critical intermediate in ATP synthesis/hydrolysis. Proc Natl Acad Sci USA 95:11065–11070
22. Gawronski-Salerno J, Freymann DM (2007) Structure of the GMPPNP-stabilized NG domain complex of the SRP GTPases Ffh and FtsY. J Struct Biol 158:122–128
23. Pylypenko O, Rak A, Reents R et al (2003) Structure of Rab escort protein-1 in complex with Rab geranylgeranyltransferase. Mol Cell 11:483–494
24. Eads JC, Ozturk D, Wexler TB et al (1997) A new function for a common fold: the crystal structure of quinolinic acid phosphoribosyltransferase. Structure 5:47–58
25. Smith RD (1999) Correlations between bound N-alkyl isocyanide orientations and pathways for ligand binding in recombinant myoglobins. Thesis, Rice.
26. Bianchetti CM, Blouin GC, Bitto E et al (2010) The structure and NO binding properties of the nitrophorin-like heme-binding protein from Arabidopsis thaliana gene locus At1g79260.1. Proteins 78:917–931
27. Hohoff C, Borchers T, Rustow B et al (1999) Expression, purification, and crystal structure determination of recombinant human epidermal-type fatty acid binding protein. Biochemistry 38:12229–12239
28. Aghajari N, Feller G, Gerday C et al (1998) Crystal structures of the psychrophilic alpha-amylase from Alteromonas haloplanctis in its native form and complexed with an inhibitor. Protein Sci 7:564–572
29. Renault L, Nassar N, Vetter I et al (1998) The 1.7 A crystal structure of the regulator of chromosome condensation (RCC1) reveals a seven-bladed propeller. Nature 392:97–101
30. Gregg KJ, Finn R, Abbott DW et al (2008) Divergent modes of glycan recognition by a new family of carbohydrate-binding modules. J Biol Chem 283:12604–12613
31. Nagae M, Nishikawa K, Yasui N et al (2008) Structure of the F-spondin reeler domain reveals a unique beta-sandwich fold with a deformable disulfide-bonded loop. Acta Cryst D 64:1138–1145
32. Mazza C (1997) Human type I 17beta-hydroxysteroid dehydrogenase: site directed mutagenesis and X-ray crystallography structure-function analysis. PhD Thesis at Universite Joseph Fourier
33. Banner DW, Bloomer A, Petsko et al (1976) Atomic coordinates for triose phosphate isomerase from chicken muscle. Biochem Biophys Res Commun 72:146–155
34. Rao-Naik C, delaCruz W, Laplaza JM et al (1998) The rub family of ubiquitin-like proteins. Crystal structure of Arabidopsis rub1 and expression of multiple rubs in Arabidopsis. J Biol Chem 273:34976–34982
35. Binda C, Coda A, Aliverti A et al (1998) Structure of the mutant E92K of [2Fe-2S] ferredoxin I from Spinacia oleracea at 1.7 A resolution. Acta Cryst D 54:1353–1358
36. Price SR, Evans PR, Nagai K (1998) Crystal structure of the spliceosomal U2B"-U2A' protein complex bound to a fragment of U2 small nuclear RNA. Nature 394:645–650
37. Lindahl M, Svensson LA, Liljas A et al (1994) Crystal structure of the ribosomal protein S6 from Thermus thermophilus. EMBO J 13:1249–1254

Chapter 3

Up, Down, and Around: Identifying Recurrent Interactions Within and Between Super-secondary Structures in β-Propellers

Søren Thirup, Vikas Gupta, and Esben M. Quistgaard

Abstract

The examination and analysis of super-secondary structures or other specific structural patterns associated with particular functions, sequence motifs, or structural contexts require that the set of structures examined shares the same feature. This seems obvious but in practice this may often present problems such as identifying complete sets of data avoiding false positives. In the case of the β-propeller structures the symmetry of the propeller creates problems for many structure similarity search programs. Here we present a procedure that will identify propeller structures in PDB and assign them to the different N-propeller types. In addition we outline methods to examine similarities and differences within and between the four stranded up-and-down blades of the β-propeller.

Key words: Beta-propellers, Structural similarity, Classification, Beta sheet

1. Introduction

The structure of a protein or a domain is described at increasing levels of complexity: primary, secondary, and tertiary structure. These descriptions are quite simple, but extensive: the sequence of amino acids, a list of hydrogen bonds between main chain atoms, and a list of atomic coordinates. For the human mind such lists are difficult to handle, so we tend to identify patterns that occur frequently and classify protein structures according to the presence of these patterns. In this way the secondary structure becomes a sequence of secondary structure elements: β-strands, α-, 3_{10}-, or π-helices and specific turn types. The tertiary structures become classified by fold, topology, motif, and super-secondary structure, but unfortunately these classifications are not completely distinct.

Alexander E. Kister (ed.), *Protein Supersecondary Structures*, Methods in Molecular Biology, vol. 932,
DOI 10.1007/978-1-62703-065-6_3,

Fold and topology are often used as equivalent (cf. CATH (1) and SCOP (2)), and so are motif and super-secondary structure. The distinction between these terms would be useful and we suggest that a super-secondary structure should be well defined in terms of sequence length and the set of defining hydrogen bonds. Originally super-secondary structure was defined as a recurring spatial arrangement of a number of secondary structural elements (3). In addition to this we would require that the number of secondary structural elements should be small, should be consecutive in sequence, and should be without intermittent secondary structural elements. The term motif could then be used for a structural arrangement similar to a super-secondary structure, as judged by root mean square deviation. In a motif a secondary structural element could be missing or there could be additional secondary structural elements inserted between secondary structural elements of the super-secondary structure. For the description of the β-propellers below only a subset of blades display the up-and-down β-sheet super-secondary structure but all blades of the propeller have the up-and-down β-sheet motif since the fourth strand in many cases is not forming hydrogen bonds with strand 3.

1.1. The β-Propeller Structures

A β-propeller is a toroidal fold with 4–8 or 10 super-secondary structural units, called propeller blades, arranged in a circular arrangement with a pseudo *N*-fold axis (Fig. 1). The blades form up-and-down antiparallel β-sheets with usually four strands. However, the inner or outermost strand is occasionally so irregular that it cannot be recognized as a β-strand by secondary structure assignment methods, and other unusual arrangements may also be seen, e.g., the blade may be extended with a fifth or even more strands. The structure of the influenza virus neuraminidase domain revealed the first example of a β-propeller structure in this case consisting of six blades (4). Since then around 1,000 structures containing β-propeller domains have been determined. Additional structures have been termed propeller structures such as the three-bladed (PDB:1N7V) and pinwheel structures (e.g., PDB:1SUU, PDB:1WP5, PDB:1ZVT). These will however not be considered here as they are not showing the classical features of a β-propeller, i.e., the consecutive up-and-down sheet and a pseudo *N*-fold axis relating the individual propeller blades.

1.2. Topology of the Propeller

The top and bottom of the propeller domain were defined in the description of the neuraminidase structure: "The first strand of each sheet, near the centre of the subunit, is entered from the top …" (4). Using this definition blade n is related to blade $n+1$ by a counterclockwise rotation of 360°/N when looking down the propeller axis (Fig. 1a). To date all propellers have been found to have this hand. This is caused by the twist of the β-sheet, which places the C-terminal of the fourth strand pointing in the counterclockwise direction. The blades are often evenly distributed to form a circular

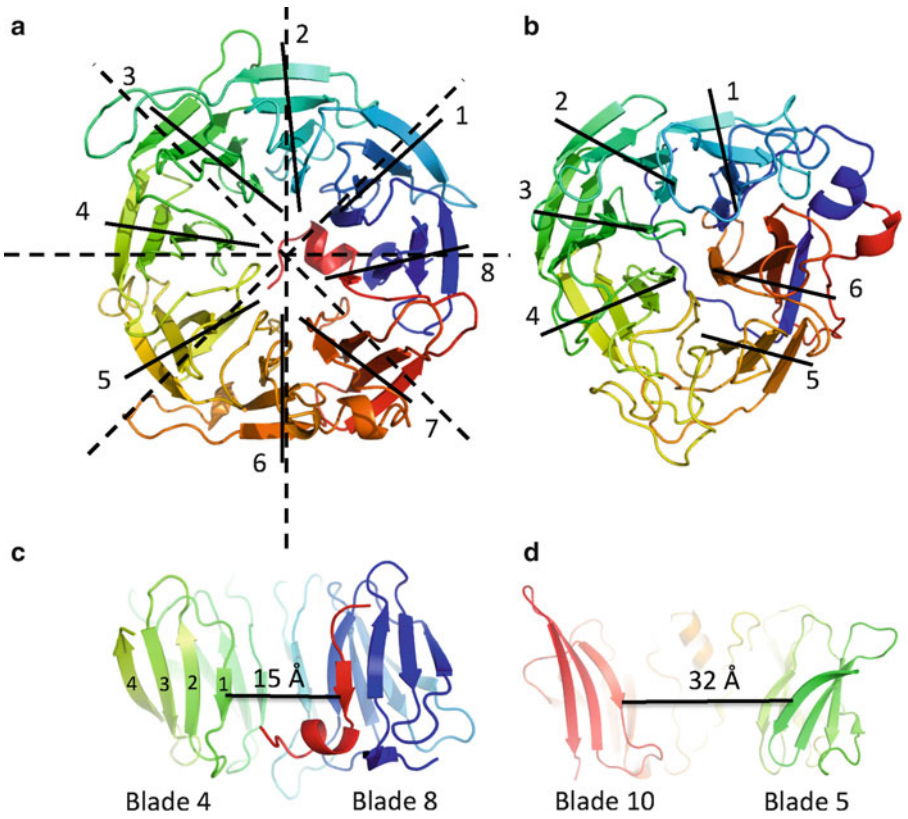


Fig. 1. (**a**) Cartoon of an eight-bladed propeller (PDB:1AOF (26)). Blades are numbered 1 through 8. *Dashed lines* indicate direction of ideal propeller blades separated by 360/8 = 45°. Full lines indicate observed direction of blades. (**b**) Highly irregular six-bladed propeller (PDB:1A14 (27)) where especially blade 5 deviates from the ideal direction. (**c**) Cross section of the eight-bladed propeller in (**a**) viewed from the side. The innermost strands of blades 4 and 8 are parallel to the propeller axis. Strands in blade 4 is numbered 1 through 4. The *bar* indicates the distance between Cα atoms in the two inner strands of blades 4 and 8. (**d**) Example of inner strands of blades deviating from the propeller axis in a ten-bladed propeller (PDB:3F6K (28)). The *bar* indicates the distance between Cα atoms of the two inner strands in the equatorial plane of the propeller.

domain, but for some eight- and the ten-bladed propellers an oval shape of the propeller is observed. In these cases the angles between blades deviate from 360°/N. In a preliminary examination of the angle between blades of propeller structures we found that the sum of the angles often exceeds 360°. This is caused by one or more of the blades being rotated around an axis parallel to, but not coincident with, the propeller axis. In extreme cases this rotation of a blade may be nearly 90° (Fig. 1b).

The inner strand of the blades is usually nearly parallel with the propeller axis (Fig. 1c). The top end is a little further from the axis, thereby creating a funnel-shaped cavity along the propeller axis with the opening at the top of the domain. Often the center of the top face or the cavity is found to be a ligand binding site. Again we find examples among the large number of propeller structures known at present where the inner strand deviates significantly from the propeller axis (Fig.1d).

1.3. Blade Interactions

The interface between consecutive blades is primarily formed by hydrophobic interactions especially of residues from strands 2 and 3. Hydrogen bonds and salt bridges between side chains of neighboring blades are also observed but they do not seem to be important for general structural stability as these residues are not part of conserved sequence patterns. The importance of the size of the hydrophobic residues in the packing of the blades was first pointed out by Murzin, who noted that the packing at the inner strand is mostly by intercalated small residues. In contrast the packing of the central strands 2 and 3 involves large hydrophobic residues with their Cβ atoms pointing towards each other, thereby creating the largest possible distance between strands (5). This is also reflected in the frequency of large hydrophobic residues in the central hydrophobic core (6).

The closure of the propeller is achieved by the N- and C-terminal blade packing together in the same fashion as internal blades. In many cases however there is a closing blade composed of strands of the N-terminal and the C-terminal of the propeller—this is usually termed Velcro closure. In some cases closure may be reinforced by the presence of disulfide bridges. The closing blade may be composed in different ways: three strands from the N-terminal + 1 from the C-terminal, two from both, or 1-N-terminal and three from the C-terminal. In all cases the C-terminal strands form the inner most part of the blade.

1.4. Loops and Connections

The binding site and catalytic site are mostly found at the top at the entry to the central tunnel (7, 8). This face of the propeller is composed of the linker between blades and the loop between strands 2 and 3. These connections are also where the largest variability is observed. For Asp-box propellers it has been shown that there is much higher structural variability in these loops than at the other face of the propeller (9). Not surprising since the feature characterizing the asp-box propellers is the conserved loop between strands 3 and 4. The short loop found between strands 1 and 2 is however also observed in other propeller types (6).

Propeller domains constitute parts of larger proteins and may occur anywhere in the sequence. Extra domains may even be inserted between blades as in *V. cholerae* neuraminidase PDB:1KIT (10). In this structure an N-terminal β-sandwich domain precedes one strand, part of a Velcro closure in the last blade, and the first two blades of the six-bladed β-propeller, which are followed by another β-sandwich domain, and finally the C-terminal part of the chain forms the four remaining blades. Domain insertions between strands of a blade are also observed, e.g., PDB:2DOV (11), PDB:3OC0 (12), and PDB:2UVK (13), but these are small domains consisting of only a few secondary structure elements.

1.5. The Up-and-Down Sheets

The generic propeller blade is a four stranded up-and-down antiparallel sheet, each strand consisting of five to six residues. The observed structures reveal that strands 2 and 3 are nearly invariably present whereas much larger variation is observed for strands 1 and 4. Strand 1 tends to be short and in many six-bladed propellers it forms a bulge in one of the blades which fills the central cavity of the propeller, e.g., PDB:1KIT (10) and PDB:1L0G (14), but in general its variation is restricted by the limited space near the propeller axis. In contrast strand 4 at the outer rim is much less restricted and consequently a much larger variation is observed. In many cases this strand is still in the extended conformation, but dislocated preventing hydrogen bond formation with strand 3. In the beta lactamase inhibitor the fourth strand is replaced by an α-helix in all seven blades. In a preliminary analysis of strand angles in the six-bladed propellers classified in CATH, we found that the average angles between strands 1 and 2 (27°) and 2 and 3 (26°) were nearly the same but the standard deviation, 19° and 14°, respectively, was higher for the innermost strands. The average angle between strands 3 and 4 was 5° higher with a standard deviation of 22°. As mentioned above the interface between blades is primarily formed by large hydrophobic side chains located in strands 2 and 3; hence it is reasonable to conclude that strands 2 and 3 are responsible for forming a rigid scaffold of the propeller.

1.6. Propeller Families

Overall the sequences of the propeller proteins do not show any significant similarity. For many propellers however the structural repeat is evident in the sequences showing a period of no less than 35 amino acids. The repeat size itself and number of repeats along with additional criteria have in some cases been successfully used to assign a protein of unknown structure to the propeller fold (15). Some sequence motifs are recurrent in different propellers and have been used to define propeller subclasses. In Pfam 39 families are defined as belonging to the propeller clan plus the additional BNR-family corresponding to the Asp-box propellers. A thorough analysis of sequences and structures suggests that the propellers, apart from the asp-box proteins, share a common origin (16). In the subclasses the conserved residues tend to include a large hydrophobic side chain involved in blade-to-blade packing, a glycine located in one of the loops, and an aspartate or asparagine side chain also located near the end of a β-strand in or the loop. The side chain of the aspartate or asparagine typically forms intra-blade interactions.

For making structural comparison of the β-propellers we need to extract the known structures from the PDB. Ideally structures would be classified when they are deposited in the PDB. This is however not the case. At present the newest release of CATH is based on PDB release of November 2010 and the latest SCOP release is from 2009. To identify propeller structures in the most

recent version of the PDB we have devised a scheme using the Dali search tool. We use classified structures of CATH and SCOP as queries and as markers to define cutoff limits for reliably assigning structures to the different propeller types.

2. Materials

2.1. Databases

PDB: http://www.rcsb.org/pdb/ (17).
CATH: http://www.cathdb.info/ (1).
SCOP: http://scop.mrc-lmb.cam.ac.uk/scop/ (2).
pFam: http://pfam.sanger.ac.uk/ (18).

2.2. Servers

DALI: http://ekhidna.biocenter.helsinki.fi/dali_server/ (19).
PDBe: http://www.ebi.ac.uk/pdbe/ (20).

2.3. Software

Spasm, mkspaz, and savant may be downloaded from USF at http://xray.bmc.uu.se/usf/ (21).
Pymol: The PyMOL Molecular Graphics System, Version 1.3, Schrödinger, LLC.
Jalview: http://www.jalview.org/ (22).
Mustang: http://www.csse.monash.edu.au/~karun/Site/mustang.html (23).
Excel: Microsoft Corporation.
In the methods description it is assumed that the software has been downloaded and installed as described in the installation instructions.

3. Methods

3.1. Identifying Propeller Structures in PDB

First get number of propellers from CATH and SCOP. This is done using the advanced search option at the PDB homepage, which offers the possibility to search for either the CATH code or the SCOP fold family. The number of hits obtained in the full PDB and the reduced PDB by removing 90% similar entries is shown in Table 1. The discrepancy between the numbers is primarily due to the different update frequencies of the two classifications.

1. For each N-propeller create list of structures classified as propellers by either CATH or SCOP using the option in PDB advanced search available at the RCSB homepage (see Note 1).
2. Import lists of PDB identifiers to spreadsheet.
3. Remove duplicates from lists.
4. Pick a structure of each type from the lists.

Table 1
No. of propeller structures in the PDB

	4	5	6	7	8	10	Total
CATH	14	21	215	136	110	0	496
Cath-pdb90	11	10	41	40	15	0	117
SCOP	12	28	174	122	92	0	428
Scop-pdb90	9	14	34	35	11	0	98
Scop + Cath	14	35	232	146	110	0	537
1.search	15	55	258	290	134	0	752
2.search	17	82	311	331	170	0	911
3.search	17	98	323	367	170	0	975
manual check	17	100	335	391	173	1	1017

5. For $N=4, 10$.
   (a) Inspect structure in Pymol and create a file only containing the propeller domain. Large inserts between blades should be removed, e.g., at the Pymol command line interface write

   create 1KIT-propA , /1KIT//A/xxx-yyy

   save 1KIT-propA.pdb , /1KIT-propA

   (b) Go to the DALI server and submit the pdb-file containing the propeller domain as query.

   (c) Import DALI results into spreadsheet (see Note 2). Use the import external data option. It is only necessary to import the pdb identifier and the *Z*-value of the hit at this point but the chain id will be needed later.

   (d) Mark result list with the presence in list of propellers. Example of hit list is shown in Table 2. Columns 1–3 contain the result from Dali. Columns 4–10 contain the result of matching the pdb identifier to the list of structures already assigned to the different propeller types. The formula used for this in excel :

   =IF(ISNA(MATCH(A3;'prop-list'!$A$2:$A$400;0));-1;MATCH(A3;'prop-list'!$A$2:$A$400;0))

   Here A3 refers to the field of the pdb identifier in the dali result, and "prop-list" is the sheet containing the list of assigned propeller structures. The A column contains identifiers of four-propellers.

Table 2
Search results from Dali imported into Excel sheet

prop6			in list					
PDB-id	chain	Z	4	5	6	7	8	10
1A14	N	78,7	−1	−1	1	−1	−1	−1
7NN9	A	74,3	−1	−1	232	−1	−1	−1
…	…	…	…	…	…	…	…	…

(e) Define top-hits: *Z*-value > *Z*-value of any structure classified as the current *N*, and *Z* > 25 (see Note 3).

(f) Add top-hits to the list of classified structures.

(g) For a new dali search for N-propellers pick the hit with lowest Dali *Z*-value marked as an N-propeller.

6. If all top-hits have previously been identified, which will be the case after 3–4 cycles, redefine the way queries are picked (step 8).
7. If a structure appears in two lists it may contain two different propeller domains. The method outlined here will in general only assign the structure to one of the propeller types. So pay special attention to, e.g., tricorn protease.
8. Redefine the rule for choosing queries: Choose as query the first structure, i.e., highest *Z*-value in the result list for N-propeller which has not been assigned. Repeat step 4.
9. The remaining hits, which have not been assigned to any propeller class, may now be inspected visually and assigned to their respective propeller class.

3.2. Download Structures

1. Create a directory for each propeller type.
2. For each propeller type create a list file for input to ftp. At the PDBe ftp-site the structures are organized in a file tree where structures are kept in separate directories according to the middle two characters of the pdb identifier, e.g., 1KIT is found in a subdirectory called KI. The list file for input to the ftp program can be made in the spreadsheet application (Table 3).

   The pdb identifier is stored in the first column and in the second column the following formula is entered: =LOWER(CONCATENATE("get ";MID(A1;2;2);"/pdb";A1; ".ent.gz";" ";A1;".pdb.gz "))

   Copy and paste column 2 into an ascii text file called list.txt or some other sensible name (see Note 4).
3. In the first line of list.txt insert:

   cd /pub/databases/rcsb/pdb/data/structures/divided/pdb (see Note 5).

Table 3
Example of creating list for ftp

1GYD	get gy/pdb1gyd.ent.gz 1gyd.pdb.gz
1GYE	get gy/pdb1gye.ent.gz 1gye.pdb.gz
1GYH	get gy/pdb1gyh.ent.gz 1gyh.pdb.gz
…	…

4. Change working directory to the directory created previously for the propeller type in question. At the command prompt enter ftp –a ftp.ebi.ac.uk < list.txt.
5. Unzip the files.
6. Use AWK to create file containing the chain with a propeller—this requires that you have kept track of chain id. Save the awk script below in a file called "chain.awk":

```
{if($1=="ATOM")
{if( $5==chain && $6>=first && $6<=last)
printf("%4 s%7i%2s%-3 s%1s%3 s%1s%1 s%4i%12
.3f%8.3f%8.3f%6.2f%6.2f\n",
$1,$2,"",$3,"",$4,"",$5,$6,$7,$8,$9,$10,$11)}
else print}
```

For example, save chain B of 1GYD.pdb in a file called 1GYDB located in a subdirectory called chains:

awk –f chain.awk chain="B" first=1 last=10000 < 1GYD.pdb > chains/1GYDB.pdb

A script that will do this for all the downloaded pdb files can be easily made using the CONCATENATE function of excel as illustrated above (see Note 6).

3.3. Creating Multiple Structural Alignment

Create an input file for MUSTANG, e.g., mustang-input.txt:

```
> ../prop5/pdb
+1gyd.pdb
1gye.pdb
1gyh.pdb
1oyg.pdb
+1 pt2.pdb
1 s18.pdb
1s1d.pdb
+1st8.pdb
+1uv4.pdb
…
```

The first line of this file describes the location of the structure files specified in the following lines. Only structures preceded by + will be read and aligned by Mustang (see Note 7).

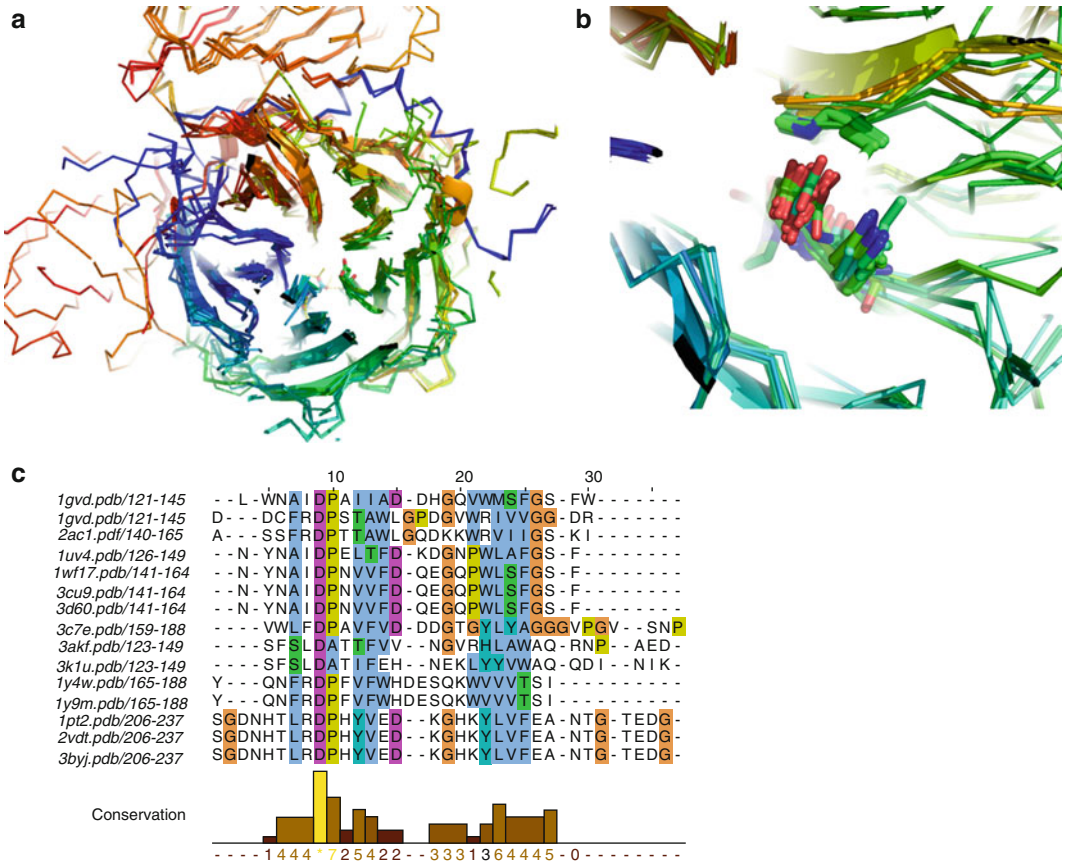


Fig. 2. (**a**) Structural superposition of 15 five-propeller structures created by Mustang. Each chain is colored *blue* to *red* from the N- to the C-terminal. The side chain of Asp 158 *C. cellulosa* alpha-l-arabinase (PDB:1GYD (24)) is shown as *sticks*. (**b**) Side chains superimposed on Asp 158 and atoms within 4 Å shown as *sticks*. (**c**) Sequence alignment corresponding to the structural alignment generated by Mustang. The excerpt shows the only absolutely conserved position among the 15 sequences, which corresponds to Asp 158 of *C. cellulosa* alpha-l-arabinase.

The second blank line is skipped but is required.

To run Mustang enter the command:

mustang-3.2.1 -f mustang-input.txt -F fasta

The option "-F fasta" specifies that the sequence alignment will be written as a fasta-formatted file. The structural alignment is written to a pdb-file containing all chains that can be displayed in Pymol (Fig. 2a) and the sequence alignment may be displayed using Jalview (Fig. 2b). The sequence alignment may be further used for analysis, e.g., calculating logos, creating a Hidden Markov Model, or it may be mapped onto the structure using consurf. An aspartate, at position 158 in PDB:1GYD (24), is conserved in the aligned structures. In some of the structures this aspartate forms (e.g., residue 189 in PDB:1YW4 (25)) a hydrogen bond with the main chain of a neighboring blade and also a salt bridge with a neighboring arginine (Fig. 2c). To examine this interaction further we will use Spasm.

3.4. Create Database for Spasm

For each of the collections of propeller structures first create a list of the pdb files that were downloaded as in Subheading 3.2.

ls *pdb > mkspaz.inp

In this list file insert at the top the name of the database for spasm, e.g., prop5.lib, and insert a carriage return after each file name:

```
prop5.lib
1gyd.pdb
1gye.pdb
1gyh.pdb
1oyg.pdb
1 pt2.pdb
1 s18.pdb
...
```

Then create the library: mkspaz < mkspaz.inp > prop5.log

3.5. Identifying Common Structures and Interactions

Several different scenarios may be envisioned here: searching for a loop occurring between any two beta strands, searching for a loop between two specific strands in a blade, searching for part of a loop, or interactions between blades. For each of these the query template has to be constructed differently. In the following example we will examine the interaction of Asp 189 of PDB:1Y4W (25) with the neighboring blade identified in Subheading 3.4.

1. Construct the template using Pymol. We will include the two inner strands of blades 3 and 4 as shown in Fig. 3a to ensure that the interaction found is in the correct structural context. Load PDB:1Y4W into Pymol and write

   ```
   create temp1, /1y4w//A/188-195+200-208+241-249+257-263
   save temp1.pdb, /temp1
   ```

2. We wish to ignore the residue type except at the Asp 189 position in the Spasm search. Save the following awk-script to a file called toxxx.awk:

   ```
   {if($1=="ATOM")
   {if( $5 == chain && $6>= first && $6<=last)
   printf ("%4 s%7i%2s%-3 s%1s%3 s%1s%1 s%4i%1
   2.3f%8.3f%8.3f%6.2f%6.2f\n",
   $1,$2,"",$3,"","XXX","",$5,$6,$7,$8,$9,$10,$11)}
   else printf ("%4 s%7i%2s%-3 s%1s%3 s%1s%1 s
   %4i%12.3f%8.3f%8.3f%6.2f%6.2f\n",
   $1,$2,"",$3,"",$4,"",$5,$6,$7,$8,$9,$10,$11)}
   else print}
   ```

   We can now change all residues except the aspartate at position 189 to residue type XXX by using the awk-script twice:

   ```
   awk -f toxxx.awk chain=A,first=188
   last=188<temp1.pdb>temp2.pdb
   awk -f toxxx.awk chain=A,first=190
   last=263<temp2.pdb>temp3.pdb
   ```

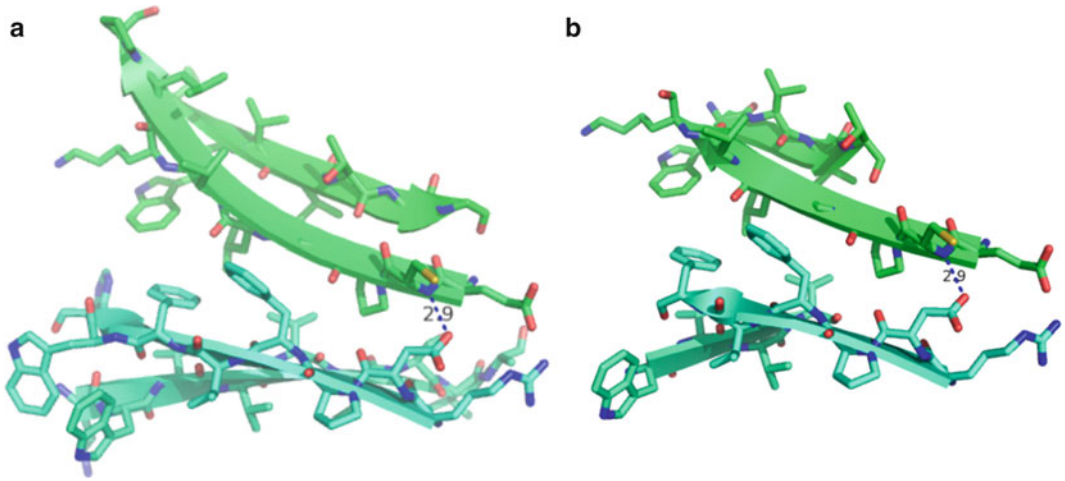


Fig. 3. (**a**) Template #1 consisting of residues 188–195, 200–208, 241–249, and 257–263 of PDB:1Y4W. The distance between Oδ of Asp189 and the peptide nitrogen of Cys242 is indicated. (**b**) Template # 2 consisting of residues 188–193, 201–204, 241–247, and 258–261 of PDB:1Y4W.

In spasm the residue type XXX is used to indicate that the side chain is insignificant. This means that only the Ca position is used in the search and any residue type is allowed at that position (see Note 8).

3. Run a Spasm search using the library of five-propellers and the template as query, i.e., temp3.pdb created in the previous step. Spasm is started from the command line and will prompt for the different input parameters. In this first run of Spasm use the suggested default values for maximum root mean square deviation (rmsd) for Cα to Cα and side chain to side chain (sc–sc) distance. When prompted for sequence substitutions choose the option that prohibits substitutions. This latter option only affects the aspartate since the rest of the residues have the type XXX. Simply by looking at the number of hits reported by Spasm, we see that some propellers have more than one match to the template (see Table 4). Although this would seem likely given the symmetry of the propellers it usually indicates that more discriminating search criteria are needed.
4. Superimpose the fragments identified by Spasm. This is done by Savant, which reads the output file from Spasm and the individual pdb files. It creates a pdb file for each fragment in the correct orientation. Inspecting the hits in pymol reveals that the multiple hits in the same propeller obtained in the previous step are due to different superpositions of the same fragment. This situation may be avoided by repeating step 3 reducing either the cutoff values for rmsd of Cα-Cα and sc-sc distance or reducing the size of the template used as query (Fig. 3b). Finding an appropriate set of search parameters usually requires a number of test runs, i.e., repeating step 3.

Table 4
Results of Spasm searches

Propeller library	5	5	5	5	5	5	6	6
Nr of entries in library	72	72	72	72	72	72	264	264
Template #	1	1	2	2	2	2	2	2
Max RMSD	2	2	2	2	1,5	1	2	2
Max CA–CA dist.	4	2	2	4	2	1	2	4
Max SC–SC dist.	5	2	2	5	1	1	2	5
Nr of entries with hit(s)	25	5	13	60	13	4	0	15
Total number of hits	45	5	13	248	13	4	0	16

The top rows report which propeller library was used, the template used, and the cutoff values for reporting a hit. The lower two rows report the result of the search

The Savant program also creates an output file containing rmsd values for superpositioning atomic positions and hits may be sorted according to this value. In Table 4 different sets of search parameters and corresponding number of hits are listed as well as the result of using template #2.

5. The interaction identified in this way needs now to be examined in further detail by relating the found structures to function. In this example it turns out that the conserved aspartate and the neighboring arginine found in the majority of the hits are involved in substrate binding. Thus it is likely that the framework of the propeller positions the aspartate and arginine for interaction with the ligand. Also we can examine if this particular interaction is specific for five-propellers by performing the same search in the other propeller libraries. In Table 4 the result of two searches in the six-propeller library is reported showing hits only when the high cutoff values are used.

4. Notes

1. During this process we found an error in the classification of 2OVP (2OVQ, and 2OVR), 1NEX, and 1R5M which were listed in PDB as SCOP 7-bladed-propellers but are in fact 8-bladed. These structures showed up in a search using an eight-bladed template with a Z value from 27 to 29.8. However these structures also show up with $25 < Z < 28.5$ in a search using a seven-bladed template.
2. In step c one may choose to only use the PDB90 subset of the hits. This is offered as an option on the DALI search result and

will reduce the number of hits by a factor of 4 and hence reduce the amount of bookkeeping considerably.

3. In step d the choice of *Z* cutoff may be set as low as 20 for four- and five-bladed propellers. But as mentioned in the above note a value of 25 may even be too low for discriminating between seven- and eight-bladed propellers.
4. Be sure that proper carriage returns are stored at the end of each line of the text file. Otherwise the file will not be accepted as input by ftp.
5. The actual directory path may vary depending on the ftp site used. The path given here is valid for PDBe at ftp.ebi.ac.uk
6. For some comparisons it may be required that the structure file only contains the propeller domain. To create such a file the sequence range covering the propeller domain first needs to be identified. These limits can be obtained from visual inspection in Pymol but can also be found in the results file from the DALI search. Once the limits have been identified the chains. awk can be used to extract the appropriate residues. Should the propeller domain have inserts that need to be excluded save the following awk script in a file called exclude.awk:

```
{if($1=="ATOM")
{if($5 == chain&& $6<first && $6>last)
printf("%4 s%7i%2s%-3 s%1s%3 s%1s%1 s%4i%12
    .3f%8.3f%8.3f%6.2f%6.2f\n",
$1,$2,"",$3,"",$4,"",$5,$6,$7,$8,$9,$10,$11)}
else print}
```

Invoke the script by typing, e.g.,

awk –f exclude.awk chain="B" first=91 last=110<1GYDB.pdb>1GYDBp.pdb

The file 1GYDBp.pdb will now contain 20 residues less than 1GYDB.pdb. By combining chain.awk and exclude.awk with the proper parameters, i.e., chain, first, and last, the required fragments of a structure can be extracted. This also holds true for creating small fragments such as a single blade or b-hairpins.

7. The input file specified 15 structures to be superimposed. Some of these structures contained additional domains. If positions of inserts in the propeller domain are to be compared this may be sensible. The time for calculating the superpositions grows with the square of the number of residues in each structure and by the square of the number of structures to be superimposed. For that reason it is an advantage to truncate the structures prior to superposition and it also limits the number of structures that can be superimposed within a reasonable time.

8. Spasm is searching for similarities in distances between Ca atoms and distances between center of gravities of side chains. When changing residues to type XXX the side chain positions are ignored. In some structural similarity searches using Spasm it may be an advantage that the query template is a poly-Alanine except in a few specific positions. This can be achieved using the following awk-script:

```
{if($1=="ATOM")
{if($5 == chain && $6>= first && $6<= last &&
$4 !="GLY")
{if($3 == "N" || $3 == "CA" || $3 == "CB" ||
$3=="C"||$3 == "O")
printf ("%4 s%7i%2s%-3 s%1s%3 s%1s%1 s%4i%1
2.3f%8.3f%8.3f%6.2f%6.2f\n",
,$2,"",$3,"","ALA"," ",$5,$6,$7,$8,$9,$10,$11)}
else printf ("%4 s%7i%2s%-3 s%1s%3 s%1s%1 s
%4i%12.3f%8.3f%8.3f%6.2f%6.2f\n",
$1,$2,"",$3,"",$4," ",$5,$6,$7,$8,$9,$10,$11)}
else print}
```

Acknowledgments

We wish to thank The Lundbeck Foundation for financial support through the MIND Centre and The Danish National Research Foundation for funding through the CARB Centre.

References

1. Orengo CA, Michie AD, Jones S et al (1997) CATH—a hierarchic classification of protein domain structures. Structure 5:1093–1108
2. Murzin AG, Brenner SE, Hubbard T et al (1995) SCOP: a structural classification of proteins database for the investigation of sequences and structures. J Mol Biol 247:536–540
3. Rao ST, Rossmann MG (1973) Comparison of super-secondary structures in proteins. J Mol Biol 76:241–256
4. Varghese JN, Laver WG, Colman PM (1983) Structure of the influenza virus glycoprotein antigen neuraminidase at 2.9 A resolution. Nature 303:35–40
5. Murzin AG (1992) Structural principles for the propeller assembly of beta-sheets: the preference for seven-fold symmetry. Proteins 14:191–201
6. Paoli M (2001) Protein folds propelled by diversity. Prog Biophys Mol Biol 76:103–130
7. Baker SC, Saunders NF, Willis AC et al (1997) Cytochrome cd1 structure: unusual haem environments in a nitrite reductase and analysis of factors contributing to beta-propeller folds. J Mol Biol 269:440–455
8. Russell RB, Sasieni PD, Sternberg MJ (1998) Supersites within superfolds. Binding site similarity in the absence of homology. J Mol Biol 282:903–918
9. Quistgaard EM, Thirup SS (2009) Sequence and structural analysis of the Asp-box motif and Asp-box beta-propellers; a widespread propeller-type characteristic of the Vps10 domain family and several glycoside hydrolase families. BMC Struct Biol 9:46
10. Crennell S, Garman E, Laver G et al (1994) Crystal structure of Vibrio cholerae neuraminidase reveals dual lectin-like domains in addition to the catalytic domain. Structure 2: 535–544

11. Nojiri M, Hira D, Yamaguchi K et al (2006) Crystal structures of cytochrome c(L) and methanol dehydrogenase from Hyphomicrobium denitrificans: structural and mechanistic insights into interactions between the two proteins. Biochemistry 45:3481–3492
12. Mattei P, Boehringer M, Di Giorgio P et al (2010) Discovery of carmegliptin: a potent and long-acting dipeptidyl peptidase IV inhibitor for the treatment of type 2 diabetes. Bioorg Med Chem Lett 20:1109–1113
13. Severi E, Muller A, Potts JR et al (2008) Sialic acid mutarotation is catalyzed by the Escherichia coli beta-propeller protein YjhT. J Biol Chem 283:4841–4849
14. Jing H, Takagi J, Liu JH et al (2002) Archaeal surface layer proteins contain beta propeller, PKD, and beta helix domains and are related to metazoan cell surface proteins. Structure 10:1453–1464
15. Springer TA (1998) An extracellular beta-propeller module predicted in lipoprotein and scavenger receptors, tyrosine kinases, epidermal growth factor precursor, and extracellular matrix components. J Mol Biol 283:837–862
16. Chaudhuri I, Soding J, Lupas AN (2008) Evolution of the beta-propeller fold. Proteins 71:795–803
17. Berman HM, Westbrook J, Feng Z et al (2000) The Protein Data Bank. Nucleic Acids Res 28:235–242
18. Finn RD, Mistry J, Tate J et al (2010) The Pfam protein families database. Nucleic Acids Res 38:D211–D222
19. Holm L, Rosenstrom P (2010) Dali server: conservation mapping in 3D. Nucleic Acids Res 38:W545–W549
20. Velankar S, Alhroub Y, Alili A et al (2011) PDBe: Protein Data Bank in Europe. Nucleic Acids Res 39:D402–D410
21. Kleywegt GJ (1999) Recognition of spatial motifs in protein structures. J Mol Biol 285:1887–1897
22. Waterhouse AM, Procter JB, Martin DM et al (2009) Jalview Version 2–a multiple sequence alignment editor and analysis workbench. Bioinformatics 25:1189–1191
23. Konagurthu AS, Whisstock JC, Stuckey PJ et al (2006) MUSTANG: a multiple structural alignment algorithm. Proteins 64:559–574
24. Nurizzo D, Turkenburg JP, Charnock SJ et al (2002) Cellvibrio japonicus alpha-l-arabinanase 43A has a novel five-blade beta-propeller fold. Nat Struct Biol 9:665–668
25. Forouhar F, Abashidze M, Conover K et al (2005) unpublished
26. Williams PA, Fulop V, Garman EF et al (1997) Haem-ligand switching during catalysis in crystals of a nitrogen-cycle enzyme. Nature 389:406–412
27. Malby RL, McCoy AJ, Kortt AA et al (1998) Three-dimensional structures of single-chain Fv-neuraminidase complexes. J Mol Biol 279:901–910
28. Quistgaard EM, Madsen P, Groftehauge MK et al (2009) Ligands bind to Sortilin in the tunnel of a ten-bladed beta-propeller domain. Nat Struct Mol Biol 16:96–98

Chapter 4

Structure Description and Identification Using the Tableau Representation of Protein Folding Patterns

Arun S. Konagurthu and Arthur M. Lesk

Abstract

We have developed a concise tableau representation of protein folding patterns, based on the order and contact patterns of elements of secondary structure: helices and strands of sheet. The tableaux provide a database, derived from the protein data bank, minable for studies on the general principles of protein architecture, including investigation of the relationship between local supersecondary structure of proteins and the complete folding topology. This chapter outlines the tableaux representation of protein folding patterns and methods to use them to identify structural and substructural similarities.

Key words: Tableau representation, Protein folding pattern, Supersecondary structure, Substructure search

1. Introduction

1.1. Tableau Representation

Tableaux are a compact and powerful two-dimensional representations of protein folding patterns introduced by Lesk (1). Underlying a tableau representation is the idea that the essence of a protein folding pattern is the order, along the amino acid chain, of secondary structural elements—helices and strands of sheet—and the geometry of interactions of pairs of secondary structural elements that are in contact.

Tableaux capture the contact information and encode the geometry of interacting pairs in a simple square symmetric matrix. The rows and columns correspond to the labels of secondary structural elements in the observed order of appearance in the amino acid sequence. Each off-diagonal element in this matrix either is a blank if the corresponding pair of secondary structural elements is

Alexander E. Kister (ed.), *Protein Supersecondary Structures*, Methods in Molecular Biology, vol. 932,
DOI 10.1007/978-1-62703-065-6_4,

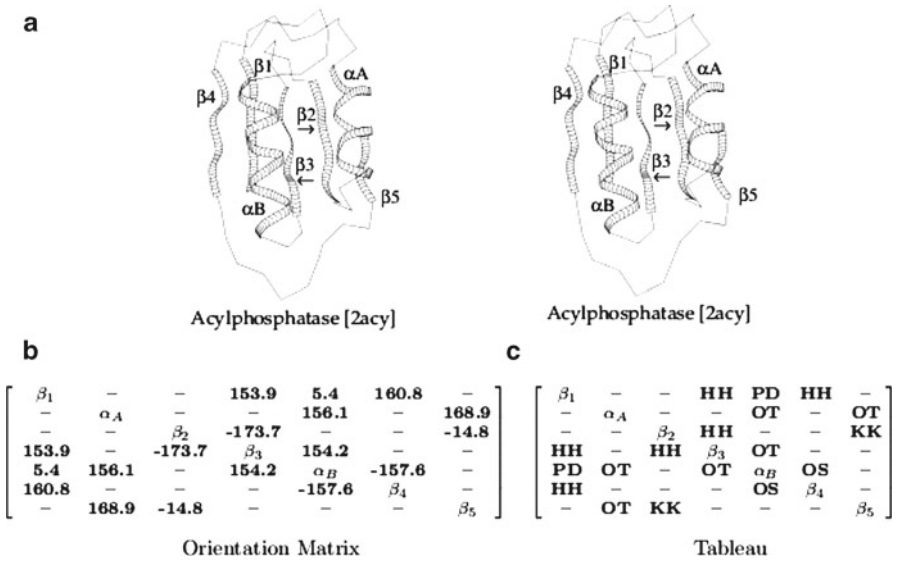


Fig. 1. (**a**) Structure of Acylphosphatase (in wall-eyed stereo), an αβ-protein (wwPDB ID: 2ACY). The chevrons indicate the direction from N- to C-terminus of the amino acid chain. (**b**) The matrix containing the relative orientations of secondary structural elements in the structure that are in contact. (**c**) Tableau representation of the structure. The labels of rows and columns are denoted in their main diagonals of the matrices.

not in contact, or encodes the relative orientation of secondary structure elements that are in contact. The relative orientation of two secondary structural elements in contact appears in the matrix as a discrete two-character code. The design of the encoding scheme takes into account the crucial observation that relative orientations of secondary structural elements in homologous proteins can change substantially even though the basic topology of the folding pattern is retained.

Figure 1 shows the structure of Acylphosphatase with its corresponding relative orientations of secondary structural elements and its tableau representation. The method of construction of a tableau representation of a protein structure is described in Subheading 3.1.

1.2. Structural and Substructural Lookup

The tableau representation permits efficient querying and retrieval of local and global similarities in protein structures (2).

Very closely related structures often result in identical tableaux. Given a tableau of a structure as a query, identical tableaux from the database can be efficiently extracted in constant-time (per match in the database). This method is discussed in Subheading 3.2.1.

Closely related tableaux that differ only in a few rows and columns can also be retrieved in constant time by preprocessing and storing the neighborhood tableaux in the database. This method is discussed in Subheading 3.2.2.

Identifying similarities between protein structures is a computationally hard problem. With tableaux, this can be rigorously framed as a maximum common subtableau extraction problem

(see Note 1). We present a quadratic programming and an equivalent integer linear programming formulation of this problem in Subheadings 3.2.3 and 3.2.4.

1.3. Connection Between Tableau and Supersecondary Structures

Supersecondary structures arise out of contacts of several secondary structure elements local in the amino acid sequence. Tableau captures these relationships elegantly. Successive diagonals of the tableau, proceeding outwards from the main diagonal, contain information about contacts between secondary structure elements local in sequence; the closer to the main diagonal, the more local the information. Of the supersecondary structures, the α-hairpin, the β-hairpin, and α–β–α unit (3), the first two involve only two consecutive elements of secondary structure and therefore appear in tableaux on the diagonal adjacent (i.e., ±1) to the main diagonal (see Note 2), and the third involves two diagonals adjacent (i.e., ±2) to the main diagonal. With this convenient representation, tableaux allow the addressing of the following question: *how little do we need to know to specify a protein fold?* (4). By retaining diagonals closer to main diagonal of tableaux, it is possible to ask whether the supersecondary structural information is enough to identify—note, we emphasize, necessarily to determine—the tertiary fold of the domain. Kamat and Lesk (5) systematically address this question using a subset of ASTRAL SCOP (6) domains. For complete tableaux (i.e., tableaux composed of the entire information in their respective matrices), in 98 % of the cases knowing the tableau uniquely identifies the SCOP ID. Simplifying tableaux by discarding some of the outer diagonals it is possible to ask how many diagonals are needed to retain the capacity to uniquely identify SCOP ID from the tableau.

Interestingly, retaining the main diagonal plus only one adjacent diagonal reduces the ability to correlate tableau uniquely from 98 to 95 %, which is a very small drop given the amount of information discarded. Keeping two diagonals in addition to the main diagonal allows identification of up to 97 % of SCOP IDs (5). The conclusion is that almost all the information required to identify a protein folding pattern is inherent in the local supersecondary structure (4). Therefore, tableau representations are useful in guiding attempts to predict protein structure from sequence using predictions of supersecondary structures with tableau representation as an intermediate (4).

2. Materials

1. ASTRAL SCOP 1.71 (6) was used in all our experiments. ASTRAL inherits its domain definitions from SCOP.
2. DSSP (7) was used to assign secondary structure.
3. MD5sum (Message-Digest algorithm 5) (8) was used to produce MD5 hashes of tableaux.

4. All our software was written in C++.
5. Quadratic and Linear integer programming formulations were solved using ILOG CPLEX Concert Technology libraries for C++.

3. Methods

3.1. Construction of Tableau

Tableau of a given protein structure is constructed as follows:

1. The three-dimensional coordinates are delineated into secondary structural elements (see Note 3). This identifies a sequence of secondary structural elements (helices and strands of sheet). All helices are labeled α not distinguishing α; 3_{10}, and π helices. Strands of sheet are labeled β (see Note 4).
2. For every pair of secondary structural elements in contact find the relative orientation of these elements. The relative orientation of two secondary structural elements specifies an angle $-180° \leq \Omega \leq 180°$. The angle and contact are computed using the following steps (see Fig. 2).
   (a) Represent each secondary structural element as a vector. For a helix, this vector corresponds to the helical axis. For a strand, the vector is represented by the least-squares line through its C_α. Each vector is directed from the N- to C-terminus of the secondary structural element.
   (b) Compute the mutual perpendicular between the two vectors.
   (c) Sighting along the mutual perpendicular find the shortest rotation required of the vector in the front (along the line of sight) to eclipse the vector at the back (see Note 5). This shortest rotation defines the orientation angle (see Note 6).
3. A tableau representation encodes the relative orientations of pairs of secondary structural elements in contact as follows:
   (a) Two secondary structural elements are defined to be in contact if at least two residues from the elements are in contact. Two residues are treated to be in contact if there exists at least one pair of atoms between the residues that are in contact. Two atoms are in contact if their distance is less than sum of their van der Waals radii plus some small constant (see Note 7).
   (b) The possible range of orientation angles ($-180° \leq \Omega \leq 180°$) is divided into quadrants in two different ways, differing in orientation by 45° (see Fig. 3).
   (c) Any orientation angle of a pair of secondary structural elements in contact lies within a quadrant corresponding to each of the two partitions.

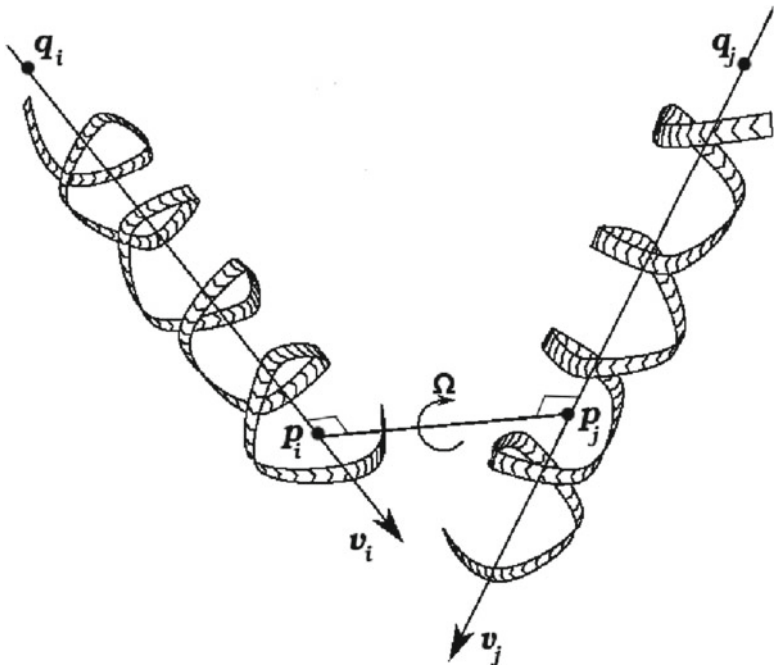


Fig. 2. Relative orientation of two secondary structural elements (in this case helices) specified by the angle between the vectors along their axes. Chevrons indicate the direction of the chain, pointing from N- to C-terminus is computed using a four position vectors: p_i, q_i, p_j, q_j. The vector $p_i - p_j$ is a mutual perpendicular of the vectors representing the secondary structural elements, represented by vectors $v_i \equiv p_i - q_i$ and $v_j \equiv p_j - q_j$. Sighting along the perpendicular vector from $p_i \rightarrow p_j$ (or equivalently from the other direction, $p_i \rightarrow p_j$), the orientation angle Ω is the shortest rotation (clockwise or anticlockwise) required to reorient v_j (or equivalently v_i) to eclipse v_i (or equivalently v_j) in a synplanar arrangement. Ω is positive for clockwise rotations and negative otherwise.

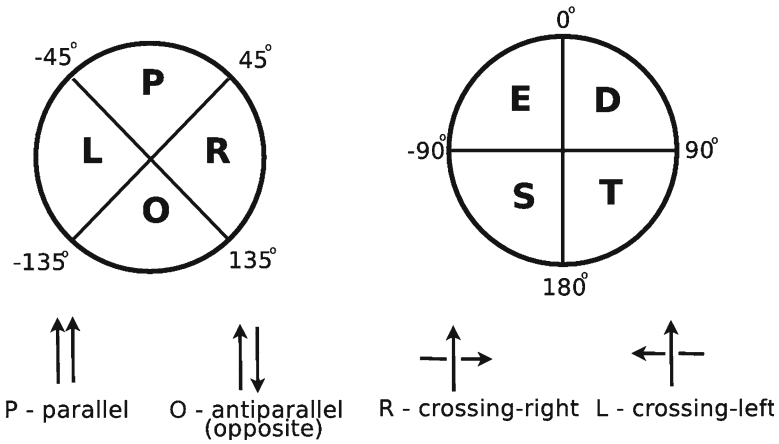


Fig. 3. Double-quadrant encoding of angles recorded in tableaux. Note that crossing-left and crossing right are distinguished; tableaux do contain enantiomorph information.

(d) This gives a discrete two-character code specifying their relative orientation. The quadrants labeled P, O, R, and L, in the partitions of the circle on the left of Fig. 3, are centered around the relative orientations shown at the bottom of the figure. E, D, T, and S label the rotated set of quadrants (see Notes 8 and 9).

3.2. Structural and Substructural Lookup Methods

3.2.1. Constant-Time Retrieval of Identical Tableau

1. Given a database of structures, preprocess their tableaux.
2. Each tableau in the database is hashed based on its MD5sum (Message-Digest algorithm 5) (8).
3. Given a query, compute its tableau.
4. Generate the MD5sum of the query tableau.
5. The tableaux in the database that are identical with the query tableau share the same MD5sum. These can be retrieved in constant time per hit in the database.

3.2.2. Constant-Time Retrieval of Closely Related Structures

1. For each structure in the database, find their tableaux.
2. Find neighborhood subtableaux created by deleting a set of rows and columns. For example, a $N \times N$ tableau has $N-1$ subtableaux generated by deleting each row (corresponding column) one at a time.
3. All the tableaux and their corresponding subtableaux are hashed independently based on their MD5sum.
4. Given a new structure as a query, compute its tableau.
5. Tableaux and subtableaux in the database that share the same MD5sum can then be retrieved in constant time. This allows us to retrieve closely related structures.

3.2.3. Quadratic Integer Programming Based Extraction of Maximally Similar Subtableau

1. Let P and Q be two proteins which contains M and N secondary structural elements respectively. Let $T^p = \left(t_{ij}^p\right)_{1 \le i,j \le M}$ and $T^q = \left(t_{ij}^q\right)_{1 \le i,j \le N}$ be their tableau matrices respectively.
2. Introduce Boolean variables $1 \le i \le M$, $1 \le j \le N$ where $y_{ij} = 1$ indicates that the ith secondary structural element in P is matched with jth secondary structural element in Q, and $y_{ij} = 0$ indicates they are not matched.
3. The QIP formulation for comparing two tableaux for similarities is as follows:

$$f(y) = \max \sum_{\substack{1 \le i,k \le M \\ 1 \le j,l \le N}} \zeta\left(t_{ik}^p, t_{jl}^q\right) y_{ij} y_{kl} \qquad (1)$$

subject to the constraints

$$\sum_{j=1}^{N} y_{ij} \le 1, \quad 1 \le i \le M \tag{2}$$

$$\sum_{j=1}^{M} y_{ij} \le 1, \quad 1 \le j \le N \tag{3}$$

$$y_{ij} + y_{kl} \le 1, \quad 1 \le i < k \le M, \quad 1 \le l < j \le N \tag{4}$$

4. In the objective function given by Eq. 1, $\zeta\left(t_{ik}^{p}, t_{jl}^{q}\right)$ represents the scoring function that scores the matching of $t_{ik}^{p} \in T^{p}$ as follows: $\zeta\left(t_{ik}^{p}, t_{jl}^{q}\right) = 2$ if $t_{ik}^{p} \equiv t_{jl}^{q}$; $\zeta\left(t_{ik}^{p}, t_{jl}^{q}\right) = 1$ if $t_{ik}^{p} \cong t_{jl}^{q}$; $\zeta\left(t_{ik}^{p}, t_{jl}^{q}\right) = 0$ if t_{ik}^{p} and t_{jl}^{q} are blank; $\zeta\left(t_{ik}^{p}, t_{jl}^{q}\right) = -2$ otherwise (see Notes 10 and 11).
5. Solving the above program identifies maximally similar subtableau.

3.2.4. Equivalent Integer Linear Programming Based Extraction of Maximally Similar Subtableau

The quadratic integer program described in the above section can be reframed as an integer linear program on quadratic Boolean variables as follows:

1. Let $x_{ijkl} = 1$ when *i*th and *k*th secondary structural element in P are matched with *j*th and *l*th secondary structural element respectively in Q. Using the notation introduced in the previous section, we have

$$x_{ijkl} = y_{ij} \wedge y_{kl} \forall i, j, k, l \quad \text{s.t.} \quad 1 \le i, k \le M, \quad 1 \le j, l \le N$$

2. The integer linear program can then be formulated as:

$$f(y) = \max \sum_{\substack{1 \le i,k \le M \\ 1 \le j,l \le N}} \zeta\left(t_{ik}^{p}, t_{jl}^{q}\right) y_{ij} y_{kl} \tag{5}$$

subject to the constraints

$$\sum_{j=1}^{N} y_{ij} \le 1, \quad 1 \le i \le M \tag{6}$$

$$\sum_{j=1}^{M} y_{ij} \le 1, \quad 1 \le j \le N \tag{7}$$

$$y_{ij} + y_{kl} \le 1, \quad 1 \le i < k \le M, \quad 1 \le l < j \le N \tag{8}$$

$$x_{ijkl} \le y_{ij}, \quad 1 \le i, k \le M, \quad 1 \le j, l \le N \tag{9}$$

$$x_{ijkl} \le y_{kl}, \quad 1 \le i, k \le M, \quad 1 \le j, l \le N \tag{10}$$

$$y_{ij} + y_{kl} \leq x_{ijkl} + 1, \quad 1 \leq i < k \leq M, \quad 1 \leq j < l \leq N \qquad (11)$$

3. Solving the above linear integer program extracts maximally similar subtableau between two proteins (see Notes 12 and 13).

4. Notes

1. Maximum common subtableau problem is equivalent to the quadratic assignment problem in computer science, which has no known polynomial time algorithm. However, typical sizes of tableaux are very small—a protein has, on an average, 15 secondary structural elements. This allows the maximum common subtableau problem to be solved exactly in practical time.
2. In a tableau, the main diagonal instead of recording self-contacts lists the secondary structural label.
3. The accuracy of the tableau representation depends on the accuracy of the method to delineate secondary structural elements. Methods like DSSP work well to assign secondary structure at a residue level, but their use to find the precise start and end points of each secondary structural element can be inexact (9, 10).
4. To uniquely identify secondary structural elements, helices are numbered $\alpha_A, \alpha_B, \ldots$ and strands are numbered $\beta_1, \beta_2, \ldots$.
5. The angle of rotation is invariant to the direction of sight along the mutual perpendicular. Both directions give the same angle (and sign) of rotation.
6. Clockwise rotations give positive angles and anticlockwise rotations give negative angles.
7. A constant of 1 Å is used in our work.
8. E, D, T, and S are mnemonics for *Elevenses*, *Dinner*, *Tea*, and *Supper* corresponding to the British meal system (regarding the circle as a clock face).
9. For adjacent strands of the same _ sheet, additional two letter codes KK and HH specify parallel and anti-parallel _ sheet interactions respectively. This is useful to distinguish strands that form β sheet, from those in different β sheets packed face to face.
10. Constraints 2 and 3 ensure that each secondary structural element in one tableau is matched with at most one secondary structural element in the other. Constraint 4 ensures that the matching preserves the order of the secondary structural elements.

11. $t_{ik}^{p} \cong t_{jl}^{q}$ implies that t_{ik}^{p} and t_{jl}^{q} differ by one symbol, for example OS and OT.
12. The objective given by Eq. 5 is equivalent to the objective given by Eq. 1 because $x_{ijkl} = y_{ij} y_{kl}$.
13. Constraints 9 and 10 ensure that the value of any x_{ijkl} cannot exceed that of y_{ij} and that of y_{kl}. Constraint 11 ensures that the values of x_{ijkl} is pushed to 1 when both y_{ij} and y_{kl} are 1. While the integer linear program objective in Eq. 5 can be relied on to push values x_{ijkl} to 1, explicitly including this constraint will allow the integer linear program to converge faster to the optimal solution. Constraints 6–8 are same as Constraints 2–4 in the previous formulation.

References

1. Lesk AM (1995) Systematic representation of protein folding patterns. J Mol Graphics 13:159–164
2. Konagurthu AS, Stuckey PJ, Lesk AM (2008) Structural search and retrieval using a tableau representation of protein folding patterns. Bioinformatics 24:645–651
3. Levitt M, Chothia C (1976) Structural patterns in globular proteins. Nature 261:552–558
4. Konagurthu AS, Lesk AM (2010) Cataloging topologies of protein folding patterns. J Mol Recogn 23(2):253–257
5. Kamat AP, Lesk AM (2007) Contact patterns between helices and strands of sheet define protein folding patterns. Proteins 66:869–876
6. Chandonia JM et al (2004) The ASTRAL compendium. Nucleic Acids Res 32:D189–D192
7. Kabsch W, Sander C (1983) Dictionary of protein secondary structure: pattern recognition of hydrogen-bonded and geometrical features. Biopolymers 22:2577–2637
8. Rivest R (1992) The MD5 message digest algorithm, RFC 1321. MIT and RSA Data Security, Inc
9. Konagurthu AS et al (2011) Piecewise linear approximation of protein structures using the principle of minimum message length. Bioinformatics 27:i43–i51
10. Konagurthu AS, Lesk AM, Allison L (2012) Minimum Message Length inference of secondary structure from protein coordinate data. Bioinformatics 28:i97–105

Part II

Supersecondary Structure Prediction

Chapter 5

Computational Prediction of Secondary and Supersecondary Structures

Ke Chen and Lukasz Kurgan

Abstract

The sequence-based prediction of the secondary and supersecondary structures enjoys strong interest and finds applications in numerous areas related to the characterization and prediction of protein structure and function. Substantial efforts in these areas over the last three decades resulted in the development of accurate predictors, which take advantage of modern machine learning models and availability of evolutionary information extracted from multiple sequence alignment. In this chapter, we first introduce and motivate both prediction areas and introduce basic concepts related to the annotation and prediction of the secondary and supersecondary structures, focusing on the β hairpin, coiled coil, and α-turn-α motifs. Next, we overview state-of-the-art prediction methods, and we provide details for 12 modern secondary structure predictors and 4 representative supersecondary structure predictors. Finally, we provide several practical notes for the users of these prediction tools.

Key words: Secondary structure prediction, Supersecondary structure prediction, Beta-hairpins, Coiled coils, Helix-turn-helix, Greek key, Multiple sequence alignment

1. Introduction

Protein structure is defined at three levels: *primary structure* which is the sequence of amino acids joined by peptide bonds, *secondary structure* that concerns regular local substructures including α-helices and β-strands, which were first postulated by Pauling and coworkers (1, 2), and *tertiary structure* which is the three-dimensional structure of a protein molecule. The supersecondary structure (SSS) bridges the two latter levels and concerns specific combinations/geometric arrangements of a few secondary structure elements. Common SSSs include α-helix hairpins, β hairpins, coiled coils, Greek key, and β-α-β, α-turn-α, α-loop-α, and

Alexander E. Kister (ed.), *Protein Supersecondary Structures*, Methods in Molecular Biology, vol. 932,
DOI 10.1007/978-1-62703-065-6_5,

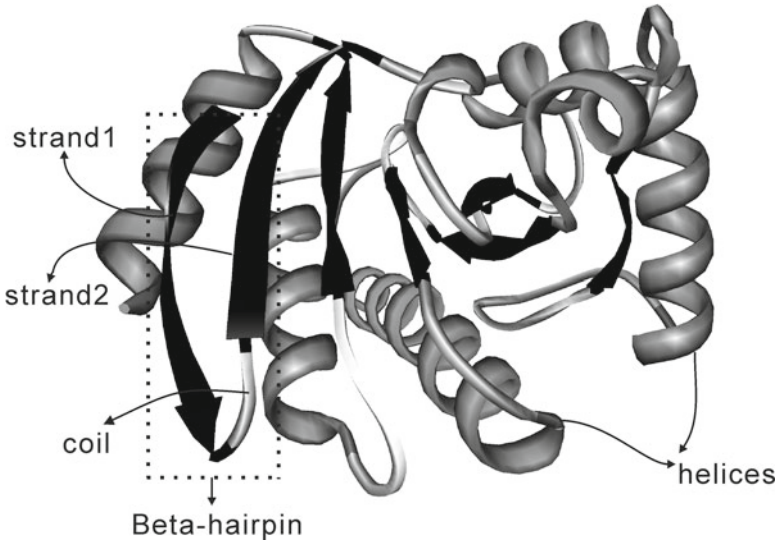


Fig. 1. Cartoon representation of the tertiary structure of chain A of AF1521 protein (PDB code: 2BFR). The α-helices are shown in *dark gray*, β-strands in *black*, and coils in *light gray*. The β hairpin supersecondary structure motif, which consists of strand1, strand2, and the coil between the two strands, is denoted using the *dotted rectangle.*

Rossmann motifs. The secondary and SSS elements are combined together, with help of various types of coils, to form the tertiary structure. An example that displays the secondary structures and the β hairpin SSS is given in Fig. 1.

In early 1970s Anfinsen demonstrated that the native tertiary structure is encoded in the primary structure (3) and this observation fueled the development of methods that predict the structure from the sequence. The need for these predictors is motivated by the fact that the tertiary structure is known for a relatively small number of proteins, i.e., as of mid-2011 about 70,000 protein structures are deposited in the Protein Data Bank (PDB) (4) when compared with 12.5 million nonredundant protein sequences in the RefSeq database (5), and the fact that experimental determination of protein structure is relatively expensive and time-consuming and cannot keep up with the rapid accumulation of the sequence data (6–9). One of the successful ways to predict the tertiary structure is to proceed in a stepwise fashion. First, we predict how the sequence folds into the secondary structure, then how these secondary structure elements come together to form SSSs, and finally the information about the secondary and SSSs is used to help in computational determination of the full three-dimensional molecule (10–15).

The last three decades have observed strong progress in the development of accurate predictors of the secondary structure, which currently provide predictions with about 82% accuracy (16).

Besides being useful for the prediction of the tertiary structure, the secondary structure predicted from the sequence is widely applied for the analysis and prediction of numerous structural and functional characteristics of proteins. These characteristics include multiple alignment (17), prediction of protein–ligand interactions (18–20), prediction of residue depth (21, 22), structural classes and folds (23–25), residue contacts (26, 27), disorder (28–30), folding rates and types (31–33), and target selection for structural genomics (34, 35), to name just a few. The secondary structure predictors enjoy strong interest, which could be quantified by the massive workloads that they handle. For instance, the Web server of the one of the most popular methods, PSIPRED, was reported in 2005 to receive over 15,000 requests per month (36). Another indicator is the fact that many of these methods receive high citations counts. A recent review (37) reported that seven methods were cited over 100 times and two of them, PSIPRED (36, 38, 39) and PHD (40, 41) were cited over 1,300 times.

The prediction of the SSS includes methods specialized for specific types of these structures, including β hairpins, coiled coils, and helix-turn-helix motifs. The first methods were developed in 1980s and to date about 20 predictors were developed. Similarly as the secondary structure predictors, the predictors of SSS found applications in numerous areas including analysis of amyloids (42, 43), microbial pathogens (44), and synthases (45), simulation of protein folding (46), analysis of relation between coiled coils and disorder (47), genome-wide studies of protein structure (48, 49), and prediction of protein domains (50). One interesting aspect is that the prediction of the secondary structure should provide useful information for the prediction of SSS. Two examples that exploit this relation are a prediction method by the Thornton's group (51) and the BhairPred method (52), both of which predict the β hairpins.

The secondary structure prediction field was reviewed a number of times. The earlier reviews summarized the most important advancements in this field, which were related to the use of sliding window, evolutionary information extracted from multiple sequence alignment, and machine-learning classifiers (53–55), and more recently due to the utilization of consensus-based approaches (56). More recent reviews concentrate on the evaluations and applications of the secondary structure predictors and provide practical advice for the users, such as the information concerning availability (16, 57, 58). The SSS prediction area was reviewed less extensively. The β hairpin and coiled coil predictors, as well as the secondary structure predictors were overviewed in 2006 (59) and a comparative analysis of the coiled coil predictors was presented in the same year (60). In this chapter, we summarize a more comprehensive set of recent secondary structure and SSS predictors.

We also demonstrate how the prediction of the secondary structure is used to implement a SSS predictor and provide several practical notes for the users.

2. Materials

2.1. Assignment of Secondary Structure

The secondary structure, which is assigned from the tertiary structure, is used for a variety of applications, including visualization (61–63) and classification of the protein folds (64–67), and as a ground truth to develop and evaluate the secondary and SSS predictors. Several annotation protocols have been developed over the last few decades. The first implementation was done in late 1970s by Levitt and Greer (68). This was followed by Kabsch and Sander who developed a method called dictionary of protein secondary structure (DSSP) (69), which is based on the detection of hydrogen bonds defined by an electrostatic criterion. Other, more recent, assignment methods include DEFINE (70), P-CURVE (71), STRIDE (72), P-SEA (73), XTLSSTR (74), SECSTR (75), KAKSI (76), Segno (77), PALSSE (78), SKSP (79), PROSIGN (80), and SABA (81). Moreover, the 2Struc Web server provides an integrated access to multiple annotation methods, which enables convenient comparison between different assignment protocols (82).

The DSSP remains to be the most widely used protocol (76), which is likely due to the fact that it is used to annotate depositions in the PDB and since it was used to evaluate secondary structure predictions in the two largest community based assessments: the Critical Assessment of techniques for protein Structure Prediction (CASP) (83) and the evaluation of automatic protein structure prediction (EVA) continuous benchmarking project (84). DSSP determines the secondary structures based on the patterns of hydrogen bonds, which are categorized into three major states: helices, sheets, and regions with irregular secondary structure. This method assigns one of the following eight secondary structure states for each of the structured residues (residues that have three-dimensional coordinates) in the protein sequence:

- G: (3-turn) 3_{10} helix, where the carboxyl group of a given amino acid forms a hydrogen bond with amid group of the residue three positions down in the sequence forming a tight, right-handed helical structure with 3 residues per turn.
- H: (4-turn) α-helix, which is similar to the 3-turn helix, except that the hydrogen bonds are formed between consecutive residues that are 4 positions away.
- I: (5-turn) π-helix, where the hydrogen bonding occurs between residues spaced 5 positions away. Most of the π-helices are right-handed.

- E: extended strand, where 2 or more strands are connected laterally by at least two hydrogen bonds forming a pleated sheet.
- B: an isolated beta-bridge, which is a single residue pair sheet formed based on the hydrogen bond.
- T: hydrogen bonded turn, which is a turn where a single hydrogen bond is formed between residues spaced 3, 4, or 5 positions away in the protein chain.
- S: bend, which corresponds to a fragment of protein sequence where the angle between the vector from C_i^α to C_{i+2}^α (C^α atoms at the ith and $i+2$th positions in the chain) and the vector from C_{i-2}^α to C_i^α is below 70°. The bend is the only non-hydrogen-bond-based regular secondary structure type.
- –: irregular secondary structure (also referred to as loop and random coil), which includes the remaining conformations.
- These eight secondary structure states are often mapped into the following three states (see Fig. 1):
- H: α-helix, which corresponds to the right or left handed cylindrical/helical conformations that include G, H, and I states.
- E: β-strand, which corresponds to pleated sheet structures that encompass E and B states.
- C: coil, which covers the remaining S, T, and - states.

The DSSP program is freely available from http://swift.cmbi.ru.nl/gv/dssp/.

2.2. Assignment of Supersecondary Structures

The SSS is composed of several adjacent secondary structure elements. Therefore, the assignment of the SSS relies on the assignment of the secondary structure. Among more than a dozen types of the SSSs, the β hairpins, coiled coils, and α-turn-α motifs received more attention due to the fact that they are present in a large number of protein structures and they have pivotal roles in the biological functions of proteins. The β hairpin motif comprises the second largest group of protein domain structures and is found in diverse protein families, including enzymes, transporter proteins, antibodies, and in viral coats (52). The coiled coil motifs mediate the oligomerization of a large number of proteins and are involved in regulation of gene expression, e.g., transcription factors (85). The α-turn-α (helix-turn-helix) motif is instrumental for DNA binding, i.e., majority of the DNA-binding proteins interact with DNA through this motif (86). The β hairpins, coiled coils, and α-turn-α motifs are defined as follows:

- β hairpin motif two strands that are adjacent in the primary structure, oriented in an antiparallel arrangement, and linked by a short loop;

- Coiled coil is build by two or more α-helices that wind around each other to form a supercoil.
- α-turn-α motif is composed of two α-helices joined by a short turn structure.

The β hairpin motifs are commonly annotated by PROMOTOF program (87), which also assigns several other SSS types, e.g., psi-loop and β-α-β motifs. Similar to DSSP, the PROMOTOF program assigns SSS based on the distances and hydrogen bonding between the residues. The coiled coils are usually assigned with the SOCKET program (88), which locates/annotates coiled-coil interactions based on the distances between multiple helical chains. The DNA-binding α-turn-α motifs are usually manually extracted from the DNA-binding proteins, since these motifs that do not interact with DNA are of lesser interest.

For users convenience, certain SSSs, such as the coiled coils and β-α-β motifs, can be accessed, analyzed, and visualized using specialized repositories such as CCPLUS (89) and TOPS (90). CCPLUS archives coiled coil structures identified by SOCKET for all structures in PDB. The TOPS database stores topological descriptions of protein structures, including the secondary structure and the chiralities of selected SSSs, e.g., β hairpins and β-α-β motifs.

2.3. Multiple Sequence Alignment

Multiple sequence alignment profile was introduced into the pipelines for the prediction of the secondary structure in early 1990s (91). Using the multiple sequence alignment profile rather than the primary sequence has led to a large improvement by 10% accuracy in the secondary structure prediction (91). The alignment profile is also often used in the prediction of the SSS (52, 59, 60). The multiple sequence alignment profile is generated from a given protein sequence in two steps. In the first step, sequences that are similar to the given input sequence are identified from a large sequence database, such as the *nr* (nonredundant) database provided by the National Center for Biotechnology Information (NCBI). In the second step, multiple sequence alignment is performed between the input sequence and its similar sequences and the profile is generated. An example of the multiple sequence alignment is given in Fig. 2 where eight similar sequences are identified for the input protein (we use the protein from Fig. 1). Each position of the input (query) sequence is represented by the frequencies of amino acid derived from the multiple sequence alignment to derive the profile. For instance, for the boxed position in Fig. 2, the counts of amino acids Tyr (Y), Ala (A) and Gly (G) are 5, 2, and 2, respectively. Therefore, this position is represented by a 20-dimensional vector (2/9, 0, 0, 0, 0, 2/9, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 5/9), where each value indicates the fraction of the corresponding amino acid type (amino acids are sorted in alphabetical order) in multiple sequence alignment at this position.

Query protein	…	**K**	**R**	**L**	**E**	**H**	**G**	**G**	**G**	**V**	**A**	**Y**	**A**	**I**	**A**	**K**	**A**	**C**	**A**	**G**	**D**	**A**	**G**	**L**	…
YP_002995377	…	K	Y	L	E	H	G	G	G	V	A	Y	A	I	A	K	A	A	S	G	D	V	R	E	…
YP_002958591	…	K	Y	L	E	H	G	G	G	V	A	Y	A	I	A	K	A	A	A	G	N	V	A	E	…
YP_003418650	…	S	Y	L	Q	H	G	G	G	V	A	Y	A	I	V	K	K	G	G	–	–	–	–	–	…
YP_002828572	…	S	Y	L	Q	H	G	G	G	V	A	Y	A	I	V	K	K	G	G	–	–	–	–	–	…
ZP_04861702	…	G	M	L	K	H	V	G	G	V	A	A	A	I	V	K	K	G	G	–	–	–	–	–	…
ZP_05391340	…	G	A	L	K	H	G	G	G	A	A	A	A	I	V	K	A	G	G	–	–	–	–	–	…
YP_003345806	…	E	Y	L	K	H	G	G	G	V	A	G	A	I	V	R	A	G	G	–	–	–	–	–	…
YP_003496764	…	S	H	L	K	M	G	G	G	V	A	G	A	I	R	R	A	G	G	–	–	–	–	–	…

Fig. 2. Multiple sequence alignment between the input (query) sequence, which is a fragment of chain A of the AF1521 protein shown in Fig. 5.1, and similar sequences identified in the *nr* database. The first row shows the query chain and the subsequent rows show the eight aligned proteins. Each row contains the protein sequence ID (the first column) and the corresponding amino acid sequence (the third and subsequent columns), where "…" denotes continuation of the chain and "–" denotes a gap, which means that this part of the sequence could not be aligned. The boxed column is used as an example to discuss generation of the multiple sequence alignment profile in Subheading 2.3.

The profile is composed of these 20-dimensional vectors for each position in the input protein chain.

The PSI-BLAST (Position-Specific Iterated BLAST) (92) algorithm was developed for the identification of distant similarity to a given input sequence. First, a list of closely related protein sequences is identified from a sequence database, such as the *nr* database. These sequences are combined into a general "profile," which summarizes significant features present in these sequences. Another query against the sequence database is run using this "profile," and a larger group of sequences is found. This larger group of sequences is used to construct another "profile," and the process is repeated. PSI-BLAST is more sensitive in picking up distant evolutionary relationships than a standard protein-protein BLAST that does not perform iterative repetitions. Since late 1990s, the PSI-BLAST is commonly used for the generation of multiple sequence alignment profile, which is named position-specific scoring matrix (PSSM) and which is often utilized in the prediction of secondary and SSSs. An example PSSM profile is given in Fig. 3. The BLAST and PSI-BLAST programs are available at http://blast.ncbi.nlm.nih.gov/.

3. Methods

3.1. Current Secondary Structure Prediction Methods

The prediction of the secondary structure is defined as mapping of each amino acid in the primary structure to one of the three (or eight) secondary structure states, most often as defined by the DSSP. Virtually all recent secondary structure predictors use a sliding window approach in which a local stretch of residues around a central position in the window is utilized to predict the secondary

		A	R	N	D	C	Q	E	G	H	I	L	K	M	F	P	S	T	W	Y	V	...
1	K	-1	2	0	-1	-3	1	1	-2	-1	-3	-3	5	-2	-3	-1	0	-1	-3	-2	-3	...
2	R	-2	2	-2	-3	-3	-1	-2	-3	1	-2	-1	0	-1	2	-3	-2	-2	2	7	-2	...
3	L	-2	-2	-4	-4	-1	-2	-3	-4	-3	2	4	-3	2	0	-3	-3	-1	-2	-1	1	...
4	E	-1	0	0	2	-4	2	5	-2	0	-4	-3	1	-2	-4	-1	0	-1	-3	-2	-3	...
5	H	-2	0	1	-1	-3	0	0	-2	8	-4	-3	-1	-2	-1	-2	-1	-2	-3	2	-3	...
6	G	0	-3	0	-1	-3	-2	-2	6	-2	-4	-4	-2	-3	-3	-2	0	-2	-3	-3	-4	...
7	G	0	-3	0	-1	-3	-2	-2	6	-2	-4	-4	-2	-3	-3	-2	0	-2	-3	-3	-4	...
8	G	0	-3	0	-1	-3	-2	-2	6	-2	-4	-4	-2	-3	-3	-2	0	-2	-3	-3	-4	...
9	V	0	-3	-3	-4	-1	-2	-3	-4	-3	3	1	-3	1	-1	-3	-2	0	-3	-1	4	...
10	A	4	-2	-2	-2	0	-1	-1	0	-2	-1	-2	-1	-1	-2	-1	1	0	-3	-2	0	...
11	Y	-2	-2	-3	-4	-2	-2	-2	-4	1	0	1	-2	0	3	-3	-2	-2	2	6	-1	...
12	A	4	-2	-2	-2	0	-1	-1	0	-2	-1	-2	-1	-1	-2	-1	1	0	-3	-2	0	...
13	I	-1	-3	-4	-3	-1	-3	-4	-4	-4	5	2	-3	1	0	-3	-3	-1	-3	-1	3	...
14	A	4	-2	-2	-2	0	-1	-1	0	-2	-1	-2	-1	-1	-2	-1	1	0	-3	-2	0	...
15	K	-1	2	0	-1	-3	1	1	-2	-1	-3	-3	5	-2	-3	-1	0	-1	-3	-2	-3	...
16	A	4	-2	-2	-2	0	-1	-1	0	-2	-1	-2	-1	-1	-2	-1	1	0	-3	-2	0	...
17	C	2	-3	-3	-3	9	-2	-3	-2	-3	-1	-1	-2	-1	-3	-2	0	-1	-3	-2	-1	...
18	A	4	-1	-1	-1	-1	-1	-1	0	-2	-2	-2	-1	-1	-3	-1	2	0	-3	-2	-1	...
19	G	0	-3	0	-1	-3	-2	-2	6	-2	-4	-4	-2	-3	-3	-2	0	-2	-3	-3	-4	...
20	D	-2	-2	1	6	-4	0	2	-1	-1	-3	-4	-1	-3	-4	-2	0	-1	-5	-3	-4	...
21	A	3	-2	-2	-2	-1	-1	-1	-1	-2	0	-1	-1	-1	-2	4	0	0	-3	-2	1	...
22	G	0	3	0	-1	-3	3	0	2	-1	-3	-3	1	-2	-3	-2	1	0	-3	-2	-3	...
23	L	-1	0	-1	0	-3	1	4	-2	-1	-1	1	2	0	-2	-2	-1	-1	-3	-2	-1	...
...																						

Fig. 3. Position-specific scoring matrix generated by PSI-BLAST for the input (query) sequence, which is a fragment of chain A of the AF1521 protein shown in Fig. 5.1. The first and second columns are the residue number and type, respectively, in the input protein chain. The subsequent columns provide values of the multiple sequence alignment profile for a substitution to an amino acid type indicated in the first row. Initially, a matrix $\{p_{i,j}\}$, where $p_{i,j}$ indicates the probability that the jth amino acid type (in columns) occurs at ith position in the input chain (in *rows*), is generated. The position-specific scoring matrix $\{m_{i,j}\}$ is defined as $m_{i,j} = \log(p_{i,j}/b_j)$, where b_j is the background frequency of the jth amino acid type.

structure state at the central position. Moreover, as one of the first steps in the prediction protocol, the state-of-the-art methods use PSI-BLAST to generate multiple alignment and/or PSSM that, with the help of the sliding window, are used to encode the input sequence. The early predictors were implemented based on a relatively simple statistical analysis of composition of the input sequence. The modern methods adopt sophisticated machine learning-based classifiers to represent the relation between the input sequence (or more precisely between the evolutionary information generated with PSI-BLAST) and the secondary structure states. In majority of cases, the classifiers are implemented using neural networks. However, different predictors use different numbers of networks (between one and hundreds), different types of networks (e.g., feed-forward and recurrent), and different sizes of the sliding window. These prediction methods are provided to the end users as standalone applications and/or as Web servers. The standalone programs are suitable for higher volume (for a large

Table 1
Summary of the recent sequence-based predictors of secondary structure

Name	Year last published	Prediction model	Availability
PSIPRED	2010	Neural network	WS + SP
SPINE	2009	Neural network	WS + SP
Frag1D	2009	Scoring function	SP
DISSPred	2009	Support vector machine + clustering	WS
SAM-T	2009	Neural network	WS + SP
PROTEUS	2008	Neural network	WS + SP
Jpred	2008	Neural network	WS
P.S.HMM	2007	Neural network + hidden Markov model	WS
Porter	2007	Neural network	WS + SP
OSS-HMM	2006	Hidden Markov model	SP
YASSPP	2006	Support vector machine	WS
YASPIN	2005	Neural network + hidden Markov model	WS
SABLE	2005	Neural network	WS + SP
SSpro	2005	Neural network	WS + SP

The "year last published" column provides the year of the publication of the most recent version of a given method. The "availability" column identifies whether a standalone program (SP) and/or a Web server (WS) is available. The methods are sorted by the year of their last publication in the descending order

number of proteins) predictions and they can be incorporated in other predictive pipelines, but they require installation by the user on a local computer. The Web servers are more convenient since they can be run using a Web browser and without the need for the local installation, but they are more difficult to use when applied to predict a large set of chains, i.e., some servers allow submission of one chain at the time and may have long wait times due to limited computational resources and a long queue of requests from other users. Moreover, recent comparative survey (16) shows that the differences in the predictive quality for a given predictor between its standalone and Web server versions depend on the frequency with which the underlying databases, which are used to calculate the evolutionary information and to perform homology modeling, are updated. Sometimes these updates are more frequent for the Web server, and in other cases for the standalone package.

Table 1 summarizes 15 methods, including PSIPRED (36, 38, 39), SPINE (93, 94), Frag1D (95), DISSPred (96), SAM-T

(97–101), PROTEUS (102, 103), Jpred (104–106), P.S.HMM (107), Porter (108, 109), OS-HMM (110), YASSPP (111), YASPIN (112), SABLE (113), and SSpro (114, 115), that predict the 3-state secondary structure and which were published since 2005 inclusive. Older methods were reviewed in refs. 53–55. We note that only a few methods, including SSpro8 (115) and SAM-T08 (101), predict the 8-state secondary structure. Following, we discuss in greater detail the methods that offer Web servers, as arguably these are used by a larger number of users. We summarize their architecture, provide location of their implementation, and briefly discuss their predictive performance. We note that the predictive quality should be considered with a grain of salt since different methods were evaluated on different datasets and using different test protocols (see Note 1). However, we primarily utilize fairly consistent results that were published in two recent comparative studies (see Note 3) (16, 37). Moreover, recent research shows that improved predictive performance could be obtained by post-processing of the secondary structure predictions (see Note 4) (116).

3.2. PSIPRED

PSIPRED is one of the most popular prediction methods (see Note 2); e.g., it received the largest number of citations as shown in (16, 37). This method was developed in late 1990s by Jones group at the University College London (38), and later improved and updated, with the most recent version 3.0 (39). PSIPRED is characterized by a relatively simple design which utilizes just two neural networks. This method was ranked as top predictor in the CASP3 and CASP4 competitions, and was recently evaluated to provide 3-state secondary structure predictions with 81% accuracy (16, 39). The current version bundles the secondary structure predictions with the prediction of transmembrane topology and fold recognition.

Inputs: PSSM generated from the input protein sequence using PSI-BLAST
Architecture: ensemble of two neural networks
Availability: http://bioinf.cs.ucl.ac.uk/psipred/

3.3. Jpred

Jpred was developed in late 1990s by Barton group at the University of Dundee (105). This method was updated a few times, with the most recent version Jpred 3 (104, 106). Similarly as PSIPRED, Jpred was demonstrated to provide about 81% accuracy for the 3-state secondary structure prediction (104). The Web server implementation of Jpred couples the secondary structure predictions with the prediction of solvent accessibility and prediction of coiled coils using COILS algorithm (117).

Inputs: hidden Markov model profiles and PSSM generated from the input protein sequence using HMMer (118) and PSI-BLAST, respectively
Architecture: ensemble of neural networks
Availability: http://www.compbio.dundee.ac.uk/www-jpred/

3.4. SSpro

SSpro was introduced in early 2000 by the Baldi group at the University of California, Irvine (115). The latest version 4.5 (114) utilizes homology modeling, which is based on alignment to known tertiary structures from PDB, and achieves over 82% accuracy (16). The SSpro 4.0 was also ranked as one of the top secondary structure prediction servers in the EVA benchmark (119). SSpro is part of a comprehensive prediction center called SCRATCH, which also includes predictions of secondary structure in 8-states using SSpro8 (115), and prediction of solvent accessibility, disorder, contact numbers and contact maps, domains, disulfide bonds, B-cell epitopes, solubility upon overexpression, antigenicity, and tertiary structure.

Inputs: sequence profiles generated from the input protein sequence using PSI-BLAST
Architecture: ensemble of recurrent neural networks
Availability: http://scratch.proteomics.ics.uci.edu/

3.5. SAM-T

SAM-T is a family of methods which are under development since late 1990s by Karplus lab at the University of California at Santa Cruz. They include SAM-T98 (97), SAM-T99 (98), SAM-T02 (99), SAM-T04 (100), and SAM-T08 (101). The server outputs secondary structure prediction using multiple annotation protocols, including the 3- and 8-state DSSP. It also offers a number of other predictions (the predicted secondary structure is used as an input to calculate some of these predictions) including the tertiary structure, solvent accessibility, residue-residue contacts, multiple sequence alignments of putative homologs, and lists and alignment to potential templates with known structure.

Inputs: multiple alignment generated from the input protein sequence using PSI-BLAST
Architecture: neural network
Availability: http://compbio.soe.ucsc.edu/SAM_T08/T08-query.html

3.6. SABLE

The SABLE predictor was developed by Meller group at the University of Cincinnati (113). The Web server that implements this method was used close to 200,000 times since it became operational in 2003. Two recent comparative studies (16, 39) and prior evaluations within the framework of the EVA initiative show that SABLE achieves accuracy of about 78%. The Web server of the current version 2 also includes prediction of solvent accessibility and transmembrane domains.

Inputs: PSSM generated from the input protein sequence using PSI-BLAST
Architecture: ensemble of recurrent neural networks
Availability: http://sable.cchmc.org/

3.7. YASPIN

The YASPIN method was developed by Heringa lab at the Vrije Universiteit in 2004 (112). This is a hybrid method that utilizes a neural network and a hidden Markov model. One of the key characteristics of this method is that, as shown by the authors, it provides accurate predictions of β-strands (112). The predictive performance of YASPIN was evaluated using EVA benchmark and more recently in two comparative assessments (16, 39), which show that this method provides predictions with accuracy in the 76–79% range.

Inputs: PSSM generated from the input protein sequence using PSI-BLAST
Architecture: Two-level hybrid design with neural network in the 1st level and hidden Markov model in the 2nd level
Availability: http://www.ibi.vu.nl/programs/yaspinwww/

3.8. PORTER

This predictor was developed by Pollastri group at the University College Dublin (109). The Web server that implements PORTER was utilized over 170,000 times since 2004 when it was released. This predictor was upgraded in 2007 to include homology modeling (108). The original and the homology-enhanced versions were recently shown to provide 79% (16) and 83% accuracy (37), respectively. PORTER is a part of a comprehensive predictive platform called DISTILL (120), which also incorporates predictors of relative solvent accessibility, residue-residue contact density, contacts maps, subcellular localization, and tertiary structure.

Inputs: PSSM generated from the input protein sequence using PSI-BLAST
Architecture: ensemble of recurrent neural networks
Availability: http://distill.ucd.ie/porter/

3.9. YASSPP

YASSPP was designed by Karypis lab at the University of Minnesota in 2005 (111). This is one of the few modern predictors that do not utilize neural network classifiers, but instead it uses multiple support vector machine learners. This method was shown to provide similar predictive quality to PSIPRED (111). The YASSPP predictor is bundled with several other predictors for transmembrane helices, disorder, solvent accessibility, contact order, and DNA-binding and ligand-binding residues in the MONSTER server at http://bio.dtc.umn.edu/monster/.

Inputs: PSSM generated from the input protein sequence using PSI-BLAST
Architecture: ensemble of six support vector machines
Availability: http://glaros.dtc.umn.edu/yasspp/

3.10. PROTEUS

This secondary structure prediction approach was developed by Wishart group at the University of Alberta around 2005 (103). PROTEUS is a consensus-based method in which outputs of three

secondary structure predictors, namely PSIPRED (37), Jnet (106), and an in-house TRANSSEC (103), are fed into a neural network. The predictions from the neural network are combined with the results based on homology modeling to generate the final output. PROTEUS is characterized by accuracy of about 81%, which was shown both the authors (102) and in a recent comparative survey (16). This predictor was incorporated into an integrated system called PROTEUS2, which additionally offers prediction of signal peptides, transmembrane helices and strands, and tertiary structure (102).

Inputs: multiple alignment generated from the input protein sequence using PSI-BLAST
Architecture: neural network that utilizes consensus of three secondary structure predictors
Availability: http://wks16338.biology.ualberta.ca/proteus2/

3.11. SPINE

The SPINE method originated at the Zhou group at the Indiana University–Purdue University in mid 2000s. The initial implementation (94), which was completed at the SUNY at Buffalo, was recently upgraded to create SPINE X (93). This predictor is characterized by relatively strong predictive performance with accuracy at about 81% (16). An important feature of this method is that it also provides predictions of backbone torsion angles, which give more detailed insights into the conformation of the backbone when compared with the secondary structure. The Web server of SPINE X also provides predictions of solvent accessibility (93) and fluctuations of the torsion angles (121).

Inputs: PSSM generated from the input protein sequence using PSI-BLAST
Architecture: ensemble of neural networks
Availability: http://sparks.informatics.iupui.edu/SPINE-X/

3.12. P.S.HMM

The P.S.HMM predictor was developed at the University of Copenhagen and University of Southampton (107). Similar to YASPIN, this is a hybrid of a neural network and a hidden Markov model. The P.S.HMM method uses the hidden Markov model to produce initial predictions that are refined with help of a small neural network, while YASPIN performs predictions in the reverse order. The unique characteristic of this method is the fact that the hidden Markov model was designed utilizing genetic algorithms. This predictor provides outputs with 69% accuracy, as recently evaluated in (16), which is consistent with results presented by the authors (107).

Inputs: sequence profiles generated from the input protein sequence using PSI-BLAST
Architecture: Two-level hybrid design with hidden Markov model in the 1st level and neural network in the 2nd level
Availability: http://wonk.med.upenn.edu/

3.13. DISSPred

The DISSPred approach was recently introduced by Hirst group at the University of Nottingham (96). Similar to SPINE, this method predicts both the 3-state secondary structure and the backbone torsion angles. The unique characteristic of DISSPred is that the predictions are cross-linked as inputs, i.e., predicted secondary structure is used to predict torsion angles and vice versa. The author estimated the accuracy of this method to be at 80% (96).

Inputs: PSSM generated from the input protein sequence using PSI-BLAST
Architecture: ensemble of support vector machines and clustering
Availability: http://comp.chem.nottingham.ac.uk/disspred/

3.14. Supersecondary Structure Prediction Methods

Since SSS predictors are designed for a specific type of the SSSs, e.g., SpiriCoil only predicts the coiled coils (48), the prediction of the SSS is defined as the assignment of each residue in the primary structure to two states: a state indicating the formation of a certain SSS type and another state indicating any other conformation. Similar to the prediction of the secondary structure, majority of the recent SSS predictors use a sliding window approach in which a local stretch of residues around a central position in the window is utilized to predict the SSS state at the central position. However, the architectures of the methods that were proposed for the prediction of different types of SSSs vary more substantially when compared with the fairly uniform architectures of the modern secondary structure predictors.

One of the early attempts for the prediction of β hairpin utilized the predicted secondary structure and similarity score between the predicted sequence and a library of β hairpin structures (51). More recent β hairpin predictors use the predicted secondary structure and some sequence-based descriptors to represent the predicted sequence (52, 122–126). Moreover, several types of prediction algorithms, including neural networks, support vector machines, quadratic discriminants, and random forests, were used for the prediction of β hairpin motifs.

The first attempt to predict coiled coils was based on scoring the propensity for formation of coiled coils in the predicted (input) sequence by calculating similarity to a PSSM derived from a statistical analysis of a coiled coil database (67). More recent studies utilize the hidden Markov models and the PSSM profile to represent the input sequence (48, 127–131).

The initial study on the prediction of α-turn-α motif was also based on scoring similarity between the predicted sequence and the α-turn-α structure library (132). Subsequently, a statistical method that utilizes a pattern dictionary of the primary sequences was developed (133). A more recent predictor exploits the potential for using structural knowledge to improve the detection of the helix-turn-helix motifs (134). This method uses a linear predictor that takes similarity scores between the input protein structure and a template library of α-turn-α structures as its inputs.

Table 2 summarizes 16 SSS prediction methods, including 6 β hairpin predictors: method by de la Cruz et al. (51), BhairPred (52), and methods by Hu et al. (125), Zou et al. (124), Xia et al. (123), and Jia et al. (122); 7 recent coiled coil predictors: MultiCoil (135), MARCOIL (131), PCOILS (130), bCIPA (129), Paircoil2 (128), CCHMM_PROF (127), and SpiriCoil (48); and 3 α-turn-α predictors: method by Dodd and Egan (132), GYM (133), and HTHquery (134) (see Note 6). The older coiled coil predictors were reviewed in ref. 60.

We note that some of the methods for the prediction of β hairpin and α-turn-α structures do not offer any implementation, i.e., neither a standalone program nor a Web server, which substantially limits their utility. Following, we discuss in greater detail the representative predictors for each type of the SSSs, with particular emphasis on the β hairpin predictors that utilize the predicted secondary structure.

3.15. BhairPred

The BhairPred predictor was developed by Raghava group at the Institute of Microbial Technology, India in 2005 (52). The predictions are performed using a support vector machine-based model,

Table 2
Summary of the recent sequence-based predictors of supersecondary structure

Supersecondary structure type	Name (authors)	Year last published	Prediction model	Availability
β Hairpin	Jia et al.	2011	Random forest	NA
	Xia et al.	2010	Support vector machine	NA
	Zou et al.	2009	Increment of diversity + quadratic discriminant analysis	NA
	Hu et al.	2008	Support vector machine	NA
	BhairPred	2005	Support vector machine	WS
	de la Cruz et al.	2002	Neural network	NA
Coiled coil	SpiriCoil	2010	Hidden Markov model	WS
	CCHMM_PROF	2009	Hidden Markov model	WS
	Paircoil2	2006	Pairwise residue probabilities	WS + SP
	bCIPA	2006	no model	WS
	PCOILS	2005	Residue probabilities	WS
	MARCOIL	2002	Hidden Markov model	SP
	MultiCoil	1997	Pairwise residue probabilities	WS + SP
α-Turn-α	HTHquery	2005	Linear predictor	WS
	GYM	2002	Statistical method	WS
	Dodd et al.	1990	Similarity scoring	NA

The "year last published" column provides the year of the publication of the most recent version of a given method. The "availability" column identifies whether a standalone program (SP), and/or a web server (WS), or neither (NA) is available. The methods are sorted by the year of their last publication in the descending order for a given type of the supersecondary structures

which is shown by the authors to outperform a neural network-based predictor. Each residue is encoded using its PSSM profile, secondary structure predicted with PSPPRED, and solvent accessibility predicted with the NETASA method (136). BhairPred was shown to provide predictions with accuracy in the 71–78% range on two independent test sets (52).

Inputs: PSSM generated from the input protein sequence using PSI-BLAST, 3-state secondary structure predicted using PSIPRED, and solvent accessibility predicted with NETASA
Architecture: support vector machine
Availability: http://www.imtech.res.in/raghava/bhairpred/

3.16. CCHMM_PROF

The CCHMM_PROF predictor was developed by Fariselli group at the University of Bologna in 2009 (127). CCHMM_PROF is the first hidden Markov model-based predictor of coiled-coils that exploits the PSSM profile to encoding the input sequence. The major difference between CCHMM_PROF and other hidden Markov models is that the states of CCHMM_PROF produce vectors instead of symbols. The CCHMM_PROF achieved accuracy of 97% when discriminating between sequence that do and do not contain coiled coils (127). This predictor finds the location of the coiled coil segments with 80% success rate and was shown to outperform older solutions (127).

Inputs: PSSM generated from the input protein sequence using PSI-BLAST
Architecture: hidden Markov model
Availability: http://gpcr.biocomp.unibo.it/cgi/predictors/cchmmprof/pred_cchmmprof.cgi

3.17. HTHquery

This method was developed by Thornton group at the European Bioinformatics Institute in 2005 (134). HTHquery takes a protein structure as input and tests whether this structure has a helix–turn–helix motif which could bind to DNA. The input protein is compared with a set of structural templates and putative α-turn-α regions with the smallest RMSD to each template in a template library are determined using Kabsch algorithm (137). The accessible surface area and the electrostatic motif score are computed for each of these putative regions using NACCESS (http://www.bioinf.manchester.ac.uk/naccess/) and the methods described in (138), respectively. Next, these inputs, i.e., the minimum RMSD, the accessible surface area, and the electrostatic motif score, are inputted into a linear predictor. HTHquery provides predictions with a true positive rate of 83.5% and a false positive rate of 0.8% (134).

Inputs: protein structure
Architecture: linear predictor
Availability: http:// www.ebi.ac.uk/thornton-srv/databases/HTHquery

3.18. Supersecondary Structure Prediction by Using Predicted Secondary Structure

Since SSS is composed of several adjacent secondary structure elements, the prediction of the secondary structures should be a useful input to predict SSS (see Note 5). Two SSS predictors, BhairPred (52) and the method developed by Thornton group (51), have utilized the predicted secondary structure for the identification of β hairpins. Following, we discuss latter method to demonstrate how the predicted secondary structure is used for the prediction of the SSS. The Thornton et al. method consists of 5 steps:

Step 1. Predict the secondary structure for a given input sequence using PHD method (40).

Step 2. Label all β-coil-β patterns in the predicted secondary structure.

Step 3. Score similarity between each labeled pattern and each hairpin structure in a template library. The similarity vector between a β-coil-β pattern and a hairpin structure consists of 14 values, including 6 values that measure similarity of the secondary structures, 1 value that measures similarity of the solvent accessibility, 1 value that indicates the presence of turns, 2 values that describe specific pair interactions and nonspecific distance-based contacts, and 4 values that represent the secondary structure patterns related to residue length.

Step 4. The 14 similarity scores are processed by a neural network that produces a discrete output, 0 or 1, indicating that the strand-coil-strand pattern is unlikely or likely, respectively, to form a β hairpin.

Step 5. For a given labeled β-coil-β pattern, a set of similarity scores is generated for each template hairpin, and therefore the neural network generates an output for each template hairpin. The labeled β-coil-β pattern is predicted as β hairpin if the outputs are set to 1 for more than 10 template hairpins.

The working of the Thornton et al. method is visualized in Fig. 4.

4. Notes

1. The predictive quality of the secondary structure predictors was empirically compared in several large-scale, world-wide initiatives including CASP (83), Critical Assessment of Fully Automated Structure Prediction (CAFASP) (139), and EVA (84, 119). Only the early CASP and CAFASP meetings, including CASP3 in 1998, CASP4 and CAFASP2 in 2000, and CASP5 and CAFASP3 in 2002, included the evaluation of the secondary structure predictions. Later on, the evaluations were carried out within the EVA platform. Its most recent release monitored 13 predictors. However, EVA was last updated in mid 2008.

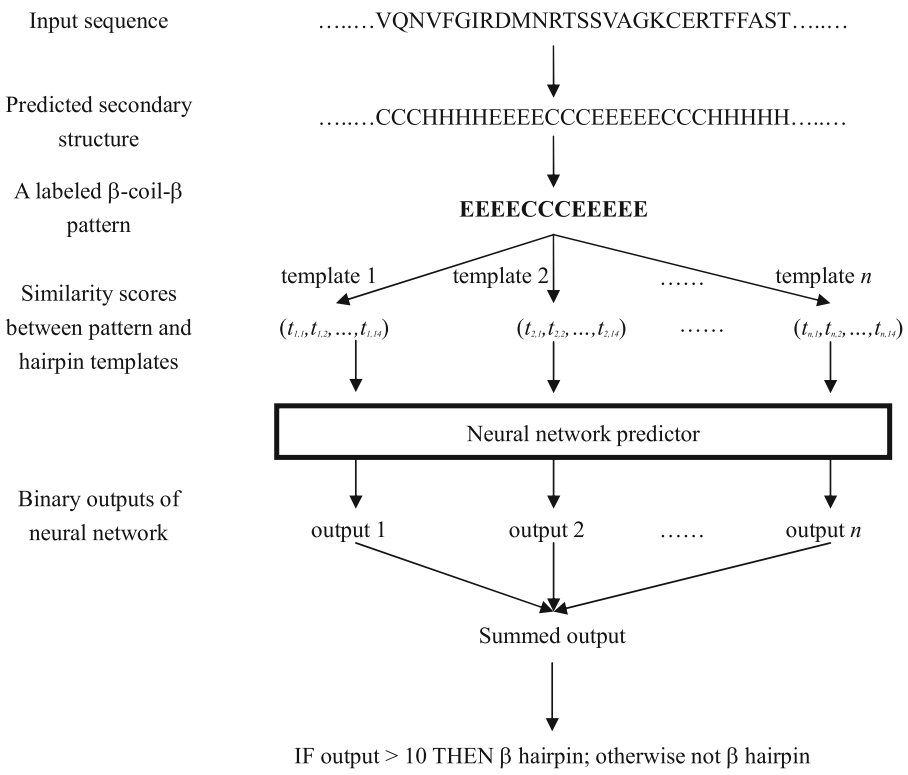


Fig. 4. The architecture of the β hairpin predictor proposed by the Thornton group (51).

2. A recent large-scale comparative analysis (16) has revealed a number of interesting and practical observations concerning state-of-the-art in the secondary structure prediction. The accuracy of the 3-state prediction based on the DSSP assignment is currently at 82%, and the use of a simple consensus-based prediction improves the accuracy by additional 2%. The homology modeling-based methods, such as SSpro and PROTEUS, are shown to be better by 1.5% accuracy than the ab initio approaches. The neural network-based methods are demonstrated to outperform the hidden Markov model-based solutions.
3. As shown in (16), the current secondary structure predictors are characterized by several drawbacks, which motivate further research in this area. They confuse 1–6% of strand residues with helical residues and vice versa (these are significant mistakes) and they perform poorly when predicting residues in the beta-bridge and 3_{10} helix conformations.
4. The arguably most popular secondary structure predictor is PSIPRED. This method is implemented as both a standalone application (version 2.6) and a Web server (version 3.0).

PSIPRED is continuously improved, usually with a major upgrade every year and with weekly updates of the databases. The current (as of June 2011) count of citations in the ISI Web of Knowledge to the paper that describes the original PSIPRED algorithm (38) is close to 1,700, which demonstrates the high utility of this method.

5. Prediction of the SSSs could be potentially improved by utilizing a consensus of different approaches. As shown in a relatively recent comparative analysis of coiled coil predictors (60), the best-performing Marcoil has generated many false positives for highly charged fragments, while the runner-up PCOILS provided better predictions for these fragments. This suggests that the results generated by different coiled coil predictors could be complementary.
6. The major obstacle to utilize the predicted secondary structure in the prediction of the SSSs, which was observed in mid 2000s, was (is) the inadequate quality of the predicted secondary structure. For instance, only about half of the native β hairpins were predicted with the strand-coil-strand secondary structure pattern (51). The use the native rather than the predicted secondary structure was shown to lead to a significant improvement in the prediction of the SSSs (52).

Acknowledgment

This work was supported by the Alberta Ingenuity and Alberta Innovates Graduate Student Scholarship to KC and the NSERC Discovery grant to LK.

References

1. Pauling L, Corey RB, Branson HR (1951) The structure of proteins; two hydrogen-bonded helical configurations of the polypeptide chain. Proc Natl Acad Sci USA 37:205–211
2. Pauling L, Corey RB (1951) The pleated sheet, a new layer configuration of polypeptide chains. Proc Natl Acad Sci U S A 37:251–256
3. Anfinsen CB (1973) Principles that govern the folding of protein chains. Science 181:223–230
4. Berman HM, Westbrook J, Feng Z et al (2000) The Protein Data Bank. Nucleic Acids Res 28:235–242
5. Pruitt KD, Tatusova T, Klimke W et al (2009) NCBI Reference sequences: current status, policy, and new initiatives. Nucleic Acids Res 37(Database issue):D32–D36
6. Gronwald W, Kalbitzer HR (2010) Automated protein NMR structure determination in solution. Methods Mol Biol 673:95–127
7. Chayen NE (2009) High-throughput protein crystallization. Adv Protein Chem Struct Biol 77:1–22
8. Zhang Y (2009) Protein structure prediction: when is it useful? Curr Opin Struct Biol 19:145–155
9. Ginalski K (2006) Comparative modeling for protein structure prediction. Curr Opin Struct Biol 16:172–177
10. Yang Y, Faraggi E, Zhao H et al. (2011) Improving protein fold recognition and template-based modeling by employing probabilistic-based matching between predicted

one-dimensional structural properties of the query and corresponding native properties of templates. Bioinformatics 27(15):2076–2082
11. Roy A, Kucukural A, Zhang Y (2010) I-TASSER: a unified platform for automated protein structure and function prediction. Nat Protoc 5:725–738
12. Faraggi E, Yang Y, Zhang S et al (2009) Predicting continuous local structure and the effect of its substitution for secondary structure in fragment-free protein structure prediction. Structure 17:1515–1527
13. Wu S, Zhang Y (2008) MUSTER: improving protein sequence profile–profile alignments by using multiple sources of structure information. Proteins 72:547–556
14. Zhou H, Skolnick J (2007) Ab initio protein structure prediction using chunk-TASSER. Biophys J 93:1510–1518
15. Skolnick J (2006) In quest of an empirical potential for protein structure prediction. Curr Opin Struct Biol 16:166–171
16. Zhang H, Zhang T, Chen K et al (2011) Critical assessment of high-throughput standalone methods for secondary structure prediction. Brief Bioinform 12(6):672–688
17. Pei J, Grishin NV (2007) PROMALS: towards accurate multiple sequence alignments of distantly related proteins. Bioinformatics 23: 802–808
18. Zhang T, Zhang H, Chen K et al (2010) Analysis and prediction of RNA-binding residues using sequence, evolutionary conservation, and predicted secondary structure and solvent accessibility. Curr Protein Pept Sci 11:609–628
19. Pulim V, Bienkowska J, Berger B (2008) LTHREADER: prediction of extracellular ligand-receptor interactions in cytokines using localized threading. Protein Sci 17:279–292
20. Fischer JD, Mayer CE, Söding J (2008) Prediction of protein functional residues from sequence by probability density estimation. Bioinformatics 24:613–620
21. Song J, Tan H, Mahmood K et al (2009) Prodepth: predict residue depth by support vector regression approach from protein sequences only. PLoS One 4:e7072
22. Zhang H, Zhang T, Chen K et al (2008) Sequence based residue depth prediction using evolutionary information and predicted secondary structure. BMC Bioinform 9:388
23. Mizianty MJ, Kurgan L (2009) Modular prediction of protein structural classes from sequences of twilight-zone identity with predicting sequences. BMC Bioinform 10:414
24. Kurgan L, Cios K, Chen K (2008) SCPRED: accurate prediction of protein structural class for sequences of twilight-zone similarity with predicting sequences. BMC Bioinform 9:226
25. Chen K, Kurgan L (2007) PFRES: protein fold classification by using evolutionary information and predicted secondary structure. Bioinformatics 23:2843–2850
26. Xue B, Faraggi E, Zhou Y (2009) Predicting residue-residue contact maps by a two-layer, integrated neural-network method. Proteins 76:176–183
27. Cheng J, Baldi P (2007) Improved residue contact prediction using support vector machines and a large feature set. BMC Bioinform 8:113
28. Mizianty MJ, Stach W, Chen K et al (2010) Improved sequence-based prediction of disordered regions with multilayer fusion of multiple information sources. Bioinformatics 26:i489–i496
29. Mizianty MJ, Zhang T, Xue B et al (2011) In-silico prediction of disorder content using hybrid sequence representation. BMC Bioinform 12:245
30. Schlessinger A, Punta M, Yachdav G et al (2009) Improved disorder prediction by combination of orthogonal approaches. PLoS One 4:e4433
31. Zhang H, Zhang T, Gao J et al. (2012) Determination of protein folding kinetic types using sequence and predicted secondary structure and solvent accessibility. Amino Acids. 42(1):271–283
32. Gao J, Zhang T, Zhang H et al (2010) Accurate prediction of protein folding rates from sequence and sequence-derived residue flexibility and solvent accessibility. Proteins 78:2114–2130
33. Jiang Y, Iglinski P, Kurgan L (2009) Prediction of protein folding rates from primary sequences using hybrid sequence representation. J Comput Chem 30:772–783
34. Mizianty M, Kurgan L (2011) Sequence-based prediction of protein crystallization, purification, and production propensity. Bioinformatics 27:i24–i33
35. Slabinski L, Jaroszewski L, Rychlewski L et al (2007) XtalPred: a web server for prediction of protein crystallizability. Bioinformatics 23:3403–3405
36. Bryson K, McGuffin LJ, Marsden RL et al (2005) Protein structure prediction servers at University College London. Nucleic Acids Res 33:W36–W38
37. Kurgan L, Miri Disfani F (2011) Structural protein descriptors in 1-dimension and their sequence-based predictions. Curr Protein Pept Sci. 12(6):470–489
38. Jones D (1999) Protein secondary structure prediction based on position-specific scoring matrices. J Mol Biol 292:195–202

39. Buchan DW, Ward SM, Lobley AE et al (2010) Protein annotation and modelling servers at University College London. Nucleic Acids Res 38:W563–W568
40. Rost B (1996) PHD: predicting one-dimensional protein structure by profile-based neural networks. Methods Enzymol 266:525–539
41. Rost B, Yachdav G, Liu J (2004) The predict protein server. Nucleic Acids Res 32(Web Server issue):W321–W326
42. O'Donnell CW, Waldispühl J, Lis M et al (2011) A method for probing the mutational landscape of amyloid structure. Bioinformatics 27:i34–i42
43. Bryan A Jr, Menke M, Cowen LJ et al (2009) BETASCAN: probable beta-amyloids identified by pairwise probabilistic analysis. PLoS Comput Biol 5:e1000333
44. Bradley P, Cowen L, Menke M et al (2001) BETAWRAP: successful prediction of parallel beta-helices from primary sequence reveals an association with many microbial pathogens. Proc Natl Acad Sci U S A 98:14819–14824
45. Hornung T, Volkov OA, Zaida TM et al (2008) Structure of the cytosolic part of the subunit b-dimer of Escherichia coli F0F1-ATP synthase. Biophys J 94:5053–5064
46. Sun ZR, Cui Y, Ling LJ et al (1998) Molecular dynamics simulation of protein folding with supersecondary structure constraints. J Protein Chem 17:765–769
47. Szappanos B, Süveges D, Nyitray L et al (2010) Folded-unfolded cross-predictions and protein evolution: the case study of coiled-coils. FEBS Lett 584:1623–1627
48. Rackham OJ, Madera M, Armstrong CT et al (2010) The evolution and structure prediction of coiled coils across all genomes. J Mol Biol 403:480–493
49. Gerstein M, Hegyi H (1998) Comparing genomes in terms of protein structure: surveys of a finite parts list. FEMS Microbiol Rev 22:277–304
50. Reddy CC, Shameer K, Offmann BO et al (2008) PURE: a webserver for the prediction of domains in unassigned regions in proteins. BMC Bioinform 9:281
51. de la Cruz X, Hutchinson EG, Shepherd A et al (2002) Toward predicting protein topology: an approach to identifying beta hairpins. Proc Natl Acad Sci U S A 99:11157–11162
52. Kumar M, Bhasin M, Natt NK et al (2005) BhairPred: prediction of beta-hairpins in a protein from multiple alignment information using ANN and SVM techniques. Nucleic Acids Res 33(Web Server issue):W154–W159
53. Barton GJ (1995) Protein secondary structure prediction. Curr Opin Struct Biol 5:372–376
54. Heringa J (2000) Computational methods for protein secondary structure prediction using multiple sequence alignments. Curr Protein Pept Sci 1:273–301
55. Rost B (2001) Protein secondary structure prediction continues to rise. J Struct Biol 134:204–218
56. Albrecht M, Tosatto SC, Lengauer T et al (2003) Simple consensus procedures are effective and sufficient in secondary structure prediction. Protein Eng 16:459–462
57. Rost B (2009) Prediction of protein structure in 1D—secondary structure, membrane regions, and solvent accessibility. In: Bourne PE, Weissig H (eds) Structural bioinformatics, 2nd edn. Wiley, New York, pp 679–714
58. Pirovano W, Heringa J (2010) Protein secondary structure prediction. Methods Mol Biol 609:327–348
59. Singh M (2006) Predicting protein secondary and supersecondary structure. In: Aluru S (ed) Handbook of computational molecular biology. Chapman and Hall/CRC Press, pp 29.1–29.29
60. Gruber M, Söding J, Lupas AN (2006) Comparative analysis of coiled-coil prediction methods. J Struct Biol 155:140–145
61. Kolodny R, Honig B (2006) VISTAL-a new 2D visualization tool of protein 3D structural alignments. Bioinformatics 22:2166–2167
62. Moreland JL, Gramada A, Buzko OV et al (2005) The molecular biology toolkit (MBT): a modular platform for developing molecular visualization applications. BMC Bioinformatics 6:21
63. Porollo AA, Adamczak R, Meller J (2004) POLYVIEW: a flexible visualization tool for structural and functional annotations of proteins. Bioinformatics 20:2460–2462
64. Murzin AG, Brenner SE, Hubbard T et al (1995) SCOP: a structural classification of proteins database for the investigation of sequences and structures. J Mol Biol 247:536–540
65. Orengo CA, Michie AD, Jones S et al (1997) CATH—a hierarchic classification of protein domain structures. Structure 5:1093–1108
66. Andreeva A, Howorth D, Chandonia JM et al (2008) Data growth and its impact on the SCOP database: new developments. Nucl Acids Res 36:D419–D425
67. Cuff AL, Sillitoe I, Lewis T et al (2011) Extending CATH: increasing coverage of the protein structure universe and linking structure with function. Nucleic Acids Res 39(Database issue):D420–D426
68. Levitt M, Greer J (1997) Automatic identification of secondary structure in globular proteins. J Mol Biol 114:181–239
69. Kabsch W, Sander C (1983) Dictionary of protein secondary structure: pattern recognition

of hydrogen-bonded and geometrical features. Biopolymers 22:2577–2637
70. Richards F, Kundrot CE (1988) Identification of structural motifs from protein coordinate data: secondary structure and first-level supersecondary structure. Proteins 3:71–84
71. Sklenar H, Etchebest C, Lavery R (1989) Describing protein structure: a general algorithm yielding complete helicoidal parameters and a unique overall axis. Proteins 6:46–60
72. Frishman D, Argos P (1995) Knowledge-based protein secondary structure assignment. Proteins 23:566–579
73. Labesse G, Colloc'h N, Pothier J et al (1997) P-SEA: a new efficient assignment of secondary structure from C alpha trace of proteins. Comput Appl Biosci 13:291–295
74. King S, Johnson WC (1999) Assigning secondary structure from protein coordinate data. Proteins 3:313–320
75. Fodje M, Al-Karadaghi S (2002) Occurrence, conformational features and amino acid propensities for the pi-helix. Protein Eng 15:353–358
76. Martin J, Letellier G, Marin A et al (2005) Protein secondary structure assignment revisited: a detailed analysis of different assignment methods. BMC Struct Biol 5:17
77. Cubellis MV, Cailliez F, Lovell SC (2005) Secondary structure assignment that accurately reflects physical and evolutionary characteristics. BMC Bioinform 6(Suppl 4):S8
78. Majumdar I, Krishna SS, Grishin NV (2005) PALSSE: a program to delineate linear secondary structural elements from protein structures. BMC Bioinform 6:202
79. Zhang W, Dunker AK, Zhou Y (2008) Assessing secondary structure assignment of protein structures by using pairwise sequence-alignment benchmarks. Proteins 71:61–67
80. Hosseini SR, Sadeghi M, Pezeshk H et al (2008) PROSIGN: a method for protein secondary structure assignment based on three-dimensional coordinates of consecutive C(alpha) atoms. Comput Biol Chem 32:406–411
81. Park SY, Yoo MJ, Shin J et al (2011) SABA (secondary structure assignment program based on only alpha carbons): a novel pseudo center geometrical criterion for accurate assignment of protein secondary structures. BMB Rep 44:118–122
82. Klose DP, Wallace BA, Janes RW (2010) 2Struc: the secondary structure server. Bioinformatics 26:2624–2625
83. Moult J, Pedersen JT, Judson R et al (1995) A large-scale experiment to assess protein structure prediction methods. Proteins. 23:ii-v.
84. Koh IY, Eyrich VA, Marti-Renom MA et al (2003) EVA: evaluation of protein structure prediction servers. Nucleic Acids Res 31:3311–3315
85. Parry DA (2008) Fifty years of coiled-coils and alpha-helical bundles: a close relationship between sequence and structure. J Struct Biol 163:258–269
86. Pellegrini-Calace M, Thornton JM (2005) Detecting DNA-binding helix-turn-helix structural motifs using sequence and structure information. Nucleic Acids Res 33:2129–2140
87. Hutchinson EG, Thornton JM (1996) PROMOTIF—a program to identify and analyze structural motifs in proteins. Protein Sci 5:212–220
88. Walshaw J, Woolfson DN (2001) Socket: a program for identifying and analysing coiled-coil motifs within protein structures. J Mol Biol 307:1427–1450
89. Testa OD, Moutevelis E, Woolfson DN (2009) CC+: a relational database of coiled-coil structures. Nucleic Acids Res 37(Database issue):D315–D322
90. Michalopoulos I, Torrance GM, Gilbert DR et al (2004) TOPS: an enhanced database of protein structural topology. Nucleic Acids Res 32(Database issue):D251–D254
91. Rost B, Sander C (1993) Improved prediction of protein secondary structure by use of sequence profiles and neural networks. Proc Natl Acad Sci U S A 90:7558–7562
92. Altschul SF, Madden TL, Schäffer AA et al (1997) Gapped BLAST and PSI-BLAST: a new generation of protein database search programs. Nucleic Acids Res 25:3389–3402
93. Faraggi E, Xue B, Zhou Y (2009) Improving the prediction accuracy of residue solvent accessibility and real-value backbone torsion angles of proteins by guided-learning through a two-layer neural network. Proteins 74:847–856
94. Dor O, Zhou Y (2007) Achieving 80% ten-fold cross-validated accuracy for secondary structure prediction by large-scale training. Proteins 66:838–845
95. Zhou T, Shu N, Hovmöller S (2010) A novel method for accurate one-dimensional protein structure prediction based on fragment matching. Bioinformatics 26:470–477
96. Kountouris P, Hirst JD (2009) Prediction of backbone dihedral angles and protein secondary structure using support vector machines. BMC Bioinform 10:437
97. Karplus K, Barrett C, Hughey R (1998) Hidden Markov models for detecting remote protein homologies. Bioinformatics 14:846–856
98. Karplus K, Karchin R, Barrett C et al (2001) What is the value added by human intervention

in protein structure prediction? Proteins 5(Suppl):86–91
99. Karplus K, Karchin R, Draper J et al (2003) Combining local-structure, fold-recognition, and new fold methods for protein structure prediction. Proteins 53:491–496
100. Karplus K, Katzman S, Shackleford G et al (2005) SAM-T04: what is new in protein-structure prediction for CASP6. Proteins 61(Suppl 7):135–142
101. Karplus K (2009) SAM-T08, HMM-based protein structure prediction. Nucleic Acids Res 37(Web Server issue):W492–W497
102. Montgomerie S, Cruz JA, Shrivastava S et al (2008) PROTEUS2: a web server for comprehensive protein structure prediction and structure-based annotation. Nucleic Acids Res 36(Web Server issue):W202–W209
103. Montgomerie S, Sundararaj S, Gallin WJ et al (2006) Improving the accuracy of protein secondary structure prediction using structural alignment. BMC Bioinform 7:301
104. Cole C, Barber JD, Barton GJ (2008) The Jpred 3 secondary structure prediction server. Nucleic Acids Res 2008(36):W197–W201
105. Cuff JA, Clamp ME, Siddiqui AS et al (1998) JPred: a consensus secondary structure prediction server. Bioinformatics 14:892–893
106. Cuff J, Barton GJ (2000) Application of multiple sequence alignment profiles to improve protein secondary structure prediction. Proteins 40:502–511
107. Won K, Hamelryck T, Prügel-Bennett A et al (2007) An evolutionary method for learning HMM structure: prediction of protein secondary structure. BMC Bioinform 8:357
108. Pollastri G, Martin AJM, Mooney C et al (2007) Accurate prediction of protein secondary structure and solvent accessibility by consensus combiners of sequence and structure information. BMC Bioinform 8:201
109. Pollastri G, McLysaght A (2005) Porter: a new, accurate server for protein secondary structure prediction. Bioinformatics 21:1719–1720
110. Martin J, Gibrat JF, Rodolphe F (2006) Analysis of an optimal hidden Markov model for secondary structure prediction. BMC Struct Biol 6:25
111. Karypis G (2006) YASSPP: better kernels and coding schemes lead to improvements in protein secondary structure prediction. Proteins 64:575–586
112. Lin K, Simossis VA, Taylor WR et al (2005) A simple and fast secondary structure prediction algorithm using hidden neural networks. Bioinformatics 21:152–159
113. Adamczak R, Porollo A, Meller J (2005) Combining prediction of secondary structure and solvent accessibility in proteins. Proteins 59:467–475
114. Cheng J, Randall AZ, Sweredoski MJ et al (2005) SCRATCH: a protein structure and structural feature prediction server. Nucleic Acids Res 33:W72–W76
115. Pollastri G, Przybylski D, Rost B et al (2002) Improving the prediction of protein secondary structure in three and eight classes using recurrent neural networks and profiles. Proteins 47:228–235
116. Madera M, Calmus R, Thiltgen G et al (2010) Improving protein secondary structure prediction using a simple k-mer model. Bioinformatics 26:596–602
117. Lupas A, Van Dyke M, Stock J (1991) Predicting coiled coils from protein sequences. Science 252:1162–1164
118. Eddy SR (1998) Profile hidden Markov models. Bioinformatics 14:755–763
119. Eyrich VA, Martí-Renom MA, Przybylski D et al (2001) EVA: continuous automatic evaluation of protein structure prediction servers. Bioinformatics 17:1242–1243
120. Bau D, Martin AJ, Mooney C et al (2006) Distill: a suite of web servers for the prediction of one-, two- and three-dimensional structural features of proteins. BMC Bioinform 7:402
121. Zhang T, Faraggi E, Zhou Y (2010) Fluctuations of backbone torsion angles obtained from NMR-determined structures and their prediction. Proteins 78:3353–3362
122. Jia SC, Hu XZ (2011) Using random forest algorithm to predict β-hairpin motifs. Protein Pept Lett 18:609–617
123. Xia JF, Wu M, You ZH et al (2010) Prediction of beta-hairpins in proteins using physicochemical properties and structure information. Protein Pept Lett 17:1123–1128
124. Zou D, He Z, He J (2009) Beta-hairpin prediction with quadratic discriminant analysis using diversity measure. J Comput Chem 30:2277–2284
125. Hu XZ, Li QZ (2008) Prediction of the beta-hairpins in proteins using support vector machine. Protein J 27:115–122
126. Kuhn M, Meiler J, Baker D (2004) Strand-loop-strand motifs: prediction of hairpins and diverging turns in proteins. Proteins 54:282–288
127. Bartoli L, Fariselli P, Krogh A et al (2009) CCHMM_PROF: a HMM-based coiled-coil predictor with evolutionary information. Bioinformatics 25:2757–2763
128. McDonnell AV, Jiang T, Keating AE et al (2006) Paircoil2: improved prediction of

coiled coils from sequence. Bioinformatics 2006(22):356–358

129. Mason JM, Schmitz MA, Müller KM et al (2006) Semirational design of Jun-Fos coiled coils with increased affinity: Universal implications for leucine zipper prediction and design. Proc Natl Acad Sci U S A 103:8989–8994
130. Gruber M, Söding J, Lupas AN (2005) REPPER—repeats and their periodicities in fibrous proteins. Nucleic Acids Res 33(Web Server issue):W239–W243
131. Delorenzi M, Speed T (2002) An HMM model for coiled-coil domains and a comparison with PSSM-based predictions. Bioinformatics 18:617–625
132. Dodd IB, Egan JB (1990) Improved detection of helix-turn-helix DNA-binding motifs in protein sequences. Nucleic Acids Res 18:5019–5026
133. Narasimhan G, Bu C, Gao Y et al (2002) Mining protein sequences for motifs. J Comput Biol 9:707–720
134. Ferrer-Costa C, Shanahan HP, Jones S et al (2005) HTHquery: a method for detecting DNA-binding proteins with a helix-turn-helix structural motif. Bioinformatics 21:3679–3680
135. Wolf E, Kim PS, Berger B (1997) MultiCoil: a program for predicting two- and three-stranded coiled coils. Protein Sci 6(6):1179–1189
136. Ahmad S, Gromiha MM (2002) NETASA: neural network based prediction of solvent accessibility. Bioinformatics 18(6):819–824
137. Kabsch W (1976) A solution for the best rotation to relate two sets of vectors. Acta Cryst A32:922–923
138. Shanahan H, Garcia M, Jones S et al (2004) Identifying DNA binding proteins using structural motifs and the electrostatic potential. Nucleic Acids Res 32:4732–4741
139. Fischer D, Barret C, Bryson K et al (1999) CAFASP-1: critical assessment of fully automated structure prediction methods. Proteins Suppl 3:209–217

Chapter 6

A Survey of Machine Learning Methods for Secondary and Supersecondary Protein Structure Prediction

Hui Kian Ho, Lei Zhang, Kotagiri Ramamohanarao, and Shawn Martin

Abstract

In this chapter we provide a survey of protein secondary and supersecondary structure prediction using methods from machine learning. Our focus is on machine learning methods applicable to β-hairpin and β-sheet prediction, but we also discuss methods for more general supersecondary structure prediction. We provide background on the secondary and supersecondary structures that we discuss, the features used to describe them, and the basic theory behind the machine learning methods used. We survey the machine learning methods available for secondary and supersecondary structure prediction and compare them where possible.

Key words: β-Hairpins, β-Sheets, Artificial neural networks, Support vector machines, Supersecondary structure feature vectors

1. Introduction

Proteins are the machinery of the cell, and their behavior is highly influenced by their shape in physical space (1). However, experimental determination of protein structure by either nuclear magnetic resonance (NMR) or X-ray crystallography is a difficult process, despite the fact that protein sequences can be obtained using high-throughput genomic methods. This situation has encouraged numerous efforts to develop computational methods for predicting protein structure from amino acid sequence.

Unfortunately, computational prediction of protein structure remains an unsolved problem, although progress continues (2). In particular, quantum mechanical approaches are intractable due to the large number of atoms involved, and methods based on empirically derived molecular force fields are only partially effective (3).

Alexander E. Kister (ed.), *Protein Supersecondary Structures*, Methods in Molecular Biology, vol. 932,
DOI 10.1007/978-1-62703-065-6_6,

For these reasons, there have been efforts towards predicting simpler protein substructures, namely, secondary and supersecondary structures such as α-helices, β-sheets, β-hairpins, and β-α-β motifs. In addition, success in predicting secondary and supersecondary structures may improve full structure prediction based on molecular force fields (4–8).

In this chapter, we survey secondary and supersecondary prediction efforts from a machine learning perspective, with an emphasis on β-hairpins and β-sheets. In Subheading 2, we describe the secondary and supersecondary motifs considered by the machine learning methods, along with the basic theory behind two of the prominent machine learning approaches. In Subheading 3, we survey the methods available for secondary and supersecondary predictions, with a focus on β-hairpin and β-sheet prediction. For β-hairpin prediction, we provide an outline that encompasses the general approach in that situation. The situation is more complicated for β-sheets and general supersecondary structures and cannot be summarized in a single outline that is applicable to all methods. In Subheading 4, we describe properties of the methods and speculate about alternative methods.

2. Materials

In this section we provide some background on the tools necessary for performing machine-learning based predictions of protein secondary and supersecondary structure. This background includes some information on the secondary and supersecondary structures that we consider and the machine learning methods used.

2.1. Secondary and Supersecondary Structures

β-hairpins. The β-hairpin is a supersecondary structure (SSS) motif formed when two β-strands close together in the sequence are hydrogen bonded in an antiparallel orientation. The loop region between the β-strands usually consists of two to five residues and may be hydrogen bonded in a 3_{10}, α, or π configuration. A β-hairpin is shown (as part of a β-sheet) in Fig. 1.

The formation of β-hairpins can play an important role in the stabilization of protein tertiary structures (9). Predicting β-hairpins can also reduce the conformational search space and increase the accuracy of ab initio tertiary structure prediction procedures (10). Furthermore, the β-hairpin is considered to be the simplest SSS motif and can serve as a building block for more complex motifs (9). Therefore, β-hairpin prediction can provide a foundation for addressing more complex structure prediction tasks (10).

β-sheets. One of the main challenges in protein structure prediction is the identification of the long-range interactions between the strands in β-sheets. These interactions provide a β-sheet with a

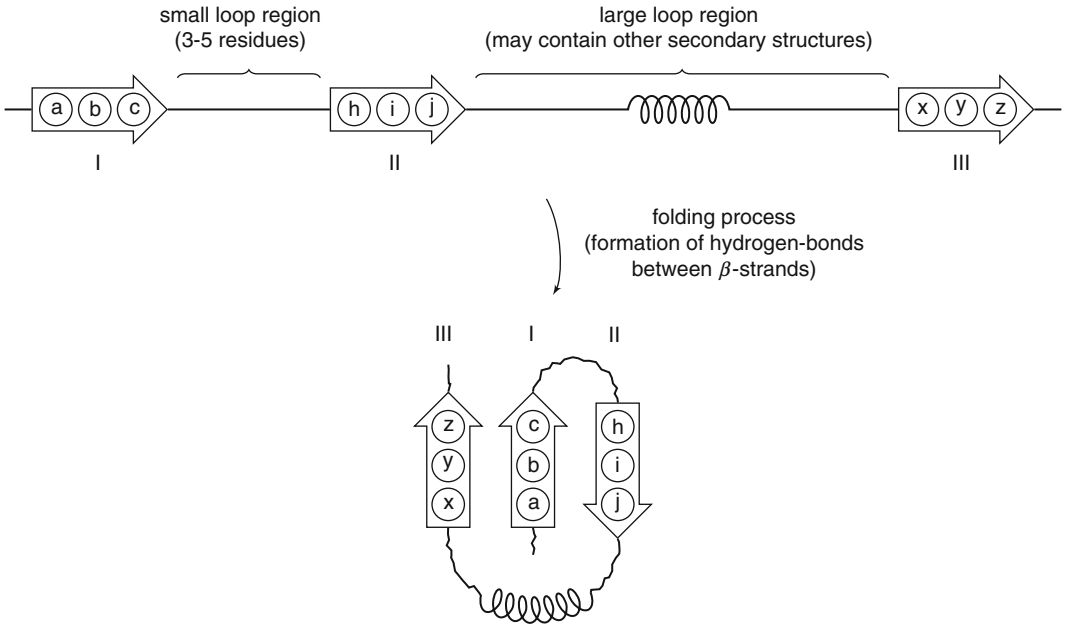


Fig. 1. β-Sheet formation. A β-sheet can form via hydrogen bonding between near and/or distantly separated β-strands. The β-sheet can contain β-hairpins separated by a small loop region, as seen in this example between strands I and II. Or, paired strands can be separated by the majority of the protein sequence, as seen in this example between strands II and III. These possibilities add to the complexity of the β-sheet topology prediction problem.

distinct topology and function. Predicting β-sheet topologies can give insight into the mechanisms of β-sheet formation (11) and supplement conventional tertiary structure prediction approaches (4, 6–8).

β-Sheet topology prediction can be viewed as a superclass of β-hairpin prediction. The key differences arise from the fact that β-hairpins are composed of only two β-strands with a limited loop segment. In contrast, β-sheets as a whole often contain more than two β-strands in varying configurations whose interstrand loop segments may span tens or even hundreds of residues. β-sheet formation is illustrated in Fig. 1.

The topology of a β-sheet is described by the ordering, pairing, and alignment between its β-strands. A pairing between β-strands refers to two adjacent hydrogen bonded β-strands. The ordering of a β-sheet with n strands is implied by its $n-1$ pairings (n pairings for a β-barrel). An alignment of a β-strand pair consists of the specific adjacent interstrand β-residue pairings. β-residue pairings need not be hydrogen bonded.

The β-sheet plays an important role in many proteins and is implicated in a number of neurodegenerative diseases (12, 13). The folding mechanisms involved in β-sheet formation are not well understood, primarily due to the long-range nature of their stabilizing hydrogen bonds (14, 15). Understanding the different factors

involved in β-sheet formation can provide insight into how β-sheets function (11).

General Supersecondary Structures. In addition to β-hairpins and β-sheets, prediction methods exist for more general supersecondary structures. These structures include SSS motifs based on strand-loop-strand configurations, protein classes such as α, β, $\alpha+\beta$, and α/β, as defined by the structural classification of proteins (SCOP) database (16), and particular classifications of loops between supersecondary structure elements (SSEs).

2.2. Machine Learning

Secondary and supersecondary structure prediction is often performed by inferring a functional model from some dataset of examples. This model is then used to make further predictions. Although there are a wide variety of such models available in machine learning, the two most commonly used models are artificial neural networks (ANNs) and Support Vector Machines (SVMs). In this subsection we provide a brief description of these two methods, a discussion of the features used to describe amino acid sequences corresponding to secondary and supersecondary structures, and a short primer on the various performance metrics used to measure the validity of a machine learning model.

Artificial Neural Networks. The philosophy behind an ANN is to represent a functional relationship in a manner analogous to the method thought to be present in a biological neural network. In theory, such a model would have many of the advantages of, for example, a human brain—able to learn, generalize, adapt, and have fault tolerance (17, 18).

Computationally, an ANN can be described as a weighted graph, where nodes are neurons and directed edges are connections between neurons. Edges are weighted according to the influence of one neuron on another. Each neuron is described by a function:

$$y_i = \theta\left(\sum_{j=1}^{n} w_j x_j - w_0\right) \tag{1}$$

Originally proposed by McCulloch and Pitts (19), the function θ is called an activation function. This function is typically a step function from 0 to 1, a piecewise linear function, a sigmoid, or a Gaussian. The weights w_j combined with the inputs x_j determine whether a neuron will be activated (return a "1" for output y_i) or inhibited (return a "0"). Thus, a positive weight corresponds to an excitatory synapse and a negative weight corresponds to an inhibitory synapse. The neurons are combined together according to the underlying graph to obtain a model function.

The ANN architecture can be used to perform various machine learning tasks, including classification, regression, clustering, and

function approximation (17). In the case of protein supersecondary structure prediction, ANNs are often used as classifiers. A classifier is a functional model that predicts a discrete value (e.g., whether or not two β-strands form a β-hairpin).

The weights in an ANN have to be learned from labeled examples, a process also known as training. The training process is generally performed by gradient descent, but the exact optimization algorithms used depends on the particular architecture of the ANN (17). One of the original training algorithms is known as error-correction (20). Other algorithms include Boltzmann learning, Hebbian learning, and competitive learning (17, 18). The training process for ANNs is a difficult nonlinear optimization, with few theoretical guarantees. Once trained, the resulting weights w_j are generally uninterpretable, especially for complex architectures. Nevertheless, ANNs are widely used in many fields (17), including prediction of protein supersecondary structure.

Support Vector Machines. The most common use of a SVM is as a linear binary classifier (21, 22). As in the case of ANNs, SVMs must be trained. Since it is relatively simple to train a SVM, we describe the process in slightly more detail than was provided for ANNs. Suppose we have a dataset $\{\mathbf{x}_i\} \subseteq R^n$, and that each point $\mathbf{x}_i$ is an n-dimensional vector in our dataset with a corresponding class label $y_i \in \{\pm 1\}$. Our goal is to separate the points in our dataset according to their class label. Since there are two classes, this is known as binary classification. A SVM attempts this classification by using a linear hyperplane $\mathbf{w}^T\mathbf{x} + b$, $\mathbf{w} \neq 0$.

A SVM uses an optimal separating hyperplane known as the maximal margin hyperplane. The hyperplane margin is twice the distance from the separating hyperplane to the nearest point in one (or the other) of the two classes. The SVM hyperplane is found by solving the quadratic programming problem (23, 24)

$$\begin{aligned} \max \ & \alpha \frac{1}{2} \sum_{i,j} y_i y_j \alpha_i \alpha_j \mathbf{x}_i^T \mathbf{x}_j - \sum_i \alpha_i \\ \text{s.t.} \ & \sum_i y_i \alpha_i = 0 \\ & 0 \leq \alpha_i \leq C \end{aligned} \tag{2}$$

where $\mathbf{w} = \sum_i y_i \alpha_i \mathbf{x}_i$ is the normal to the SVM hyperplane. Using $\mathbf{w}$ we form the SVM decision function $f(\mathbf{x}) = \text{sign}(\mathbf{w}^T\mathbf{x} + b)$, where b is obtained implicitly (25, 26). We note that $\alpha_i \neq 0$ only when $\mathbf{x}_i$ is a support vector.

The SVM problem given in Eq. 2 only applies to datasets $\{\mathbf{x}_i\} \subseteq R^n$. Often, however, we want to use a SVM on a dataset that is not a subset of R^n. This occurs in the case of secondary and supersecondary protein structure prediction, when we use amino acid sequences to describe our data. Fortunately, there is a ready

solution to this problem, formalized for SVMs in the use of kernel functions.

Suppose our data $\{\mathbf{x}_i\} \subseteq S$, where S might be the set of all finite length protein sequences. We can then define a kernel function as a map $k : S \times S \rightarrow R$ such that

$$k(\mathbf{x}_i, \mathbf{x}_j) = \Phi(\mathbf{x}_i)^T \Phi(\mathbf{x}_j) \quad (3)$$

where Φ: $S \rightarrow F$ is a map from our original data space S into a space F with a defined dot product such as R^n. In fact, the map Φ: $S \rightarrow F$ is a key component of most, if not all, machine learning methods for secondary and supersecondary protein structure prediction. Such a map is used for both SVMs and ANNs and describes the manner in which an amino acid sequence is encoded as a vector of numeric quantities for input to a classifier. Such an encoding is known in machine learning as a feature vector and is discussed in the following sub-section.

Once we have defined a kernel function, we simply replace the dot product $\mathbf{x}_i^T \mathbf{x}_j$ in Eq. 2 with the kernel $k(\mathbf{x}_i, \mathbf{x}_j)$ to obtain the full SVM quadratic programming problem. (A similar procedure can be used for any method that is written in terms of dot products, thus giving rise to the moniker kernel methods.)

SVMs have a number of advantages over competing methods, including a unique solution of a fairly straightforward quadratic programming problem (compared to a nonunique solution of a nonlinear optimization used in a neural network), ability to employ nonlinearity by choice of kernel, and widespread availability of software (27, 28). On the other hand, SVMs can be memory intensive $O(n^2)$ and the flexibility in choice of kernel can make SVMs difficult for beginners. Neither of these disadvantages, however, has slowed the widespread use of SVMs in bioinformatics, including secondary and supersecondary protein structure prediction.

Feature Vectors. One of the primary challenges of casting a secondary or supersecondary prediction problem as a machine learning problem is the encoding of the underlying protein amino acid sequences. These sequences must be represented as feature vectors in R^n for input to the machine learning algorithm. Choosing an encoding appropriate to the physical system under study is a major factor in the success or failure of the resulting method. Here we describe some of the more widely used feature vectors for amino acid sequences.

One of the simplest representations is Amino Acid Composition (AAC). AAC counts the occurrences of each amino acid in a sequence and has been used by Nakashima et al. (29) to determine protein classes. A derived representation, Pseudo Amino Acid Composition (PseAAC) is often considered to be a better choice for use in prediction tasks (30–32). PseAAC introduces additional

features based on hydrophobicity and side-chain masses of different amino acids. Another feature, the increment of diversity (ID) has been used to predict protein classes (33, 34). The ID measure compares a query sequence to a precomputed profile for each prediction class based on a quantification of diversity.

The previously described features, however, do not include residue order. To address this potential shortcoming, a few sequence preserving methods have been developed. Markov chain based feature spaces can preserve sequential information, including the dipeptide frequency method used by Lin and Li (34). A Markov chain model modified by using n-gap transition probability has been used to predict protein classes (35). Both Baldi et al. (36) and Brown et al. (37) predict β-sheet topology using features based on short subsequences of an amino acid sequence. These subsequences are obtained via a "sliding window" which travels over the amino acid sequence.

Another common class of methods for encoding amino acid sequences uses multiple sequence alignment. A primary example is the position specific scoring matrix (PSSM) (38). PSSMs measure how similar a given amino acid sequence is to a library of sequences known to have certain properties (e.g., SSS motifs). PSSMs have been used for SSS motif prediction (39, 40).

Performance Metrics. The performance of a machine learning method is typically assessed using either a predetermined training/test set split or cross-validation. In the case of the training/test set split, the original dataset is split into two mutually exclusive sets, a training set and a test set. The training set is usually a large fraction of the original dataset (say 80%) and the test set contains the remaining fraction (e.g., 20%) of the data. The machine learning model is trained on the training set and then used to make predictions on the test set. Model performance is assessed by computing statistics on the test set predictions, including accuracy, sensitivity, specificity, and precision (to be defined shortly).

These same steps are performed in the case of cross-validation, but multiple training/test sets are generated and model performance is averaged. For n-fold cross-validation, the original dataset is split into n equal subsets. Each subset becomes a test set. Then, for each test set, a model is trained on the complement of that set (the training set) and predictions are made on the test set using the resulting model. Performance is again assessed by computing accuracy, sensitivity, specificity, and precision, now averaged over all test sets.

To define accuracy, sensitivity, specificity, and precision, we denote true positive predictions by tp, true negative predictions by tn, false positive predictions by fp, and false negative predictions by fn. Now accuracy can be written $(tp+tn)/(tp+tn+fp+fn)$, sensitivity can be written $tp/(tp+fn)$, specificity can be written $tn/(tn+fp)$, and precision can be written $tp/(tp+fp)$. Accuracy measures how well the model

performs overall, sensitivity measures how well the model performs on positive predictions, specificity measures how well the model performs on negative predictions, and precision measures how relevant the positive predictions are relative to the total number of positive samples.

3. Methods

In this section we survey machine learning methods available for supersecondary structure prediction. Our focus is β-hairpin prediction and β-sheet prediction, followed by general supersecondary structure prediction. We compare methods where possible.

3.1. β-Hairpins

Survey. One of the first β-hairpin prediction methods was proposed by de La Cruz et al. (41). This method is carried out using the following steps: (1) predict the secondary structures using the PHD web server (42); (2) label the β-loop-β occurrences in the sequence; (3) scan a database of known β-hairpins for similar instances (similarity is

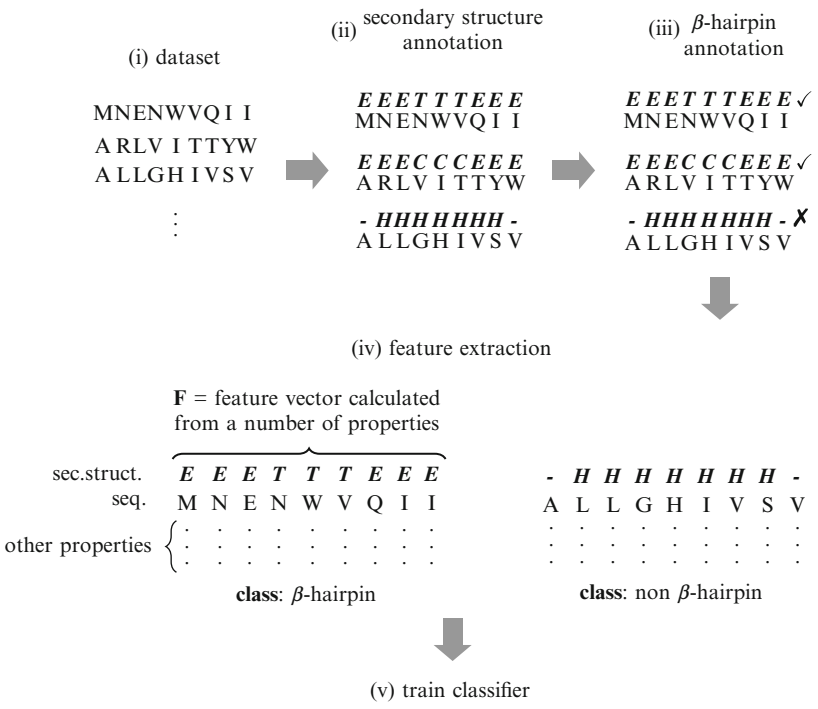


Fig. 2. Training a β-hairpin predictor. The general workflow for training a β-hairpin predictor: (**i**) a dataset containing the sequences of a number of observed β-hairpin and non β-hairpin instances is obtained from the protein data bank (PDB) (www.pdb.org); (**ii**) each sequence is annotated with secondary structure assignments, typically using the dictionary of protein secondary structure (DSSP) (50); (**iii**) the β-hairpins are identified according to a set criteria (which may vary between methods) based on the observed secondary structures; (**iv**) the feature vectors for each instance are obtained (note that the chosen properties vary greatly between methods and can be a major determinant of predictive performance); (**v**) a well known machine algorithm (e.g., ANN or SVM) is used to learn a predictive model from each of the training instances via their feature vectors.

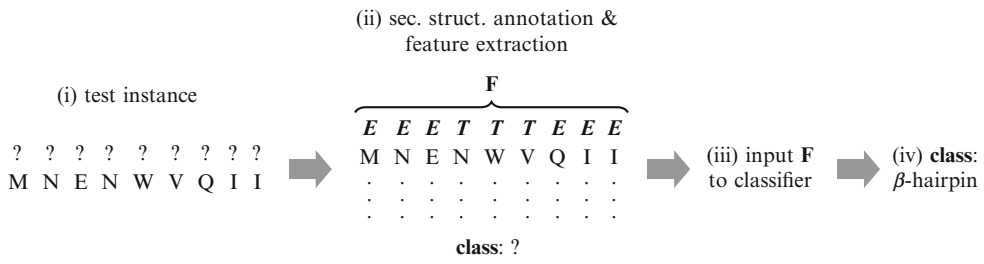


Fig. 3. Classification of a New Sample. The general workflow for classification of a new sample: (**i**) a test instance is obtained which consists of only a protein sequence; (**ii**) secondary structures are assigned to the sequence (the features of the sequence are extracted using the same procedure as in Fig. 2.); (**iii**) the feature vector is input to the classifier; (**iv**) a class of either "β-hairpin" or "non β-hairpin" is assigned to the test instance using the trained classification model.

calculated from structure and sequence information and produces 14 scores for each β-hairpin in the database); and (4) use these scores as input features to an ANN that returns a "1" or "0" denoting whether or not a sample is likely or unlikely to form a β-hairpin, respectively.

Subsequent approaches in the literature have followed the prediction model set by de La Cruz et al. (41). The general approach is illustrated in Fig. 2. After training, prediction is performed according to the illustration in Fig. 3. These approaches use well-known machine learning algorithms with new or improved feature vectors for discriminating β-hairpins. The machine learning algorithms used typically include the previously mentioned ANN or the more recent SVM.

Kuhn et al. (10) uses homology information from the PSI-BLAST web server (43), predicted secondary structures, and sequence profiles from PSIPRED (44) (another web server) as inputs to an ANN. This method is unique in that it uses separate ANNs for predicting the start and end locations of a loop region, respectively.

Kumar et al. (45) use feature vectors similar to those of Kuhn et al. (10), but include predicted solvent accessibility information. The authors compared two predictors, an ANN and a SVM. The SVM was shown to outperform the ANN.

Kuhn et al. (10) predicted only the turns between strands and found that reducing the sequence identity of the dataset from 50 to 25% resulted in only a small reduction in prediction accuracy. This may be explained by the use of PSI-BLAST profiles that are capable of capturing evolutionary information from distantly related sequences with low sequence identity.

Hu and Li (46) proposed a method that was not reliant on homology or secondary structural information. Instead, position specific residue probabilities and the increment of diversity (ID) were calculated from each segment and used as input features to a SVM. The ID measure was used to compute the similarity between the dipeptide frequency distributions in a case with known β-hairpins

and one without β-hairpins. The ID measure was also used in subsequent methods (47, 48).

Zou et al. (48) used quadratic discriminant analysis (QDA) to develop a classifier that uses a quadratic combination of ID-encoded features based on the amino acid substring distributions of each example. A quadratic classifier was also implemented by Hu et al. (47). This approach used features based on position specific residue probabilities, ID-encoded dipeptide frequency and hydropathy values, pseudo-amino acid composition (30), and accessible surface area autocorrelation. The classifier took the form of a quadratic discriminant function similar to that used by Zou et al. (48).

The most recent approach, proposed by Xia et al. (49), uses a SVM with a feature vector that includes physicochemical and secondary structure information along with the conformational propensity of adopting a β-hairpin structure. The conformational propensity is empirically derived from their dataset. The five physicochemical properties (hydrophobicity, hydrophilicity, polarity, polarizability, and average accessible surface area) are encoded into the feature vector using five values. Each value represents the auto covariance (AC) of a particular physicochemical property calculated from the entire sequence. The AC procedure accounts for the possible interactions between neighboring residues.

Comparisons. An objective comparison between each of the methods described for β-hairpin prediction cannot be easily made because they have been evaluated on different datasets and have differing β-hairpin definitions. For example, de La Cruz et al. (41) described the only method capable of predicting variable-length β-hairpins, while others require β-hairpins to be of a fixed length. De la Cruz et al. (41) were able to achieve this by transforming each β-hairpin into a fixed-size feature vector prior to being processed by an ANN. In contrast, other methods extract the feature vectors directly from the β-hairpin sequences.

Generally, each of the methods described uses the dictionary of protein secondary structure (DSSP) (50) or PROMOTIF algorithm (51) to determine the strand-loop-strand segments of each protein sequence. Alternatively, a secondary structure prediction algorithm such as PHD or PSIPRED can be used. However, these alternatives may misclassify strands even before the application of a machine learning method. Work done by de La Cruz et al. (41) predicted 542 β-hairpins according to PHD but 1,031 β-hairpins according to DSSP. This shows that nearly half of the β-hairpins were missed by using PHD instead of DSSP. However, these figures were generated in 2002 and do not reflect the state of the art in secondary structure prediction algorithms (52).

Table 1 compares the results obtained using the original method of de La Cruz with the with the βhairPred algorithm of Kumar et al. (45) when run on the same dataset. In this

Table 1
A comparison of the results obtained using the dataset from de La Cruz et al. (41)

Method	No. β-hairpins	Variant	Sens. (%)	Spec. (%)	Prec. (%)	Acc. (%)
De La Cruz et al. (41)	1,031	*Ideal*	55.9	73.6	64.2	–
		Test	47.7	77.4	30.1	–
Kumar et al. (45)	1,076	ANN	64.9	74.3	75.7	69.1
		SVM	78.8	70.6	76.6	75.1

The *ideal* and *test* variants are those that used the observed and predicted secondary structures, respectively. The performance metrics were reproduced from their respective publications

Table 2
A comparison of results between methods using the dataset from Kumer et al. (45)

Method	No. β-hairpins	Sens. (%)	Spec. (%)	Prec. (%)	Acc. (%)
Kumar et al. (45)	5102	82.6	75.7	77.2	79.2
Hu and Li (46)	4817	90.6	77.7	84.1	85.0
Zou et al. (48)		92.4	78.8	87.6	86.7
Hu et al. (47)	4884	83.4	77.4	81.8	80.7

Values reproduced from the papers cited

comparison, the β-hairpin counts differed between each method. De La Cruz et al. (41) counted 1,031 β-hairpins, while Kumar et al. (45) found 1,076. The difference between these two counts does not allow a strict comparison between the two methods. Nevertheless, it would be reasonable to infer from these results that βhairPred is a significant improvement over the method of de La Cruz et al. in predictive performance. βhairPred exhibits superior sensitivity and precision values, especially in light of the relatively minor difference in observed β-hairpins.

The dataset described by Kumar et al. (45) has been used by later studies. Table 2 compares the results obtained by each method on this dataset. It is again noted that the number of observed β-hairpins differs considerably between each method. However, the variance in performance between the methods is small, and we cannot make a confident conclusion about the superiority of any given method.

The CASP6 dataset (53) is the most frequently used dataset in studies of β-hairpin prediction, and can therefore provide some degree of objectivity when comparing performance between methods.

Table 3
Prediction results of different methods using the CASP6 dataset (53)

Method	Accuracy (%)	Exact matches	Inexact matches	Non-β-hairpins
Hu and Li (46)	71.8	21/26 = 80.8%	35/46 = 76.1%	61/91 = 67.0%
Zou et al. (48)	74.2	22/26 = 84.6%	37/46 = 80.4%	62/91 = 68.1%
Kumar et al. (45)	73.3	22/27 = 81.5%	25/51 = 49.0%	85/102 = 83.3%
Hu et al. (47)	75.6	23/27 = 85.2%	32/51 = 62.7%	81/102 = 79.4%

The exact matches represent the number of β-hairpins identified by PROMOTIF that exactly matched those determined by secondary structure prediction. The inexact matches represent the number β-hairpins identified by PROMOTIF that overlap those found by secondary structure prediction. For matches, we indicate (number correct)/(total number of given type) and the corresponding percentage

The main limitation of this dataset is its relatively small size when compared to the previously described datasets. Comparisons between methods using the CASP6 dataset are shown in Table 3.

3.2. β-Sheets

Survey. Existing methods for predicting β-sheet topology are usually based on well-known techniques from machine learning, dynamic programming, or integer linear programming (ILP). These approaches may use statistically derived information from the Protein Data Bank (PDB) (www.pdb.org) or a priori rules and constraints.

Early approaches to β-strand prediction were only capable of discriminating native β-strand alignments from nonnative alternatives and were based entirely on statistically derived interstrand residue pair propensities (54–56). Predictions were made by selecting the best alignments ranked by pseudoenergy, calculated as the normalized sum of constituent residue pair propensities. These studies demonstrated that nonrandom native alignment selectivity can be obtained from pairwise propensities. Consequently, later methods have used both single and pairwise propensities in their algorithms (14, 57, 58).

Modeling β-strand formation using pairwise statistics alone is simplistic and ignores important structural and physicochemical influences, external to the residue pair (14, 15). Machine learning methods address this problem by encoding information about the surrounding environment into the feature vector of each instance.

These approaches treat β-sheet topology prediction as a binary classification task and can be used to predict β-residue or strand pairing. Baldi et al. (36) describe an ANN that can predict β-residue pairs using the surrounding amino acid windows as features. Brown et al. (37) used a SVM to predict β-strand pairings and their orderings in β-sheets given that the set of β-strands for each β-sheet is known. In this approach, the feature vector of each β-sheet is generated using a sliding window that traverses the amino acid sequences.

Binary classifiers, such as SVMs, can only predict one pairing at a time and additional steps are required to predict strand orderings. This is commonly done using heuristics that build the orderings from the best scoring predicted strand pairs. Brown et al. (59) select the best scoring ordering from all possible combinations. BetaPro (described later in this section) uses a greedy algorithm that builds an ordering one strand pair at a time. The latter approach is computationally more efficient since it does not rely on exhaustive enumeration but may result in a locally optimal solution.

Globally optimal predictions have been obtained by formulating β-sheet topology prediction as an ILP problem (57, 60). These approaches solve a pseudoenergy objective function according to a set of constraints that do not allow physically impossible or unfavorable topological configurations. These constraints can greatly reduce the search space of possible solutions (5). Klepeis and Floudas (60) describe several ILP formulations that maximize the hydrophobic interaction pseudoenergies between individual β-residue and strand pairs. The authors attempted to simulate the hydrophobic collapse model of β-sheet nucleation and are capable of predicting a complete β-sheet topology. However, the accuracy of these formulations was only examined in six case studies. Systematic performance evaluations on large datasets were not made.

The method that has received the most attention in recent times is BetaPro (14). BetaPro adopts the unique approach of using different algorithms for predicting each aspect of β-sheet topology. BetaPro consists of three distinct stages:

1. Calculate the matrix of β-residue pairing probabilities, O, using a recurrent neural network (61) where the feature sets for describing the β-residue pairings contain the sequence, relative solvent accessibilities, and secondary structures of the residues immediately surrounding the pair. The sequence separation of the pair is also used and is considered to be one of the most important features (36).
2. Calculate the matrix of β-strand pairing pseudoenergies, W, using the Needleman–Wunsch (62) global sequence alignment algorithm where alignments are scored using the pairing probabilities in O. This stage finds the optimal alignment between all possible β-strand pairs using a well-known dynamic programming algorithm conventionally used to align DNA and protein sequences (62).
3. Generate the β-strand pairing graph, G, using a greedy algorithm that builds the graph one pair at a time using the best scoring pairs in W according to a set of structural constraints. Each vertex in G represents a β-strand and an edge represents a pairing between two β-strands.

The modularity of this approach has allowed several improvements to be made.

Table 4
The reported results for β-strand pairing prediction between BetaPro (14) and the work in Jeong et al. (57)

Method	Variant	Sens. (%)	Spec. (%)	Prec. (%)
BetaPro (14)	–	61.6	59.9	60.0
Jeong et al. (57)	*ILP*	62.3	61.0	61.0
	Greedy	62.3	62.6	62.6

These values were reproduced from Table 6.1 of Jeong et al. (57)

Jeong et al. (57) describe two solutions to stage 3. The first is an ILP formulation similar to that of Klepeis and Floudas (60) that maximizes the β-strand pairing pseudoenergies. The second is a greedy algorithm similar to that of BetaPro. This approach increases the weighting of the β-strand pairings surrounding an already matched pair, modeling the folding pathway theory of protein folding (2).

BetaZa (58) represents the most recent improvement to BetaPro. In this approach, stage 2 is replaced with a version of the Needleman-Wunsch algorithm (62) that permits an unlimited number of gaps, allowing β-bulges to be modeled. Some β-sheet topologies are physicochemically unstable and therefore rarely observed (63). Conversely, certain topologies are more frequent than others, allowing their probability distributions to be calculated (63). BetaZa replaces stage 3 with an algorithm that enumerates and scores all topologies of a β-sheet. The scoring function is a Bayesian model that considers the residue pairing and topology probability distributions. The algorithmic search space is significantly reduced by heuristics that ignore highly unlikely topologies or topologies with low residue pairing probabilities according to O.

Comparisons. The BetaPro family of predictors is considered the de facto state of the art. However, an objective comparison on the same dataset between all the described methods has yet to be made. It is important to consider that earlier methods were trained and evaluated on versions of the PDB that were significantly smaller than the current version. An evaluation of the previously described methods on the latest PDB snapshots would be required for an accurate comparison.

The only objective comparisons have been within the BetaPro family, where each method is evaluated using tenfold cross-validation on the BetaSheet916 dataset (14) using identical fold compositions. Jeong et al. (57) reported results only for β-strand pairing prediction, as shown in Table 4, where the performances of their ILP and greedy algorithms are compared with the conventional BetaPro implementation. The methods of Jeong et al. (57) provide small increases in specificity and precision, with the improved greedy algorithm providing the greatest increase of 3%.

Table 5
The reported prediction results compare BetaPro and BetaZa

Problem	Measure	BetaPro (%)	BetaZa (%)
β-Strand pairing	Sens.	68.9	69.1
	Prec.	61.9	61.9
β-Residue pairing	Sens.	63.4	66.5
	Prec.	54.4	57.2
Pair directionality	Sens.	66.1	66.2
	Prec.	59.4	59.4

The sensitivity and precision values for β-strand pairing differ from those in Table 4 because a slightly modified version of the BetaSheet916 dataset was used by Aydin et al. (10)

Aydin et al. (58) reported results for β-strand pairing, β-residue pairing, and pair directionality; where pair directionality prediction is the task of determining whether or not a β-strand pair is parallel or antiparallel. The dataset in this comparison was modified by removing all proteins with bifurcated β-sheets. Only the sensitivity and precision values were reported. These results are shown in Table 5 and demonstrate that BetaZa is comparable to BetaPro for β-strand pairing and directionality prediction. However, BetaZa provides a 3% increase in β-residue pairing prediction sensitivity and precision.

Unfortunately, β-sheet prediction methods do not report any ordering prediction accuracies, since there exists no standard metric for this task. The method described by Brown et al. (37) achieved a strand ordering accuracy of 49.3%, albeit on a completely different dataset (not BetaSheet916). They consider the order prediction to be a binary classification task where a correct ordering is viewed as a true prediction and false otherwise, even if only one strand is out of place.

3.3. General Supersecondary Structures

Taylor and Thornton published some of the original SSS motif prediction work in 1983 (64). In this work, the SSS motif considered consisted of two parallel β-strands connected by an α-helix. Taylor and Thornton analyzed 62 α-helices for 18 known structures and designed an optimal sequence template consisting of a strand–loop–helix–loop–strand SSS motif with 5–5–12–5–5 residues, respectively. Using this template, 70% of the $\beta\alpha\beta$ motifs were found in a dataset of 16 β/α type proteins (65).

In 1997, Sun et al. (39) published work on SSS motif prediction using Ramachandran plots. In this work, amino acids are classified into seven groups based on their spatial conformations. These class labels are used to mark the connecting residues between regular SSEs. Using this scheme, SSS motifs with the same loop properties

can be grouped together based on the Ramachandran plot class labels. For example, a β-loop-α motif might be labeled β-lba-α, where "lba" indicates the class labels of three amino acids in the loop between the strand and helix (40). Sun et al. (39) obtained eleven different SSS motifs using their scheme and trained a corresponding number of ANNs to predict the motifs. Inputs to the ANNs were obtained by representing amino acid sequences numerically using PSSMs (38), and outputs were conserved amino acid patterns for a particular loop class (e.g., "lba"). The resulting ANNs produced 70% prediction accuracy on a 240 protein dataset.

A year later, Boutonnet et al. (66) proposed an approach for prediction of $\alpha\beta\beta$ and $\beta\beta\alpha$ motifs, where the direction of the hydrogen bonded strands are differentiated. In this study, a graphical visualization method was developed and applied to 141 structures. Although not a classification from the point of view of machine learning, Boutonnet et al. provide an in-depth analysis of SSS motif prediction tasks. One conclusion of their analysis is that physicochemical properties of amino acids result in conserved sequences in SSS motifs. From a machine learning perspective, this implies and that physicochemical properties should be incorporated into algorithms designed to predict SSS motifs.

More recently, Zou et al. (31) have used a diversity measure based discrimination method to predict four types of SSS motifs ($\alpha\alpha$, $\alpha\beta$, $\beta\alpha$, and $\beta\beta$). This method includes a variety of protein sequence descriptions, including the ID measure mentioned previously in the context of β-hairpin prediction (46), AAC (29), PseACC (30), and dipeptide components. Using a combination of these features, Zou et al. trained a novel machine learning classifier based on a SVM and a quadratic discriminant using the ArchDB40 dataset (67). They report a 69.4% success rate for correctly predicting SSS motifs.

In addition to machine learning approaches for predicting SSS motifs, Tran et al. (68) have formalized the SSS prediction task as a graph problem. In their work, dynamic programming is used to find the shortest closed path in a certain graph, allowing prediction of β-barrels and α-helical bundles in transmembrane proteins.

Comparisons. The SSS motif definitions used by the surveyed machine learning methods are all slightly different, and the methods themselves are different as well. This disparity makes a comparison between the methods infeasible.

4. Notes

In this section we include observations on the behavior of, potential difficulties with, and future work on the different machine learning methods discussed. The notes are divided according to

β-hairpin prediction, β-sheet prediction, and general supersecondary structure prediction.

1. β-Hairpins
   The β-hairpin prediction methods surveyed were designed to predict β-hairpins formed by β-strands that occur consecutively in the sequence. However, β-hairpins can also be formed between nonconsecutive β-strands (59), for example, in a β-barrel. The prediction of multiple simultaneous β-hairpins remains a challenging problem since it requires the consideration of global as well as local interactions between β-strands and residues.

   The highest prediction accuracies reported by the surveyed β-hairpin prediction methods do not exceed 80%. This suggests that the β-hairpin prediction accuracy may be limited (perhaps to 80%). This may be due to the fact that the correct assignment of SSEs is the first step in many methods. This difficulty is discussed by de La Cruz et al. (41).

2. β-Sheets
   A major bottleneck of β-sheet topology prediction methods is their reliance on secondary structure prediction algorithms. State of the art secondary structure prediction algorithms have accuracies of up to 80% (52). The results presented by each β-sheet topology prediction method surveyed were obtained using known secondary structure assignments made by DSSP. Predicting β-sheet topology from predicted secondary structures serves as a greater challenge, considering that any incorrect input will likely lead to an incorrect prediction.

   The methods surveyed also largely ignore the interactions external to the β-sheet that can play a major role in its formation (23). While the sliding window feature sets of the ANN based approaches are capable of capturing some of the surrounding environment of a β-residue pair (36, 37), extending the window beyond the β-strand could potentially lead to over fitting and long training times (36).

   While β-sheet topology prediction cannot replace experimental structure determination methods, they do offer the researcher a means to inspect the likely conformations of a β-sheet. This information can be used to guide the distance geometry calculations required by experimental structure determination methods (24) and to constrain the search space of ab initio tertiary structure prediction methods (7). Additionally, the importance of each β-sheet feature used in prediction can provide an insight into some of the mechanisms of β-sheet nucleation (11, 15).

3. General supersecondary structures
   The central difficulty for prediction of supersecondary structure using machine learning is the variety of definitions of SSS motifs. Definitions range from very specific, where similar SSS motifs can be distinguished by different loop characteristics (e.g., different cases of $\beta\alpha$ using Ramachandran plots (39)), to very general, where motifs are divided into only four classes (e.g., $\alpha\alpha$, $\alpha\beta$, $\beta\alpha$, and $\beta\beta$ (31)). This lack of consensus means that each machine learning method is solving a different problem. As a result each method uses different feature vectors, different algorithms, different datasets, and has different goals. One possible solution to this problem is a definition based on relative orientation of SSEs rather than loop attributes. Definitions based on relative orientation between SSEs have been considered using tableau (69), and an algorithm for protein structure motif search (ProSMos) (70).

Acknowledgments

Hui Kian Ho and Lei Zhang are supported by the Australian National Information and Communications Technology Research Centre (NICTA).

References

1. Branden C, Tooze J (1999) Introduction to protein structure. 2nd edn. Garland Publishing, New York
2. Dill KA, Ozkan SB, Shell MS et al (2008) The protein folding problem. Annu Rev Biophys 37:289–316
3. Moult J, Fidelis K, Kryshtafovych A et al (2009) Critical assessment of methods of protein structure prediction—round VIII. Proteins: Struct Function Bioinform 77:1–4
4. Cui Y, Chen RS, Wong WH (1998) Protein folding simulation with genetic algorithm and supersecondary structure constraints. Proteins 31:247–257
5. Fonseca R, Helles G, Winter P (2010) Ranking beta sheet topologies of proteins. In: Proceedings of the world congress on engineering and computer science, San Francisco, CA, pp 624–628
6. Klepeis JL, Floudas CA (2003) ASTRO-FOLD: a combinatorial and global optimization framework for ab initio prediction of three-dimensional structures of proteins from the amino acid sequence. Biophys J 85:2119–2146
7. Porwal G, Jain S, Babu SD et al (2007) Protein structure prediction aided by geometrical and probabilistic constraints. J Comput Chem 28:1943–1952
8. Rajgaria R, Wei Y, Floudas CA (2010) Contact prediction for beta and alpha-beta proteins using integer linear optimization and its impact on the first principles 3D structure prediction method ASTRO-FOLD. Proteins: Struct Function Bioinform 78:1825–1846
9. Blanco FJ, Rivas G, Serrano L (1994) A short linear peptide that folds into a native stable beta-hairpin in aqueous solution. Nat Struct Biol 1:584–590
10. Kuhn M, Meiler J, Baker D (2004) Strand-loop-strand motifs: prediction of hairpins and diverging turns in proteins. Proteins 54:282–288
11. Parisien M, Major F (2007) Ranking the factors that contribute to protein beta-sheet folding. Proteins 68:824–829
12. Marshall KE, Serpell LC (2009) Structural integrity of beta-sheet assembly. Biochem Soc Trans 37:671–676
13. Kajava AV, Baxa U, Steven AC (2010) Beta arcades: recurring motifs in naturally occurring

and disease-related amyloid fibrils. FASEB J 24:1311–1319
14. Cheng J, Baldi P (2005) Three-stage prediction of protein beta-sheets by neural networks, alignments and graph algorithms. Bioinformatics 21(Suppl 1):i75–84
15. Wathen B, Jia Z (2009) Folding by numbers: primary sequence statistics and their use in studying protein folding. Int J Mol Sci 10:1567–1589
16. Murzin AG, Brenner SE, Hubbard T et al (1995) SCOP: a structural classification of proteins database for the investigation of sequences and structures. J Mol Biol 247:536–540
17. Jain AK, Mao J, Mohiuddin KM (1996) Artificial neural networks: a tutorial. Computer 29:31–44
18. Haykin S (1998) Neural networks: a comprehensive foundation. Prentice Hall, New Jersey
19. McCulloch WS, Pitts W (1943) A logical calculus of the ideas immanent in nervous activity. Bull Math Biophys 5:115–133
20. Rosenblatt F (1962) Principles of neurodynamics. Spartan Books, New York
21. Shawe-Taylor J, Cristianini N (2000) Support vector machines and other Kernel-based learning methods. Cambridge University Press, Cambridge
22. Vapnik V (1998) Statistical learning theory. Wiley, New York
23. Takano K, Katagiri Y, Mukaiyama A et al (2007) Conformational contagion in a protein: structural properties of a chameleon sequence. Proteins 68:617–625
24. Li W, Zhang Y, Kihara D et al (2003) TOUCHSTONEX: protein structure prediction with sparse NMR data. Proteins: Struct Function Bioinform 53:290–306
25. Bennett K, Campbell C (2000) Support vector machines: hype or hallelujah? SIGKDD Explorations 2:1–13
26. Burges C (1998) A tutorial on support vector machines for pattern recognition. Data Min Knowl Discov 2:121–167
27. Chang C-C, Lin C-J (2001) LibSVM: a library for support vector machines. http://www.csie.ntu.edu.tw/~cjlin/libsvm.
28. Joachims T (1999) Making large-scale SVM learning practical. In: Scholkopf B, Burges C, Smola A (eds) Advances in Kernel methods—support vector learning, MIT Press. pg 169–184, Cambridge, MA
29. Nakashima H, Nishikawa K, Ooi T (1986) The folding type of a protein is relevant to the amino acid composition. J Biochem 99:153–162
30. Chou K-C (2001) Prediction of protein cellular attributes using pseudo-amino acid composition. Proteins: Struct Function Bioinform 43:246–255
31. Zou D, He Z, He J et al (2011) Supersecondary structure prediction using Chou's pseudo amino acid composition. J Comput Chem 32:271–278
32. Xiao X, Shao SH, Huang ZD et al (2006) Using pseudo amino acid composition to predict protein structural classes: approached with complexity measure factor. J Comput Chem 27:478–482
33. Li QZ, Lu ZQ (2001) The prediction of the structural class of protein: application of the measure of diversity. J Theor Biol 213:493–502
34. Lin H, Li QZ (2007) Using pseudo amino acid composition to predict protein structural class: approached by incorporating 400 dipeptide components. J Comput Chem 28:1463–1466
35. Ji X, Bailey J, Ramamohanarao K (2008) g-MARS: protein classification using gapped Markov chains and support vector machines. In: Chetty M, Ngom A, Ahmad S (eds) Pattern recognition in bioinformatics. Springer, Berlin/Heidelberg, pp 165–177
36. Baldi P, Pollastri G, Andersen CA et al (2000) Matching protein beta-sheet partners by feedforward and recurrent neural networks. Proc Int Conf Intell Syst Mol Biol 8:25–36
37. Brown WM, Martin S, Chabarek JP et al (2006) Prediction of beta-strand packing interactions using the signature product. J Mol Model 12:355–361
38. Jones DT (1999) Protein secondary structure prediction based on position-specific scoring matrices. J Mol Biol 292:195–202
39. Sun Z, Rao X, Peng L et al (1997) Prediction of protein supersecondary structures based on the artificial neural network method. Protein Eng 10:763–769
40. Zhirong S, Blundell T (1995) The pattern of common supersecondary structure (motifs) in protein database. In: System sciences, 1995. Vol. V. Proceedings of the twenty-eighth Hawaii international conference on, vol 315, pp 312–318.
41. de la Cruz X, Hutchinson EG, Shepherd A et al (2002) Toward predicting protein topology: an approach to identifying β hairpins. Proc Natl Acad Sci 99:11157–11162
42. Rost B, Sander C, Schneider R (1994) PHD-an automatic mail server for protein secondary structure prediction. Comput Appl Biosci 10:53–60
43. Altschul SF, Madden TL, Schäffer AA et al (1997) Gapped BLAST and PSI-BLAST: a new generation of protein database search programs. Nucleic Acids Res 25:3389–3402

44. McGuffin LJ, Bryson K, Jones DT (2000) The PSIPRED protein structure prediction server. Bioinformatics 16:404–405
45. Kumar M, Bhasin M, Natt NK et al (2005) BhairPred: prediction of β-hairpins in a protein from multiple alignment information using ANN and SVM techniques. Nucleic Acids Res 33:W154–W159
46. Hu X, Li Q (2008) Prediction of the β-hairpins in proteins using support vector machine. Protein J 27:115–122
47. Hu X-Z, Li Q-Z, Wang C-L (2010) Recognition of β-hairpin motifs in proteins by using the composite vector. Amino Acids 38:915–921
48. Zou D, He Z, He J (2009) β-Hairpin prediction with quadratic discriminant analysis using diversity measure. J Comput Chem 30:2277–2284
49. Xia JF, Wu M, You ZH et al (2010) Prediction of beta-hairpins in proteins using physicochemical properties and structure information. Protein Pept Lett 17:1123–1128
50. Kabsch W, Sander C (1983) Dictionary of protein secondary structure: pattern recognition of hydrogen-bonded and geometrical features. Biopolymers 22:2577–2637
51. Hutchinson EG, Thornton JM (1996) PROMOTIF—a program to identify and analyze structural motifs in proteins. Protein Sci 5:212–220
52. Pirovano W, Heringa J (2010) Protein secondary structure prediction. In: Carugo O, Eisenhaber F (eds) Data mining techniques for the life sciences. Humana Press, Totowa, NJ, pp 327–348
53. Lattman EE (2005) Sixth meeting on the critical assessment of techniques for protein structure prediction. Proteins: Struct Function Bioinform 61:1–236
54. Hubbard TJP (1994) Use of B-strand interaction pseudo-potentials in protein structure prediction and modelling. In: Hunter L (ed) System sciences, 1994. Proceedings of the twenty-seventh Hawaii international conference on systems science. IEEE Society Press, Maui, Hawaii, pages 169–184
55. Steward RE, Thornton JM (2002) Prediction of strand pairing in antiparallel and parallel beta-sheets using information theory. Proteins 48:178–191
56. Zhu H, Braun W (1999) Sequence specificity, statistical potentials, and three-dimensional structure prediction with self-correcting distance geometry calculations of beta-sheet formation in proteins. Protein Sci 8:326–342
57. Jeong J, Berman P, Przytycka T (2007) Bringing folding pathways into strand pairing prediction. In: Giancarlo R, Hannenhalli S (eds) Algorithms in bioinformatics. Springer, Berlin/Heidelberg, pp 38–48
58. Aydin Z, Altunbasak Y, Erdogan H (2011) Bayesian models and algorithms for protein B-sheet prediction. Comput Biol Bioinform IEEE/ACM Trans 8:395–409
59. Hutchinson EG, Thornton JM (1993) The Greek key motif: extraction, classification and analysis. Protein Eng 6:233–245
60. Klepeis JL, Floudas CA (2003) Prediction of beta-sheet topology and disulfide bridges in polypeptides. J Comput Chem 24:191–208
61. Baldi P, Pollastri G (2003) The principled design of large-scale recursive neural network architectures—dag-rnns and the protein structure prediction problem. J Mach Learn Res 4:575–602
62. Needleman SB, Wunsch CD (1970) A general method applicable to the search for similarities in the amino acid sequence of two proteins. J Mol Biol 48:443–453
63. Ruczinski I, Kooperberg C, Bonneau R et al (2002) Distributions of beta sheets in proteins with application to structure prediction. Proteins 48:85–97
64. Taylor WR, Thornton JM (1983) Prediction of super-secondary structure in proteins. Nature 301:540–542
65. Taylor WR, Thornton JM (1984) Recognition of super-secondary structure in proteins. J Mol Biol 173:487–512
66. Boutonnet NS, Kajava AV, Rooman MJ (1998) Structural classification of $\alpha\beta\beta$ and $\beta\beta\alpha$ supersecondary structure units in proteins. Proteins: Struct Function Bioinform 30:193–212
67. Espadaler J, Fernandez-Fuentes N, Hermoso A et al (2004) ArchDB: automated protein loop classification as a tool for structural genomics. Nucleic Acids Res 32:D185–188
68. Tran VD, Chassignet P, Steyaert J-M (2009) Prediction of super-secondary structure in alpha-helical and beta-barrel transmembrane proteins. BMC Bioinform 10:O3
69. Lesk AM (1995) Systematic representation of protein folding patterns. J Mol Graph 13:159–164
70. Shi S, Zhong Y, Majumdar I et al (2007) Searching for three-dimensional secondary structural patterns in proteins with ProSMoS. Bioinformatics 23:1331–1338

Chapter 7

Beyond Supersecondary Structure: The Global Properties of Protein Sequences

S. Rackovsky

Abstract

Analysis of the global properties of protein sequences, rather than single-site or local properties, has been shown to lead to new understanding of folding and function. Here we describe the use of software which can describe sequences numerically in an orthonormal fashion, Fourier-analyze those sequences, and verify the statistical significance of the resulting Fourier coefficients. The resulting parameters can be used to study problems involving sequences from a unique perspective.

Key words: Sequence analysis, Global representation, Fourier analysis

1. Introduction

The rapidly growing number of protein structures available, and the even more rapidly growing number of protein sequences, have precipitated a revolution in biology. Investigators in every area of biomedical science routinely utilize sequences and structures to understand biological processes, and sequence and structure homology searches have become everyday tools. At the same time, scientists interested in the basic processes of protein folding are able to mine sequence and structure databases in order to gain insight into the mechanism of architecture choice.

The evaluation of the degree of structural homology between proteins from their sequences is one of the significant outstanding problems in biomedical research. The fact that this problem is still open is apparent from the persistence of interest in the "remote homolog" problem—the observation that, in any reasonably large group of sequences which fold to a specified, common architecture there will be pairs of sequences which are not related by any currently known criterion.

Alexander E. Kister (ed.), *Protein Supersecondary Structures*, Methods in Molecular Biology, vol. 932,
DOI 10.1007/978-1-62703-065-6_7,

The basic computation which underlies this field is the determination of a distance between two sequences of interest. This requires a numerical calculation designed to compare sequence attributes. This calculation, in turn, requires a physically meaningful quantitative representation of the sequences, which must reflect two distinct attributes of proteins:

1. The physical properties of the amino acids whose arrangement constitutes the sequence.
2. A length scale, which must be chosen by the investigator to describe appropriately the linear, extended nature of the sequence.

These attributes must be included properly in order to give useful comparisons between molecules.

The calculation methods in common use are based on arbitrarily chosen sequence representations, and on a strictly local choice of length scale. Amino acid substitution matrices are constructed based on representations of the amino acids by their names, using empirical alignment studies, or sequences are represented numerically by arbitrarily chosen sets of physical properties. The length scale of the problem is set by the use of alignment, which compares sequences on the basis of single-residue correspondences.

The disadvantages of these choices are clear. Establishment of substitution matrices by alignment biases alignment results in a circular fashion. The use of arbitrarily chosen amino acid properties results in representations which are, in general, both incomplete and correlated. A single-residue length scale results in the loss of information about the structure of the sequence as a whole.

Because of these problems, we have developed [1, 2] representations of protein sequences which are based on a complete orthonormal representation of amino acid physical properties, and which describe the structure of protein sequences in terms of parameters which are derived from the global (i.e., end-to-end) properties of the sequences. Before presenting a detailed prescription for the method, I briefly outline the approach.

1. The amino acid property representation is taken from the work of Kidera et al. [3, 4], who carried out a factor analysis of all known property sets attributed to the 20 amino acids, and demonstrated that the data could be represented by a set of 10 property factors, which, together, account for 86 % of the variance of the entire dataset. Therefore, to a very good approximation, an amino acid X can be represented numerically as a ten-vector of property factors f:

$$X = (f_X^{(1)}, f_X^{(2)}, \ldots, f_X^{(10)}) \tag{1}$$

2. It follows that an N-residue sequence can be written as a set of ten numerical strings of length N, each of which describes the variation of one of the property factors along the length of the protein.

The property factors are linearly independent by construction, and therefore the ten strings together give a complete, uncorrelated description of the physical properties of the sequence.

3. Once the sequences are represented numerically in this way, it is possible to Fourier transform each of the ten strings, which gives a set of sine and cosine Fourier coefficients:

$$a_k^{(l)} = N^{-1} \sum_{m=1}^{N} f_m^{(l)} \sin(2\pi km / N) \tag{2}$$

$$b_k^{(l)} = N^{-1} \sum_{m=1}^{N} f_m^{(l)} \cos(2\pi km / N) \tag{3}$$

where k is a wave number, l indicates the property factor, and m is a sequence position. Note that each Fourier coefficient contains information from the *entire* sequence.

4. It is not sufficient to calculate only the Fourier coefficients for the sequence of interest. In order to establish the statistical significance of any conclusions which are drawn from these studies, it is necessary to compare the observed Fourier coefficients to those which would be expected from randomly generated sequences of the same amino acid composition as the sequence under study[2]. We have shown[2] that the average and standard deviation of the Fourier coefficients and power spectra over the ensemble of permuted sequences can be calculated analytically and exactly. In the case of the Fourier coefficients, it can be shown that the ensemble average equals zero so that only the standard deviation need be calculated.

5. Once the Fourier coefficients and the associated ensemble quantities have been calculated, it is possible to study the global properties of protein sequences in a number of ways. In earlier work we have shown that this approach can be used to classify protein sequences [5], to understand the implications of global properties for folding [6], and to understand conformation switching mechanisms which are not amenable to analysis using local methods [7]. The interested reader will find novel uses for this alternative approach to sequence description.

2. Materials

The "materials" required for this Method consist of software and databases. The software for carrying out the Fourier analysis is available from the author upon request. The sequence databases will be constructed by the investigator, from data of interest in his/her ongoing investigation. The following are required:

1. *A FORTRAN compiler*: All software for this computation is written in FORTRAN. Compilers are readily available for every known computing platform, many of them at no cost. The software provided by the author is available as source code, rather than compiled executables, in order to make it possible to compile and run the software on any system.
2. *A Sequence Database*: The end user must provide a database of the sequences of interest, formatted to be readable by the software (see Note 1).
3. *Fourier Software*: This consists of a set of programs available from the author:
   (a) *sfac.dat*- This is a file which contains the Kidera et al. property factors [3, 4] for the 20 amino acids, together with the numerical identification code for the amino acids.
   (b) *k3fourier_3.f*- This is a FORTRAN program which calculates both the Fourier coefficients arising from the sequence of interest, and the permutation-ensemble average of the Fourier coefficients and the associated standard deviation.
   (c) *k3fourier_3.par*- This is a parameter file, which contains the name of the file containing the sequence database referred to previously. The name is read as an alphanumeric string. [Format A80].

3. Methods

1. Place all files in the same directory, in order to avoid the use of long pathnames.
2. Compile the program file (k3fourier_3.f) (see Note 2).
3. Place the database file name in the parameter file (k3fourier_3.par).
4. Run the program (see Notes 3–6).
5. Use the output to calculate properties of interest.

4. Notes

1. The database should be in the following format:
   (a) Two lines of text which describe the database. These can be left blank if the user prefers, but the lines must be present at the beginning of the database [Format A80].
   (b) For each protein in the database, the following information must be present:

- One line containing the PDB name of the sequence [Format A12].
- One line containing the number of residues [Format I7].
- One line containing the CATH code numbers for the protein [Format 4I7].
- The sequence, written in numerical form. The numerical sequence code is read from the file sfac.dat. The numerical representation of the sequence takes as many lines as necessary, depending on the sequence length [Format 36I2].

Here is a sample of the database format, for two sequences:

```
/fortran/CathDomainSeqs.S35.ATOM.
v3.1.0
/fortran/CathDomainList.S35.v3.1.0
>pdb|12asA00
327
3 30 930 10
120 8 1 9141514 816 518 916 7 516151410
4 41510 610 8 41814 113 8101615
18 6 2 61714 2121016 6 1 4 9 1181418 918
9 11013 2 114 5 41818 71610 1 9
19 915141710 614 7 2 516 1 6 4 6102017
711 9 1101513 2 4 21510161310 716
182018 21419 219 4151811 6 2 6 41514
5161710 9161718 4 1 819 1 6 8 9 117
4 1 11816 4 4 5 610 113 51013 214 8 7
518 71614 4101016152013 210 2 1 9
615 415 1 8 1 9 210 6 118 51018 6 8 6 6
91016 2 6 715 7 21815 113 220 2
21916171316 410 6 7 1 61012 6 2
810181912131810 4 2 1 5 410161611 6 815
18 2 1 21710 9 71410 11017 6 2 4 21510
410 419 714 1101015 6 411131417 8
6 6 6 8 614161510171110101014 1013 7 8
6141814 1 6181913 1 11815 4161813
161010
>pdb|1531000
185
1 10 530 10
1517 2 320 612181215 8 21717 6 116 3 917
1 913 4 6101620 3 61816 116 9 9
```

```
8 1 415 21014 111 21520 917 8 8 9 918 6
4 910 318 413 118 8 1 6 8 81615
416 7 1 6 91810 912 619 6 215 612 6 5
610111418 2 91516 7 91314 6171912
6 418 7 81714 61717 810 812 5 8 917 814
9 9 513161917 9 2141410 9 6 6 8
16 12012 1 6 1 61218151620 11511 2 8
61717 7 2 220 112 21818 115 1142020
914 7 620
```

2. The code in the program file is quite straightforward, and should compile readily on any FORTRAN compiler. To the best of the author's knowledge, even FORTRAN 77 compilers should be adequate.
3. The program is reasonably fast. Calculations for a 7056-sequence database were run on a 4-processor Macintosh G5 computer, after compilation with the open-source gfortran compiler. The run was completed in less than 1 h.
4. The output of this program will appear in the form of two files. They are named k3fourier_3.dat and k3fouriersd_3.dat. The first file contains the values of the sine and cosine Fourier coefficients for each sequence in the database, for each value of the wave number k and the property factor l. The ranges of indices for which the Fourier coefficients are calculated are $1 \leq k \leq 60$ and $1 \leq l \leq 10$.
5. The format of the file k3fourier_3.dat is as follows:
   (a) One line with the name of the sequence database file [Format A80].
   (b) Two lines describing the database, rewritten from the input database file [Format A80].
   For each protein:
   (c) One line with the PDB designation for the protein [Format A12].
   (d) One line with the number of residues in the sequence [Format I7].
   (e) One line containing the CATH code numbers for the protein [Format 4I7].
   (f) A label for the sine coefficients [Format A80].
   (g) Sixty lines of sine Fourier coefficients for the ten property factors [Format I4,1x,10E12.4].
   (h) A label for the cosine coefficients [Format A80].
   (i) Sixty lines of cosine Fourier coefficients for the ten property factors [Format I4,1x,10E12.4].

Here is a sample of the output for this file, for one protein:

```
/fortran/kdbase.dat
/fortran/CathDomainSeqs.S35.ATOM.v3.1.0
/fortran/CathDomainList.S35.v3.1.0
>pdb|12asA00
327
3 30 930 10
SINE COEFFICIENTS:
1-0.2803E-01    0.3257E-01    0.2385E-01
0.4227E-01    0.3886E-01    -0.6137E-02
-0.9909E-02
2   0.3373E-01   -0.2681E-01   0.7338E-02
0.2507E-01    0.9410E-02    -0.4803E-01
-0.3721E-01
.
.
.
60   0.8521E-01   -0.4041E-01   0.5282E-01
0.3138E-01    -0.2321E-01    -0.7009E-02
-0.6327E-01
COSINE COEFFICIENTS:
1-0.3482E-01    -0.6637E-02    0.6473E-01
0.3383E-02 0.4088E-01 -0.8058E-02 0.5585E-
01
2-0.2784E-01    -0.1630E-01    0.5950E-01
-0.2092E-01    -0.3894E-02    -0.2582E-01
0.2176E-01
.
.
.
60-0.1290E-03   -0.2255E-02   -0.3854E-01
-0.7149E-01    -0.6677E-02    -0.9370E-01
-0.1016E-01
```

The format of the file k3fouriersd_3.dat is as follows:

(a) One line with the name of the sequence database file [Format A80].

(b) Two lines describing the database, rewritten from the input database file [Format A80].
For each protein:

(c) One line with the PDB designation for the protein [Format A12]

(d) One line with the number of residues in the sequence. [Format I7].

(e) One line containing the CATH code numbers for the protein [Format 4I7].

(f) One line of standard deviations [Format 1x, 10E12.4]. Note that the standard deviations depend only on l, are independent of k, and are the same for the sine and cosine Fourier coefficients.

6. A separate version of the program is available for cases where it is of interest to calculate power spectral values, rather than Fourier coefficients. The power spectral values are the squares of the Fourier coefficients, and their permutation-ensemble averages are not zero. Therefore, in that program, a separate file is created containing the values of the ensemble averages of the spectral elements.

References

1. Rackovsky S (1998) Hidden sequence periodicities and protein architecture. Proc Natl Acad Sci USA 95:8580–8584
2. Rackovsky S (2006) Characterization of architecture signals in proteins. J Phys Chem B 110:18771–18778
3. Kidera A, Konishi Y, Oka M et al (1985) Statistical analysis of the physical properties of the 20 naturally occurring amino acids. J Prot Chem 4:23–55
4. Kidera A, Konishi Y, Ooi T et al (1985) Relation between sequence similarity and structural similarity in proteins: role of important properties of amino acids. J Prot Chem 4:265–297
5. Rackovsky S (2009) Sequence physical properties encode the global organization of protein structure space. Proc Natl Acad Sci USA 106:14345–14348
6. Rackovsky S (2010) Global characteristics of protein sequences and their implications. Proc Natl Acad Sci USA 107:8623–8626
7. Rackovsky S (2011) Spectral analysis of a protein conformational switch. Phys Rev Lett 106:248101

Chapter 8

Creating Supersecondary Structures with BuildBeta

Silvia Crivelli and Nelson Max

Abstract

BuildBeta is a feature of the ProteinShop software designed to thoroughly sample a protein conformational space given the protein's sequence of amino acids and secondary structure predictions. It targets proteins with beta sheets because they are particularly challenging to predict due to the complexity of sampling long-range strand pairings. Here we discuss some of the most difficult targets in the recent 9th Community Wide Experiment on the Critical Assessment of Techniques for Protein Structure Prediction (CASP9) and show how BuildBeta can leverage some of the most successful methods in the category "template-free modeling" by augmenting their sampling capabilities. We also discuss ongoing efforts to improve the quality of the supersecondary structures it generates.

Key words: Protein structure prediction, Template-free modeling, Sample conformational space, Beta sheets

1. Introduction

In 2001 we began the development of ProteinShop, an interactive modeling tool designed to manually create protein structures using a combination of human knowledge and intuition with computational capabilities (1). ProteinShop permits the user to manipulate secondary or supersecondary structures (SSS) with the mouse using inverse kinematics (IK), allowing him/her to form H-bonds, to break existing H-bonds and create new ones, to change dihedral angles guided by a Ramachandran plot, to create new protein folds guided by a potential energy function, and much more. We have used ProteinShop to create initial configurations for our protein structure prediction method and have participated in the Critical Assessment of Protein Structure Prediction Methods (CASP) competition for many years (2–12).

Alexander E. Kister (ed.), *Protein Supersecondary Structures*, Methods in Molecular Biology, vol. 932,
DOI 10.1007/978-1-62703-065-6_8,

We have also developed an automatic feature for the creation of a variety of SSS called BuildBeta (13). BuildBeta is a function in ProteinShop that takes the human out of the loop and lets the computer automatically generate all the beta topologies that are possible for a given sequence with a specific secondary structure prediction. It was designed for the template-free modeling (TFM) prediction of proteins with mostly beta strands. Given a prediction file, i.e., a file that contains the sequence of amino acids and secondary structure predictions for each amino acid, BuildBeta can automatically create a variety of possible arrangements of beta strands into beta sheets. This seems an important feature in light of the post-CASP9 analysis, which is publicly available (14). According to this document, both the assessors in the TFM category and the predictors argue that methods tend to predict common local cores and are usually unable to generate beta sheets of long-range contact order. This problem is sometimes overcome when the methods find a distant template or when the target structure corresponds to a common topology (perhaps due to fragment use and/or knowledge-base potentials). However, the quality of the predictions decreases rapidly as the number of strands increases or when the topology is less common. Zhang, who led 3 very successful groups in the competition (Zhang_server, Quark, and Zhang_ab_initio), suggested that they should work on enumerating all beta-scaffolds to fold beta proteins (14).

Given this challenge, we posit that BuildBeta has great potential to become a tool that complements existing methods by helping them to create a variety of beta topologies that they can use to increase the sampling of a target's conformational space. In this chapter, we address this issue by discussing BuildBeta in the context of the CASP9 competition. We show the best models created by CASP9 participants for some of the most difficult targets and discuss what BuildBeta can do to help these groups improve their predictions. We also describe recent work we have done aimed at creating more realistically looking SSS.

This chapter is organized as follows. In Subheading 2 we provide an overview of the BuildBeta feature, analyze the CASP9 results corresponding to beta proteins in the TFM category, and discuss how BuildBeta can be used to improve some of those results. In Subheading 3 we present the methods developed to create more realistically looking beta sheets and barrels.

2. Materials

One approach to template-free or ab initio structure prediction, called hierarchical, starts with the sequence of amino acids, determines the secondary structure prediction, brings those elements of

secondary structure together into SSS, and then packs those SSS to form the final 3D shape. BuildBeta was originally designed to support this approach by generating SSS that contain beta strands. Using basic packing principles, inverse kinematics, and beta pairing scores, BuildBeta samples a protein conformation space by creating all possible arrangements of beta strands into beta sheets. Sampling is extremely challenging for beta sheet proteins because they involve long-range interactions that lead to combinatorial complexity. Thus, BuildBeta may generate an enormous number of models (see Note 1), which may take hours or even days to complete on a single processor. However, the time to completion can be reduced by parallel implementation, which can be done straightforwardly by assigning different topologies to different processors (see Note 2).

BuildBeta's input may be either a prediction file or a coordinate file with PDB format. A prediction file contains three lines: one with the sequence of amino acids, another with predictions of whether each amino acid is part of an alpha helix, a beta strand, or a coil region, and the third with the confidence scores for those predictions. BuildBeta assigns ideal values to the backbone dihedral angles of residues predicted to be alpha helices and beta strands (see Note 3) and rotates the backbone dihedral angles of the flexible coil regions to construct beta sheets fully automatically using IK. The resulting structures should be subjected to alpha helix repacking, coil modeling, and overall refinement (see Note 4).

BuildBeta also permits one to prespecify rigid core regions of the sequence, each containing one or more beta strands, and then build automatically around those prespecified sheets. Currently, BuildBeta makes only single beta sheets unless a core is provided that has at least one strand on each sheet. In our paper (13), we discuss in detail the BuildBeta approach to generate single beta sheet topologies as well as multiple sheet topologies using cores. In this chapter, we present more recent features of BuildBeta as well as discuss our tests using CASP9 results.

2.1. CASP9 Targets

CASP has two main categories of 3D structure predictions: the "template-based modeling" (TBM) in which a template is identified that is used to model the target and the "TFM". However, there are cases in the latter category in which human experts have been able to identify a distant relative of the target, and so the TBM/TFM distinction has become blurred. In this chapter, we classify the targets according to their *GDT_TS* scores, where *GDT_TS* is defined as $GDT\_TS = (GDT\_P1 + GDT\_P2 + GDT\_P4 + GDT\_P8)/4$, and *GDT_Pn* denotes percent of residues under distance cutoff $\leq n$Å (15). The *GDT_TS* score has been used by the CASP assessors for many years and provides a good measure of the overall quality of the predictions. In general, high *GDT_TS* indicates good-quality predictions. There are 30 TFM targets out of 116

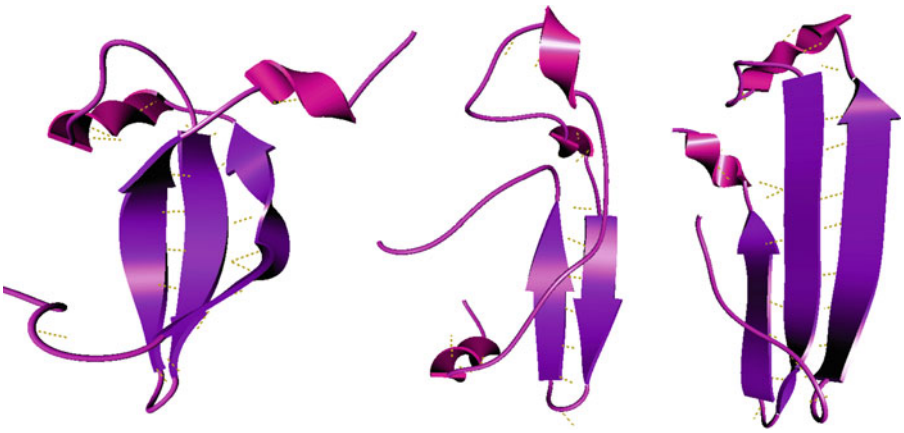


Fig. 1. (*Left*) native structure for T0531, (*center*) the best-ranked CASP9 model, (*right*) the second best-ranked model.

total CASP9 targets and 11 of those targets are alpha + beta or beta-proteins. Targets in this category are T0531, T0537, T0550_D2, T0571_D2, T0578, T0581, T0604_D1, T0604_D3, T0621, T0624, and T0629. Next, we briefly describe the targets whose best models have *GDT_TS* > 40.

T0531 (65 *residues*). The top model submitted for this target was generated by group Chicken_George (*GDT_TS* = 44.83) closely followed by MUFOLD_MD (*GDT_TS* = 43.53). According to the REMARK lines of their submitted models, these groups did not use a template. The Root Mean Square Deviations (RMSDs) are large considering the small size of the target (8 and 7.7 Å, respectively). Figure 1 shows (left) the native structure, (center) the best-ranked model submitted by the group Chicken_George, and (right) the second best model submitted by MUFOLD_MD.

T0537_D1 (381 *residues*). Several groups in the humans and servers categories were successful at finding a template for this target and achieved a good *GDT_TS* (58.06 for domain 2 and 38.46 for domain 1). The best-ranked model for domain 1 (286 residues) submitted by group prmls has *RMSD* = 8 Å. Figure 2 shows (left) the native structure and (right) the best-ranked model.

T0571_D2 (135 *residues*). Group KnowMIN submitted the highest ranked model for this target with a *GDT_TS* = 35.56 without a template (according to REMARK lines of their model). This group is closely followed by a number of human groups. All these groups seem to have used the model created by the Baker-Rosetta server, which used 3E9T_D2 as template. The Baker-Rosetta model does not have the overall barrel topology or strand arrangement of the target but it has some common features and the *RMSD* = 10.5 Å. Figure 3 shows (left) the native structure and (right) the Baker-Rosetta model.

Fig. 2. (*Left*) native structure for T0537_D1, (*right*) the best-ranked CASP9 model.

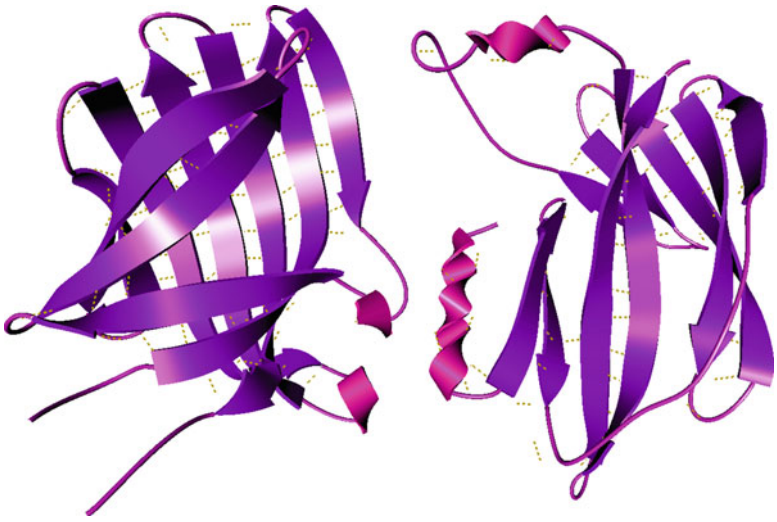


Fig. 3. (*Left*) native structure for T0571_D2, (*right*) the Baker-Rosetta model.

T0581 (105 *residues*). The Baker-Rosetta server generated a good model for this target (*GDT_TS*=67.86, *RMSD*=4.4 Å) and then the "humans" in the FoldIT project improved it (*GDT_TS*=70.48, *RMSD*=3.9 Å). This model was considered the winner of the competition because it produced the largest improvement over the closest template and because the predictions from the secondary structure prediction servers were very wrong. Figure 4 shows (left) the native structure and (right) the best-ranked model submitted by FoldIt.

T0604_D1 (84 *residues*). The Zhang server produced a very good model (*GDT_TS*=67.50, *RMSD*=2.66 Å) for this target using information from templates and contact predictions. Figure 5 shows (left) the native structure and (right) the best-ranked model submitted by the Zhang server.

Fig. 4. (*Left*) native structure for T0581, (*right*) the best-ranked CASP9 model.

Fig. 5. (*Left*) native structure for T0604_D1, (*right*) the best-ranked CASP9 model.

T0624 (*69 residues*). The Baker group submitted the best prediction, which was obtained without a template. Strand 3 is missing in the model but the overall topology is correct with a *GDT_TS*=56.16 and *RMSD*=5 Å. Figure 6 shows (left) the native structure and (right) the model generated by the Baker group.

In the next section, we discuss the targets for which the best predicted model had a *GDT_TS*< 30. This set includes T0550_D2 (PDB code 3NQK, 162 residues) from Bacteroides ovatus, T0578_D1 (PDB code 3NAT, 155 residues) from Enterococcus faecalis, T0604_D3 (PDB code 3NLC, 202 residues for the fragment that corresponds to the beta sheet) from Vibrio parahaemolyticus, and T0621 (PDB code 3NKG_D1, 169 residues) from

Fig. 6. (*Left*) native structure for T0624, (*right*) the best-ranked CASP9 model.

Sulfurospirillum deleyianum. T0629 is a non-globular protein with a high alpha-helical content and so outside the BuildBeta domain. We discuss how BuildBeta can help the most successful groups by sampling the protein conformation space more thoroughly after they have created their models, thus creating new models, some of which closer to the native structure than the original ones. We show that the SSS created by BuildBeta are of comparable quality to those of the "more successful" models described in this section.

2.2. Improving CASP9 Results with BuildBeta

CASP participants can submit up to 5 models for each target, with model 1 being their best model according to their own ranking method. Unlike the targets discussed in the previous section, most of the best models submitted by different groups for the targets discussed in this section had incorrect secondary structure or incorrect beta topology. Our goal in this paper is to show that BuildBeta could have helped some of these groups achieve considerably better models by producing better SSS and by sampling in the area of the correct beta topology. To that end, we downloaded the model 1 submitted for each one of these targets and analyzed the BuildBeta results obtained by using a prediction file that has their secondary structures as input. Notice that BuildBeta focuses on the beta strands and how to form beta sheets and simply places the helices away from the sheet. Therefore, the structures it generates should be subjected to further modeling (see Note 4). Moreover, as mentioned in Subheading 2.1, BuildBeta generates a large number of configurations corresponding to all possible topologies for a given input file and only ranks them according to a simple (and outdated) scoring function (16) (see Note 5) leaving the use of a more sophisticated scoring function to the user (BuildBeta code is set up to include a scoring function if the user so desires). Consequently, the models shown here are handpicked and are compared against the best models of the CASP9 competition simply to show that BuildBeta can complement other methods by enhancing their sampling phase. We discuss the experiments in more detail next.

```
AA: TGYVVDNNSIFFYAGLINEDMDKDMRKKYKINVHFKEDGTLDMKQDDPSNEMEFELIGTPTYSSTSVMDA
Exp:CCEEEECCEEEEECCCCCCCCCCCCCHHHEEEEEECCCCCEEEEECCCCCCCCEEEEEEEEEEEEECC
Mod:CEEEEECCEEEEECCCCCCCCHHHCCCCCEEEEEEECCCEEEEEEEECCCCEEECCCCCCCCCCCCCCCC

AA: TRPYLERRYVQIMFEYDFQDFTYGGSGTEVIPIKYRVAGSMTLLRNINTQIPDEDQQIEW
Exp:CCCCEEEEEEEEEEEEEECCCCCCCCCCCCEEEEEEEEEEEEEEEECCCCCCCCCCCC
Mod:CCCCCCCCEEEEEEEEEEEECCCCCCCCCCCCEEEEEEEEEEEEEECCCCCHHHHHHHC
```

Fig. 7. Primary sequence for target T0550_D2 (line beginning with AA), secondary structures obtained from the native structure (line beginning with Exp), and secondary structures according to the model submitted by MUFOLD_Server (line beginning with Mod). The secondary structures represented with "E" (strands) are shown on *light gray* background (*yellow* in the color version) and those represented with "H" (*helices*) are shown on *darker gray* background (*blue* in the color version).

2.2.1. T0550_D2

The best-ranked model 1 for this target was generated by MUFOLD_Server for 87% of the structure (residues:210–339) with *GDT_TS*=26.85 and *RMSD*=9.2 Å. Because our goal is to show what BuildBeta could have done to improve this model, we converted the pdb file of the model generated by the MUFOLD_Server into a prediction file. In other words, we took the secondary structures of the model and created an extended structure that has alpha helices and beta strands according to the model but extended coil regions. Also, the values for the dihedral angles in the alpha helices and beta strands are the ideal values assigned by ProteinShop rather than those in the model (see Note 3). Figure 7 shows the predicted secondary structures according to the model submitted by the MUFOLD_Server (line beginning with Mod) and the secondary structures according to the experimental structure (line beginning with Exp).

BuildBeta cannot align strands that are connected by coils that are 3 residues long or shorter. Consequently, we modified the MUFOLD_Server-based prediction file so that all the coil regions have at least 4 residues. Because the MUFOLD_Server model suggests that there are two sheets, we ran BuildBeta with core, which allows us to create two sheets (we created cores by considering pairs of consecutive strands and assigning one strand to each core). Figure 8 shows (left) the native structure 3NQK_D2, range:210–339, (center) the best model created by the MUFOLD_Server, and (right) one of the best models created by BuildBeta using the secondary structure predictions according to the MUFOLD_Server model (line Mod in Fig. 7).

Although the strand predictions were reasonable, the arrangement of the beta strands in the MUFOLD_Server model is different from that in the experimental structure. In fact, the arrangement of the beta strands in both structures is as follows:

3NQK_D2	MUFOLD_Server
I: 1 2 3 4	I: 1 2 7 6
II: 5 6 7	II: 3 4 5

Fig. 8. Experimental structure and models for target T0550. (*Left*) native structure 3NQK.pdb for residues 210–339, (*center*) best CASP9 model for T0550 submitted by the MUFOLD_Server, *RMSD* = 9.2 Å, (*right*) a BuildBeta-generated model based on the secondary structure of the MUFOLD server model, *RMSD* = 8 Å.

where the roman numbers denote the sheet number and the Arabic numbers represent the strand position in the protein sequentially numbered starting at the N-terminal. Thus, in this case, BuildBeta could have helped the predictors by sampling the topology that has the correct arrangement of the beta strands. The *RMSD* between one of those BuildBeta samples and the experimental structure is about 8 Å. Notice that, because we took the secondary structures from the MUFOLD_Server model, our prediction file has strand #5 shorter than in the native structure and an alpha helix at the end.

2.2.2. T0578_D1

The best model for this target was submitted by the group prmls in the human category. The *GDT_TS* score was 28.72 and the group used a template, 2EMB_D1. The beta sheet in the model, as that in the template, has the strand ordering 1 2 3 4 all antiparallel. However, strands 3 and 4 in the native structure are parallel. We converted the pdb file of the prmls-generated model into a

Fig. 9. Experimental structure and models for T0578. (*Left*) native structure 3NAT.pdb, (*center*) best CASP9 model submitted by group prmls, *RMSD* = 7.5 Å, (*right*) BuildBeta model, *RMSD* = 4 Å.

prediction file and, because we are interested in the beta sheet supersecondary structure, we only considered the sequence that corresponds to the beta sheet. We ran BuildBeta with this input and it generated structures for 96 topologies including the topology that has the correct alignment of beta strands. Figure 9 shows the native structure (left), the model submitted by the prmls group (center), and one of the models generated by BuildBeta (right). *RMSD* between the structure generated by group prmls and the native structure is 7.5 Å whereas *RMSD* between the BuildBeta-generated model and the native structure is 4 Å.

2.2.3. T0604_D3

Group SWIFT-Human submitted the highest ranked model 1 for this target using 2BRY_D1 as template (*GDT_TS* = 17.44, *RMSD* = 15 Å). A quick look at this model (see Fig. 10, center) shows that there is a potential strand rendered as a coil that is not recognized as such by STRIDE (17) but that has dihedral angles in the beta-sheet area of the Ramachandran plot and is H-bonded with strand 2 in the model. To better identify the residues in that potential strand, we used a ProteinShop feature that, given a pdb file (in this case, the pdb file of the SWIFT-Human model), permits a user to visualize the dihedral angles of the residues as dots in a Ramachandran plot. While doing that, we discovered that there were other regions of the structure that had dihedral angles in the

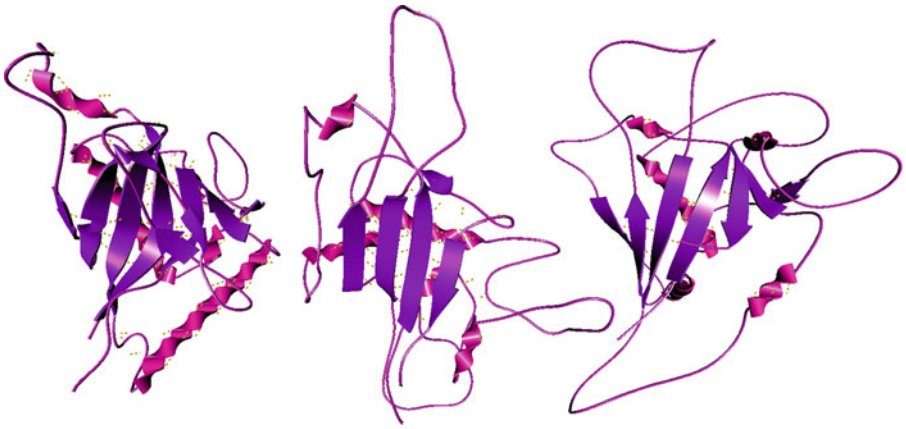


Fig. 10. Experimental structure and models for T0604_D3. (*Left*) Domain 3 of the native structure 3NLC.pdb, (*center*) best CASP9 model submitted by group SWIFT-Human, *RMSD* = 15 Å, (*right*) BuildBeta-generated model, *RMSD* = 9 Å.

```
AA:  PFSVGFRIEHKQSMIDEARFGPN AGHPILGAADYKLVHHCKNGRTVYSFCMCPGGTVVAATSEEGRVVT
Exp: CEEEEEEEEEECCCHHHHHHCCCCCCCCCCCCCCC EEEECCCCCEEEEEEEEEECCEECCCCCCCCCE
Mod: CCCEEEEEEEECCCCHHHHCCCCCCCCCCCCCC EEEECCCCHHHHHHHHHHCCCCC EEEEEEECCCCCCC

AA:  NGMSQYSRAERNANSAIVVGISPEVDYPGDPLAGIRFQRELESNAYKLGGEN YDAPAQKIGDFLKGRDP
Exp: ECCCCCCCCCCEEEEEEEEECCCCCCCCCCHHHHHHHHHHHHHHHHHHCCCCC EEEEEHHHHHHCCCC
Mod: CCCCCCCCCCEEEEEEEEEECCCCCCCCCCCCCCCCCCCCCHHHHHCCCCC EEEECCHHHHHCCCCC

AA:  SQLGDVEPSFTPGIKLTDLSKALPPFAVEAIREAIPAFDRKIKGFASEDGLLTGVETRTSSPVC
Exp: CCCCCCCCCCCCCEEECCHHHCCCHHHHHHHHHHHHHHHHCCCCCCCCCC EEEEEEECCCCCCCC
Mod: CCCCCCCCEEEEEEECCCCCCCCCHHHHHHHHHHHHHCCCCCCCCCCCCCCCCCC EEEECCCC
```

Fig. 11. Primary sequence for target T0604_D3 (line beginning with AA), secondary structures obtained from the native structure (line beginning with Exp), and secondary structures according to the model submitted by SWIFT_Human (line beginning with Mod). The latter shows the beta strands of the SWIFT_Human model on gray background (*yellow* background in the color version) and those that we added based on their dihedral angle values on *darker gray* background (*orange* in the color version).

beta sheet region of the Ramachandran plot and so we created a prediction file that has the original beta strands in the model and also some potential strands detected by the analysis of the dihedral angles. Figure 11 shows the sequence of amino acids for Target T0604_D3 (line AA), the secondary structures corresponding to the experimental structure (line Exp), and the secondary structure predictions that resulted from the SWIFT-Human model and our observations of the dihedral angle values (line Mod). The latter shows the beta strands of the SWIFT_Human model on a light gray background (yellow background in the color version) and those that we added based on their dihedral angle values on a darker gray background (orange in the color version).

We ran BuildBeta using the modified prediction file as input. BuildBeta created a large number of files corresponding to 161,280 topologies. Figure 10 shows a model created by BuildBeta for the

native topology (right), compared to the native structure (left), and the structure created by the SWIFT-Human group (center). The native structure has 7 strands forming a beta sheet and 2 very short ones in front of that main sheet. The SWIFT-Human model has 4 strands aligned in the order 2 3 1 4 all antiparallel. The BuildBeta-generated model (using the predictions shown in Fig. 11) has 7 strands and *RMSD*=9 Å. Notice that the helices in the BuildBeta model are the same as in the SWIFT_human model although the arrangement is different given that our model has additional beta strands. For example, the helix at the bottom of our model, which is located between strands 2 and 3, is on the left-hand side of the SWIFT-Human model between its strands 1 and 2.

2.2.4. T0621

T0621 is a beta sandwich and also a case in which human expertise was successful at selecting distant templates. In fact, the prmls group submitted a model 1 with the correct topology. Its *GDT_TS*=30 and *RMSD*=9.9 Å. Considering all the 5 models submitted by each group (rather than their model 1 only), two servers submitted models with better *GDT_TS* (33.43) than prmls although one has an incorrect topology and the other misses strands 1 and 8. An interesting observation about this target was made by the Zhang group, one of the top performers of the competition, which did not find a template and could not produce a good model for this target. In their CASP9 presentation (14), this group said that their methods were unable to generate beta sheets of long-range contact order like those in T0621 and illustrated this point with the model 1 submitted by their Zhang-Ab-Initio group in which all beta sheets have short-range hairpins. BuildBeta can help overcome this limitation. To emphasize this point, we did an experiment in which we took the secondary structure from the model 1 of the Zhang-Ab-Initio group (*RMSD* with native =15.7 Å) and ran BuildBeta which generated a structure that has the correct beta sheet topology and *RMSD*= 10 Å. BuildBeta does not take care of loop or alpha-helix modeling and the poorly placed alpha helices are a big factor in the large *RMSD* of our model. We roughly moved them to the position they have in the native structure so that the *RMSD* is more indicative of the quality of the BuildBeta-generated beta sheets. The model with the roughly replaced helices but unmodeled loops has *RMSD*= 7.5 Å from the native structure.

2.3. Dealing with Complexity

One of the main limitations of BuildBeta is the large number of configurations it creates. The number gets even larger when one tries not only different arrangements of the predicted beta strands into a beta sheet but also two or more sheets as in the case of a beta sandwich. Chiang et al. (18) provide an interesting result that could substantially limit the complexity of BuildBeta to generate beta sandwiches. By looking at protein structures as an arrangement of supermotifs, Chiang et al. could find specific patterns of

strand packing in sandwich-like proteins and demonstrate that only a few possible arrangements of beta strands are found in this type of proteins. Based on these findings, they developed a supersecondary database http://binfs.umdnj.edu/sssdb/ that focuses on sandwich-like proteins with two main beta sheets. It shows that sandwich-like beta proteins can be described by a very small number of supermotifs. Thus, the database contains 703 proteins, which are classified into 38 supermotifs, each containing a number of motifs and proteins that are representatives for each motif. If we had limited BuildBeta to simply generate the motifs in the SSS database rather than all possible ones, then it would have found a reasonably good model for target T0621 within a few minutes.

Using the prediction file obtained from the Zhang-Ab-Initio model, which contains 7 beta strands, BuildBeta would have found the motif 1AO in the SSS database with the arrangement:

Motif 1AO	Target T0621
I: 1 7 3 5	I: 1 8 3 6
II: 2 6 4	II: 2 7 4 5

The representative for this motif is protein 1CWP which contains 8 beta strands rather than 7 and has the same arrangement of strands as target T0621. Moreover, Kister and Gelfand (19) contend that a small number of residues in a sequence are critical to structure formation whereas others play minor structural roles. They show that two beta sandwiches are alike (i.e., they have the same number of strands as well as ordering and orientation of the strands) if and only if they share a unique set of supersecondary structure-determining residues (conserved residues). To find these residues among proteins that have very low sequence similarities, they propose an algorithm that is secondary structure based rather than sequence based. In their algorithm, the units of alignment are strands and loops. Residue similarity is defined based on whether the residues are hydrophobic or hydrophilic. As Kister et al. point out, there are different possible alignments but the goal is to find the optimal variant which affords the greatest number of conserved positions.

We aligned the sequence of T0621 with that of 1CWP using the secondary structure of the Zhang-Ab-Initio model and the alignment rules of (19), and the result is shown in Fig. 12. Residues with dark background (red in the color version) represent those considered interchangeable hydrophobic in (19) whereas those with white background are those considered interchangeable hydrophilic. Also shown in lighter gray background are identical residues (shown in blue in the color version) as well as the beta strand predictions (shown in yellow in the color version). The alignment in Fig. 12 suggests that the Zhang-Ab-Initio group predicted an

```
SS  :CCCCCCCCEEEEEEEEECCCCCCCEEEEEECCCCHHHCCHHHHHCCEEEEEECCEECCC--------------
1CWP:KAIKAWTGYSVSKWTASCAAAEAKVTSAITISLPNELSSERNKQLKVGVLLWLGLLPS---------------
3NKG:     SNAPNPISIPIDLSQAGSVVEKEVKIEES----------WSYHLILQFAVHDRKEDGGLDGKRVWKF
SS  :     CCCCCCEEEEEECCCCCEEEEEEEEEEC----------CEEEEEEEEEECCCCCCCCCHHHHHHC

SS  :---------------------------------------- CCCCEEEEEEECCCCHHHHHHHCCCEEECCCC
1CWP:---------------------------------------- VSGTVKSCVTETQTTAAASFQVALAVADNSKD
3NKG:LGFNSYDPRDGKQVGYVDYRLAKSELGDLIDETYDCDGTVVPIKITIHQINQDN-------TKKLIADNLY-
SS  :CCCCEECCCCCCEECCHHHHHHHHHHCCCCCCCCCCCCCC EEEEEEEEECCCC-------CEEEEEEEE-

SS  :CCC---------EEECCCCCCCCHHHHHHHEEEEEEECCCCCCCCCEEEEEEEEEECCCHHH
1CWP:VVA---------AMYPEAFKGTTLEQLAADLTIYLYSSAA--LTEG-DVIVHLEVEHVRPT
3NKG:MTKGNGSGAYTRDITTISLDK---------GKYIFRIENIEAFSEMIGRKVDFTIYINKR
SS  :CCCCCCCCCCCCHHHCCCCCC---------CEEEEECCCCCCCCCCCCCCEEEEEECCCC
```

Fig. 12. Alignment between 1CWP (representative for motif 1AO found in supersecondary database http://binfs.umdnj.edu/sssdb/) and the native structure for T0621, 3NKG, according to alignment rules of (19). The SS line at the top corresponds to the secondary structure of 1CWP and the SS line at the bottom corresponds to the secondary structure of 3NKG. Residues with *dark gray background* (*red* in the color version) represent those considered interchangeable hydrophobic in (19) whereas those with white background are those considered interchangeable hydrophilic. Residues with light gray background (*blue* in the color version) are identical in both sequences. Also, the line SS shows the beta strand predictions with light gray background (*yellow* in the color version.).

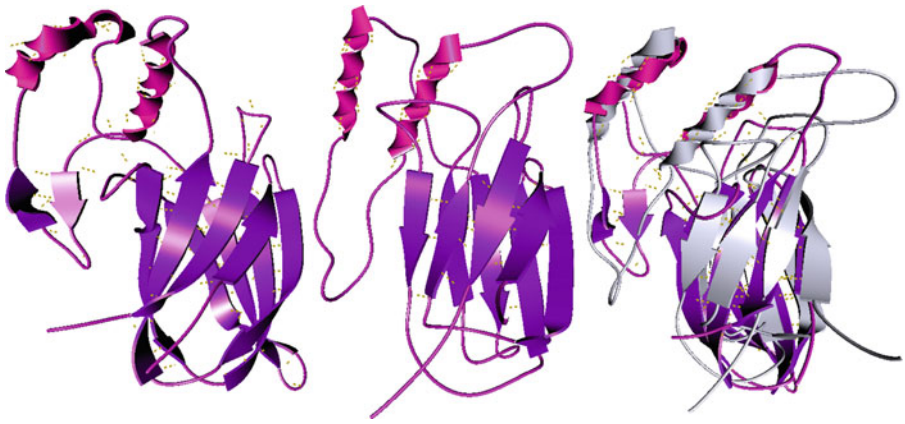


Fig. 13. Experimental structure and model for T0621. (*Left*) Native structure 3NKG.pdb, (*center*) BuildBeta model created using secondary structure predictions based on the Zhang-Ab-Initio model 1, (*right*) the BuildBeta model (*light gray*) superimposed on the *darker gray* native structure (*magenta* in the color version). *RMSD* = 6 Å.

alpha helix (last SS line in Fig. 12) instead of a strand. We added the missing strand to our prediction file and generated the structure that is shown in Fig. 13 (center) superimposed (right) on the native structure of T0621 (left). The RMSD between our model and the native structure is 6 Å. Moreover, if we considered rule 1 in (19) for the alignment of beta strands that states, "If residue a in strand A of protein P is H-bonded to residue b in strand B then residue a′ in corresponding strand A′ of protein Q is H-bonded to residue b′ of corresponding strand B′", then we could have created a model with the same H-bonds as in the native structure.

In this section we discussed a fairly common weakness of protein structure prediction methods: they can seldom predict the

correct beta topology for those targets that present unusual beta folds and do not have a close template. Modeling long-range interactions is an extremely difficult problem with high combinatorial complexity. BuildBeta can help those methods by thoroughly sampling the conformation space. The structures thus generated are not final and should be further processed to pack helices and model coils, but these results show that even these "raw" structures are better than the best-ranked models submitted to CASP9 for targets in the TFM category.

The number of structures generated by BuildBeta depends on the number of beta strands as well as on the length of the coil regions and other geometric constraints that limit some of the achievable topologies. For example, there were 96 possible topologies for target T0578_D1 and 161,280 topologies for target T0604_D3 (see Note 1).

The next challenges for BuildBeta are how to reduce the number of structures created and how to pick the correct topology among the many possible ones. We need to replace the almost brute-force sampling of strand arrangements with a more information-guided sampling and we need to replace the old zipping tables with more up-to-date information (see Note 6), but we still want to generate enough variety not to miss the new folds.

3. Methods

Another frequent problem discussed by the CASP9 assessors for the TFM category (14) is the poor quality of the secondary and SSS. Models with beta strands tend to have incorrect backbone torsion angles or no H-bonds with a neighboring strand. Some methods also fail to close barrels even though the model generated has beta strands that seem to form a barrel but do not come all the way around to close it. BuildBeta could also help these methods to tackle these problems. First, it assigns dihedral angles that are within the beta sheet region of the Ramachandran plot to the beta strands (see Note 3) and zips adjacent strands through H-bonds. We are currently working on increasing the number of H-bonds to improve the SSS. Second, BuildBeta provides a barrel feature that generates closed barrels and that could be invoked by predictors whenever their method produces models that look like open barrels. We discuss these features next.

3.1. Optimized Sheet Building

In our paper (13), we showed how to arrange rigid beta strands into beta sheets, by specifying the hydrogen bonding geometry for two central "zipping" hydrogen bonds on two adjacent strands of the proposed sheet, and using inverse kinematics to move the flexible coils to achieve this specified geometry when possible. If the beta strands have their backbone dihedral angles set to the

standard values at the center of the beta strand region of the Ramachandran plots, they will be twisted, and the potential hydrogen bonds joining other residues besides the central pair on the two strands may not be formed properly. Perfectly flat sheets with no twist can achieve all potential hydrogen bonds, but are not the best approximation to strands in a native protein. We develop here an optimal backbone geometry for the strands, and optimum hydrogen bonding geometry for the zipping pair of residues, for various configurations of parallel and antiparallel strands, in order to create most of the potential hydrogen bonds. Since the parameters for these optimal configurations will be used independent of the residues in each strand, we assumed that all residues were alanine in order to include at least the steric effects of the beta carbon of the side chain in the energy terms being optimized. To the standard Amber energy terms (20), we added extra artificial energy terms in the form of springs on the length of the desired hydrogen bonds, and on their N_H…O and H…O_C angles, in order to favor their formation, and then decreased the weights of these terms as the LBFGS (21) energy minimization iterations proceeded. Also, to keep the backbone dihedral angles near values that give a native-like twist to the strands and the sheets, we added springs to keep these dihedral angles near the values $\phi = -119$ and $\varphi = 132$, which are the averages of the values given for parallel and antiparallel sheets in Subheading 3.2.2 of (22) (see Note 7).

We optimized the repeated ϕ and φ angles for strands of $N = 3$, 5, 7, or 9 residues, assumed to be either in a parallel hydrogen bonding configuration on both edges of the strand, in an antiparallel configuration on both edges, or in a mixed configuration, with a parallel strand on one edge and an antiparallel strand on the other. Since our zipping residues are at the center of the potentially hydrogen-bonded region where the strands overlap in the proposed alignment, it is sufficient to optimize only for the odd overlap lengths, and then use the next largest odd number for an even-length overlap region.

For the parallel strand case, we consider three parallel strands of length N, as shown in Fig. 14 (left), so that the hydrogen bonds

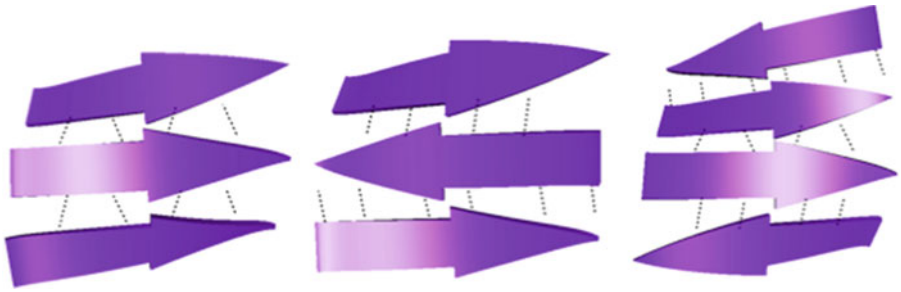


Fig. 14. (*Left*) all-parallel strand case, (*center*) antiparallel strand case, and (*right*) mixed parallel/antiparallel case.

on both edges of the central strand can be considered. The two central zipping hydrogen bonds are identical on both edges of the strand, so there are six rigid body degrees of freedom for specifying the position of one strand with respect to the other. Adding the two degrees of freedom for the strand backbone φ and ϕ angles gives 8 parameters in the optimization. The detailed definition of the rigid body parameters is given in Subheading 3.2.

For antiparallel strands, we again consider three strands as shown in Fig. 14 (center). Note that in this case the two instances of zipping geometry are not identical. Between the bottom and middle strands, the two central zipping hydrogen bonds are between hydrogen and oxygen atoms on the two central residues, but since these atoms are already "used up," the two zipping hydrogen bonds between the middle and top strands are between the atoms on the residues preceding and succeeding the two central residues. In addition, there is the potential for twofold strand-pair rotational symmetry about axes perpendicular to the plane of this figure, through a point between each of the two pairs of central hydrogen bonds just discussed. We enforce this potential symmetry, resulting in only four degrees of freedom for the geometric relationship for each of these two pairs of strands. Adding the two degrees of freedom for the strand backbone, this results in 10 parameters in the optimization. Their definition is given in the following section.

Figure 14 (right) shows a configuration with two mixed strands. The strand second from the bottom has an antiparallel strand below it, and a parallel strand above it. The strand above it has a parallel strand below it, and an antiparallel strand above it. However, these two strands are not equivalent, because the central hydrogen bonds between the two top antiparallel strands join hydrogen and oxygen atoms on the two central residues, while that is not the case for the ones between the bottom two antiparallel strands. Therefore, we consider all four strands in the optimization. Note that if more strands were added to the sheet, continuing to alternate between parallel and antiparallel, the three hydrogen bonding patterns shown between pairs of adjacent strands in this figure would just repeat over and over again, so this figure contains all the relevant geometry. As discussed above, there are six degrees of freedom for the parallel hydrogen bonding geometry between the central two strands, and two different sets of four degrees of freedom for the antiparallel strand relationships in the top and bottom pairs of strands, when twofold rotational symmetry is assumed. Together with the two degrees of freedom for the strand backbone dihedral angles, this gives 16 parameters for the optimization.

There may be other arrangements than the cases discussed above, for example in a five strand sheet, when the first four strands from the top are in the alternating parallel/antiparallel arrangement, but the bottom strand does not continue this pattern.

In this case, there are conflicting assignments to the backbone configuration of the fourth strand from the top, so we assign the backbone configuration to the overlap regions of the fourth strand twice, first for the alternating case, and then for the anti-parallel case. The last assignment will take priority.

3.2. Parameters

First we describe the construction of the ideal zipping geometry, when all the rigid body degrees of freedom parameters have the value zero, and then we describe the motions specified by these parameters.

3.2.1. Parallel Strands

For the parallel case, we assume that the geometry is as shown in Fig. 14 (left), with the "anchoring" strand in the middle fixed, and the "manipulated" strand below it is to be moved into the ideal zipping position. The ideal zipping is arranged to align as closely as possible ideal "bond sites" which would be the midpoints of straight hydrogen bonds of a standard 2.025 Å length, if the C, O, N, and H atoms involved were all in the same straight line. These bond sites are shown as black dots in Fig. 15. Let $amide_1$ and $carboxyl_1$ denote the positions of the N and C backbone atoms of the manipulated residue R_i, as show on the left of Fig. 15. Similarly, define $amide_2$ and $carboxyl_2$ on the anchoring strand. Note that as shown in Fig. 15, these atoms are on the successor and predecessor residues of the central zipping residue R_j on the anchoring strand.

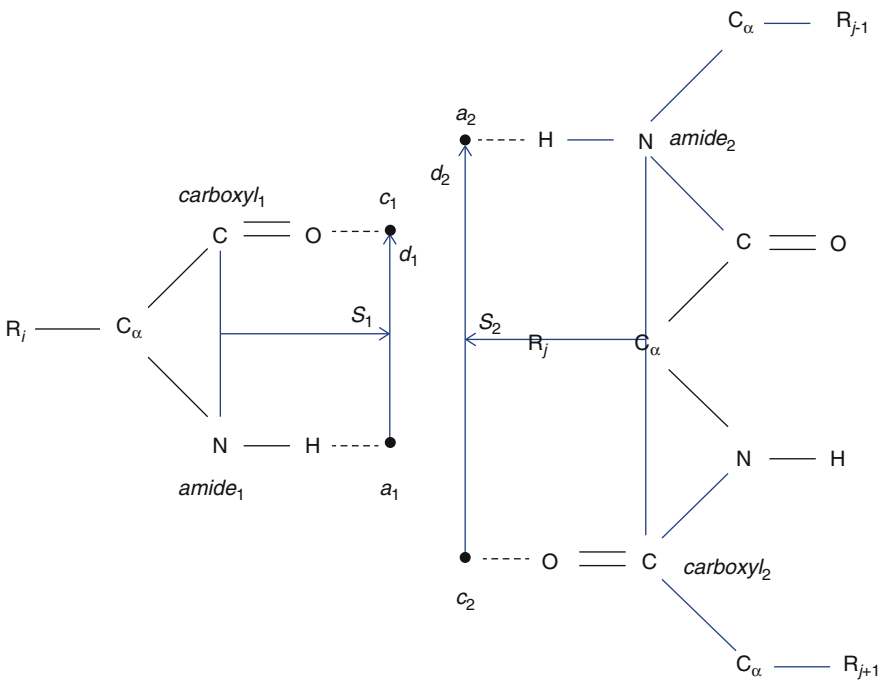


Fig. 15. Zipping geometry for parallel strands.

Extend the NH bond vector from $amide_1$ past the hydrogen atom a distance of 1.0125 Å (half the length of an ideal hydrogen bond) to get the location a_1 of the bond site for the hydrogen bond from this amide group. Similarly, extend the CO bond from $carboxyl_1$ past the oxygen atom a distance of 1.0125 Å to get the bond site c_1, extend the CO bond from $carboxyl_2$ past the oxygen atom a distance of 1.0125 Å to get the bond site c_2, and extend the NH bond vector from $amide_2$ past the hydrogen atom a distance of 1.0125 Å to get the bond site a_2. The zipping motion should align as closely as possible the position of a_1 with c_2, and of c_1 with a_2, and also align as closely as possible the half-bond vectors to these ideal bond sites.

We first want to align the two vectors $d_1 = c_1 - a_1$ and $d_2 = a_2 - c_2$ in the same direction, so we rotate the manipulated strand around an axis which is the cross product of d_1 and d_2 by the angle between these two vectors. Next we want to make the hydrogen bonds as straight as possible by aligning the planes of the vectors to the ideal bond sites. We define a normal $plane_1$ to the average plane of the ideal zipping bonds of the manipulated residue by letting $s_1 = (a_1 - amide_1 + c_1 - carboxyl_1)/2$ and taking $plane_1$ to be the cross product of d_1 and s_1. (Note that we use the coordinates after the above rotation of the manipulated strand to compute s_1 and $plane_1$.) Similarly, we define $s_2 = (a_2 - amide_2 + c_2 - carboxyl_2)/2$, and take $plane_2$ to be the cross product of s_2 and d_2. The cross products are in the opposite order as for $plane_1$, since s_1 and s_2 point in approximately opposite directions. In order to align these two planes, we rotate the manipulated strand around an axis, which is the cross product of $plane_1$ and $plane_2$, by the angle between these vectors, to align the two average bond vector planes. Both of these rotations use an axis through the origin, which may be far from the zipping bonds in question. However we compensate by a final translation, which moves the midpoint of the rotated positions of a_1 and c_1 to the midpoint of the unrotated positions of a_2 and c_2, creating the approximately equal-length hydrogen bonds. Because these bonds are diagonal rather than aligned with the ideal straight half-bond vectors to the bond sites, they will be longer than the ideal length, but this will be corrected during the optimization procedure.

The six rigid-motion degrees of freedom for the zipping geometry define the following additional rotations and translations of the manipulated strand away from the above ideal configuration. Let p_2 be the vector from the origin to the midpoint of a_2 and c_2. We first translate by the vector $-p_2$ (and later, after the rotations, back by p_2) so that the rotations will be about this common midpoint as center. We define three orthogonal unit rotation axis vectors, v_d along d_2, v_p along $plane_2$, and v_{dp} along the cross product of v_d and v_p. The second, fifth, and sixth degrees of freedom define rotations about the axes, v_{dp}, v_d, and v_p, taken in that order. Then we translate back by p_2 so that the midpoint of a_2 and c_2 ends up fixed. Finally we

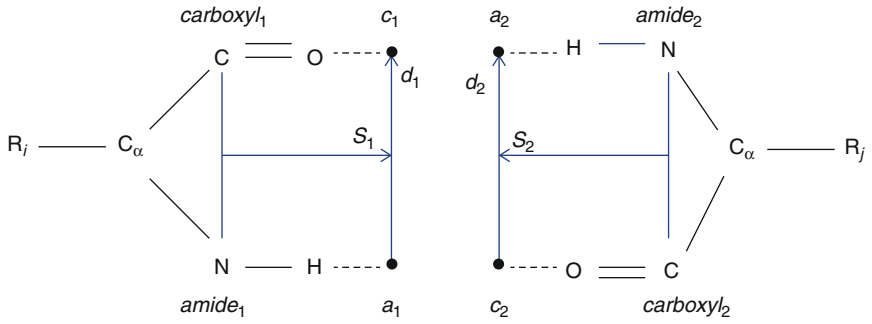


Fig. 16. Zipping geometry for antiparallel strands.

translate along the axes v_d, v_p, and v_{dp} by distances specified in the first, third, and fourth degrees of freedom, respectively.

As discussed in (13), there are two different parallel zipping cases, the one described above, and the converse case, when the backbone NH group of residue R_j on the fixed anchoring strand is hydrogen-bonded to the CO group of the predecessor R_{i-1} of R_i on the moving manipulated strand, and the CO group on R_j is hydrogen-bonded to the NH group on the successor R_{i+1} to R_i. In this case the roles of R_i and R_j are reversed compared to the discussion above, so the inverse of the transformation, which positions R_j with respect to R_i, is used to position R_i with respect to R_j.

3.2.2. Antiparallel Strands

For antiparallel strands, the two zipping cases are more fundamentally different, since in the first case both the backbone CO and NH groups of R_i are hydrogen-bonded to those of R_j, while in the second case the backbone NH group of R_{i+1} is hydrogen-bonded to the CO group of R_{j-1} and the CO group of R_{i-1} is hydrogen-bonded to the NH group of R_{j+1}. As before, for the first zipping case, let $amide_1$ and $carboxyl_1$ denote the positions of the N and C backbone atoms of the manipulated residue R_i, as shown in Fig. 16.

Similarly, define $amide_2$ and $carboxyl_2$ at residue R_j on the anchoring strand. Let a_1, c_1, a_2, and c_2 be the respective bond sites, as defined in the parallel case by extending the bond between the two atoms in each of these groups by 1.025 Å. Then, the ideal transformations, making these bond sites coincide and aligning the average planes of their hydrogen bonds, are constructed as follows. Let $s_1 = (a_1 - amide_1 + c_1 - carboxyl_1)/2$ and $s_2 = (a_2 - amide_2 + c_2 - carboxyl_2)/2$. To make s_1 point in the same direction as $-s_2$, we rotate the manipulated residue about an axis in the direction of the cross product of s_1 and $-s_2$, by an angle equal to the angle between them. Renaming a_1 and c_1 to their new positions, let $d_1 = c_1 - a_1$ and $d_2 = a_2 - c_2$, and let $plane_1$ be the cross product of s_1 and d_1, and $plane_2$ be the cross product of s_2 and d_2. To make these the planes with these two normals parallel, we rotate the manipulated strand

about an axis in the direction of the cross product of $plane_1$ and $plane_2$ by the angle between these two vectors.

Finally we translate so that the midpoint between the (doubly) new a_1 and c_1 coincides with the midpoint between a_2 and c_2. If the backbone geometry of these two strands is the same, these coinciding midpoints will become a center of twofold rotation, with axis in the direction of $plane_2$. When we apply rigid body degrees of freedom away from this ideal case, we want to preserve the twofold rotational symmetry, and there are only four degrees of freedom that do this, defined as follows. Let m be the vector from the midpoint of a_1 and c_2 (which should now actually coincide) and the midpoint of a_2 and c_1. The third degree of freedom represents a rotation about the axis m. It transforms the vector s_1 associated with the manipulated residue to a new vector, which we still call s_1.

Another symmetry-preserving motion would rotate the manipulated strand about the axis s_1 and the anchoring about s_2, by an angle, which is the first degree of freedom, to give a propeller twist to the planes defined by the two normals $plane_1$ and $plane_2$. However, the anchoring strand is supposed to remain fixed, so we instead rotate the manipulated strand by the first rotation, followed by the inverse of the second. We revise the locations of the bond sites a_1 and c_1 to their positions after this rotation, and define a revised vector m from the midpoint of a_1 and c_2 (which no longer coincide) to the midpoint of a_2 and c_1 so that m is perpendicular to the new axis of twofold rotational symmetry. Another such perpendicular vector is $d = s_2 - s_1$, where s_1 is again revised according to the last rotation. We normalize the vector m and d to unit vectors. They are not necessarily perpendicular to each other, but since they are independent, and are both perpendicular to the new twofold symmetry axis, they form a basis for the set of translations of the manipulated strand that would preserve this symmetry (if the symmetry axis were moved by half the translation). The second degree of freedom is the translation along d, and the fourth is the translation along m.

The ideal zipping geometry for the second zipping case is the same, except that the residue numbers used to define the atoms involved differ by +1 or −1 from R_i and R_j, as discussed above. Figure 14 (center) shows three antiparallel strands, with the bottom pair joined by hydrogen bonds according to the first antiparallel zipping case, and the top pair joined according to the second zipping case. This configuration includes all the hydrogen bonding from the middle strand, and since the top and bottom strands are equivalent, it could be repeated to make as large an antiparallel beta sheet as desired. Therefore, our optimization minimized the energy for this three-strand case. There are four rigid body degrees of freedom for the first antiparallel zipping case between the bottom two strands, four more for the second antiparallel zipping

case between the top two strands, and two for the ϕ and φ dihedral angles of the strand backbones, giving a total of ten.

3.2.3. Mixed Strands

A mixed case is also possible, with some strands parallel, and some strands antiparallel. It is impractical to optimize all possible such patterns, but we have optimized the case of alternating parallel and antiparallel strands shown in the four strands in Fig. 14 (right). The first and second strands from the top, hydrogen-bonded by the first antiparallel zipping case, are equivalent by twofold rotation, and the third and fourth strands, hydrogen-bonded by the second antiparallel zipping case, are also equivalent by twofold rotation. The second strand would be moved to a position below the fourth strand by the latter of these twofold rotations, and could then take the role of a translated copy of the bottom strand, so this configuration has enough information to produce an alternating parallel and antiparallel sheet of any size. The second and third strands are not in equivalent environments, but this configuration contains all their backbone hydrogen bonds, so it is sufficient to minimize its energy. The degrees of freedom in order are the four for the antiparallel zipping between the first and second strands, the six for the parallel zipping between the second and third strands, the four for the antiparallel zipping between the third and fourth strands, and the two for the backbone φ and ϕ dihedral angles, giving a total of sixteen.

3.3. Beta Barrels

In (13), we constructed generic all-parallel and all-antiparallel barrels with the six degrees of freedom defining the geometry of all the "zipping" hydrogen bond pairs optimized to close the barrel, using a fake hydrogen bond energy term with springs on the H…O distances and the N_H…O and H…O_C angles. We have now generalized this procedure to allow 6 separate degrees of freedom for the separate zipping geometries of all the adjacent strand pairs so that we can consider mixed parallel and antiparallel strand pairs in the barrel. We optionally include the fake hydrogen bond energy for all the potential hydrogen bonds between the aligned strands in the barrel, not just the zipping pair, to increase the number of hydrogen bonds that are formed. We modified the rules for deciding which side of a sheet to place an alpha helix so that all alpha helices associated with a barrel are placed outside it. Also, long strands in a barrel can form more hydrogen bonds if their standard beta strand dihedral angles are modified to allow the strands to curl in a spiral around the barrel cylinder, so we allow the user to optionally specify an amount of such curling, or to specify that it be one of the variables to be optimized.

The *S* value of Murzin (23) specifies the total shift in the strand alignment around the barrel, and the sign of *S* determines the side of the sheet that will be on the outside. This side is selected by the user, and the alignments proposed by the Zhu and Braun (24) alignment scores are biased to give an *S* value of the appropriate sign.

Fig. 17. Experimental structure and model for T0571. (*Left*) Native structure 3N91.pdb, (*right*) a BuildBeta-generated model using secondary structure predictions obtained from the model 1 submitted by the Baker-Rosetta server. *RMSD* = 6.2 Å.

The natural twist of the strands then partially curls the sheet into a barrel, and it can optionally be further curled by setting the initial zipping geometry degrees of freedom. Then these degrees of freedom are optimized to minimize the fake hydrogen bond energy, moving just the strands, without considering the loops between them. Finally, inverse kinematics is applied to these loops, to achieve the optimized zipping geometry, and then to place any helices next to the barrel. We used this new barrel feature to generate structures for target T0571, which we discussed in Subheading 2.1. Figure 17 shows a structure created by BuildBeta using secondary structure predictions obtained from the model 1 submitted by the Baker-Rosetta server. The RMSD between the BuildBeta model and the native structure is 6.2 Å. The Web site http://proteinshop.org/main/PShop_BuildBeta.html contains videos of BuildBeta "in action" including two that show the formation of barrels.

4. Notes

1. A sheet conformation for an n-strand sheet is defined by a permutation of the numbers from 1 to n, representing the order of the strands in the sheet, and a binary orientation sequence

of n zeroes or ones, specifying whether the successive strands point up or down. If two consecutive digits in the binary sequence are the same, the corresponding adjacent strands are parallel, and if they are different, the strands are antiparallel. For a given sheet of n strands, BuildBeta may generate up to $n!/2$ valid permutations, and for each permutation, up to 2^{n-1} valid orientation sequences.

2. Some BuildBeta parameters can be set in the ProteinShop configuration file, which is read at start-up time. Among those parameters are *start_topology*, *end_topology,* and *try_topologies*, which define the start and end topology as well as the number of topologies to be calculated at runtime. If the user would like to generate n topologies using k processors then he/she can write a simple script that assigns n/k topologies to each processor.
3. When ProteinShop reads a prediction file that contains the sequence of amino acids and secondary structure prediction, it automatically assigns standard values to the ϕ and φ backbone dihedral angles. The standard values for an alpha helix are –60 and 45, respectively. The standard values for a beta strand are –119 and 132, respectively.
4. When beta sheets are built using the automatic zipping function, the dihedral angles of the residues in the coil regions are changed in a concerted way using IK. This function makes no use of contact forces or potential energy functions and, therefore, the resulting structures may have coils crossing other coils or sheets, or knotted topologies. We are exploring different coil refinement methods to be added to the ProteinShop software as plug-ins so that they can be applied to the structures generated by BuildBeta before they are output. Also, the zipping process drags along the alpha helices (unless they are part of a core region) and may leave them in a position that intersects the sheet. BuildBeta attempts to move them parallel to the sheet, at a specified distance from it. In the case of an alpha helix between two parallel beta strands, it chooses the side of the sheet that makes a right-hand screw turn, as in Fig. 2 of Richardson (25). This side is determined according to the up/down direction of the strand before the helix, and the placement (to the left or the right of this strand) of the strand after the helix, so it makes sense even if these two strands are antiparallel. If there are two or more helices in this same backbone segment, they are placed at successively farther distances from the sheet (assuming that the coils are long enough) so that they do not overlap.
5. Ruczinski (16) studied the statistical distribution of the topological structures of proteins in the Protein Data Bank, and came up with rules for fitting the probability of a given

structure based on whether the lengths of the loops (including alpha helices) between the strands are long (more than 10 residues) or short. BuildBeta computes the probabilities of each beta topology by using Eq. (9.38) of (16) and retains a list of the *try_topologies* (set in the ProteinShop configuration file) highest probability ones. However, given that these probabilities are outdated and that we can run BuildBeta in parallel as described in Note 2, we usually set *try_topologies* to $2^{n-2}\ n!$ (where n is the number of strands) so that all topologies are considered, regardless of their probability.

6. Zhu and Braun's alignment scores are based on statistics from a training data set of only 169 structures in the Protein Data Bank (24). These scoring functions do not take into account that side-chain packing in parallel and antiparallel sheets is different and so, the tables need to be redone and updated.
7. We had hoped that this optimization would produce dihedral angles that were close to the average values for antiparallel strands ($\varphi = -122$ and $\phi = 136$) and for parallel strands ($\varphi = -116$ and $\phi = 128$) reported in (22) but we noticed no such trend. Therefore we might get more native-like structures if our springs pulled the backbone dihedral angles to these separate values, depending on a strand's environment.

Acknowledgements

The computations were performed using NERSC machines. This work was supported by the Director, Office of Science, Office of Advanced Scientific Computing Research, of the US Department of Energy under Contract No. DE-AC02-05CH11231. The authors would like to thank Dr. Daniel Rodrigues for his helpful comments.

References

1. Crivelli S et al (2004) ProteinShop: a tool for interactive protein manipulation. J Comput Aided Mol Des 18:271–285
2. Azmi A et al (2000) Predicting protein tertiary structure using a global optimization algorithm with smoothing. In: Floudas CA, Pardalos PM (eds) Optimization in comp. chemistry and molecular biology: local and global approaches. Kluwer, Netherlands, pp 1–18
3. Crivelli S et al (2000) A global optimization strategy for predicting protein tertiary structure: α-helical proteins. Comput Chem 24:489–497
4. Head-Gordon T et al (2000) Predicting protein tertiary structure using a global optimization algorithm. In: Moult J, Fidelis K, Zemla A, Hubbard T (eds) Proc. of CASP4—fourth meeting on the critical assessment of techniques for protein structure prediction. Pacific Grove, California, pp A43–A44
5. Crivelli S et al (2002) A physical approach to protein structure prediction. Biophys J 82(36–49):2002
6. Kreylos O, Max N, Crivelli S (2002) ProtoShop: interactive design of protein structures. In: Moult J, Fidelis K, Zemla A, Hubbard T (eds) Proc. of

CASP5—fifth meeting on the critical assessment of techniques for protein structure prediction. Pacific Grove, California, pp A213–A214

7. Eskow E et al (2004) An optimization approach to the problem of protein structure determination. Math Program Ser A 101:497–514
8. Ding J (2004) Protein structure prediction using physics-based global optimization with knowledge-guided fragment packing. In: Moult J, Fidelis K, Hubbard T, Rost B, Tramontano A (eds) Proc. of CASP6—sixth meeting on the critical assessment of techniques for protein structure prediction. Gaeta, Italy, pp A111–A113
9. Ding Jinhui et al (2005) Protein structure prediction using physical-based global optimization and knowledge-guided fragment packing. IEEE Computational Systems Bioinformatics (CSB 2005), Stanford, CA, pp 211–213
10. Max N, Crivelli S (2006) Protein structure prediction using proteinshop. In: Proceedings of CASP7 – seventh meeting on the critical assessment of techniques for protein structure prediction, Pacific Grove, California, 26–30 Nov 2006, pp 101–102.
11. Hu C, Max N, Crivelli S (2008). Protein structure prediction using proteinshop. In: Proceedings of CASP8—eight meeting on the critical assessment of techniques for protein structure prediction, Sardinia, Italy, Dec 2008, pp 85–86.
12. Refugio Scott et al (2010) Structure prediction of beta proteins using BuildBeta. In: Proceedings of CASP9—ninth meeting on the critical assessment of techniques for protein structure prediction, California, Dec 2010.
13. Max N et al (2010) BuildBeta—a system for automatically constructing beta sheets. Proteins 78:559–574
14. http://www.predictioncenter.org/casp9/doc/presentations/CASP9_FM.pdf
15. Zemla A (2003) LGA: a method for finding 3D similarities in protein structures. Nuclei Acids Res 31(13):3370–3374
16. Ruczinski I et al (2002) Distributions of beta sheets in proteins with application to structure prediction. Proteins 48:85–97
17. Frishman D, Argos P (1995) Knowledge-based protein secondary structure assignment proteins 23:566–579
18. Chiang Y-S et al (2007) New classification of supersecondary structures of sandwich-like proteins uncovers strict patterns of strand assemblage. Proteins 68:915–921
19. Kister A, Gelfand I (2009) Finding of residues crucial for supersecondary structure formation. PNAS 106(45):18996–19000
20. Cornell WD et al (1995) A second generation force field for the simulation of proteins, nucleic acids, and organic molecules. J Am Chem Soc 117:5179–5197
21. Liu DC, Nocedal J (1989) On the limited memory method for large scale optimization. Math Program B 45(3):503–528
22. Hovmoller S, Zhou T, Ohlson T (2002) Conformation of amino acids in proteins. Acta Chrystallogr D58:768–776
23. Murzin AG, Lesk AM, Chothia C (1994) Principles determining the structure of beta-sheet barrels in proteins. A theoretical analysis. J Mol Biol 236:1369–1381
24. Zhu H, Braun W (1999) Sequence specificity, statistical potentials, and three-dimensional structure prediction with self-correcting distance geometry calculations of beta-sheet formation in proteins. Protein Sci 8: 326–342
25. Richardson J (1976) Handedness of crossover connections in beta sheets. PNAS 73:2619–2623

Chapter 9

A Modular Perspective of Protein Structures: Application to Fragment Based Loop Modeling

Narcis Fernandez-Fuentes and Andras Fiser

Abstract

Proteins can be decomposed into supersecondary structure modules. We used a generic definition of supersecondary structure elements, so-called *Smotifs*, which are composed of two flanking regular secondary structures connected by a loop, to explore the evolution and current variety of structure building blocks. Here, we discuss recent observations about the saturation of Smotif geometries in protein structures and how it opens new avenues in protein structure modeling and design. As a first application of these observations we describe our loop conformation modeling algorithm, ArchPred that takes advantage of Smotifs classification. In this application, instead of focusing on specific loop properties the method narrows down possible template conformations in other, often not homologous structures, by identifying the most likely supersecondary structure environment that cradles the loop. Beyond identifying the correct starting supersecondary structure geometry, it takes into account information of fit of anchor residues, sterical clashes, match of predicted and observed dihedral angle preferences, and local sequence signal.

Key words: Secondary structure, Supersecondary Structure, Smotif, Loop modeling, Protein structure evolution, Protein structure modeling, Protein structure design

1. Introduction

1.1. Recent Advances in the Analysis and Definition of Supersecondary Structure Motifs

Protein structures can be decomposed into folds (1) that are determined by the number, arrangement, and connectivity (topology) of secondary structure elements (2). An intermediate structural level between folds and secondary structures are the supersecondary structure elements that are composed of a number of regular secondary structure elements linked by loops (e.g., Rossmann, helix-turn-helix, four strand Greek key, β-meander motifs). Folds can be perceived as an overlapping combination of various supersecondary elements (Fig. 1). These supersecondary structure elements are sometimes shared among different folds

Alexander E. Kister (ed.), *Protein Supersecondary Structures*, Methods in Molecular Biology, vol. 932,
DOI 10.1007/978-1-62703-065-6_9,

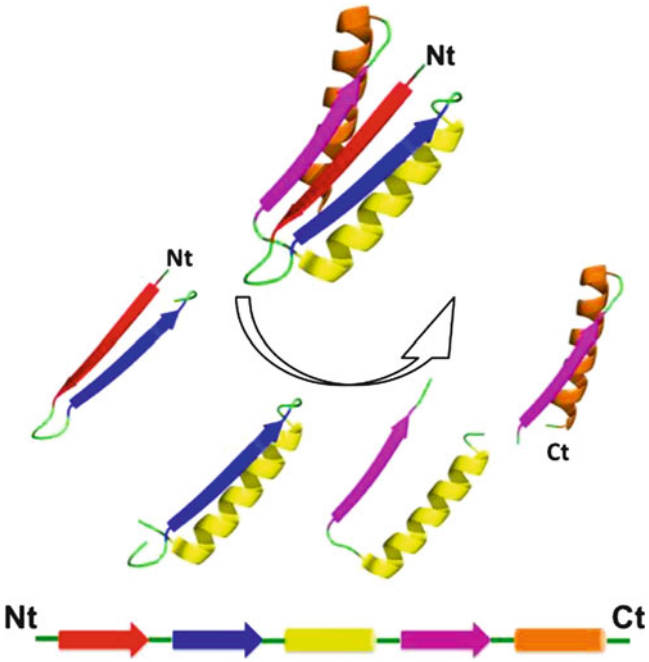


Fig. 1. A protein structure decomposed into a set of overlapping Smotifs. Secondary structure elements at the bottom are shown in *cylinders* and *arrows* representing α-helices and β-strands, respectively.

and sometimes highly repeated within the same one. Some of these supersecondary structure elements are frequently observed because these are either connected to a general functional role (e.g., DNA or cofactor binding) or they present a thermodynamically advantageous structural arrangement, which might have emerged several times during evolution. The latter observation prompted the theory of relic peptide world (3), which proposes that modern, thermodynamically stable proteins are the results of duplication, mutation, shuffling and fusion of a limited set of relic peptides.

Several reports in the recent literature have explored the idea of describing protein structures as superstructures formed by the assembly of a limited number of supersecondary elements. A recent update on the idea of ancient relic peptides found that protein fragments of up-to 20–40 residues long can co-occur in different structural contexts suggesting that these to be an ancestral pool of peptide modules (4). Various efforts tried to explore possible toolsets of supersecondary elements, such as antiparallel β-sheets (5), αββ and ββα motifs (6), αα-turn motifs (7), or four helix bundles (8). Other studies identified similar structural elements as possible building blocks of structural hierarchy using different approaches. The so-called *Closed Loops* were identified by their close C_α–C_α contacts from solution structures and found to have a nearly standard

size (27 residues ±5) in agreement with the theoretical optimal size for loop closure derived from polymer statistics (9–11). In another approach, dynamic Monte Carlo simulations of a C_α lattice model was used to identify the nearest neighbor residues, labeled as *most interacting residues*, which serve as anchors for protein folding (12). It was found that anchor residues are conserved hydrophobic clusters that keep together the so-called *Tightened End Fragments*, which essentially correspond to the Closed Loop definition. A related, more generic analysis identified fragments that are shared by different folds using an arbitrary length of 5, 10, 15, 20 residues (13) and those were used to establish structural and functional relationships among folds. Voigt et al. introduced peptide "schemas" as subunits of the protein structure that interact the least with their environment. It was assumed that these schemas can be easily swapped among proteins, as these will have the least impact in disturbing the interaction network within a fold (14). Peptide schemas are composed of 20–30 residues and typically are bundled α-helices, α-helices combined with β-strands and β-strands connected by a hairpin turn. The computationally predicted schemas were explored experimentally in five proteins and were found to be viable building units for recombinant hybrid proteins. Nussinov and coworkers defined possible building blocks to better explain the hierarchical nature of folding process (15, 16). The building blocks were determined computationally by a stepwise dissection procedures and a scoring function that was assessing compactness, isolation, and hydrophobicity. Building blocks were found sometimes to be as small as a single secondary structure or more complex supersecondary structures and in general these were assumed to be the most highly populated conformations in solution.

2. Materials

2.1. Smotif, a General Supersecondary Structure Building Block

In order to systematically explore modularity in all known protein structures, we used a general, supersecondary structure classification (17). In this approach a basic supersecondary motif, which we refer to as *Smotif,* is composed of two regular secondary structure elements linked by a loop. These motifs may or may not serve as possible units for structural evolution; however, the definition carries the advantage that one can employ automated algorithms to systematically explore the shape and occurrence of these motifs in all known folds (Fig.1).

Smotifs are characterized in protein structures by the types of sequential secondary structures and the geometry of the orientation of the secondary structures with respect to each other, as described by four internal coordinates, introduced by Oliva et al. (17, 18).

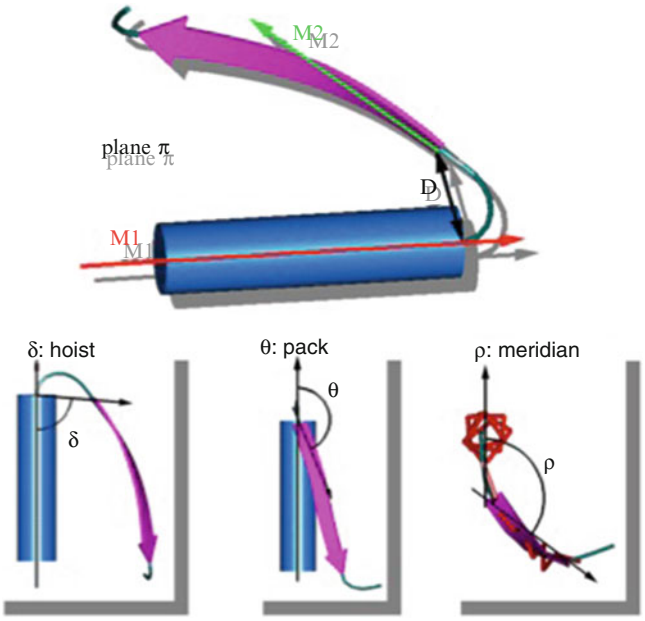


Fig. 2. Definition of the geometry of Smotifs. The geometry of each Smotif is defined by four internal coordinates: a distance, *D* and three angles: hoist (δ), pack (θ), and meridian (ρ). *D* is the distance between the last and first Cα of the Nt and Ct secondary structure respectively. M1 and M2 are defined by the shortest of the principal moments of inertia of the flanking secondary structures: δ, the angle between axis M1 and vector of *D*; θ, the angle the angle between M1 and M2; and ρ, the angle between M2 and the plane that contains the vector M1.

Therefore, a protein structure can be expressed as a string of overlapping Smotifs. If we used a definition that has higher number of connected secondary structures, e.g., 3 or more, the number of combinations would be very large and would prevent us from a systematic classification. Meanwhile, any combination of Smotifs is possible, both sequentially and spatially, which can recapitulate earlier described, more complicated supersecondary structure motifs.

The geometry of Smotifs captures the local structural arrangement of the two flanking secondary structures SS1 and SS2 and is defined by four internal coordinates: a distance and three angles (Fig. 2) (17, 18). *D* is a distance, expressed in Angstroms, and the typical *D* values range from 2 and 16 Angstroms. As expected, *D* mainly depends on the number of residues contained in the loop region. The hoist (δ), pack (θ), and rho (ρ) angles range from 0 to 180, 180, and 360°, respectively.

The geometrical values are distributed in a continuous space, e.g., the hoist angle ranges between 0 and 180°. In order to compare Smotif geometries, the parameter space of geometrical values were binned, where the four internal coordinates define each bin. A range of binning sizes and parameter intervals were explored for

the four variables in order to get the sharpest partitioning of the geometrical space with the smallest number of possible bins. The quality of the binning was assessed by calculating the Root Mean Square Deviation (RMSD) and the LGA scores (19) upon structural superposition for all Smotifs that were classified in the same or different geometrical bins. The optimal bin partitioning for each parameter was obtained by studying the distribution of distance and angle values of Smotifs in SCOP 1.71 database proteins (20) and resulted in only 324 types of Smotif definitions using the following binning values: 4 Å bins for distance, 60° bins for δ and θ starting at 0°, and 60° bins for ρ, starting at 30°. At this level of bin resolution the RMSD upon structural superposition of more than 75% of Smotifs that belong to the same geometrical bin falls below 1 Å.

Smotifs extracted from protein structures are organized in a library using a two-level hierarchical classification. In the first hierarchical level, Smotif are grouped by type, i.e., Smotifs are identified according to the type of bracing secondary structures: αααββα and ββ according to the definition of secondary structure by the DSSP program (21). In the second hierarchical level, Smotifs are grouped according to their geometry bin, as described above. We have used this library of Smotifs to explore the modular nature of fold organization, both in terms over evolutionary time and the current fold space varieties. Subsequently, we utilized the spatial restraints enforced by the bracing secondary structures within a Smotif we set up a prediction algorithm to model the conformation of the connecting loop segments.

3. Methods

3.1. Saturation of Smotifs in Known Folds

As we described above, folds can be dissected into their Smotif building blocks, and one can express protein folds as a unique string of overlapping Smotifs (Fig. 1). The frequency of occurrence of Smotifs in all known protein folds was explored, and in addition we studied the increase of coverage of Smotifs in Protein Data Bank (PDB) over time (Fig. 3). Briefly, all protein structures that were identified as "new folds" from SCOP (20) releases 1.73 and 1.75 and CASP 3–6 meetings (22) were decomposed into Smotifs and these were compared to libraries of Smotifs extracted from backdated versions of the PDB. Within the pairs of datasets we evaluated the existence of identical Smotifs in the novel folds and the previously defined folds. The first comparison was based on the type of secondary structures and the geometry (*D*, hoist, packing, and meridian) of Smotifs. In a second, stricter comparison, the lengths of the flanking secondary elements (SS1 and SS2) were also compared. If these lengths differed by more than 2 or 4

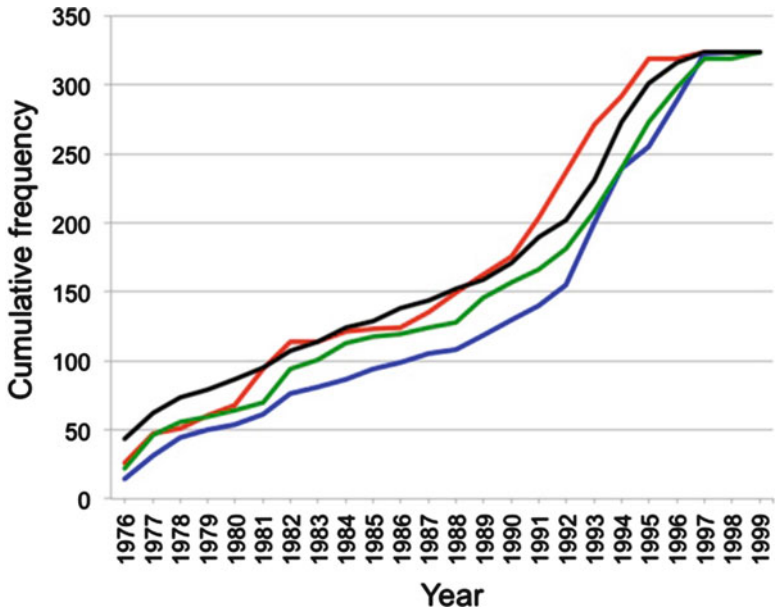


Fig. 3. Emergence of new Smotif geometries in PDB. The plot shows the cumulative frequency distribution as a function of time for α–α (*red*), α–β (*blue*), β–α (*green*), and β–β (*black*) Smotifs.

residues in the case of strands or helices, respectively, the Smotifs were considered different.

We found that proteins that were considered novel folds at CASP 3–6 meetings (years 1998–2004) and in SCOP 1.73, 1.75 (years 2007–2009) do not have any novel Smotif geometries that were not present in previously solved structures. In other words, none of the Smotifs of novel folds have a unique geometry. For instance, as early as the third round of CASP Meetings in 1998, all of the targets identified as novel folds by the experts could have been reconstructed using Smotifs from known protein structures. In an even stricter comparison, i.e., requiring not just a match in the geometry but also identical lengths of the flanking secondary structures, still less than 6% of the Smotifs in novel folds at CASP meetings would not have a match in already known structures. Similarly, we checked the motif composition of new folds from the archives of SCOP in the 1.73 (2007 November) and 1.75 (2009 June) releases. These contain a total of 233 new folds from 1,140 proteins. Similar to the CASP targets, none of these novel folds had a Smotif whose geometry was not already observed in a previously known fold and only 1% of Smotifs has a flanking secondary structure with unique length. Initially, we found 47 Smotifs (out of the 8,056 analyzed) that appeared to be new. However, after manual inspection, it turned out that these were artifacts originating from replacing obsolete PDB entries with newer ones (and thus have a newer deposition date assigned).

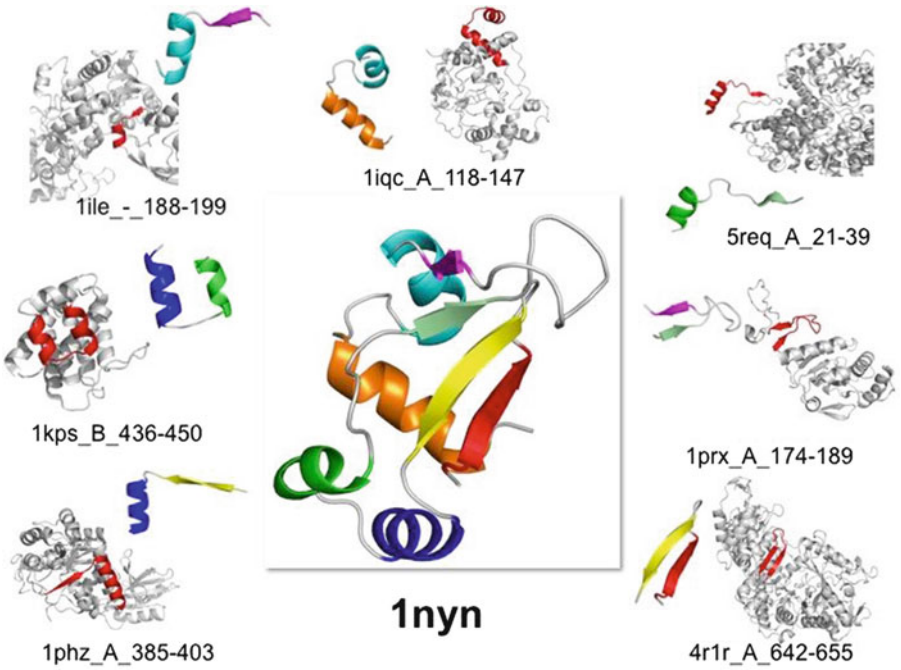


Fig. 4. Reconstruction of protein YHR087W (PDB code 1nyn) from *Saccharomyces cerevisiae* using Smotifs from unrelated structures. 1nyn was submitted to CASP5 meeting in the category of "*New Folds*"; however, all the Smotifs were available from structures that were solved before.

The above observations suggest that recently solved new folds do not imply the emergence of new Smotifs, and that a protein with a novel fold can be reconstructed using Smotifs from already existing protein folds. Approximately 10 years ago all categories of Smotifs were already represented by at least one example in a known fold. For instance, T0181 (PDB code: 1nyn), a new fold submitted to CASP5, at the time could have be constructed from seven overlapping Smotifs, all of which can be located in previously solved structures of other proteins representing a variety of different folds (Fig. 4).

3.2. What Makes a Protein Fold New?

Since the repertoire of Smotifs seems to have come close to saturation (Fig. 3) (23), this prompts the question of what is really unique about a fold structure when it is identified as "novel". When we explored the frequency of occurrence of Smotifs in the nonredundant set of known folds, we observed that novel folds have a larger fraction of Smotifs that have a low frequency of occurrence in the PDB. On the other hand, superfolds (24), those that are adopted by many different sequences (TIM barrels, OB fold, IG superfamily, etc.) often with different functions, are built by Smotifs that occur with medium or high frequencies in existing folds. This implies that novel folds are composed of a new permutation of

existing Smotifs and, specifically, a structure will have a greater likelihood of being "novel" if the structure is enriched with rarely occurring Smotifs.

Another plausible explanation would be to combine, otherwise common Smotifs, in an unusual sequence, and thus resulting in a new topology. To explore this, we calculated a Novelty Z-score for each protein (25), which was obtained from the product of individual Smotif frequencies. The hypothesis is that if the Novelty Z-score of some novel folds is similar to that of known folds, then the novelty for these cases must be a consequence of a never before seen combination of otherwise common Smotifs rather than a result of being constructed from rare Smotifs. Most novel folds are indistinguishable from already known structures in terms of their overall Novelty Z-scores, which indicates that these structures may indeed be just a new topological arrangement of common Smotifs. This means that although novel folds are often built using a higher proportion of rare Smotifs, in many cases these folds are novel because their Smotifs are assembled in an unusual sequence.

These observations open up new venues for structure modeling and design. Fragment-based approaches in structure prediction are becoming the most successful and preferred approaches, especially in the case of new folds (26, 27). These approaches rely on a library of short fragments, which are not necessarily biologically meaningful (usually too short). Using Smotifs for this purpose would drastically decrease the degree of freedom that needs to be explored in the sampling procedure. The remaining bottleneck in all these approaches is to establish a detectable sequence signal, which can relate the sequence of Smotifs to their conformation. Another venue where Smotifs may prove to be useful is protein design. If one assumes that all building blocks of known and to be discovered folds are already known and present in the current Smotif library, then the remaining challenge is to identify those new combinations of Smotifs which can result in a stable fold for a protein.

3.3. Saturation of Loops in Databanks

We also approached the previous question on completeness from the perspective of loop fragments, based on the hypothesis that similar structural environments within a Smotif probably also enforce similar connecting segments, i.e., loop conformations. We explored the fraction of loops that is extracted from all known protein sequences, called the *Sequence Space*, that are covered by loops extracted from all known protein structures, called the *Structure Space*. This will estimate the current structural coverage of short segments in the Sequence Space, i.e., in the entire set of known sequences.

Smotifs from Structure Space were structurally clustered after an all-to-all comparison and sequence identity cutoffs that ensured structural similarities were identified for each loop length. In the

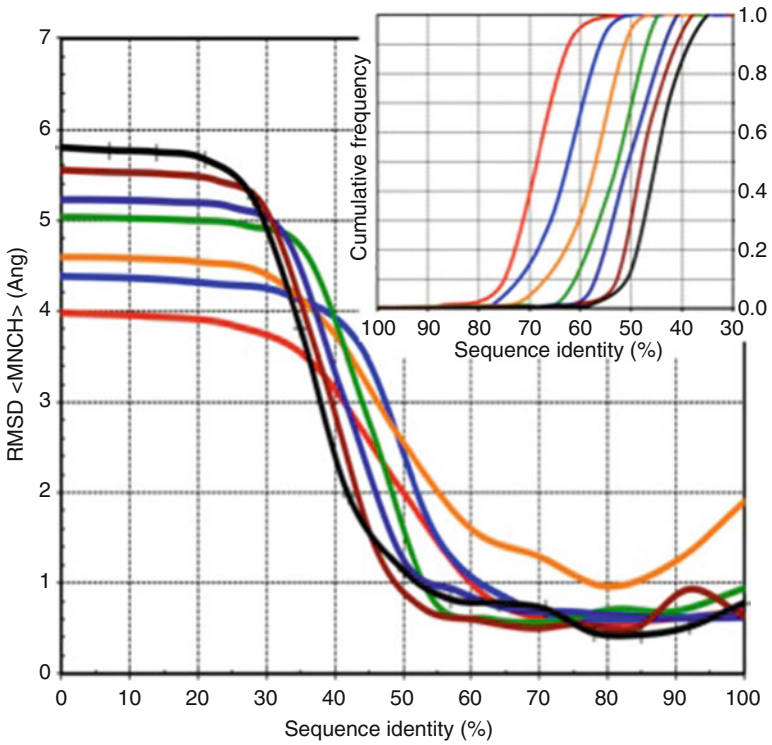


Fig. 5. Saturation of loop conformations in databases. The plot shows the relationship between structure similarity (*y*-axis, measured as the RMSD of main chain atoms) as a function of sequence identity (*x*-axis) for loops of length 8, 9, 10, 11, 12, 13, and 14 shown in *red, blue, orange, green, dark blue, brown,* and *black*, respectively. *Inset* shows the cumulative frequency distribution of loops that can be matched up between all known sequences (sequence space) and the available structural conformations (structure space) at a given sequence identity.

range of 42–55% pair-wise sequence identity we found a sharp transition between high to low RMSD values at all lengths (4–14 residue long loops were investigated), suggesting that a 50% sequence identity generally guarantees structural similarity (Fig. 5). Next, all possible loops from the Sequence Space were matched with the sequences from Structure Space, and the coverage assessed. In year 2005, below 40% sequence identity only 20 and 10% of long loops (of length 13 and 14) from Sequence Space could not be matched to at least one loop from Structure Space, while all other loop lengths matched at 100%. Meanwhile all loops (100%) of length 8 from Sequence Space 2005 have at least one loop in Structure Space at 50% or larger sequence identity. The percentages of coverage at 50% sequence cutoff dropped to 96, 94, 68, 53, 33, and 13% for loops of length 9, 10, 11, 12, 13, and 14, respectively. There were no loops at any length up to 14 residues that did not match with a conformational segment that shared at least 20% of sequence identity (Fig. 5 inset).

Finally, we also investigated the growth and change in the databases by repeating these exhaustive comparisons between

sequence and structure databases that were available in 2001 and later. We focused our analysis on "medium" and "long" loops that were in the range of 8–14 residues. Following the same approach described above, we compared Sequence Space 2005 against Structure Space 2001 and we found that while sequence databases kept growing at an exponential rate, there were almost no unique conformational segments deposited up to 12 residues long fragments during the last 5 years.

3.4. Smotifs and Loop Modeling

In the absence of an experimentally described structure, computational methods such as comparative modeling, threading or ab initio can be used to provide a useful 3D model and fill the gap between the number of sequences and structures. Among these, comparative modeling is currently the most accurate but it is applicable only for that part of the target protein where a suitable template is found (28, 29). Most notably, insertions in the target sequence or segments with very different sequences to the template are not possible to model with this technique. Even above 40% sequence identity, where the core of the fold is well preserved and can be aligned accurately, the surface exposed variable loops can vary substantially among the homologues. Meanwhile loops often represent an important part of the protein structure, and functional differences between the members of the same protein family and especially among members of superfamilies are usually a consequence of structural differences of exposed loop regions (30). Active and binding site residues are more likely to be found in loop regions (31). Text book examples of functionally important loops include antibody complementary determining regions (32), ligand binding sites (ATP (33), calcium binding sites (34), NAD(P) (35)), DNA binding (36) or enzyme active sites (e.g., Ser-Thr kinases (37), or serine proteases (38)). Therefore, the accuracy of loop conformations is often a key determinant of the usefulness of computational or experimental models.

Many loop-modeling procedures have been described in the past (39). Similarly to the prediction of protein structures there are ab initio (conformational search) methods (40–42), and database search (or knowledge-based) methods (43–45). There are also hybrid procedures that combine the two (7, 46, 47). In ab initio prediction a conformational search or enumeration of conformations is conducted in a given environment, guided by a scoring or energy function (41, 42). There are many such methods, exploiting different protein representations, sampling methods, energy function terms and optimization or enumeration algorithm (39). Knowledge-based methods (48), (also known as database search methods) are essentially a loop modeling application of comparative modeling that rely on a library of fragments as templates extracted from known protein structures and on a selection algorithm. Usually, many different alternative segments are obtained,

and then sorted according to a variety of criteria, such as geometrical fit or sequence similarity between the template and target loop sequences. We took advantage of the geometrical restraints that are imposed by Smotifs to drive the selection of fragments in loop modeling.

3.5. Overview of the ArchPRED Loop Modeling Method

ArchPRED is a knowledge-based approach to loop modeling that relies on a library of Smotifs and an algorithm containing selection, filtering and ranking steps. In a first step, Smotifs are selected from the library using the geometrical restraints imposed by the bracing secondary structures of the missing loop. Loop length and the type of flanking secondary structure (e.g., α–β) are also used during the selection step. In the second step, at the filtering stage, loops are discarded if the RMSD of their stem residues and the interactions between candidate loops and the new protein environment are unfavorable (i.e., sterical clashes). Finally in the third step, at the ranking stage, the remaining candidate loops are sorted by a composite Z-score. The Z-score function combines a sequence score using a conformational similarity weight matrix, and a Φ/Ψ main chain dihedral angle propensities score (Fig. 6).

The Smotif library currently classifies 430,000 high quality loop structures. Smotif Library is organized in a three level hierarchy: loops are identified and grouped according to (1) the type of

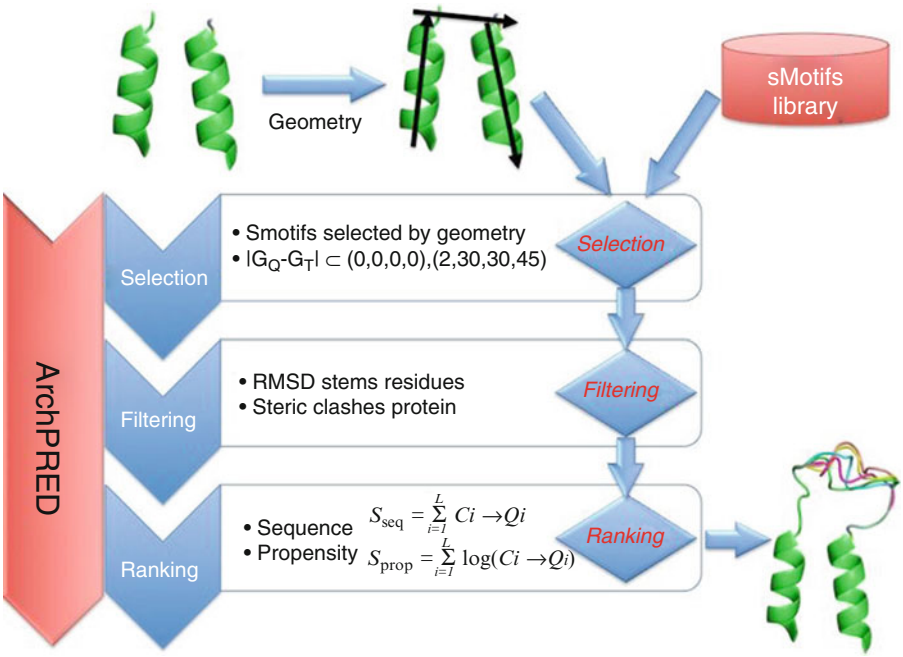


Fig. 6. Overview of ArchPRED algorithm. The prediction algorithm is composed of two major components: (1) a library of Smotifs and (2) a selection, filtering and scoring algorithm. Smotifs that fit the geometry of the query loop are fitted in the new protein and filtered by RMSD of stem residues and steric clashes and finally scored by a composite Z-scores that combines sequence and propensity information.

bracing secondary structures, (2) the length of the loop region, and (3) four internal coordinates of the bracing secondary structures as defined by a distance vector between the anchor points and three angles: hoist (δ), packing (θ), and meridian (ρ) (18), which bins loops into $20 \times 6 \times 6 \times 8 = 5{,}760$ possible cells or geometrical combinations. Not all cells are equally populated, e.g., short loops cannot have large distance values or ββ-hairpin loops have a restricted geometry in terms of possible angle combinations due to strict hydrogen bond requirements that defines them (49).

3.6. Selecting, Filtering, Ranking of Candidate Loops in ArchPRED

The search algorithm is composed of three consecutive steps: (1) selection, (2) filtering, and (3) ranking of the suitable segments from the Smotif library. During the (1) selection step, loops queried in a stepwise manner: first, by similar bracing secondary structures, and those having a similar length (±1 residue) to the query loop. The next selection step in the lookup process involves comparing the four internal coordinates of Smotifs. This initial selection of candidate loops by geometrical requirements quickly narrows the space to be explored by subsequent, more elaborate structural comparisons. For instance, for loops of lengths 4, 8, and 12, the average number of selected loops by stem residue distances comparison on 50 randomly chosen examples (with a tolerance of 1 Å) is 1,534, 683, and 430; while the selected number of loops after geometrical comparison is only 181, 85, and 25, respectively. In the meantime, this filtering step does not eliminate good candidate loops. Comparing the average $RMSD_{local}$ of the best fragment between loops that are selected by end point distances and loops selected by geometry, the differences are less than 0.05, 0.09, and 0.11 Å for the test sets of 4, 8, and 12 residue long loops, respectively.

After the initial selection, a two-stage (2) filtering step follows, which checks for the fit of stem residues by superposition of main chain atoms and RMSD calculation and evaluates of steric clashes between the loop and the rest of the protein environment. RMSD cutoffs for superposed stem residues have been applied before in loop structure prediction method either for ranking (50) or filtering (51). RMSD fit of stem residues correlate strongly with the accuracy of prediction of short loops, but this correlation becomes less pronounced for longer loops. The reason is that longer loops (8–14 residues) have more flexibility and their conformations are less restricted by the stem residues than in case of short loops (1–7 residues). Therefore, we applied a range of RMSD stem cutoff values as a function of loop length. The second descriptor to filtering of loops explores the conformational fit of candidate loops in the new protein environment. Each candidate loop is structurally fitted in the new protein environment of the query protein and the steric hindrances between the loop and its structural environment are

assessed. After these steps the average number of loop candidates in the test sets decreased to 81, 35, and 5 for loops of length 4, 8, and 12, respectively.

In the last, third step 3, the remaining candidate loops are ranked according to sequence similarity and amino acid $\tilde{\Phi}\Psi$ dihedral angle propensities. Sequence and propensity scores have their own range and correlation with prediction accuracy; therefore, these scores were converted into Z-scores in order to unify both scores with a comparable and dimensionless criterion. Sequence Z-score gauges the similarity between the sequence of the query and candidate loops and compares it to a reference distribution of randomly selected pairs of loops with similar lengths. A number of different substitution matrices were tested to score sequence similarity and the K3 weight matrix proved to be the most efficient (52), as it was derived from comparisons of Ramachandran maps and was developed to select protein fragments with similar conformations. The second quantitative measure to rank the set of candidate loops is the propensity of amino acids to adopt a specific $\tilde{\Phi}\Psi$ main chain dihedral angle conformation. Propensity is defined as the likelihood that an amino acid residue is found in specific backbone dihedral angles Φ and Ψ. The expected propensity values were obtained from a table that divides the Ramachandran plot into 15 different regions ("p15 propensity" table) (53). The logarithm of the propensity approximates the free energy of a specific residue conformation. The free energy for each position is assumed to be additive, so the score for a sequence fragment is the sum of the log of the propensities at each position (53). The composite Z-score is defined as the sum of the two types of Z-scores.

3.7. Performance of ArchPRED

Benchmarking loop prediction approaches using knowledge-based methods is not straightforward. Some sort of artificially filtered input library needs to be prepared to avoid trivial hits and consequently the overestimation of performance. However, if one overly ambitious in getting rid of all segments in a database that show any level and type of similarity to a query may end up with seriously underestimated method performance. For benchmarking purposed a filtered library was derived by removing any Smotif extracted from protein that share the same SCOP (20) superfamily with the query loop. The performance was compared to a competitive and freely available ab initio prediction method: ModLoop (54) and with the theoretical minimum RMSD, i.e., selecting the best candidate in the library, and thus informing on the practical limits of the method.

The minimum value of RMSD that can be obtained with loops available in the Smotif library (i.e., the loop with the smallest RMSD) are on average 0.25, 0.5, and 1 Å more accurate (for 4, 8, and 12 residues long loops, respectively) than the best results

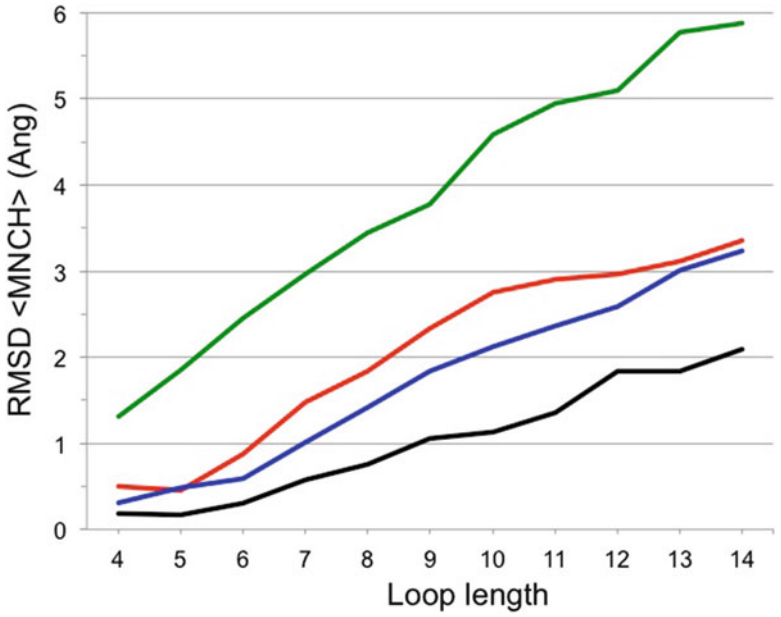


Fig. 7. Prediction accuracy (average RMSD of mainchain atoms in angstrom) of ArchPRED, Modloop and theoretical limit of prediction as a function of loop length. *Black* line represents the practical upper limit of the prediction accuracy, i.e., selecting the best available Smotif in the library. *Red* and *blue* represent ArchPRED and ModLoop prediction respectively and *solid green line* shows the average RMSD for random prediction, i.e., a random selection of Smotifs from the library.

obtained by ModLoop, thus indicating that there are good candidate loops in the library and for all loop lengths (Fig. 7). This observation agrees with our analysis in database completeness (see Subheading 3.3) and also agrees with the conclusions by Du et al. (55) and Choi et al. (56), who found that even for long loops there are suitable candidates in the current database of structures with and RMSD of 2 Å or less. Therefore, the bottleneck in knowledge-based loop modeling does not appear to be the sampling (completeness of database segments), but the search algorithm and scoring function to locate these segments. ModLoop on average outperformed ArchPRED for all loop lengths if we force the current search algorithm to locate a segment for all possible queries even if these are not very good candidates (i.e., falling below a certain cutoff Z-score). However, averages of both methods fall within the boundaries of 1 Å standard deviation. Finally, the accuracy obtained with ArchPRED is clearly much higher than the accuracy obtained with a random prediction (Fig. 7).

3.8. Coverage Versus Accuracy: Identifying Confidence Z-score Thresholds

It is important to assign confidence values to a prediction and for that we explored the performance of the method as a function of Z-scores. Z-score cutoffs were defined in such a way that Smotifs selected with more significant Z-scores will have equal or better accuracy than the average accuracy of fragments obtained by

Table 1
Accuracy of prediction and coverage for different loop lengths

Loop length	Zscore[a]	Average RMSD[b] (Å)	Coverage[c] (%)
4	1	0.22	98
5	1	0.15	96
6	1	0.34	98
7	1	0.93	94
8	2	1.38	78
9	3	1.93	60
10	3	2.11	46
11	3	2.30	44
12	4	2.47	28
13	4	2.85	4
14	4	2.88	6

[a]Z-score cutoff to ensure higher accuracy than what is expected from ModLoop
[b]Average RMSD of predicted loops for the given Z-score cutoff
[c]Percentage of query loops that are modelable (i.e., a suitable Smotif can be found) at the give Z-score threshold

ModLoop As expected, the RMSD values decrease as the composite Z-scores and the accuracy of predictions increase; however, the corresponding coverage of the prediction decreases. For loops between lengths of 4–7 residues a Z-score of 1.0 gives an equal or better performance than ModLoop with a corresponding coverage of around 90%. In the case of loops between 8 and 11 residues a Z-score larger than 2–3 is required and the average coverage is around 50–60%. For longer loops, beyond 12 residues long the coverage rapidly drops (Table 1).

3.9. ArchPRED Web Server

ArchPRED is implemented as Web server for the modeling of missing protein loops in protein structures. Users are required to provide the query structure in PDB format and define the sequential location of the missing loop. The user can also select whether to query the Smotif database by geometry or by Euclidian distance of the stem residues. If geometry is selected, then the type of bracing regular secondary structural elements (e.g., α–β) have to be defined and if these elements are beta strands than further distinguish between hairpin or link types. Once the prediction is completed, results are sent by email in form of a link pointing to a temporary Web page. Optionally, users can select to rebuild the side chains of predicted loops and to perform a limited minimization (conjugate

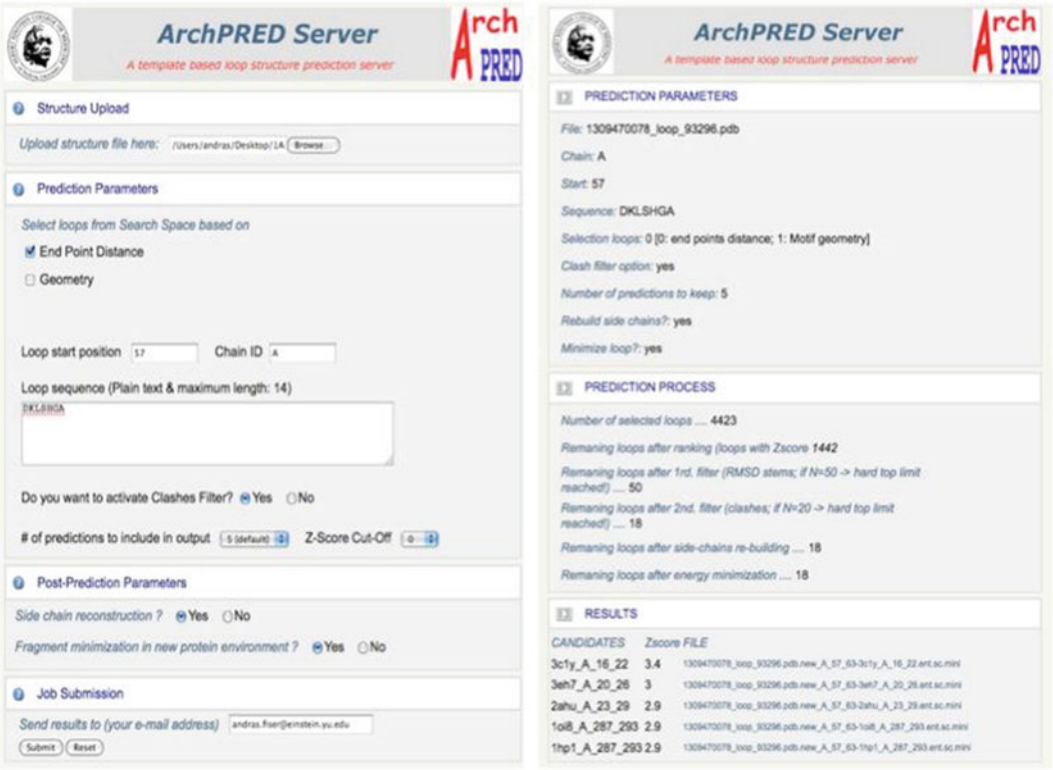


Fig. 8. Snapshots of the Archpred Web server. *Left panel*, submission page, *right panel*, results page.

gradient minimization) to anneal the stems in the protein framework. The server is accessible at http://www.fiserlab.org/servers/ArchPRED (Fig. 8).

Acknowledgment

This work was supported by NIH grant R01GM096041. This review is partially based on our previous publications of refs. 17, 23, 25, 57. NFF acknowledges support from the Research Councils UK under the RCUK Academic Fellowship scheme.

References

1. Murzin AG, Brenner SE, Hubbard T et al (1995) SCOP: a structural classification of proteins database for the investigation of sequences and structures. J Mol Biol 247:536
2. Hadley C, Jones DT (1999) A systematic comparison of protein structure classifications: SCOP, CATH and FSSP. Struct Fold Des 7:1099
3. Lupas AN, Ponting CP, Russell RB (2001) On the evolution of protein folds: are similar motifs in different protein folds the result of convergence, insertion, or relics of an ancient peptide world? J Struct Biol 134:191
4. Alva V, Remmert M, Biegert A et al (2010) A galaxy of folds. Protein Sci 19:124–130

5. Holm L, Sander C (1993) Protein structure comparison by alignment of distance matrices. J Mol Biol 233:123
6. Boutonnet NS, Kajava AV, Rooman MJ (1998) Structural classification of alphabetabeta and betabetaalpha supersecondary structure units in proteins. Proteins 30:193–212
7. Wintjens RT, Rooman MJ, Wodak SJ (1996) Automatic classification and analysis of alpha alpha-turn motifs in proteins. J Mol Biol 255:235–253
8. Presnell SR, Cohen BI, Cohen FE (1992) A segment-based approach to protein secondary structure prediction. Biochemistry 31:983
9. Berezovsky IN, Grosberg AY, Trifonov EN (2000) Closed loops of nearly standard size: common basic element of protein structure. FEBS Lett 466:283–286
10. Trifonov EN, Frenkel ZM (2009) Evolution of protein modularity. Curr Opin Struct Biol 19:335–340
11. Chintapalli SV, Yew BK, Illingworth CJ et al (2010) Closed loop folding units from structural alignments: experimental foldons revisited. J Comput Chem 31:2689–2701
12. Papandreou N, Berezovsky IN, Lopes A et al (2004) Universal positions in globular proteins. Eur J Biochem 271:4762–4768
13. Friedberg I, Godzik A (2005) Connecting the protein structure universe by using sparse recurring fragments. Structure 13:1213–1224
14. Voigt CA, Martinez C, Wang ZG et al (2002) Protein building blocks preserved by recombination. Nat Struct Biol 9:553–558
15. Tsai CJ, Maizel JV Jr, Nussinov R (2000) Anatomy of protein structures: visualizing how a one-dimensional protein chain folds into a three-dimensional shape. Proc Natl Acad Sci USA 97:12038–12043
16. Tsai CJ, Polverino de Laureto P et al (2002) Comparison of protein fragments identified by limited proteolysis and by computational cutting of proteins. Protein Sci 11:1753–1770
17. Fernandez-Fuentes N, Oliva B, Fiser A (2006) A supersecondary structure library and search algorithm for modeling loops in protein structures. Nucleic Acids Res 34:2085–2097
18. Oliva B, Bates PA, Querol E et al (1997) An automated classification of the structure of protein loops. J Mol Biol 266:814
19. Zemla A (2003) LGA: a method for finding 3D similarities in protein structures. Nucleic Acids Res 31:3370–3374
20. Andreeva A, Howorth D, Chandonia JM et al (2008) Data growth and its impact on the SCOP database: new developments. Nucleic Acids Res 36:D419–425
21. Kabsch W, Sander C (1983) Dictionary of protein secondary structure: pattern recognition of hydrogen-bonded and geometrical features. Biopolymers 22:2577–2637
22. Moult J (2005) A decade of CASP: progress, bottlenecks and prognosis in protein structure prediction. Curr Opin Struct Biol 15:285–289
23. Fernandez-Fuentes N, Fiser A (2006) Saturating representation of loop conformational fragments in structure databanks. BMC Struct Biol 6:15
24. Orengo CA, Pearl FM, Bray JE et al (1999) The CATH Database provides insights into protein structure/function relationships. Nucleic Acids Res 27:275
25. Fernandez-Fuentes N, Dybas JM, Fiser A (2010) Structural characteristics of novel protein folds. PLoS Comput Biol 6:e1000750
26. Zhang Y (2007) Template-based modeling and free modeling by I-TASSER in CASP7. Proteins 69(Suppl 8):108–117
27. Das R, Baker D (2008) Macromolecular modeling with rosetta. Annu Rev Biochem 77:363–82
28. Fiser A, Feig M, Brooks CL III, Sali A (2002) Evolution and physics in comparative protein structure modeling. Acc Chem Res 35:413–421
29. Baker D, Sali A (2001) Protein structure prediction and structural genomics. Science 294:93–96
30. Blouin C, Butt D, Roger AJ (2004) Rapid evolution in conformational space: a study of loop regions in a ubiquitous GTP binding domain. Protein Sci 13:608–616
31. Fiser A, Simon I, Barton GJ (1996) Conservation of amino acids in multiple alignments: aspartic acid has unexpected conservation. FEBS Lett 397:225
32. Kim ST, Shirai H, Nakajima N et al (1999) Enhanced conformational diversity search of CDR-H3 in antibodies: role of the first CDR-H3 residue. Proteins 37:683–696
33. Saraste M, Sibbald PR, Wittinghofer A (1990) The P-loop–a common motif in ATP- and GTP-binding proteins. Trends Biochem Sci 15:430–434
34. Kawasaki H, Kretsinger RH (1995) Calcium-binding proteins 1: EF-hands. Protein Profile 2:297–490
35. Wierenga RK, Terpstra P, Hol WG (1986) Prediction of the occurrence of the ADP-binding beta alpha beta-fold in proteins, using an amino acid sequence fingerprint. J Mol Biol 187:101–107
36. Tainer JA, Thayer MM, Cunningham RP (1995) DNA repair proteins. Curr Opin Struct Biol 5:20–26

37. Johnson LN, Lowe ED, Noble ME et al (1998) The eleventh datta lecture. The structural basis for substrate recognition and control by protein kinases. FEBS Lett 430:1–11
38. Wlodawer A, Miller M, Jaskolski M et al (1989) Conserved folding in retroviral proteases: crystal structure of a synthetic HIV-1 protease. Science 245:616–621
39. Fiser A, Do RK, Sali A (2000) Modeling of loops in protein structures. Protein Sci 9:1753
40. Fine RM, Wang H, Shenkin PS et al (1986) Predicting antibody hypervariable loop conformations. II: Minimization and molecular dynamics studies of MCPC603 from many randomly generated loop conformations. Proteins 1:342
41. Moult J, James MN (1986) An algorithm for determining the conformation of polypeptide segments in proteins by systematic search. Proteins 1:146
42. Bruccoleri RE, Karplus M (1987) Prediction of the folding of short polypeptide segments by uniform conformational sampling. Biopolymers 26:137
43. Jones TA, Thirup S (1986) Using known substructures in protein model building and crystallography. EMBO J 5:819
44. Chothia C, Lesk AM (1987) Canonical structures for the hypervariable regions of immunoglobulins. J Mol Biol 196:901
45. Fidelis K, Stern PS, Bacon D, Moult J (1994) Comparison of systematic search and database methods for constructing segments of protein structure. Protein Eng 7:953
46. Deane CM, Blundell TL (2001) CODA: a combined algorithm for predicting the structurally variable regions of protein models. Protein Sci 10:599
47. Martin AC, Cheetham JC, Rees AL (1989) Modeling antibody hypervariable loops: a combined algorithm. PNAS 86:9268–9272
48. Greer J (1981) Comparative model-building of the mammalian serine proteases. J Mol Biol 153:1027
49. Gunasekaran K, Ramakrishnan C, Balaram P (1997) Beta-hairpins in proteins revisited: lessons for de novo design. Protein Eng 10:1131–1141
50. Michalsky E, Goede A, Preissner R (2003) Loops in proteins (LIP)–a comprehensive loop database for homology modelling. Protein Eng 16:979
51. Heuser P, Wohlfahrt G, Schomburg D (2004) Efficient methods for filtering and ranking fragments for the prediction of structurally variable regions in proteins. Proteins 54:583–595
52. Kolaskar AS, Kulkarni-Kale U (1992) Sequence alignment approach to pick up conformationally similar protein fragments. J Mol Biol 223:1053–1061
53. Shortle D (2002) Composites of local structure propensities: evidence for local encoding of long-range structure. Protein Sci 11:18–26
54. Fiser A, Sali A (2003) ModLoop: automated modeling of loops in protein structures. Bioinformatics 19:2500
55. Du P, Andrec M, Levy RM (2003) Have we seen all structures corresponding to short protein fragments in the Protein Data Bank? An update. Protein Eng 16:407
56. Choi Y, Deane CM (2010) FREAD revisited: accurate loop structure prediction using a database search algorithm. Proteins 78:1431–1440
57. Fernandez-Fuentes N, Zhai J, Fiser A (2006) ArchPRED: a template based loop structure prediction server. Nucleic Acids Res 34:W173–176

Chapter 10

Residue–Residue Contacts: Application to Analysis of Secondary Structure Interactions

Vladimir Potapov, Marvin Edelman, and Vladimir Sobolev

Abstract

Protein structures and their complexes are formed and stabilized by interactions, both inside and outside of the protein. Analysis of such interactions helps in understanding different levels of structures (secondary, super-secondary, and oligomeric states). It can also assist molecular biologists in understanding structural consequences of modifying proteins and/or ligands. In this chapter, our definition of atom–atom and residue–residue contacts is described and applied to analysis of protein–protein interactions in dimeric β-sandwich proteins.

Key words: Contact surface area, Protein structure analysis, Protein structure prediction, Force field, Protein–protein interactions

1. Introduction

For most proteins, structure strictly defines function. Protein structures are formed and stabilized by interactions both inside and outside of the protein. Theoretically, the most accurate description of such interactions (particularly strong ones) can be achieved using quantum mechanics or its approximation, quantum chemistry. However, the former can only be applied to extremely simple systems of two–three particles (for example, atoms H and He, or ion H_2^+), while the latter is restricted to a few hundred atoms. Therefore, for description of protein structures and their complexes, molecular mechanics (MM) and quantum mechanics/molecular mechanics (QM/MM) empirical schemes were developed (1, 2). Different types of force fields have appeared (3–6) and are still in use (7–9). However, such approaches summarize "physical" contribution to

Alexander E. Kister (ed.), *Protein Supersecondary Structures*, Methods in Molecular Biology, vol. 932,
DOI 10.1007/978-1-62703-065-6_10,

the "energy" function and are hardly applied to qualitative analysis of interactions between different structural elements. For such analysis, researchers are considering interactions or contacts between different elements such as atoms, residues, secondary structures, super-secondary structures, and domains. For example, to define if two residues are in contact, thresholds were introduced between nearest atoms or C_α atoms. Sometimes, more complicated schemes (particularly for description of H-bonds) were used (10–13).

In this chapter, we describe our definition of atom–atom and residue–residue contacts (14, 15) and some consequences of the definition. We apply this approach for analysis of protein–protein interaction in resolved dimeric β-sandwich proteins from the Protein Data Bank (PDB (16)) to study the principles involved in stabilizing super-secondary structures formed by β-sheets.

2. Materials

Beta-sandwich proteins are a large heterogeneous group of proteins comprising 114 superfamilies in 57 protein folds (SCOP version 1.75 (17)). This group includes enzymes, transport and muscle proteins, antibodies, cell surface proteins, viral coat proteins, and many others. In spite of this diversity, all these proteins have a similar overall fold and many of them are biologically active as oligomers. The true oligomeric state of a protein is often difficult to determine, even though the correct identification of the state could be critical to an understanding of the protein's physiological function (18). Due to a similar overall fold for β-sandwich proteins, their oligomeric interfaces may share general properties. To what extent can surface properties of dimeric β-sandwich proteins be used for predicting the interface region? Two major modes of interaction of β-sandwich proteins, sheet–sheet interfaces (SSIs; Table 1) and extended sheet interfaces (ESIs; Table 2), were studied.

3. Methods

3.1. Atom–Atom and Residue–Residue Contacts

3.1.1. Contact Surface Definition

"Contact surface area" between atoms A and B is defined as the area of a sphere whose center is the center of atom A and whose radius equals the sum of the van der Waals radii of atom A and a solvent molecule. This area consists of the points where a solvent molecule, if placed there, would overlap with the van der Waals sphere of atom B (Fig. 1a). If a solvent molecule cannot be placed at some particular point because it will penetrate several

Table 1
List of proteins forming sheet–sheet interfaces

PDB ID[a]	Resolution (Å)	Domain 1[b]	Domain 2[b]	Description[c]	Interface area[d] (Å^2)
1a3q	2.10	A:227-327	B:227-327	p52 subunit of NFκB	938
1a6z	2.60	A:182-275	B	Hemochromatosis protein β2-Microglobulin	889
1bfs	1.90	A	X	p50 subunit of NFκB	956
1ddt	2.00	A:381-535	X:381-535	Diphtheria toxin	695
1dqi*	1.70	A	B	Superoxide reductase	1,656
1dqt	2.00	A	B	Immunoreceptor CTLA-4	917
1epf	1.85	A:1-97	B:98-189	Neural cell adhesion molecule	496
1f41*	1.30	A	X	Transthyretin	464
1f5w	1.70	A	B	Coxsackie virus receptor	939
1fat*	2.80	A	C	Phytohemagglutinin-L	747
1fny*	1.81	A	X	Legume lectin	902
1gzc	1.58	A	X	Legume lectin	1,008
1ic1	3.00	A:1-82	X:1-82	Intercellular adhesion molecule-1	604
1imh	2.86	C:368-468	D:368-468	T-cell transcription factor NFAT5	758
1k5n	1.09	A:182-276	B	Class I MHC β2-Microglobulin	780
1mvq*	1.77	A	X	Legume lectin	1,403
1my7	1.49	A	B	p65 subunit of NFκB	902
1nez	2.10	G	H	CD8	1,387
1nls*	0.94	A	X	Concanavalin A	1,399
1nqd	1.65	A	B	Class 1 collagenase	976
1oga	1.40	D:3-117	E:5-118	T-cell antigen receptor (V_α domain) T-cell antigen receptor (V_β domain)	1,074
1onq	2.15	A:184-280	B	CD1 β2-Microglobulin	870
1ous*	1.20	A	B	Fucose-binding lectin II	1,709
1pqz	2.10	A:144-242	B	Immunomodulatory protein β2-Microglobulin	689
1py9	1.80	A	X	Myelin oligodendrocyte glycoprotein	1,174
1qr4	2.55	A:88-175	X:88-175	Tenascin	685

(continued)

Table 1 (continued)

PDB ID[a]	Resolution (Å)	Domain 1[b]	Domain 2[b]	Description[c]	Interface area[d] (Å^2)
1r3h	2.50	A:1181-1276	B	Class I MHC homolog β2-Microglobulin	620
1spp	2.40	A	B	Spermadhesin PSP-I Spermadhesion PSP-II	959
1uqx*	1.70	A	X	Mannose-specific lectin RS-IIL	1,720
1uvq	1.80	A:85-183	B:95-191	Class II MHC α-chain Class II MHC β-chain	492
2bb2	2.10	A:2-85	X:86-175	Beta-Crystallin	1,125
3fru	2.20	A:179-269	B	Fc (IgG) receptor β2-Microglobulin	859

[a]Tetrameric structures are starred
[b]The full chain makes up the β-sandwich domain unless otherwise indicated by the residue numbering. Chains marked X were obtained by applying crystal symmetry operations
[c]Heterodimeric structures include a description of both β-sandwich domains
[d]The interface area was calculated as the sum of the atom–atom contacts, as defined by CSU software

neighboring atoms, it was postulated that this point belongs to the contact area of atom A and the nearest of these neighboring atoms (Fig. 1b). We know that complexes are stabilized upon formation of hydrophobic contacts (hydrophobic–hydrophobic, hydrophobic–aromatic, and aromatic–aromatic) and hydrogen bonds (including weak hydrophilic–aromatic). A procedure was introduced, based on contacting atom types, to estimate the "legitimacy" of an atom–atom contact. Legitimacy depends on the physicochemical nature of contacting atoms (termed "complementarity"). For this, we divided atom types into 8 classes: I. hydrophilic N and O that can donate and accept hydrogen bonds; II. acceptor N or O that can only accept a hydrogen bond; III. donor N that can only donate a hydrogen bond; IV. hydrophobic, Cl, Br, I, and all C atoms that are not in aromatic rings and do not have a covalent bond to hydrophilic, donor or acceptor atoms; V. aromatic C atoms in aromatic rings; VI. neutral C atoms that have a covalent bond to at least one hydrophilic atom or two or more acceptor or donor ones; VII. neutral–donor C atoms that have a covalent bond with only one donor atom; VIII. neutral–acceptor C atoms that have a covalent bond with only one acceptor atom (Table 3).

Table 2
List of proteins forming extended β-sheet interface[a]

PDB ID	Resolution (Å)	Domain 1	Domain 2	Description	Interface area (Å²)
1c1l	1.50	A	X	Congerin I	1,763
1cf1	2.80	A:10-182	D:183-386	Arrestin	1,119
1d2s	1.55	A	X	Sex hormone-binding globulin	959
1dhk	1.85	B	X	Phytohemagglutinin-L	1,048
1dqi*	1.70	A	C	Superoxide reductase	1,666
1f86*	1.10	A	B	Transthyretin	1,254
1gzw	1.70	A	B	Galectin-1	712
1hlc	2.90	A	B	S-lac lectin	689
1is3	1.45	A	X	Congerin II	976
1jk6	2.40	A	C	Neurophysin II	757
1k2f	2.60	A	B	SIAH, seven in absentia homolog	1,366
1kzq	1.70	A:3-131	B:3-131	Major surface antigen p30	791
1mvq*	1.77	A	X	Legume lectin	1,358
1nls*	0.94	A	X	Concanavalin A	1,475
1ous*	1.20	A	D:183-386	Fucose-binding lectin II	574
1p53	3.06	A:283-366	B:283-366	Intercellular cell adhesion molecule-1	1,267
1pzs	1.63	A	X	Cu,Zn superoxide dismutase	2,263
1qfh	2.20	A:750-857	B:750-857	F-actin cross-linking gelation factor	2,244
1sfp	1.90	A	X	Acidic seminal fluid protein	650
1ukg	1.70	A	B	Legume lectin	943
1uqx*	1.70	A	X	Mannose-specific lectin RS-IIL	528

[a]Legend as in Table 1

3.1.2. H-Bonds and Extended Distance Between Contacting Atoms

Tools were developed to analyze ligand–protein contacts (LPC software) and contacts of structural units (CSU software) such as helices, sheets, strands, and residues (15). An LPC/CSU contact is listed as a putative hydrogen bond if it is formed by two atoms of class I, or an atom of class I with one of classes II or III, or between two atoms of class II and III. Our list of putative H-bonds is very similar to the list obtained by HBPLUS software (10), which takes into account distances and angles within a triangle of atoms

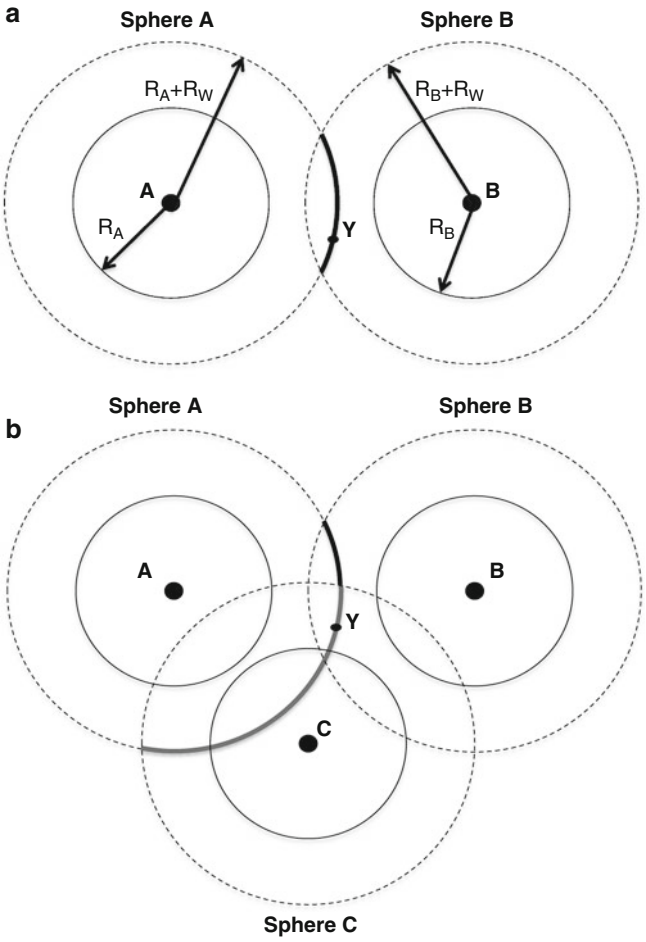


Fig. 1. Definition of atomic contact surfaces (*adapted from* [14]). Contact surface area of atom A. R_A, R_B, and R_W are van der Waals radii of atom A, B, and the solvent molecule, respectively. In (**a**) *black arc* of sphere A shows the contact surface area with atom B. In (**b**) if a solvent molecule at point Y of sphere *A* is so situated as to penetrate several atoms, we postulate that this point will be in surface contact only with the atom which is nearest to atom A. Therefore, in spite of the fact that the distance between atoms A and B is the same in (**a**) and (**b**), in (**a**), point Y is considered to be in surface contact with atom B, while in (**b**), it is considered to be in surface contact with atom C. Therefore, in (**b**), *black arc* shows the surface contact area with atom B, and *gray* with atom C.

forming H-bonds (including hydrogen atom). However, LPC/CSU software includes contacts with extended distances, up to ~6 Å. Let us consider two atoms in solution approaching each other. Contact surface between them appears at distance R, where $R = R_A + R_B + 2R_W$, and R_A and R_B are radii of atoms A and B, and R_W is the radius of a water molecule. For two oxygen atoms, contact initiates at distance $2R_O + 2R_W$, where R_O is the van der Waals radius of an oxygen atom (1.5 Å), and R_W is the radius of a water molecule (1.4 Å). So, the contacting distance is 5.8 Å!

Table 3
Legitimacy of contacts between atoms of different classes[a] (adapted from (19))

Atom class		I	II	III	IV	V	VI	VII	VIII
I	Hydrophilic	+	+	+	–	+	+	+	+
II	Acceptor	+	–	+	–	+	+	+	–
III	Donor	+	+	–	–	+	+	–	+
IV	Hydrophobic	–	–	–	+	+	+	+	+
V	Aromatic	+	+	+	+	+	+	+	+
VI	Neutral	+	+	+	+	+	+	+	+
VII	Neutral-donor	+	+	–	+	+	+	–	+
VIII	Neutral-acceptor	+	–	+	+	+	+	+	–

[a]Legitimate (+) or illegitimate (–) contact

However, contact surface depends also on environment (Fig. 1). Contact will not exist at such an extended distance if the region is crowded with other atoms of the structure. There is physical sense behind this. Almost invariably, at such a distance other protein (or ligand) atoms are found in the structure between these two, and they will not be listed as being in contact, even at considerably smaller distances.

In fact, it is quite rare in our procedure to see putative hydrogen bonds listed at distances of ~5–6 Å. But what does such a putative H-bond distance (let us say, 3.5 Å or more) imply? First of all, that in the crystal structure this region is not crowded, and even may have "empty space" (in terms of the presented atomic coordinates of the protein). It may also mean that the two atoms are surface located, and their contact is by means of a water-mediated hydrogen bond (even if no resolved molecule of water is listed at this position in the structure). Furthermore, protein surface is more flexible than protein core and its structure is defined with less accuracy. Therefore, in the snapshot of flexible protein structure the real distance between the two putatively contacting atoms could be (and probably is) considerably less. See Note 1 for suggestions how to deal with such extended distances.

3.1.3. Affinity

Originally we used a simple complementarity function for scoring. A contact was given a weight +1 or –1 depending on its legitimacy or illegitimacy (19). This function was efficient for predicting ligand position (19, 20) and was used for side chain placement (21). Later on, further development was done to also describe quantitative changes in protein–protein binding energy (22).

3.1.4. Non-symmetry and Non-smoothness

The complementarity function is not symmetrical. There are two sources responsible for this. Contact surface of atom A with B need not be the same as B with A if they have different van der Waals radii, but this difference is not large. Larger asymmetries are introduced by neighboring atoms (see definition of contact surfaces between atoms in Fig. 1). Researchers using this program deal with this issue in one of the two ways: (1) Asymmetry is considered as the limit of accuracy of the approach (usually differences are within 1–4 Å^2). They use the program mainly for qualitative analysis of structures, to determine if there are contacts and if so, which type. (2) Asymmetry is undoubtedly behind problems encountered when the function is used for scoring (see Note 2).

The neighboring atoms also make the function not smooth (even with jumps). Because of this, we do not use gradient methods for optimizing ligand position during docking; instead, the simplex method is used. In Fig. 1b, consider a situation when all atomic radii are the same and interatomic distances AC and AB are *almost* the same, for example, when AC is a little smaller than AB. In this case, atom C screens some part of the contact between A and B. However, if BA is a little smaller than AC, the presence of atom C will not screen any part of the contact between B and A! Thus, small changes in the position of atom C can cause a jump in the contact surfaces of AB and BA (see Note 3).

3.2. Protein–Protein Interaction in Dimeric β-Sandwich Proteins

Potapov et al. (23) analyzed super-secondary structure formed by two β-sheets in immunoglobulins and other sandwich-like proteins. We found that roughly half of contacts forming the interfaces are conserved, both for intra- and inter-domain interactions. Analysis of protein–protein interaction in resolved dimeric β-sandwich proteins was applied in order to further clarify the principles involved in stabilizing super-secondary structures formed by β-sheets.

3.2.1. Interaction of β-Sandwich Proteins

The sheet–sheet mode of interaction involves a face-to-face packing of β-sheets from both interacting domains. For example, 10 proteins in Table 1 form heterodimers while the other 22 are built up of identical subunits. Among the latter, there are 14 homodimers and 8 homotetramers. The characteristics of these protein interfaces are represented in Table 4. The average contact area is 956 Å^2 with 83% contributed by contacts where at least one residue is from the β-sheet. The average contact area among the analyzed homodimers is about 200 Å^2 larger than among the heterodimers. Additionally, contacts where both residues are from β-sheets were more frequent in homodimers, while loop–loop contacts were equally infrequent.

The extended β-sheet interfaces are created through the formation of the main chain hydrogen bonds between edge strands in interacting domains. For example, in Table 2, 21 protein complexes interact in an extended β-sheet mode. All complexes except

Table 4
Characteristics of protein–protein interfaces in β-sandwich proteins

Interface type	Average interface area (min; max) (Å²)	Contribution by secondary structure, Å² (%)		
		Strand–strand[a]	Strand–loop[b]	Loop–loop[c]
Sheet–sheet	956 (464; 1,720)	386 (40)	403 (43)	163 (17)
Extended sheet	1,162 (528; 2,263)	479 (45)	427 (36)	256 (19)
Other	789 (420; 1,274)	88 (11)	301 (36)	394 (53)

The average interface area contributed by:
[a]Contacts where residues in both interacting proteins originate from β-strands
[b]Contacts where one of the residues originates from β-strands
[c]Contacts where none of the residues originates from β-strands

Arrestin (1cf1) are homodimeric (15 dimers, 6 tetramers). On average ESI interfaces are about 200 Å² larger than SSI ones. In spite of the fact that the most distinctive feature of the ESI interfaces is the formation of main chain hydrogen bonds, contacts between main chain atoms make up only a quarter of the total, while contacts with side chain atoms contribute three quarters. The minimal number of hydrogen bonds among all cases of extended β-sheet interfaces in Table 2 is four, suggesting that this is the minimal number required to form this kind of interface. On average, the number of hydrogen bonds in Table 2 entries is 7 with the maximal one having 10.

3.2.2. Properties of the Interfaces

Analysis of interface contacts in SSI and ESI interfaces indicates that different physicochemical properties favor formation of these two types of complexes. In SSI, the interface area is more hydrophobic than the rest of the protein surface. In ESI, there is no clear difference between interface area and protein surface in terms of hydrophobicity/hydrophilicity. However, taking into account that the main feature of ESI interfaces is the formation of main chain hydrogen bonds, it was found that the interface area has a larger fraction of accessible surface of main chain nitrogen and oxygen atoms than the non-interface area.

Why does a particular β-sheet face take part in the SSI interface and not the opposite one? Similarly, why does a particular edge take part in the ESI interface and not the opposite one? In SSI interfaces, analysis of the contacting β-sheet versus the non-contacting one shows that hydrophobicity is still a major determinant, though variation in hydrophobic area is large (Fig. 2). Similarly, the contacting edge in ESI interfaces has a larger main chain nitrogen and oxygen accessible surface than the non-contacting one. In comparing the two, absolute areas are used because sheets and edges may have different sizes and relative areas.

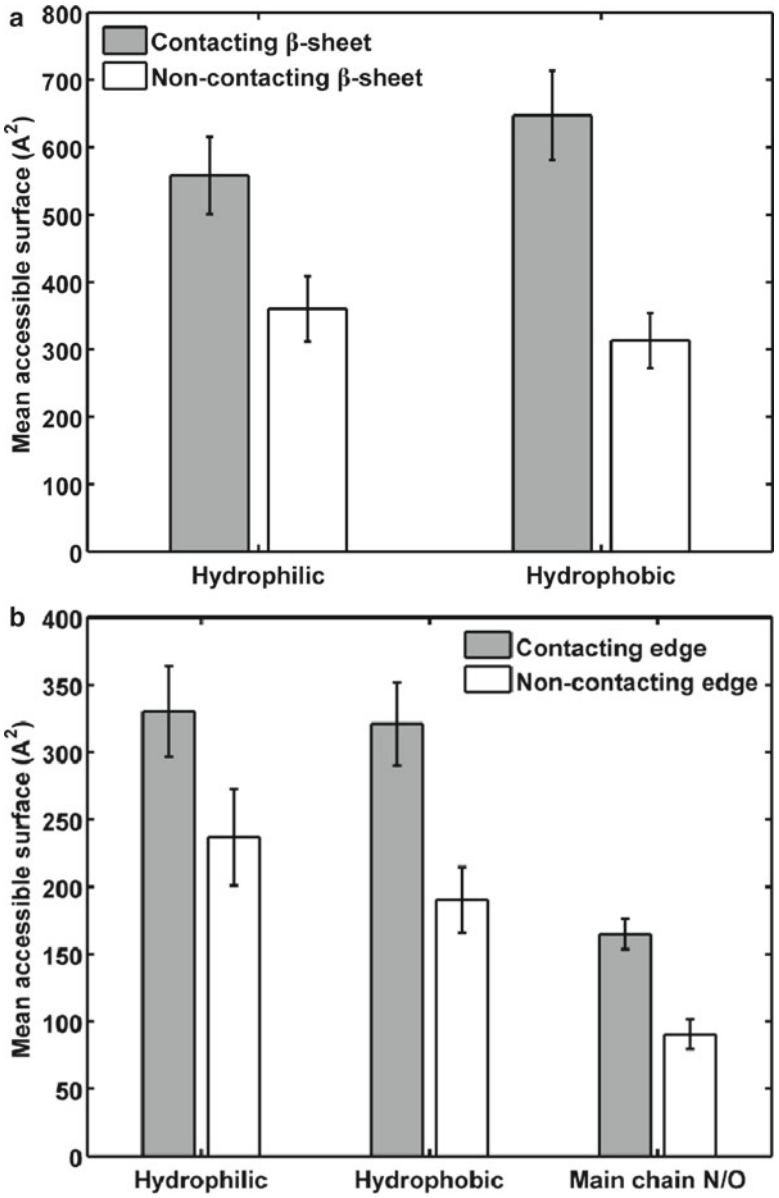


Fig. 2. Comparison of physicochemical properties of contacting and non-contacting elements in SSI (**a**) and ESI (**b**) interfaces. The contacting β-sheet in SSI interfaces, on average, is more hydrophobic than the non-contacting one. The contacting edge in ESI interfaces has a larger main chain nitrogen and oxygen accessible surface than the non-contacting one. *Error bars* show standard error of the mean.

The amino acid composition of the dimeric β-sandwich interface (Fig. 3) is similar to that reported for homodimers (24). Aromatic and hydrophobic residues are more abundant at interfaces, while polar and charged residues are more abundant on the exposed surface (except Arg, which is more common at the interface). We note that Met and Cys are very abundant in ESI interfaces and not in other types of interfaces.

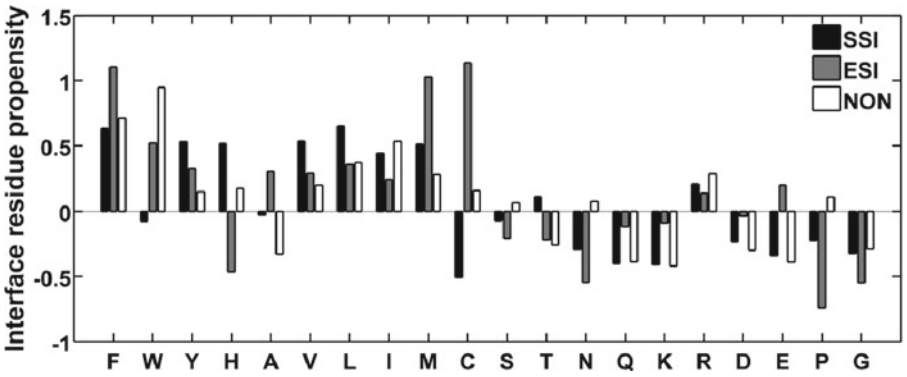


Fig. 3 Interface residue propensities. Propensity is calculated as the logarithm of the ratio of occurrence of an amino acid in the interface to occurrence of the same amino acid on the protein surface. A positive number indicates that the amino acid is observed more frequently at the interface than on the protein surface.

Analysis of residue conservation did not reveal significant insights (see Note 4).

3.2.3. Predicting the Contacting Surface

Properties for known interfaces may serve as a basis for prediction: of interacting β-sheet in the case of sheet–sheet mode of interaction, and interacting edge in case of extended β-sheet interaction. In 75% of the cases we described above, the contacting β-sheet had a larger hydrophobic area than the non-contacting one. Among proteins forming an extended β-sheet interface only in lectin-like alpha-amylase inhibitor (1dhk) did the contacting edge show a smaller portion of solvent accessible area for main chain oxygen and nitrogen atoms.

The size of the β-sheet has almost the same predictive value as the hydrophobic area. In about 70% of the cases, the contacting β-sheet is larger than the non-contacting one and more hydrophobic. Training with support vector machines using a combination of these properties did not produce a higher predictive value.

If the mode of interaction is known, the contacting surface can be accurately determined, especially in the case of the extended β-sheet interface. If the mode of interaction is unknown it is necessary to predict it first, and then, the contacting surface. However, the first prediction is problematic. The difference in hydrophobic area between β-sheets in SSI interfaces for the cases we analyzed was 370 ± 206 s.d. Å^2, while the difference in hydrophobic area between β-sheets in proteins forming ESI interfaces was 155 ± 191 s.d. Å^2. In spite of this more than twofold difference, there is a great variation in this parameter for both types of interfaces which does not allow one to state with confidence whether a given protein will or will not form an SSI. Similar analysis of solvent accessible area of main chain nitrogen and oxygen atoms shows that in ESI interfaces this difference is almost twice higher than in SSI interfaces (69 ± 48 s.d. Å^2, versus 36 ± 48 s.d. Å^2). However, there is a great variation in this parameter within the two groups here as well.

3.2.4. Predicting Mode of Interaction Based on Homology

As the mode of interaction, if not known, is hard to predict, structures of existing complexes might serve as templates for this purpose. In this approach, two steps are necessary. In the first step, one finds a homologous protein whose structure of complex is known. However in spite of the similar fold, properties of the contacting surface in a homologous protein might differ from properties of the "homologous" surface in a protein of interest. Therefore, in the second step, one estimates compatibility of a protein with the "found" mode of interaction. In other words, are relevant properties of the protein of interest similar to those of the homologous protein, so that we can assume it will interact in the same way?

To answer this question it is necessary to analyze both "positive" and "negative" examples (in other words, proteins that do and do not form a complex) to reveal the difference between them. For this purpose, extended sets of proteins forming SSI and ESI interfaces were compared to structurally similar monomeric proteins. As was shown above, hydrophobicity in the case of SSI interfaces, and accessible surface of main chain nitrogen and oxygen atoms in the case of ESI interfaces, has predictive value. These properties were used to score the differences between β-sheets and edges. It is not enough just to analyze properties of the β-sheet or edge in isolation, because, for example, a large hydrophobic surface of the β-sheet will not necessarily mean that it forms an interface if its partner is as hydrophobic as it is. Thus, the differential in hydrophobic area (ΔS_{PHO}) between β-sheets was compared in SSI and monomeric proteins, and the differential in accessible surface of main chain nitrogen and oxygen atoms ($\Delta S_{N,O}$) between edges was compared in ESI and monomeric proteins.

Analysis shows a clear difference in ΔS_{PHO} for SSI and structurally similar monomeric proteins (Fig. 4): whereas monomeric proteins tend to have a low ΔS_{PHO} (median value, 151 Å^2), proteins forming SSI interfaces have on average a more than twice-higher median value (380 Å^2). Similar results were obtained for ESI and monomeric proteins by looking at $\Delta S_{N,O}$. The median value of $\Delta S_{N,O}$ in ESI proteins is 76 Å^2, whereas in monomeric proteins it is 42 Å^2. The hydrophobic area in monomeric proteins is more evenly spread on the protein surface, resulting in relatively low ΔS_{PHO} values. In contrast, the large value of ΔS_{PHO} in SSI proteins reflects a situation where one of the β-sheets is involved in interaction with another protein and has a large hydrophobic area. Similar conclusions can be reached for ESI proteins.

The difference in distribution of ΔS_{PHO} for the two groups of proteins allows one to use this characteristic for predicting whether the protein will form a complex. Taking the percentage of cases in a given interval for SSI proteins and dividing it by the sum of percentages of SSI and monomeric proteins gives the probability for protein in that interval to form a complex. This is represented in the lower plot in Fig. 4.

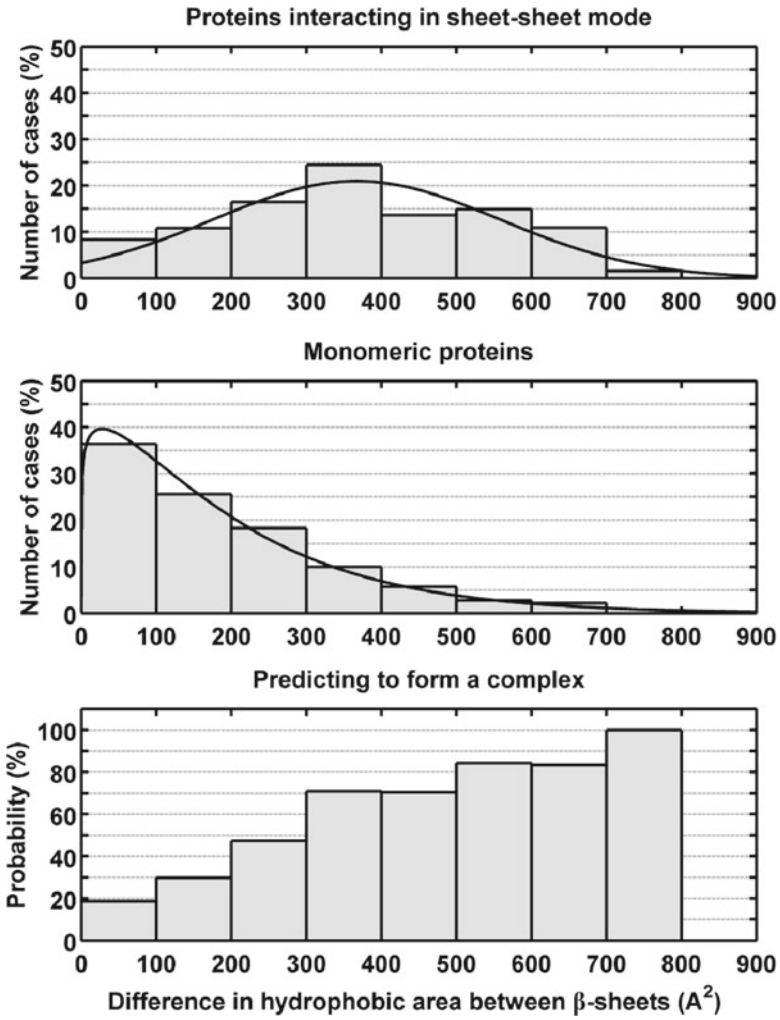


Fig. 4. Distribution of ΔS_{PHO} in SSI and monomeric proteins. The median value of ΔS_{PHO} in SSI proteins (380 Å^2) is more than twice that in monomeric ones (151 Å^2). Whereas ΔS_{PHO} in both situations ranges from 0 Å^2 to about 800 Å^2, the distribution of ΔS_{PHO} in SSI interfaces is close to Gaussian around a mean value, but it is skewed sharply to low values for monomeric proteins. The *lower plot* gives the probability to form a dimer for each interval.

The validity of the plot for prediction purposes can be further tested. The probability to form a complex was estimated for each protein in groups of SSI and monomeric proteins. The average probability to form a complex for SSI proteins equaled 71% while for monomeric proteins it was 30%. The applicability of the prediction in the general case can be tested by a tenfold cross-validation procedure. This resulted in very similar values (68% and 31%, respectively). This validation indicates that in cases of unknown proteins, the prediction plot will have a statistically similar level of accuracy as that of the training set in Fig. 4.

The validation procedure was repeated for the ESI proteins. Whereas ESI proteins are predicted with an average probability of 67%, monomeric proteins are predicted on average with a probability of 45%. The tenfold cross-validation procedure resulted in similar values (62% and 44%, respectively). The lower average level of separation for ESI proteins compared to monomeric proteins is due to the lower absolute difference 76 Å^2 versus 42 Å^2 compared to the case of SSI proteins where the median values of ΔS_{PHO} are 380 and 151 Å^2.

In summary, by analyzing physicochemical properties of monomers and proteins forming either SSI or ESI interfaces, a difference in their surface can be discerned; the protein surface in monomers tends to be more uniform in terms of hydrophobic and main chain nitrogen and oxygen atom accessible surface. The difference in distribution of ΔS_{PHO} in SSI proteins, and $\Delta S_{N,O}$ in ESI proteins, versus monomeric proteins can be used to deduce the probability of a protein adapting a similar mode of interaction as that of a homologous one whose structure of complex is known.

4. Notes

1. When accessing the CSU server (http://ligin.weizmann.ac.il/lpccsu) for analysis of a structure, a user may consider the full list of contacts and then restrict consideration, for example, to contacts up to a distance of 3.5 Å. If a user is using the CSU program as a subroutine in an automated analysis of many structures, the output can be extracted with any desired restriction (threshold). Note: LPC/CSU was originally created for use in docking procedures. For such cases, listing contacts at extended distances is a very useful property.
2. Asymmetry can be mitigated by using average values; however, you have to first calculate contacts of all residues (or atoms) at once and then derive the average of the two contacts—A with B and B with A.
3. There is an option to use a constrained Voronoi procedure for calculating atom–atom contacts (25). This behaves gradually as a function of atom coordinates and can be more conveniently used as a scoring function (26).
4. Only in about 60% of the cases were interface residues more conserved than those in the rest of the protein surface for both types of interfaces. Similar percentage was obtained in the analysis of contacting versus non-contacting elements in β-sandwich domains. This is in agreement with reports in literature (27) and with our analysis of sheet–sheet mode interfaces (23).

References

1. Warshel A, Levitt M (1976) Theoretical studies of enzymatic reactions: dielectric electrostatic and steric stabilization of the carbonium ion in the reaction of lysozyme. J Mol Biol 103:227–249
2. Beierlein FR, Michel J, Essex JW (2011) A simple QM/MM approach for capturing polarization effects in protein-ligand binding free energy calculations. Phys Chem B 115:4911–4926
3. Hendrickson JB (1961) Molecular geometry. I. Machine computing of the common rings. J Am Chem Soc 49:4537–4547
4. Kitaygorodsky AI (1961) The interaction of non-bonded carbon and hydrogen atoms and its application. Tetrahedron 14:230–236
5. Scott RA, Scheraga HA (1966) Conformational analysis of macromolecules. III. Helical structures of polyglycine and poly-l-alanine. J Chem Phys 45:2091–2101
6. Lifson S, Warshel A (1968) Consistent force field for calculations of conformations, vibrational spectra, and enthalpies of cycloalkane and n-alkane molecules. J Chem Phys 49:5116–5129
7. Ponder JW, Case DA (2003) Force fields for protein simulations. Adv Prot Chem 66:27–85
8. Christen M et al (2005) The GROMOS software for biomolecular simulation: GROMOS05. J Comput Chem 26:1719–1751
9. Brooks BR et al (2009) CHARMM: The biomolecular simulation program. J Comput Chem 30:1545–1614
10. McDonald IK, Thormton JM (1994) Satisfying hydrogen bonding potentials in proteins. J Mol Biol 238:777–793
11. Wallace AC, Laskowski RA, Thornton JM (1995) LIGPLOT: a program to generate schematic diagrams of protein-ligand interactions. Prot Eng 8:127–134
12. Word JM et al (1999) Visualizing and quantifying molecular goodness-of-fit: small-probe contact dots with explicit hydrogen atoms. J Mol Biol 285:1711–1733
13. Doncheva NT et al (2011) Analyzing and visualizing residue networks of protein structures. Trends Biochem Sci 36:179–182
14. Sobolev V, Edelman M (1995) Modeling the quinone-B binding site of the photosystem-II reaction center using notions of complementarity and contact surface between atoms. Proteins 21:214–225
15. Sobolev V et al (1999) Automated analysis of interatomic contacts in proteins. Bioinformatics 15:327–332
16. Bernstein FC et al (1977) The protein data bank: a computer based archival file for macromolecular structures. J Mol Biol 112:535–542
17. Murzin AG et al (1995) Scop - a structural classification of proteins database for the investigation of sequences and structures. J Mol Biol 247:536–540
18. Elcock AH, McCammon JA (2001) Identification of protein oligomerization states by analysis of interface conservation. Proc Natl Acad Sci USA 98:2990–2994
19. Sobolev V et al (1996) Molecular docking using surface complementarity. Proteins 25: 120–129
20. Sobolev V et al (1997) CASP2 molecular docking predictions with the LIGIN software. Proteins 29(Suppl 1):210–214
21. Eyal E et al (2004) Importance of solvent accessibility and contact surfaces in modeling side-chain conformations in proteins. J Comput Chem 25:712–724
22. Potapov V et al (2008) Computational redesign of a protein–protein interface for high affinity and binding specificity using modular architecture and naturally occurring template fragments. J Mol Biol 384:109–119
23. Potapov V et al (2004) Protein–protein recognition: juxtaposition of domain and interface cores in immunoglobulins and other sandwich-like proteins. J Mol Biol 342:665–679
24. Bahadur RP et al (2003) Dissecting subunit interfaces in homodimeric proteins. Proteins 53:708–719
25. McConkey BJ, Sobolev V, Edelman M (2002) Quantification of protein surfaces, volumes and atom–atom contacts using a constrained Voronoi procedure. Bioinformatics 18:1365–1373
26. McConkey BJ, Sobolev V, Edelman M (2003) Discrimination of native protein structures using atom-atom contact scoring. Proc Natl Acad Sci USA 100:3215–3220
27. Caffrey DR et al (2004) Are protein–protein interfaces more conserved in sequence than the rest of the protein surface? Protein Sci 13:190–202

Part III

Supersecondary Structure and Protein Folding

Chapter 11

Super-secondary Structures and Modeling of Protein Folds

Alexander V. Efimov

Abstract

A characteristic feature of the polypeptide chain is its ability to form a restricted set of commonly occurring folding units composed of two or more elements of secondary structure that are adjacent along the chain. Some of these super-secondary structures exhibit a unique handedness and a unique overall fold irrespective of whether they occur in homologous or nonhomologous proteins. Such super-secondary structures are of particular value since they can be used as starting structures in protein modeling. The larger protein folds can be obtained by stepwise addition of other secondary structural elements to the starting structures taking into account a set of simple rules inferred from known principles of protein structure.

Key words: Handedness, Protein modeling, Structure comparison, Structural tree, Super-secondary structure, Unique fold

1. Introduction

Super-secondary structures of globular proteins can be defined as commonly occurring folding units consisting of two or more elements of secondary structure that are adjacent along the polypeptide chain. While many different super-secondary structures have been observed to recur within unrelated proteins (1–3), only some of the structures exhibit unique handedness and a unique overall fold (4). A unique overall fold is defined by the number and type of secondary structure elements, the three-dimensional arrangement of these elements, and their connectivity. As a rule, super-secondary structures having a unique overall fold have a unique handedness. Super-secondary structures of a given type found in unrelated proteins may have the same overall fold despite their α-helices and/or β-strands being of different lengths, their connection regions differing in length and conformation, and their sequences lacking homology.

Alexander E. Kister (ed.), *Protein Supersecondary Structures*, Methods in Molecular Biology, vol. 932,
DOI 10.1007/978-1-62703-065-6_11,

The first example of such super-secondary structures was observed by Rao and Rossmann in α/β-proteins (1). Later on, a number of commonly occurring super-secondary structures with unique overall folds have been found in other classes of proteins (for reviews, see, e.g., (3–6)). The high frequency of occurrence of the super-secondary structures in unrelated proteins and the fact that many small proteins and domains merely consist of such structures suggest that they are relatively stable and can fold into unique structures per se. On the other hand, since the secondary structural elements are adjacent along the polypeptide chain they can associate rapidly to form compact folds. Thus, many super-secondary structures have unique folds themselves and each of them can act as a core around which the remainder of the protein molecule or the domain is folded. Alternatively, the super-secondary structures with unique folds can be used as starting structures in protein modeling. The larger protein folds can be obtained by a stepwise addition of α-helices and/or β-strands to the corresponding starting structure taking into account a restricted set of rules inferred from known principles of the protein structure (7).

2. Materials

Databases for all the structural groups of proteins containing the corresponding super-secondary structures were compiled using the Protein Data Bank (PDB) (http://www.rcsb.org/pdb/) and the SCOP database (http://scop.mrc-lmb.cam.ac.uk/scop/) (8). Proteins were manually selected from the PDB and visually examined using the RasMol molecular graphics program (9). Possible homologies were revealed by the BLAST 2 SEQUENCES (10) and PROTEIN-PROTEIN BLAST programs (http://blast.ncbi.nlm.nih.gov/Blast.cgi/). Now our database includes ten structural groups of proteins organized as the PCBOST database (http://strees.protres.ru/) (11). In total, our database contains about 4,900 proteins and domains (among them more than 1,800 are nonhomologous) and includes more than 14,000 PDB entries. The work on extension of the database and construction of novel structural trees is in progress.

3. Methods

3.1. Search for and Analysis of Super-secondary Structures

In accordance with the definition, super-secondary structures are commonly occurring folding units consisting of two or more secondary structural elements found in unrelated proteins. Currently, super-secondary structures and structural motifs are

often used as synonymous terms. The main methods of searching for novel super-secondary structures and structural motifs are visual inspection and structure comparison of known proteins.

In proteins, there is a number of super-secondary structures composed of two connected α-helices such as αα-hairpins, L-shaped and V-shaped structures, and αα-corners. Of these possibilities, only the αα-corner has a unique handedness and a unique overall fold. Its two α-helices are packed approximately crosswise so that, in three dimensions, the polypeptide chain passes through almost one complete turn of a left-handed superhelix (Fig. 1) (12). Variants of this structure were initially found in two homologous protein families, "EF-hands" in calcium-binding proteins (13) and "helix-turn-helix" motifs in the DNA-binding proteins (14). Now it is known that αα-corners are widespread in both homologous and nonhomologous proteins (12, 15) and occur practically always in one form. For comparison, there are three possible arrangements of α-helices in αα-hairpins. These are right-turned and left-turned αα-hairpins in which α-helices are packed side by side and αα-hairpins with α-helices packed face to face (16, 17).

A flat β-sheet would have no handedness, but in proteins, β-sheets are invariably twisted in a right-handed sense when viewed in the direction of the polypeptide chain (18). This is a characteristic of protein β-sheets that is independent of the arrangement and connectivity of their β-strands. It means that β-sheets may have a unique handedness without necessarily having a unique fold. For example, ββ-hairpins can be right- or left-turned depending on whether the second β-strand runs on the right or the left relative to the first one when viewed from the same side. Similarly, triple-stranded β-sheets having up-and-down topology can exist in two forms, as S-like or Z-like β-sheets.

Uniqueness appears at the level of higher order structures. If a ββ-hairpin is strongly twisted and coiled into a right-handed double-stranded superhelix (Fig. 1), it is always right-turned when viewed from the concave side (19). If a long ββ-hairpin folds onto itself so that the β-strands of the two halves are packed orthogonally in the two different layers, it is also right-turned when viewed from the concave side. This structure called the ββ-corner (Fig. 1) is right-handed in proteins since the strands rotate about an imaginary axis in the right-handed direction when passing from one layer to the other.

The 3β-corner can be represented as a Z-like β-sheet folded onto itself so that the two ββ-hairpins are packed approximately orthogonally in different layers and the central β-strand bends by ~90° in a right-handed direction when passing from one layer to the other (20). Two representative super-secondary structures including S-like β-sheets, the βSβ-superhelix and βαS-unit, are shown in the bottom row of Fig. 1. The first structure can be represented as a right-handed superhelix if the S-like β-sheet is replaced

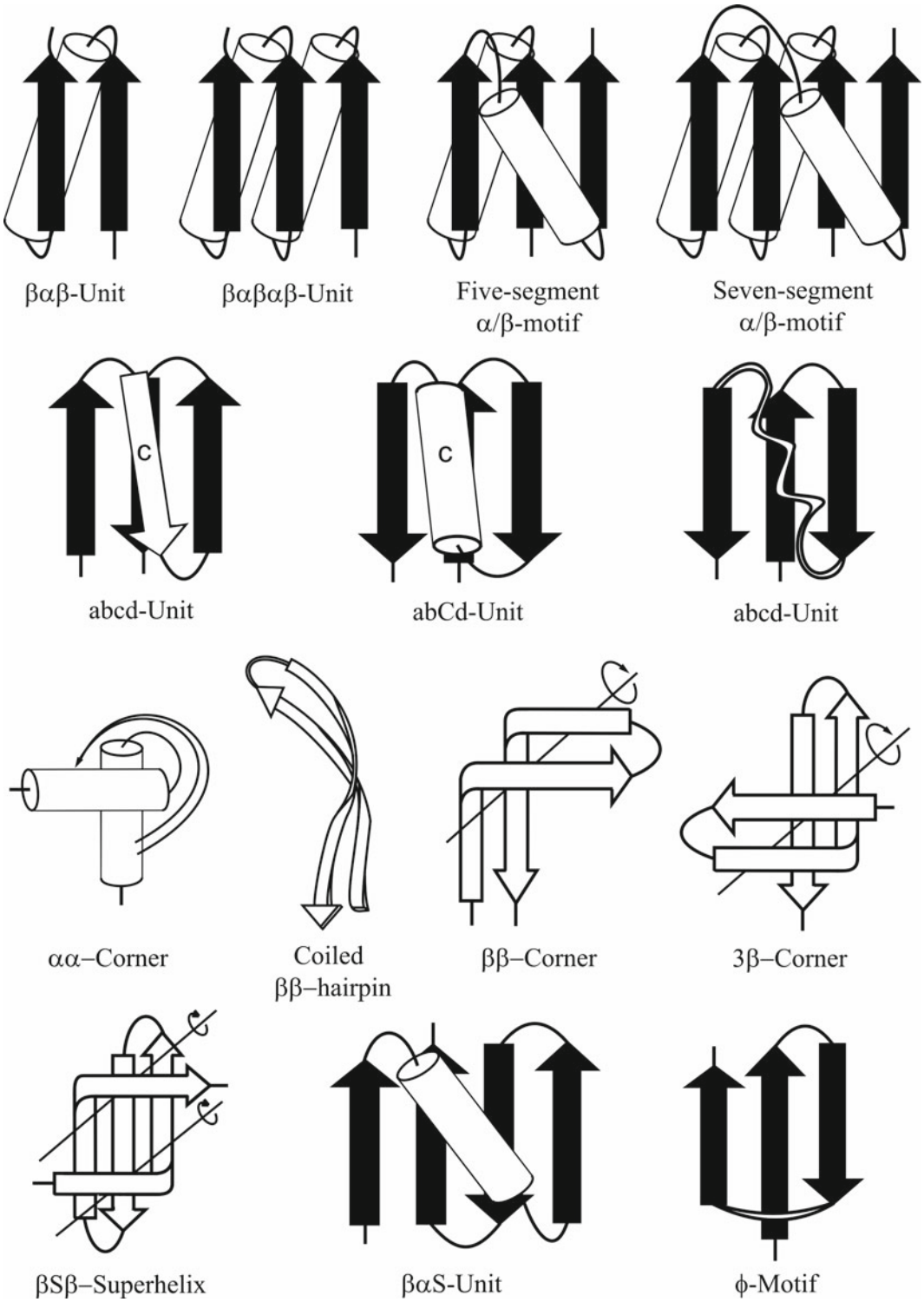


Fig. 1. Representative super-secondary structures having unique overall folds and handedness. β-Strands are shown as *arrows* and α-helices as *cylinders*. Imaginary axes of superhelices are represented as *straight lines*. See also the text.

by one imaginary strand. In the βαS-unit, the split βαβ-unit forms a right-handed superhelix. A distinctive feature of these super-secondary structures is that they include only S-like β-sheet and cannot include Z-like β-sheets (21). On the other hand, the 3β-corner is formed by a Z-like β-sheet and cannot be formed by an S-like β-sheet.

The simplest φ-motif is formed by three adjacent β-strands connected by loops and packed in one β-sheet so that its overall fold resembles the Greek letter φ (22). The loop which connects the two edge β-strands and crosses over the central β-strand or its extension is referred to as the crossover loop. There are right-handed and left-handed φ-motifs. In the right-handed φ-motif, the polypeptide chain runs from the N- to the C-end in the clockwise direction when viewed from the crossover loop (Fig. 1). In known proteins, φ-motifs occur predominantly in the right-handed form (22).

The βαβ- and βαβαβ-units are composed of α-helices and β-strands that alternate along the polypeptide chain. Their chains are folded into right-handed superhelices so that the β-strands form parallel β-sheets and the α-helices are packed in separate layers. These βαβ-superhelices are the main "building blocks" of α/β-proteins and almost always occur in the right-handed form (1, 23). In triple-layered α/β-proteins, there are two recurring motifs composed of five or seven elements of secondary structure which can be represented as combinations of the simple and split βαβ-units (see (4, 7) and the upper row of Fig. 1). Different combinations of the ψ-motif (24, 25) and the βαβ-unit are also widespread in α/β- and (α+β)-proteins (26).

The abcd-unit that is a commonly occurring folding unit in two-layered β-proteins (27) can be represented as a combination of a β-hairpin formed by β-strands a and b and a right-handed superhelix formed by strands b, c, and d. In the abCd-unit that occurs in (α+β)-proteins (27, 28) element C is an α-helix and the right-handed superhelix bCd is a split βαβ-unit (see Note 1). It is of interest that simplified depiction of the abcd-unit on a plane, using Richardson's approach (29), results in the so-called Greek key topology. However, a long β-hairpin bent in half and several other super-secondary structures also have the Greek key topology (see, e.g., 3, 27). So it is necessary to distinguish between the topology and the three-dimensional arrangement of β-strands.

An inspection of known proteins shows that structural motifs having unique overall folds tend to be located at the edges of two- or three-layered protein molecules with additional α-helices and/or β-strands arranged on one side of each motif (4, 7, 12, 27, 28). The larger protein folds can be obtained by a stepwise addition of α-helices and/or β-strands to the corresponding starting structural motif taking into account a restricted set of rules inferred from known principles of the protein structure. At each step, there

are several pathways of structure growth, but the number of allowed pathways is limited since the rules drastically reduce it (see Note 2).

A general scheme that represents the root or starting structural motif and all the intermediate and completed structures connected by lines showing allowed pathways of structure growth is referred to as the structural tree. The first versions of structural trees were constructed more than 15 years ago (7, 15). The number of solved protein structures in the PDB has substantially increased over this time. Hence, it is necessary to construct updated trees with all proteins available from PDB for each given class. Now our database includes ten computer versions of updated structural trees which are available at http://strees.protres.ru/. Structural trees can be used for solving several problems such as protein structure comparison, structural classification of proteins, protein folding and modeling, searching for all possible protein folds both known and unknown, etc.

3.2. General Rules Used in Construction of Structural Trees

Modeling of protein folds and folding pathways is based on the construction and analysis of structural trees taking into account the following set of general rules:

1. The structural motif having a unique overall fold and handedness is taken as the starting structure in modeling or the root structure of the tree.
2. Overall folds of protein and intermediate structures are taken into account and details of the structures are ignored. If the polypeptide chain direction is not shown each structure in the tree can have both the directions of the chain.
3. The larger protein and intermediate folds are obtained by stepwise addition of α-helices and/or β-strands to a growing structure so that a structure obtained at the preceding step is maintained. In some cases, "ready building blocks," e.g., β-hairpin or S-like β-sheet, are added.
4. At each step, the α-helix or the β-strand nearest to the growing structure along the polypeptide chain is the first to be attached to it (7, 12, 27).
5. The α-helices and β-strands cannot be packed into one layer because of dehydration of the free NH and CO groups of the β-strands; thus, an α-helix should be packed into the α-helical layer and a β-strand into the β-layer of a growing structure (28, 30).
6. The obtained structures should be compact; α-helices and β-sheets should be packed in accordance with the rules that govern their packing (17, 30–32).
7. Crossing of connections (33) and formation of knots (34) are prohibited, but formation of the φ- and ψ-motifs (22, 24, 25) is permitted (see Note 3).

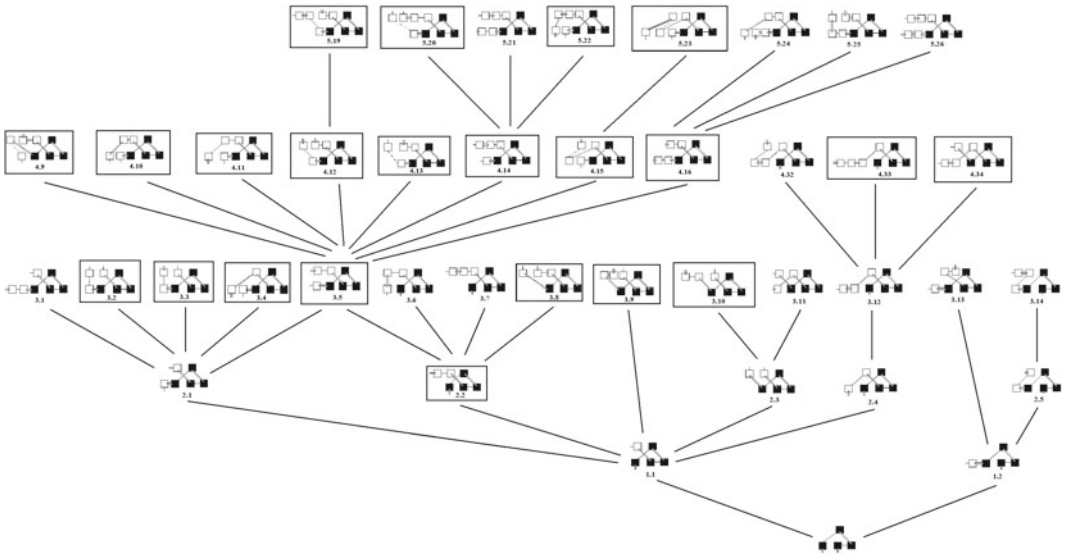


Fig. 2. A fragment of the structural tree for β-proteins containing abcd-units. The structures are viewed end-on with β-strands shown as rectangles. Near connections are shown by *double lines* and far connections by *single lines*. Long connections are simplified and drawn by *dashed lines*. In each level, the folds are enumerated from the *left* to the *right*. The folds actually found in proteins and domains as completed structures are framed. A complete version of the tree is available at http://strees.protres.ru/.

8. All the structural motifs (not only the root motifs) of the intermediate and completed structures should have the corresponding overall folds and handedness (see Note 4).

3.3. Modeling of Protein Folds Containing abcd-Units

The abcd-unit is a structural motif recurring in two-layered β-proteins and β-domains with the aligned β-sheet packing (7, 27, 35). Analysis of known proteins and domains shows that the abcd-unit is always located at the edge of the double layer, and the other β-strands are situated on the side of the d strand. So, when other β-strands are added to the root abcd-unit step by step, it looks as if the abcd-unit grows in one direction (see Fig. 2). A distinctive feature of β-proteins containing the abcd-units is that not only strands b, c, and d but also some other three β-strands adjacent along the chain can form a right-handed βββ-superhelix analogous to the split βαβ-superhelix. It means that the first and third β-strands of such βββ-superhelices do not directly interact, and there is at least one additional β-strand in the β-layer between them. For example, the right-handed superhelix bcd of the abcd-unit has strand a between strands b and d.

Let us label the strands joined to strand a of the abcd-unit as a_1, a_2, a_3, … and the strands joined to strand d as d_1, d_2, d_3, … according to their distance from strands a or d in the polypeptide chain

irrespective of the chain direction. Their possible three-dimensional arrangements can be obtained by stepwise addition to the abcd-unit taking into account the rules listed above (Fig. 2). Addition of strand a_1 (joined to strand a) to the root abcd-unit results in fold 1.1 shown on the left in the bottom level. Strands a_1 are arranged in this way in all the known proteins in which they are present. Strand a_1 cannot be packed in the bottom layer next to strand d as crossing of loops aa_1 and dd_1 would occur (rule 7). Strand a_1 also cannot be packed on the other side of strand c in the upper layer or next to strand b in the bottom layer (i.e., at the edge) as loops aa_1 and bc would cross. Addition of strand d_1 to the root abcd-unit results in fold 1.2 shown on the right of the first level of the tree. Strand d_1 cannot be packed in the upper layer next to strand c since obtained superhelix cdd_1 would not be split and strands c and d_1 would interact with each other. However, strand d_1 can be packed in the upper layer next to strand a_1 as in fold 2.3. It is possible as there is strand a_1 between strands c and d_1 in the right-handed split superhelix cdd_1. Similarly, arrangements of other β-strands relative to the root abcd-unit can be obtained if the next strands are added to it step by step. Note that fold 3.9 is obtained by addition of β-hairpin d_1d_2 to fold 1.1 and fold 3.13 is obtained by addition of β-hairpin a_1a_2 to fold 1.2 (see Note 5).

3.4. Modeling of Protein Folds Containing abCd-Units

The abCd-unit is a variant of the abcd-unit that has α-helix C instead of strand c (3, 27, 28). β-Proteins containing abcd-units and (α+β)-proteins containing abCd-units have very much in common. First of all, many proteins and domains of these classes have very similar overall folds if segment conformations are ignored (28). In both classes the abcd- and abCd-units tend to be located at the edges of molecules and most other secondary structural elements are situated on the d-strands of the units.

Possible pathways of growth of the root abCd-unit are represented in Fig. 3 (7, 36). There are more ways of addition of the first secondary structural element to the root abCd-unit as compared with the first step of the abcd-unit growth. If there is α-helix A_1 joined to strand a, it is packed next to helix C in the α-helical layer (fold 1.6) or below the β-sheet giving rise to another α-helical layer (fold 1.3) in accordance with rule 5. Helix A_1 cannot be packed on the other side of helix C in the α-helical layer as loops aA_1 and bC would cross. Helix A_1 also cannot be packed into the β-layer (rule 5). If helix A_1 is absent from a molecule, while helix D_1 joined to strand d is present, it is also packed in the α-helical layer next to helix C (fold 1.2) or below the β-sheet (folds 1.4, 1.5). In accordance with the rules, β-strand d_1 is packed next to strand d at the edge of the β-sheet (fold 1.1). The ways of addition of other α-helices and β-strands to the growing structures can easily be observed in Fig. 3. Note that all the obtained βαβ-units form right-handed superhelices in accordance with rule 8.

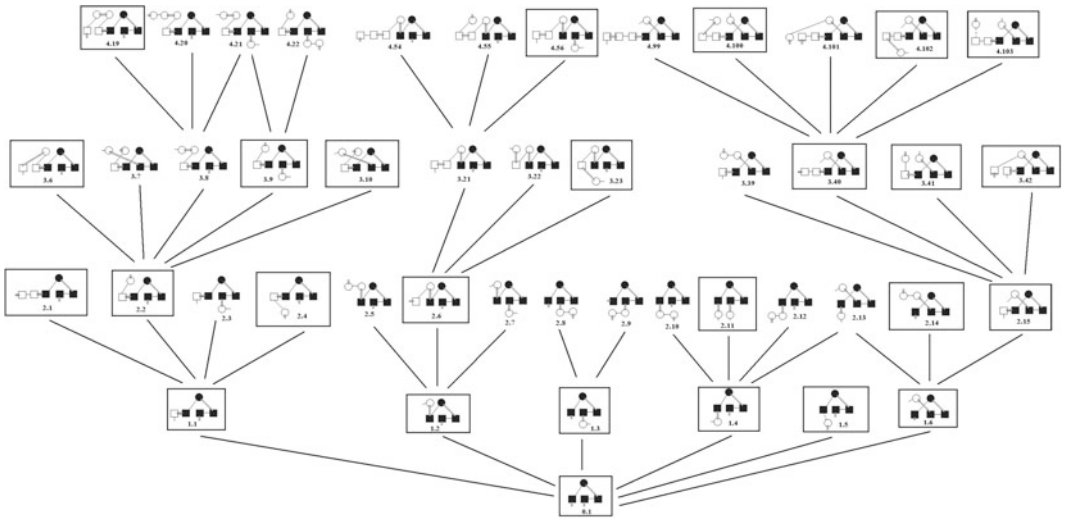


Fig. 3. A fragment of the structural tree for (α + β)-proteins containing abCd-units. The structures are viewed end-on with α-helices shown as *circles* and β-strands as *rectangles*. Other designations are as in Fig. 2.

3.5. Modeling of Protein Folds Containing Five-Segment α/β-Motifs

The five-segment α/β-motif is a structural motif consisting of three β-strands and two α-helices folded into two βαβ-units and arranged in a three-layered structure (3, 4, 7). Similar to other structural motifs considered here, this motif tends to be located at the edges of the three-layer structures of known proteins. Possible ways of stepwise growth of this motif into larger protein folds are shown in Fig. 4. Note that the branches in the right part of the tree show how combinations of the five-segment α/β-motif with the ψ-motif (folds 1.5, 2.15, 2.16, and higher; see also folds 2.7, 3.20) and with the φ-motif (folds 2.14, 3.37) can be formed (see also Note 5).

3.6. Modeling of Protein Folds Containing 3β-Corners

The 3β-corner is a structural motif that can be represented as a Z-like triple-stranded β-sheet folded upon itself so that its two halves are packed approximately orthogonally in different layers and the central β-strand is bent by 90° when passing from one layer to the other to form a half-turn of the right-handed superhelix (20, 37). A fragment of the updated structural tree for proteins containing 3β-corners is represented in Fig. 5 (38). A complete version of the tree is available at http://strees.protres.ru/. All the structures in the tree are oriented in a similar way so that root 3β-corners are localized in their bottom right corners and the β-strands of the near β-sheets are oriented horizontally and those of the far β-sheets vertically. There are two β-layers packed approximately orthogonally in the root 3β-corner. So each additional β-strand can be packed into one or the other β-layer of a growing structure. This can be done in three ways. The β-strands can be added to the root 3β-corner or another growing structure without passing of the polypeptide chain from one β-layer to the

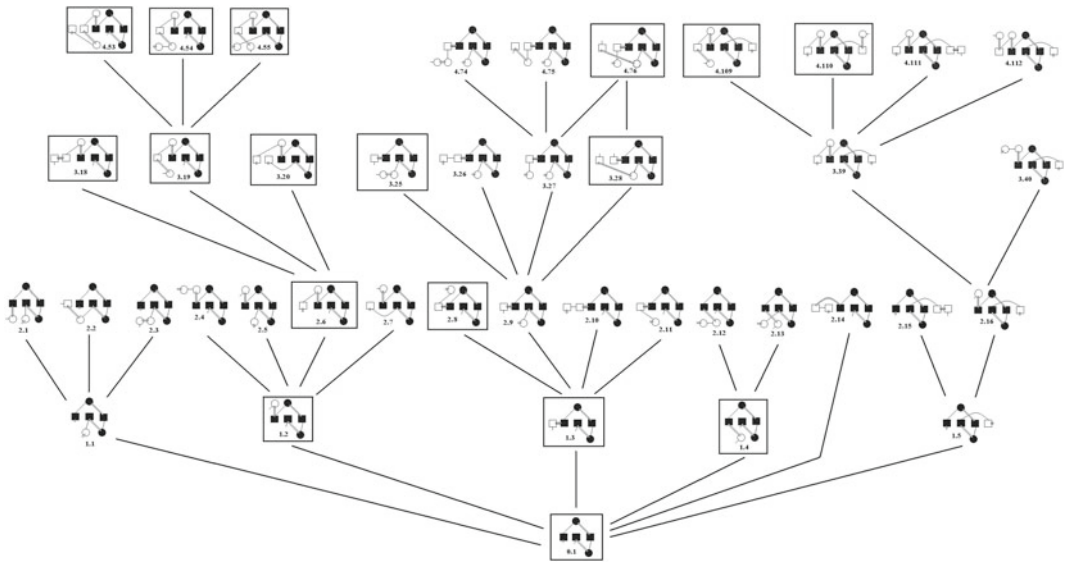


Fig. 4. A fragment of the structural tree for α/β-proteins containing five-segment α/β-motifs. The folds are represented similar to that in Fig. 3. A complete version of the tree is available at http://strees.protres.ru/.

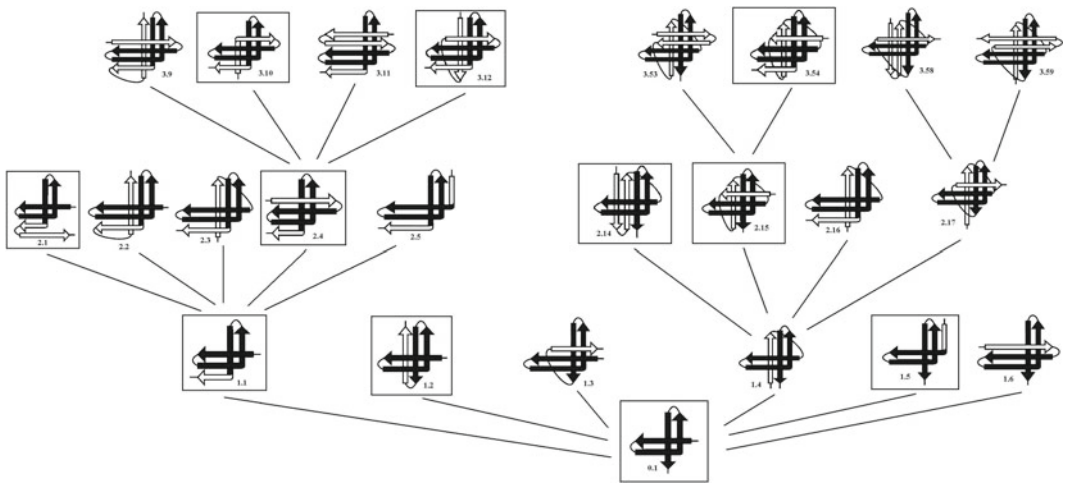


Fig. 5. A fragment of the structural tree for β-proteins containing 3β-corners. β-Strands are shown as *arrows* directed from the N- to C-ends. The β-strands of the near β-sheets are oriented horizontally and those of the far β-sheets vertically. β-Strands forming the 3β-corners are shown as *filled arrows* and the others by *open arrows*. See also the text.

other, as, for example, in folds 1.2, 1.6, 2.1, 2.14, etc. in Fig. 5. The additional β-strand can be packed in the other β-layer to form the so-called β-bend (39). In these cases, the polypeptide chain passes from one β-sheet to the other while bending by 90° to form a right-handed superhelix similar to the central β-strands in 3β-corners (see folds 1.1, 1.5, 2.5, 3.10 in Fig. 5 as well as the ββ-corner, 3β-corner, and βSβ-superhelix in Fig. 1). One more way to add a β-strand to a growing structure is represented in folds 1.3, 1.4,

2.3, 2.15, 2.17, etc. in Fig. 5. In these cases, the polypeptide chain forms a left-handed superhelix when passing from one β-sheet to the other. Stereochemical analysis shows that formation of such left-handed superhelices results in steric constraints at the crossover sites and to reduce them in proteins the constraint α_L- and ε-positions are occupied by glycines or residues with flexible side chains (38).

4. Notes

1. An inspection of known super-secondary structures shows that β-hairpins, triple-stranded β-sheets, and βαβ-units represent simple structural motifs closed into cycles by systems of hydrogen bonds. Secondary closing of these simple motifs into large cycles by means of different superhelices, split β-hairpins, or SS-bridges results in the formation of more complex structural motifs having unique overall folds and unique handedness such as abcd-units, φ-motifs, five- and seven-segment α/β-motifs, etc. Apparently, the complex structural motifs are more cooperative and stable and this may be one of the main reasons of high frequencies of occurrence of the motifs in proteins (6, 38).
2. In proteins, there are several commonly occurring folds such as the jelly roll structure (29), the β-trefoil fold (40), the OB-fold (41), the double-psi β-barrel (25), $(\alpha/\beta)_8$-barrels, the SH3-like fold, the Ig-like fold, etc. which represent the complete protein or domain structures. As a rule, these large folds contain simpler structural motifs and their structures can be obtained by stepwise addition of other elements to the corresponding simple motif. For example, the jelly roll and Ig-fold contain the abcd-units, the OB-fold includes a βSα-superhelix, the SH3-like fold contains the 3β-corner, etc.
3. As a rule, crossing of connections results in dehydration of the free NH- or CO-groups and, consequently, is prohibited in proteins (33). Formally, the crossover loops of the φ-motifs (22) and ψ-motifs (24, 25) are crossing connections. However, a detailed stereochemical analysis shows that the crossover loops in these motifs do not have dehydrated free NH- and CO-groups because of specific conformations and sequences of the loops (unpublished results).
4. In total, the left-handed βαβ-units occur very rarely in proteins (less than 1 %); however, in combinations with the ψ-motifs about 11 % of the βαβ-units are left-handed (26). The reason of this is still poorly understood and should be investigated further.

5. It should be noted that arrangements of β-strands in two-layered β-proteins can be obtained using another approach based on computer analysis of protein structure (42–44). Possible folds of α/β-proteins and the corresponding structural tree can also be generated automatically (45).

Acknowledgements

I thank Ms. E.A. Boshkova and Dr. A.B. Gordeev for help in drawing the figures. This work was supported by the Russian Foundation for Basic Research (Project No. 10-04-00727).

References

1. Rao ST, Rossmann MG (1973) Comparison of super-secondary structures in proteins. J Mol Biol 76:241–256
2. Levitt M, Chothia C (1976) Structural patterns in globular proteins. Nature 261:552–558
3. Efimov AV (1993) Standard structures in proteins. Prog Biophys Mol Biol 60:201–239
4. Efimov AV (1994) Favoured structural motifs in globular proteins. Structure 2:999–1002
5. Efimov AV (1994) Common structural motifs in small proteins and domains. FEBS Lett 355:213–219
6. Efimov AV (2010) Structural motifs are closed into cycles in proteins. Biochem Biophys Res Commun 399:412–415
7. Efimov AV (1997) Structural trees for protein superfamilies. Proteins 28:241–260
8. Murzin AG et al (1995) SCOP: a structural classification of proteins database for the investigation of sequences and structures. J Mol Biol 247:536–540
9. Sayle RA, Milner-White EJ (1995) RASMOL—biomolecular graphics for all. Trends Biochem Sci 20:374–376
10. Tatusova TA, Madden TL (1999) Blast 2 sequences—a new tool for comparing protein and nucleotide sequences. FEMS Microbiol Lett 174:247–250
11. Gordeev AB, Kargatov AM, Efimov AV (2010) PCBOST: protein classification based on structural trees. Biochem Biophys Res Commun 397:470–471
12. Efimov AV (1984) A novel super-secondary structure of proteins and the relation between the structure and the amino acid sequence. FEBS Lett 166:33–38
13. Kretsinger RH, Nockolds CE (1973) Carp muscle calcium-binding protein. II. Structure determination and general description. J Biol Chem 248:3313–3326
14. Pabo CO, Sauer RT (1984) Protein-DNA recognition. Annu Rev Biochem 53:293–321
15. Efimov AV (1996) A structural tree for α-helical proteins containing αα-corners and its application to protein classification. FEBS Lett 391:167–170
16. Efimov AV (1991) Structure of αα-hairpins with short connections. Prot Eng 4:245–250
17. Efimov AV (1999) Complementary packing of α-helices in proteins. FEBS Lett 463:3–6
18. Chothia C (1973) Conformation of twisted β-pleated sheets in proteins. J Mol Biol 75:295–302
19. Efimov AV (1991) Structure of coiled ββ-hairpins and ββ-corners. FEBS Lett 284:288–292
20. Efimov AV (1992) A novel super-secondary structure of β-proteins. A triple-strand corner. FEBS Lett 298:261–265
21. Efimov AV (1993) Super-secondary structures involving triple-strand β-sheets. FEBS Lett 334:253–256
22. Efimov AV (2008) Structural trees for proteins containing φ-motifs. Biochemistry (Moscow) 73:23–28
23. Sternberg MJE, Thornton JM (1976) On the conformation of proteins: the handedness of the β-strand-α-helix-β-strand unit. J Mol Biol 105:367–382
24. Suguna K et al (1987) Structure and refinement at 1.8 Å resolution of the aspartic proteinase from Rhizopus chinensis. J Mol Biol 196:877–900

25. Castillo RM et al (1999) A six-stranded double-psi beta barrel is shared by several protein superfamilies. Structure 7:227–236
26. Kargatov AM, Efimov AV (2010) A novel structural motif and structural trees for proteins containing it. Biochemistry (Moscow) 75:249–256
27. Efimov AV (1982) Super-secondary structure of β-proteins. Mol Biol (Moscow) 16: 799–806
28. Efimov AV (1995) Structural similarity between two-layer α/β and β-proteins. J Mol Biol 245:402–415
29. Richardson JS (1981) The anatomy and taxonomy of protein structure. Adv Protein Chem 34:167–339
30. Efimov AV (1977) Stereochemistry of packing of α-helices and β-structure in a compact globule. Dokl Akad Nauk SSSR 235:699–702
31. Crick FHC (1953) The packing of α-helices: simple coiled-coils. Acta Crystallogr 6:689–697
32. Chothia C, Levitt M, Richardson D (1977) Structure of proteins: packing of α-helices and pleated sheets. Proc Natl Acad Sci USA 74:4130–4134
33. Lim VI, Mazanov AL, Efimov AV (1978) A stereochemical theory of globular protein tertiary structure. I. Highly helical intermediate structures. Mol Biol (Moscow) 12:206–213
34. Richardson JS (1977) β-Sheet topology and the relatedness of proteins. Nature 268:495–500
35. Gordeev AB, Kondratova MS, Efimov AV (2008) Novel structural tree for β-proteins containing abcd-units. Mol Biol (Moscow) 42:323–326
36. Gordeev AB, Efimov AV (2009) Novel structural tree for (α+β)-proteins containing abCd-units. Mol Biol (Moscow) 43:521–526
37. Efimov AV (1997) A structural tree for proteins containing 3β-corners. FEBS Lett 407:37–41
38. Boshkova EA, Efimov AV (2010) Structures closed into cycles in proteins containing 3β-corners. Biochemistry (Moscow) 75:1258–1263
39. Chothia C, Janin J (1982) Orthogonal packing of β-pleated sheets in proteins. Biochemistry 21:3955–3965
40. Murzin AG, Lesk AM, Chothia C (1992) β-Trefoil fold. Patterns of structure and sequence in the Kunitz inhibitors, interleukin-1β and 1α and fibroblast growth factors. J Mol Biol 223:531–543
41. Murzin AG (1993) OB(oligonucleotide/oligosaccharide binding)-fold: common structural and functional solution for non-homologous sequences. EMBO J 12:861–867
42. Kister AE, Finkelstein AV, Gelfand IM (2002) Common features in structures and sequences of sandwich-like proteins. Proc Natl Acad Sci USA 99:14137–14141
43. Focas AS et al (2005) A geometric construction determines all permissible strand arrangements of sandwich proteins. Proc Natl Acad Sci USA 102:15851–15853
44. Kister AE et al (2006) Strict rules determine arrangements of strands in sandwich proteins. Proc Natl Acad Sci USA 103:4107–4110
45. Johannissen LO, Taylor WR (2003) Protein folds comparison by the alignment of topological strings. Prot Eng 16:949–955

Chapter 12

Computational Simulations of Protein Folding to Engineer Amino Acid Sequences to Encourage Desired Supersecondary Structure Formation

Bernard S. Gerstman and Prem P. Chapagain

Abstract

The dynamics of protein folding are complicated because of the various types of amino acid interactions that create secondary, supersecondary, and tertiary interactions. Computational modeling can be used to simulate the biophysical and biochemical interactions that determine protein folding. Effective folding to a desired protein configuration requires a compromise between speed, stability, and specificity. If the primary sequence of amino acids emphasizes one of these characteristics, the others might suffer and the folding process may not be optimized. We provide an example of a model peptide whose primary sequence produces a highly stable supersecondary two-helix bundle structure, but at the expense of lower speed and specificity of the folding process. We show how computational simulations can be used to discover the configuration of the kinetic trap that causes the degradation in the speed and specificity of folding. We also show how amino acid sequences can be engineered by specific substitutions to optimize the folding to the desired supersecondary structure.

Key words: Protein folding, Secondary structure, Tertiary structure, Kinetic traps, Amino acid substitution, Alpha helix, Energy landscape, Stability, Speed

1. Introduction

A major goal in molecular biology and biophysics is to predict if a sequence of amino acids will fold to a stable native state configuration (1–11). A practical aim associated with this quest is to develop the ability to design a sequence of amino acids to fold to a specific desired configuration. The capability to produce a wide range of "designer proteins" will have tremendous applications in molecular pharmaceutical medicine, nanoscience and nanotechnology, and commercial chemistry. Effective folding to a desired protein configuration requires a compromise between speed, stability, and

Alexander E. Kister (ed.), *Protein Supersecondary Structures*, Methods in Molecular Biology, vol. 932,
DOI 10.1007/978-1-62703-065-6_12,

specificity (12–17). We describe how computational modeling can be used to simulate the biophysical and biochemical interactions that determine protein folding. We provide examples of computational simulations on naturally occurring peptides and variants to illustrate how amino acid sequences can be engineered to encourage folding to the desired supersecondary structure.

Supersecondary structural elements in proteins are large enough to involve most of the different types of interactions experienced by amino acids. These include hydrogen bonds, dipole interactions, electrostatic interactions, hydrophobic and hydrophilic interactions, steric repulsion, as well as inherent propensities that some amino acids have for forming secondary structure such as α-helices or β-strands. There is strong coupling between the various types of amino acid interactions that create secondary, supersecondary, and tertiary interactions. For a system composed of more than a few amino acids, the multiple types of interactions and the coupling between the interactions make it extremely difficult to predict the structural folding dynamics of a protein by solving the equations of motion for the amino acids in an analytical fashion. Instead, computational simulations are used to find relationships between amino acid sequences and folding patterns. Computational approaches can be divided into two categories. Molecular Dynamics (MD) simulations calculate the forces exerted on each atom, or a group of atoms, at each of a series of time steps (18–20). The forces are used to calculate the acceleration experienced by each atom during that time step, and the accelerations are used to determine the changes in position for each of the atoms during each time step. Time increments are customarily on the order of a femtosecond. MD simulations are valuable for determining fine-scale details of structural changes for small protein motions that occur for timescales less than a microsecond. For large-scale structural changes that require longer times and involve many amino acids, such as folding into supersecondary structure, another computational approach involving lattice models is more useful. The data generated by lattice model computational simulations can be used to construct free energy surfaces which provide insight into the dynamics of the folding process. This allows the investigation of how strategic placement of specific amino acids can remove mis-folded structural traps and increase both the speed and reliability of the folding process. Such a design strategy is used by nature, such as in the GCN4 leucine zipper, and is an important consideration in the engineering of synthetic proteins.

As an example of supersecondary structure, we will focus on the two-helix bundle, shown in its correctly folded native state in Fig. 1a, c. The sequence of amino acids in the peptide chain is the same for both Fig. 1a, b, and is denoted as Sequence A. Sequence A produces two helices in which all four interfacial sidechains from each helix are hydrophobic. This results in a very stable native

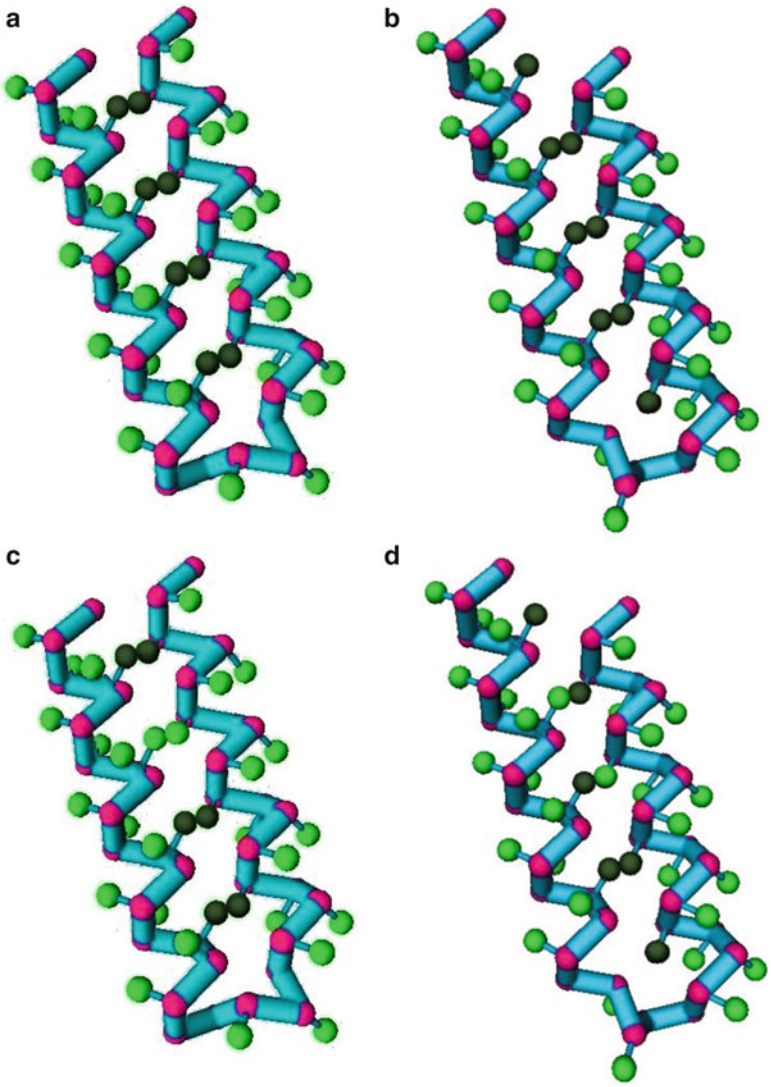


Fig. 1. Two-helix coil supersecondary structures (reproduced from (13) with permission from Wiley Periodicals, Inc.). The dark sidechains in the inside at the interface are hydrophobic and the light sidechains on the outside are hydrophilic. (**a**) Native state structure of Sequence A with all inter-helical sidechains are hydrophobic. (**b**) Nonnative, mis-folded configuration of Sequence A that can act as a kinetic trap and prevent efficient folding. (**c**) Native structure of Sequence B that has one inter-helical sidechain substituted to be hydrophilic. (**d**) Nonnative, mis-folded configuration of Sequence B that is no longer sufficiently stable to act as a kinetic trap.

state, if the native state configuration is attained, but has a detrimental effect on the kinetics and specificity of folding. The main complication for efficient folding is the possibility of mis-folding to the similar, but distinctly nonnative, structure shown in Fig. 1b. Computational simulations can be used to guide specific substitutions of amino acids to engineer peptide chains that avoid the

kinetic trap configuration of Fig. 1b. This produces peptides that are better folders, i.e., better for the combined attributes of speed, specificity, and stability. Figure 1c, d is the native structure and mis-folded structure of Sequence B, which is a modified version of Sequence A that requires only two amino acid substitutions. By replacing one interfacial amino acid on each helix with hydrophilic sidechains, Sequence B has improved folding performance compared to Sequence A. In Subheading 3, we describe these substitutions.

In Subheading 2 we describe how the information generated from computational simulations of protein folding can be used to determine the speed, specificity, and stability of folding into supersecondary structure of a specific sequence of amino acids. In Subheading 3 we explain how information from the computer simulations is used to guide amino acid substitutions to enhance the folding of the peptide. In Subheading 4 we include observations and tips for implementing the computations and calculations.

2. Materials

Implementing a computer lattice model for studying protein dynamics requires a choice for the underlying lattice, a representation of an amino acid on the lattice, a Hamiltonian (energy function) for calculating the energy of a peptide configuration, and a set of rules that determines how amino acids move on the lattice and thereby change the structural configuration of the chain. The rules for changing the peptide configuration are implemented in intervals using Monte Carlo algorithm. Therefore, the time steps are given in units of Monte Carlo (MC) steps. A variety of lattice computer models have been used for investigating protein folding (21–24). The model that we use (25) was developed from an earlier model (26, 27). The model has been shown to be effective at representing realistic protein dynamics (28–32) and more recently has been applied to protein dimerization and aggregation (33, 34).

Here we describe how the information generated from computational simulations of protein folding can be used to determine the speed, specificity, and stability of folding into supersecondary structure of a specific sequence of amino acids.

2.1. Thermodynamic Stability of the Native State: Heat Capacity Calculations

In order to determine if a sequence of amino acids is a good folder, it is necessary to calculate its free-energy landscape ($F = E - TS$) at different temperatures, both above and below the transition temperature T' at which the peptide is equally likely to be in either its native state configuration or nonnative configuration. To determine T', and subsequently F, we first plot the heat capacity C_v over

a range of temperatures and determine T' by the location of the peak of C_v. The equation for determining C_v at a given temperature T is

$$C_v(T) = \frac{\overline{E^2} - \overline{E}^2}{T^2} \tag{1}$$

The most accurate way to determine $C_v(T)$ throughout a range of temperatures, $\overline{E^2}$ and $\overline{E}^2$ must be determined at each temperature. Alternatively, $C_v(T)$ can be calculated throughout a range of temperatures less accurately but more quickly by using the data generated from simulations at a single temperature and employing the histogram technique. The histogram technique allows the determination of the average of any thermodynamic quantity Q (such as E or E^2) at many temperatures using the data generated from a single simulation at temperature T_s:

$$\overline{Q(T)} = \frac{\sum_r Q(E_r) h(E_r;T_s) e^{\frac{E_r}{R}\left(\frac{1}{T_s}-\frac{1}{T}\right)}}{\sum_r h(E_r;T_s) e^{\frac{E_r}{R}\left(\frac{1}{T_s}-\frac{1}{T}\right)}} \tag{2}$$

where the gas constant $R = 1.99 \times 10^{-3}$ kcal/mol K, E_r denotes any of the energy values that the peptide can have, and $h(E_r;T_s)$ is the number of times that a specific E_r occurs during a simulation. Using the results of the computational simulations, the following steps will create a plot of C_v as a function of T.

1. At any simulation temperature T_s, generate a time series of the energy of the peptide chain that encompasses multiple folding and unfolding processes (see Notes 1 and 2).
2. Sort the energy time series to determine the number of times $h(E_r;T_s)$ a specific energy E_r occurs (see Note 3).
3. Using Eq. 2, calculate $\overline{E^2}$ and $\overline{E}^2$ for a series of T below and above T_s.
4. Using Eq. 1, plot $C_v(T)$. For the two-helix bundle of Fig. 1 (sequence A), the C_v plot is shown in Fig. 2. The location of the peak is an approximate value of T' (see Note 4).

The peak in C_v that represents the transition T' also provides information on the stability of the native state. The lower temperature (left) side of the peak is the region in which the native state is the preferred stable configuration. Therefore, a higher value for T' means that the native state remains stable at higher temperatures.

2.2. Free-Energy Landscapes

Engineering proteins by making amino acid substitutions to increase the speed, reliability, and stability of the native state requires knowledge of the free-energy landscape. The free-energy at a given temperature is a function of various structural parameters

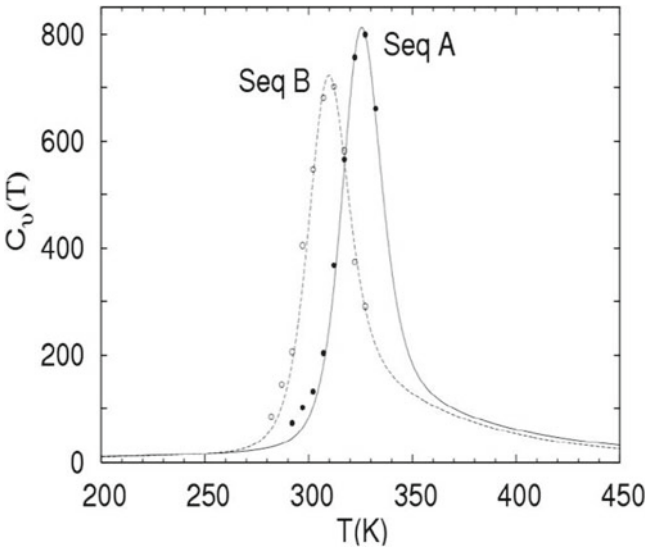


Fig. 2. Heat capacity C_v as a function of temperature for peptide Sequence A and for peptide Sequence B.

denoted as x_1, x_2,…; $F(T) = F(T;x_1,x_2)$. To picture the free-energy landscape, we construct graphs of F as a function of only two structural parameters in order to be able to depict and understand the graphs easily. For the two-helix bundle, we find that the most important structural parameters are q (the fraction of amino acids that have taken on a helical secondary structure configuration), Q (the fraction of native (correct) interhelix supersecondary contacts that are formed), and d_{cc} (the end-to-end distance of the chain). Other structural parameters, such as nonnative contacts, can also be helpful for clarifying the configuration of nonnative kinetic traps, as discussed in Subheading 3.2. When the peptide has attained the native two-helix secondary structure, $q \sim Q \sim 1$ and d_{cc} are small. In order to create 3-D graphs of $F(q,d_{cc})$ and $F(Q,d_{cc})$ at a specific temperature from the computer simulations, we use the following expression:

$$F(T;x_1,x_2) = -RT \ln P(T;x_1,x_2) \qquad (3)$$

where $P(T;x_1,x_2) = N(T;x_1,x_2)/N$ is the probability that a configuration with specific values of structural parameters x_1,x_2 (e.g., $Q = 0.83$, $d_{cc} = 27$ lattice units) appears during a simulation. N is the total number of frames in the simulation and $N(T;x_1,x_2)$ is the number of frames in which a configuration with specific values of the structural parameters occurs. To produce plots of $F(T;x_1,x_2)$, the following procedure is used.

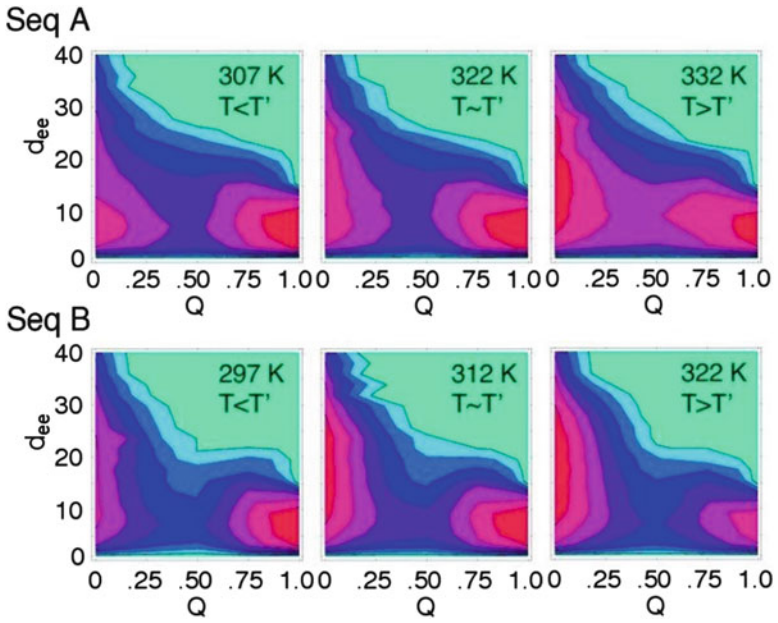


Fig. 3. Three-dimensional contour plots of the free-energy of a configuration as a function of end-to-end distance d_{ee} and native inter-helical contacts Q (reproduced from (13) with permission from Wiley Periodicals, Inc.). Red denotes low free-energy, blue denotes high free-energy. The extended, unstructured random coil configuration has $d_{ee} > 20$, $Q < 0.25$. The native state has $d_{ee} < 15$, $Q > 0.75$. A compact, nonnative structure that acts as a kinetic trap for Sequence A has $d_{ee} < 15$, $Q < 0.25$.

1. Run simulations at three different temperatures: $T_s = T' - 15$ K, T', and $T_s + 15$ (see Note 5).
2. For each T_s, generate time series for E and for the structural parameters q, Q, and d_{cc}. It is also useful to generate time series for nonnative structural parameters (see Notes 6 and 7).
3. For each T_s, create tables of $F(q,Q)$, $F(q,d_{cc})$, and $F(Q,d_{cc})$.
4. For each T_s, create three-dimensional plots of $F(q,Q)$, $F(q,d_{cc})$, and $F(Q,d_{cc})$. For peptide Sequence A, these plots are displayed in Fig. 3. The importance of these plots is explained in Subheading 3.1 (see Note 8).

2.3. Kinetics of Folding and Unfolding: Median First Passage Time

The heat capacity plot of Subheading 2.1 is useful for determining the stability of the native state, and the free-energy plots of Subheading 2.2 are useful for determining the stability and also the rate of folding and unfolding. To get precise information about the kinetics of folding and unfolding, we directly use the time information obtained from the simulations. As explained in Note 1, we suggest that many simulations be used in which each simulation stops at the frame at which the peptide has achieved its native state (see Note 9), and similarly for unfolding simulations. In addition to creating tables of the evolution of E, q, Q, and d_{cc} for each simulation, separate files should be created for time data.

1. Create a file, e.g., foldingtime.data. For each T_s, when each folding simulation finishes, the length of time of the simulation is written to the file.
2. Create a file, e.g., unfoldingtime.data. For each T_s, when each unfolding simulation finishes, the length of time is written to the file.
3. For each T_s, order the times in the files in ascending order. This will create two files: orderedfoldingtime.data and orderedunfoldingtime.data.
4. If N folding (unfolding) simulations were used (e.g., $N=100$), choose the entry in place $N/2$ (i.e., place 50). This is a representative time for folding (unfolding) known as the median first passage time (MFPT).
5. Plot the folding MFPT as a function of temperature.
6. Plot the unfolding MFPT as a function of temperature.

3. Methods

The information obtained from the various subsections of the Subheading 2 can be used to discern the underlying dynamics of the folding (unfolding) process. This information can then be used to guide amino acid substitutions to enhance the folding of the peptide.

3.1. Detecting Kinetic Trap Configurations Using Free-Energy Landscapes and Survival Probability

Figure 1b shows a nonnative, compact structure of the two-helix bundle that acts as a trap that slows down the folding process to the native state structure of Fig. 1a. The existence of a kinetic trap structure can be detected by using the free-energy landscapes described in Subheading 2.2. A compact, nonnative configuration can act as a kinetic trap if it is a local free-energy minimum and there is a straightforward route in the free-energy landscape. Inspection of the free-energy plots and how they change at different T_s should expose folding routes as well as kinetic trap structures. In Fig. 3, the initial, extended, unstructured random coil configuration has $d_{cc}>20$, $Q<0.25$ and the native state has $d_{cc}<15$, $Q>0.75$. Sequence A also displays a local minimum of the free-energy in the region of $d_{cc}<15$, $Q<0.25$ that is a compact, nonnative structure that acts as a kinetic trap. This is most apparent in the plot at $T<T'$.

Determining if this nonnative free-energy minimum is actually a kinetic trap that slows down the folding rate requires investigation of the kinetics. The MFPT described in Subheading 2.3 provides a representative time for the folding process and the unfolding process, and how the rates vary with temperature. Additional and

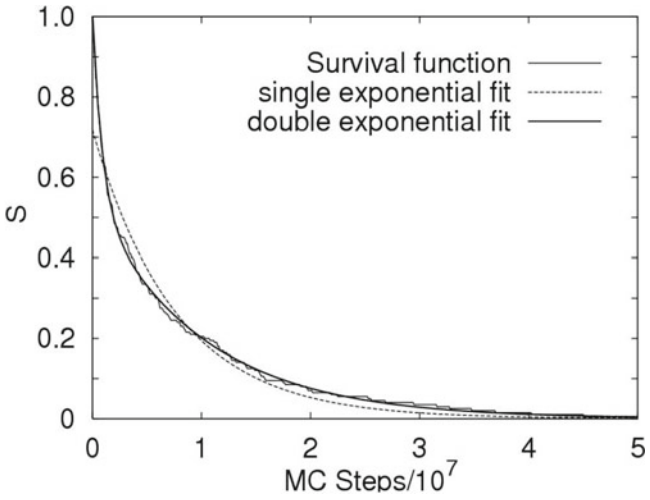


Fig. 4. Representative fits with single and double exponential functions to the survival function (*wavy line*) obtained from computer simulations of Sequence A at 307 K ($T < T'$).

more detailed kinetic information can be obtained, and the detection of routes leading to kinetic traps, if survival probabilities are employed. Survival probability is calculated using

$$S(t,T) = 1 - n(t,T_s) \tag{4}$$

where $n(t,T)$ is the fraction of simulations that have succeeded in folding after time t. For example, if 36 of 100 folding simulations have succeed after 23,000 frames for simulations run at temperature $T_s = 310$ K, and then $S(23 \times 10^3, 310\ \text{K}) = 1 - (36/100) = 0.64$. The information needed is contained in the files created in **step 3** of Subheading 2.3: orderedfoldingtime.data.

1. Create a table and graph of $S(t,T)$ for folding (unfolding). See Fig. 4.
2. Attempt to fit $S(t,T)$ using a single exponential: $S(t,T) = ae^{-t/\tau}$. If the fit is good, then the folding (unfolding) process likely involves only one stable compact configuration.
3. If a single exponential does not produce a good fit (not a two-state system), try fitting with other appropriate functions. As an example of kinetics other than a single exponential, we show a fit using a double exponential, $S(t,T) = a_1 e^{-t/\tau 1} + a_2 e^{-t/\tau 2}$ (see Table 1 and Fig. 4). A good fit using a double exponential function implies that the process may involve two compact configurations, one of which is the native state, and the other a nonnative kinetic trap (see Note 10). This information can be corroborated with local minima in the free-energy landscapes.

Table 1
Fit parameters used for the double exponential fit $S = a_1 e^{-t/\tau_1} + a_2 e^{-t/\tau_2}$ for the survival function S in Fig. 4

T(K)	N_1	a_1	τ_1 (MC)	N_2	a_2	τ_2 (MC)
297	96	0.99	6.08×10^5	104	1.02	1.13×10^7
302	106	1.02	8.85×10^5	94	1.14	1.02×10^7
307	93	1.01	8.86×10^5	107	1.42	5.72×10^6
312	95	1.01	9.49×10^5	105	1.24	6.12×10^6
317	95	0.99	1.09×10^6	105	1.38	4.69×10^6
322	99	1.04	1.41×10^6	101	1.56	4.25×10^6
327	108	1.01	1.91×10^6	92	1.10	7.72×10^6

N_1 is the number of routes, out of a total of 200 runs, following route 1, and N_2 is the number of remaining runs following route 2. The characteristic times are given in MC steps

3.2. Determining Structural Elements That Stabilize Kinetic Traps

If the free-energy landscape and analyses of the kinetics provide evidence of a kinetic trap configuration, it may be possible to remove the kinetic trap by modifying a small number of amino acids so that the free-energy of the kinetic trap configuration is increased and it no longer acts as a trap. There are two ways to determine the structural elements that provide stability for the kinetic trap. The first requires a frame-by-frame movie of the computer simulation of the folding process. The second method requires the files with time series of nonnative structural elements (see step 2 in Subheading 2.2 and Note 7).

1. Plot the energy time series, native structural parameters' time series, and the nonnative structural parameters' time series.
2. Determine specific times during the folding process when the energy of the peptide is low but the configuration of the peptide contains little native structural elements.
3. At these times, determine which nonnative structural elements (see Note 7) have formed.
4. Inspect the corresponding frame in a visualization application to confirm the presence of the stabilizing nonnative structural elements.

4. Amino Acid Substitutions to Remove the Kinetic Trap

If the nonnative structural elements that stabilize the kinetic trap involve a small number of amino acids, it may be possible to make substitutions for these amino acids to remove the interactions that

stabilize the kinetic trap structure. We were able to accomplish this for the two-helix bundle. After following the steps listed in this section, we determined that the kinetic trap structure is the configuration displayed in Fig. 1b in which the two helices are misaligned. Instead of the four native, hydrophobic inter-helical contacts of Fig. 1a, the kinetic trap structure of Fig. 1b contains three stabilizing, nonnative, hydrophobic inter-helical contacts.

We realized that we could greatly destabilize the nonnative structure of Fig. 1b without significantly destabilizing the native structure of Fig. 1a by substituting only one hydrophobic side chain on each helix with a hydrophilic sidechain. This modified peptide is called Sequence B. The resulting structures are shown in Fig. 1c, d. The native structure of Sequence B shown in Fig. 1c has three stabilizing hydrophobic–hydrophobic inter-helical contacts which maintains its free-energy almost as low as the native state of Sequence A which has four hydrophobic–hydrophobic inter-helical contacts. The small shift in the peak of C_v shown in Fig. 2 shows that the native structure of Sequence B is almost as stable as the native structure of Sequence A. However, the nonnative configuration of Sequence B shown in Fig. 1d has only one stabilizing hydrophobic–hydrophobic inter-helical contact and, as shown in Fig. 3, its free-energy is significantly higher than that of the kinetic trap structure of Sequence A shown in Fig. 1b which has three stabilizing hydrophobic–hydrophobic inter-helical contacts. The free-energy of the nonnative configuration in Fig. 1d for Sequence B is high enough that it does not act as a kinetic trap.

Evidence that the configuration of Fig. 1d for Sequence B does not act as a kinetic trap was also obtained from the kinetics. After repeating the steps of Subheading 2.3 for Sequence B, we found that at all temperatures, the MFPT for folding was shorter for Sequence B than for Sequence A. Also, after repeating the steps of Subheading 3.1 for Sequence B, we found that we could fit the survival probability with only one exponential, and the characteristic time parameter for the single exponential process of Sequence B was similar to the characteristic time of the faster of the two exponential processes used to fit the survival probability for Sequence A. Thus, the engineered peptide of Sequence B has almost the same stability as Sequence A, but is a much faster and more reliable folder.

5. Notes

1. A times series that encompasses multiple folding and unfolding events (e.g., 100 folding and 100 unfolding) can be accomplished in two ways: (a) a single, long computer simulation or (b) combining 100 separate folding simulations interleaved

with 100 separate unfolding simulations. It is better to use the second method. The first method that uses one, very long simulation has the serious problem that during a long simulation, the peptide may get trapped in a configuration because it is constrained to move on a lattice. This can result in the chain spending an enormous time during a simulation, and incorrectly skewing the data, in a configuration that a real protein could easily escape. This can happen in a short simulation, but since many short simulations are used, the data is skewed to a much smaller extent.

2. "Short" simulations are terminated at the first frame at which they have successfully folded (unfolded) to (away from) the native state. To prevent a "short" simulation run from spending too much time in an unphysical, lattice-induced configuration which will prevent folding from ever occurring, folding (unfolding) simulations should be terminated after a fixed end time, τ_c (35). For computational efficiency, τ_c should be set to approximately four times the median folding (unfolding) time. An estimate of the median folding time can only be made after several simulations have been allowed to run long enough to fold (unfold).
3. There may be so many different E_r that each one may appear only a small number of times in the energy time series and $h(E_r;T_s)$ is never larger than 10. In this case, the summations required by Eq. 2 will involve many terms. The process can be sped-up with little decrease in accuracy by collecting the E_r into bins. If the largest bins have sizeable $h(E_r;T_s)$, there will be fewer terms in the summations when Eq. 2 is employed.
4. The value for C_v that is obtained by the histogram method becomes increasingly unreliable for T far from T_s. Therefore, if the first time $T' = T'(1)$ is calculated and it is far (>20 K) from the first $T_s = T_s(1)$, this value of $T'(1)$ may be unreliable. To obtain a more reliable value for T' $[T'(2)]$, the entire process should be repeated using a computer simulation that is run at a new $T_s(2) = T'(1)$. The new $T' = T'$ (2) will be more reliable if $|T_s(2) - T'(2)| < |T_s(1) - T'(1)|$.
5. If the free-energy landscapes at these three temperatures display a variety of important features, simulations at additional interpolated temperatures may be necessary to understand folding routes.
6. The structural parameters of importance depend on the super-secondary structure of the native state. If β-structure is important in the native state, then the fraction of amino acids in β-structure should be used. If α-helix content does not play a role in any important configurations, then q is not worth monitoring.

7. Later, it will be important to clarify the structure of nonnative structures that act as kinetic traps. To help with this clarification, it is helpful at this stage to also monitor nonnative structural elements within a configuration. For example, for the two-helix bundle, important structural elements to monitor are interhelical contacts that are nonnative, such as those displayed in Fig. 1b. If important nonnative structural elements are not anticipated and tabulated at this stage along with the native structural elements (e.g., Q), it may become necessary to rerun the exact same simulations in order to tabulate the nonnative information.
8. In Fig. 3, we only display the plot of $F(Q,d_{cc})$ because the plots of $F(q,Q)$ and $F(q,d_{cc})$ did not reveal information that was useful for perceiving kinetic trap structures. This shows the importance of creating free-energy landscapes as functions of many different structural parameters because it is difficult to know *a priori* which will be useful.
9. Achieving the native state is defined computationally as having achieved or exceeded specific values of q, Q, and E.
10. There is a limit to the level of detail about the underlying dynamics that can be obtained by fitting the $S(t,T)$ functions. Care must be used in distinguishing between fits employing multiple exponentials versus stretched exponentials, and their associated underlying dynamics.

References

1. Eaton WA, Muñoz V, Hagen SJ et al (2000) Fast kinetics and mechanisms in protein folding. Annu Rev Biophys Biomol Struct 29:327–359
2. Gerstman BS, Chapagain PP (2008) Self-organizing dynamics in protein folding. In: Conn PM (eds) Molecular biology of protein folding, part B, progress in molecular biology and translational science, vol 84, p 1–37. Elsevier, ISBN 978-0-12-374595-8, ISSN 0079–6603
3. Dobson CM, Karplus M (1999) The fundamentals of protein folding: bringing together theory and experiment. Curr Opin Strcut Biol 9:92–101
4. Gerstman BS, Chapagain PP (2005) Self-organization in protein folding and the hydrophobic interaction. J Chem Phys 123(054901):1–6
5. Alm E, Baker D (1999) Matching theory and experiment in protein folding. Curr Opin Struct Biol 9:189–196
6. Onuchic JN, Luthey-Schulten Z, Wolynes PG (1997) Theory of protein folding: the energy landscape perspective. Annu Rev Phys Chem 48:545–600
7. Mayor U, Johnson CM, Daggett V et al (2000) Protein folding and unfolding in microseconds to nanoseconds by experiment and simulation. Proc Natl Acad Sci U S A 97:13518–13522
8. Honig B (1999) Protein folding: from the levinthal paradox to structure prediction. J Mol Biol 293:283–293
9. Mirny LA, Abkevich VI, Shakhnovich EI (1998) How evolution makes proteins fold quickly. Proc Natl Acad Sci U S A 95:4976–4981
10. Shakhnovich EI, Gutin AM (1993) Engineering of stable and fast-folding sequences of model proteins. Proc Natl Acad Sci USA 90: 7195–7199
11. Scalley-Kim M, Baker D (2004) Characterization of the folding energy landscapes of computer generated proteins suggests high folding free energy barriers and cooperativity may be consequences of natural selection. J Mol Biol 338:573–583
12. Vendruscolo M, Paci E, Dobson CM et al (2001) Three key residues form a critical contact network in a protein folding transition state. Nature 409:641–645

13. Chapagain PP, Gerstman BS (2006) Removal of kinetic traps and enhanced protein folding by strategic substitution of amino acids in a model α-helical hairpin peptide. Biopolymers 81:167–178
14. Gilmanshin R, Williams S, Callender RH et al (1997) Fast events in protein folding: relaxation dynamics of secondary and tertiary structure in native apomyoglobin. Proc Natl Acad Sci U S A 94:3709–3713
15. Mayor U, Guydosh NR, Johnson CM et al (2003) The complete folding pathway of a protein from nanoseconds to microseconds. Nature 421:863–867
16. Myers JK, Oas TG (2001) Preorganized secondary structure as an important determinant of fast protein folding. Nat Struct Biol 8:552–558
17. Zhou Y, Karplus M (1999) Interpreting the folding kinetics of helical proteins. Nature 401:400–403
18. Brooks BR, Bruccoleri RE, Olafson BD et al (1983) CHARMM: a program for macromolecular energy, minimization, and dynamics calculations. J Comp Chem 4:187
19. Eswar N, Webb B, Marti-Renom MA et al (2006) Comparative protein structure modeling using MODELLER. Curr Protoc Bioinform. Chapter 5, Unit 5 6.
20. Feig M, Karanicolas J, Brooks CL (2004) MMTSB tool set: enhanced sampling and multiscale modeling methods for applications in structural biology. J Mol Graph Model 22:377
21. Dill KA, Chan HS (1997) From levinthal to pathways to funnels. Nat Struct Biol 4:10–19
22. Dinner AR, Sali A, Karplus M (1996) The folding mechanism of larger model proteins: role of native structure. Proc Natl Acad Sci U S A 93:8356–8361
23. Muñoz V, Eaton WA (1999) A simple model for calculating the kinetics of protein folding from three-dimensional structures. Proc Natl Acad Sci U S A 96:11311–11316
24. Dill KA, Bromberg S, Yue K et al (1995) Principles of protein folding—a perspective from simple exact models. Protein Sci 4:561–602
25. Liu Y, Chagagain PP, Parra JL et al (2008) Lattice model simulation of interchain protein interactions and the folding dynamics and dimerization of the GCN4 leucine zipper. J Chem Phys 128(045106):1–10
26. Skolnick J, Kolinski A (1990) Simulations of the folding of a globular protein—science. Science 250:1121–1125
27. Skolnick J, Kolinski A (1991) Dynamic Monte Carlo simulations of a new lattice model of globular protein folding, structure and dynamics. J Mol Biol 221:499–531
28. Chapagain PP, Gerstman BS (2003) Finite size scaling of structural transitions in a simulated protein with secondary and tertiary structure. J Chem Phys 119:1174–1180
29. Kolinski A, Milik M, Skolnick J (1991) Static and dynamic properties of a new lattice model of polypeptide chains. J Chem Phys 94:3978–3985
30. Kolinski A, Skolnick J (1994) Monte Carlo simulations of protein folding. I. Lattice model and interaction scheme. Proteins 18:338–352
31. Chapagain PP, Gerstman BS (2004) Excluded volume entropic effects on protein unfolding times and intermediary stability. J Chem Phys 120(5):2475–2481
32. Gerstman B, Garbourg Y (1998) J Polym Sci, Part B: Polym Phys 36:2761–2769
33. Chagagain PP, Liu Y, Gerstman BS (2008) The trigger sequence in the leucine zipper: α-helical propensity dependence of folding and dimerization. J Chem Phys 129(175103):1–9
34. Liu Y, Chapagain PP, Gerstman BS (2010) Stabilization of native and non-native structures by salt bridges in a lattice model of the GCN4 leucine dimer. J Phys Chem B 114(2):796–803
35. Chapagain PP, Gerstman BS, Bhandari Y et al (2011) Free energy landscapes and thermodynamic parameters of complex molecules from non-equilibrium simulation trajectories. Phys Rev E 83(6):061905

Chapter 13

Protein Folding at Atomic Resolution: Analysis of Autonomously Folding Supersecondary Structure Motifs by Nuclear Magnetic Resonance

Lorenzo Sborgi, Abhinav Verma, Mourad Sadqi, Eva de Alba, and Victor Muñoz

Abstract

The study of protein folding has been conventionally hampered by the assumption that all single-domain proteins fold by an all-or-none process (two-state folding) that makes it impossible to resolve folding mechanisms experimentally. Here we describe an experimental method for the thermodynamic analysis of protein folding at atomic resolution using nuclear magnetic resonance (NMR). The method is specifically developed for the study of small proteins that fold autonomously into basic supersecondary structure motifs, and that do so in the sub-millisecond timescale (folding archetypes). From the NMR experiments we obtain hundreds of atomic unfolding curves that are subsequently analyzed leading to the determination of the characteristic network of folding interactions. The application of this approach to a comprehensive catalog of elementary folding archetypes holds the promise of becoming the first experimental approach capable of unraveling the basic rules connecting protein structure and folding mechanism.

Key words: Protein structure, Protein stability, Protein folding, Folding mechanisms, Folding interaction networks, Nuclear magnetic resonance

1. Introduction

Protein folding is an inherently complex process that involves coordination of the networks of weak interactions that stabilize native 3-D structures. Such complexity owes to the rich variety of structural patterns observed in natural proteins, which suggests a wide range of folding behaviors. Another important factor is the vast heterogeneity of microscopic folding routes that are explored by any given protein according to theory (1) and computer simulations (2). To make matters even more difficult, the conventional paradigm states that single-domain proteins fold in a two-state fashion in which only two

Alexander E. Kister (ed.), *Protein Supersecondary Structures*, Methods in Molecular Biology, vol. 932,
DOI 10.1007/978-1-62703-065-6_13,

populations of molecules are ever observed—native and unfolded (3). The two-state model implies that folding is an all-or-none process with mechanisms that cannot be resolved experimentally. All these factors combined explain why protein folding is still an unresolved problem in spite of many decades of intensive research.

However, in the last years there have been critical developments that are quickly changing this state of affairs. The first one is the realization that supersecondary structure motifs built by simple combinations of α-helices, β-hairpins, β-turns, and short loops can fold autonomously into stable native structures. Therefore, we now know several examples of amino acid sequences that fold into helix bundles, helix-loop-helix, α-helix-β-hairpin, Greek-key motifs, etc. (4, 5). These molecules make building a comprehensive catalog of autonomously folding supersecondary structure motifs, or folding archetypes, feasible. Thus, given that protein structure is organized hierarchically, the folding mechanism of any given protein could perhaps be described as a combination of the mechanisms observed on such archetypical catalog (6).

The application of ultrafast kinetic methods, and most notoriously the laser-induced temperature-jump technique, to the experimental study of the folding kinetics of several folding archetypes has shown that these small proteins (30–80 residues) with simple topologies fold in the microsecond timescale (5), and very closely to the empirical estimates of the folding speed limit (4, 7). By the same token, calorimetric analysis of the thermal unfolding process indicates that such fast-folding archetypes can be classified within the downhill folding scenario (8, 9). In other words, this implies that their folding free energy barrier is less than 3 *RT* at all conditions (10). As a side effect of their downhill folding character, stable supersecondary structure motifs tend to exhibit broad equilibrium thermal unfolding transitions with low cooperativity (5) that look different depending on the structural probe used to monitor unfolding (i.e., probe dependence) (11, 12).

Probe dependence is a particularly interesting feature because it suggests that folding mechanisms might be resolvable at atomic resolution in standard equilibrium unfolding experiments using nuclear magnetic resonance (NMR) spectroscopy (11). By building on this idea, we recently developed an NMR approach to monitor protein thermal unfolding at the level of individual atoms, and an analytical procedure to determine the folding interaction network and mechanism from pairwise comparisons between the hundreds of atomic unfolding curves obtained by NMR (13). One of the critical requirements for this method is that folding is fast relative to the characteristic NMR timescale, so that the protein is in the fast conformational exchange regime. Thus, the method seems perfectly suited for the microsecond folding kinetic processes that are found on folding archetypes. Indeed, the applicability of the method was originally demonstrated on the helix-loop-helix downhill folding protein BBL (13). More recently, we extended it

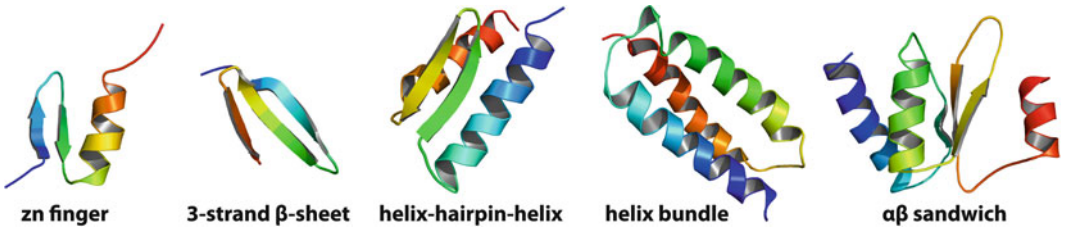


Fig. 1. Examples of small proteins that fold autonomously into elementary supersecondary structure motifs and can be used to build a catalog of folding archetypes.

to the study of gpW, another microsecond-folding archetype with a helix-hairpin-helix topology (14). Here, we describe the experimental method in detail, delving into all the technical and practical issues that need to be considered towards achieving a successful implementation. We also discuss how to perform the computational analysis of atomic unfolding curves to calculate the folding interaction matrix, and introduce a Web application (available at http://www.tmg.cib.csic.es/servers) that performs the complete atom-by-atom analysis of folding archetypes.

2. Materials

2.1. Candidate Proteins

2.1.1. Autonomously Folding Supersecondary Structure Motifs

The ideal targets for this analysis are elementary supersecondary structure motifs that are able to fold autonomously and are stable (folding archetypes). Typical archetypes are α-helix bundles, β-hairpins, helix-loop-helix, helix-hairpin motifs (e.g., zinc fingers), and minimal α-β parallel folds (Fig. 1). Suitable examples can be fished out from the protein structure database (PDB), or designed de novo using bioinformatic tools such as Modeller (15).

2.1.2. Microsecond Folders

To be suitable for the NMR analysis of folding at atomic resolution candidate proteins should fold–unfold in times faster than the characteristic NMR timescale so that the unfolding process is in the fast conformational-exchange regime at all conditions (Note 1). A simple procedure to ascertain whether a given candidate for folding archetype is likely to fold in the appropriate timescale is to carry out a prediction of folding and unfolding rates using the bioinformatics tool PREFUR, which only requires protein size and structural class ascription as input (16). PREFUR is freely available at http://www.tmg.cib.csic.es/servers.

2.2. NMR Samples and Experiments

2.2.1. NMR Samples

- Ethylene glycol (0.6 ml) in a regular 5 mm diameter NMR tube for temperature calibration of a 5 mm diameter triple resonance NMR probe.
- [^{15}N, ^{13}C]-uniformly enriched protein sample (>95 % pure) at approximately 1 mM concentration and 0.5 ml volume.

- Deuterium-enriched buffers and salts to alleviate interference from the buffer NMR signals.
- Deuterium oxide as reference signal for the NMR spectrometer.
- DSS at 0.01 mM concentration for chemical shift referencing.
- Helium or other inert gas.
- Pressure valve NMR tubes of 5 mm diameter and medium thick wall (0.77 mm) (Note 2).

2.2.2. NMR Experiments

- High-field spectrometers for protein NMR techniques equipped for multidimensional NMR, including triple-resonance probes (^{1}H, ^{15}N, ^{13}C) that can operate at high temperatures (beyond 373 K) preferably with a fine temperature control. For example Bruker 5 mm TXI-probes with z-axis gradient and BTO temperature control units can stand up to 423 K.
- Implemented standard NMR experiments for protein backbone and side chain chemical shift assignment including 2D-[^{1}H-^{15}N]-HSQC, 3D-CBCA(CO)NH, 3D-HNCACB, 3D-HNCO, 3D-HBHA(CO)NH, 3D-H(CCO)NH, 3D-(H)C(CO)NH, 3D-HCCH-TOCSY, 3D-[^{15}N]-NOESY-HSQC, and 4D-[^{1}H-^{13}C]-HMQC-NOESY-HSQC (17, 18).
- Software for NMR experiment processing and analysis, for example NMRPipe (19) and PIPP (20), respectively.

2.3. Computational Analysis

For all the computational analyses described in this work we have used the Matlab package for numerical data analysis with custom-made programs. We have implemented the whole data analysis routine (including the analysis of atomic unfolding curves, the clustering procedures, and the calculation of the TCI matrix) into a Web application that is freely available for academic use at http://www.tmg.cib.csic.es/servers. If preferred, the fitting of atomic unfolding curves to a two-state or a three-state model can also be carried out with other data analysis programs such as Sigmaplot or Origin. The network graphs were performed with the graph plotting software Visone (21) (available freely at http://www.visone.info).

3. Methods

3.1. Determining Suitability for the Atomic-Resolution Analysis of Folding by NMR

3.1.1. Reversible Equilibrium Thermal Unfolding

The analysis we describe here requires a fully reversible thermal unfolding process so that the process can be considered under thermodynamic equilibrium at all conditions. In general, the reversibility of the thermal unfolding process should be assessed before performing all the NMR experiments. This can be done preparing an NMR sample at typical protein concentrations. The sample is then heated up to 100 C for a period of 2 h and then cooled back down. Using a simple fingerprint NMR experiment such as

[^{1}H-^{15}N]HSQC (which can be performed even on non-isotopically labeled samples using the SO-FAST pulse sequence (22)), a reversible unfolding process would result in superimposable signals when the spectra acquired before and after heating are overlaid.

3.1.2. Folding Kinetics in the Fast Conformational-Exchange NMR Regime

A second important initial test to carry out is to determine whether the folding archetype folds in the fast conformational-exchange regime. This can be simply assessed on the NMR magnet by acquiring [^{1}H-^{15}N]HSQC spectra at low temperature, at the global denaturation midpoint (determined previously on a standard thermal unfolding experiment monitoring folding by a low-resolution probe such as circular dichroism), and at very high temperature. If the protein folds in the fast conformational exchange regime, the [^{1}H-^{15}N]HSQC spectrum at the denaturation midpoint will produce a single set of crosspeaks without significant line broadening. This set of crosspeaks should be approximately halfway between those measured at low and high temperatures.

3.2. NMR Experiments to Monitor Protein Unfolding with Atomic Resolution

3.2.1. Temperature Calibration of the NMR Probe

The NMR probe is calibrated for each temperature used in the study at an airflow rate of 535 l/h following the ethylene glycol chemical shift temperature dependence as described in Amman et al. (23) for the 272–416 K range. Standard one-pulse 1D ^{1}H-NMR experiments are used to monitor chemical shift changes with temperature. The ethylene glycol sample is equilibrated for 30 min at each temperature before the spectrum is acquired. The calibration typically ranges from 273 to 370 K.

3.2.2. Chemical Shift Referencing

One-pulse 1D ^{1}H-NMR experiments are used to calibrate the position of the water signal relative to DSS (0.0 ppm) at each temperature. The resonance frequency of the water determined this way is used as reference in all multidimensional triple-resonance NMR experiments for chemical shift assignment, as the isotope filtering of these experiments precludes the observation of the DSS NMR signal.

3.2.3. Chemical Shift Assignment

Protein backbone amide ^{15}N, $^{13}C_{\alpha}$, and $^{13}C_{\beta}$ chemical shifts can be obtained with a basic set of experiments: 2D-[^{1}H-^{15}N]-HSQC, 3D-CBCA(CO)NH, and 3D-HNCACB. Side chain ^{1}H and ^{13}C chemical shifts are assigned with the experiments: 3D-HBHA(CO)NH, 3D-H(CCO)3D-H(CCO)NH, and 3D-HCCH-TOCSY. All assignments are confirmed with the experiments: 3D-[^{15}N]-NOESY-HSQC and 4D-[^{1}H-^{13}C]-HMQC-NOESY-HSQC. A special procedure might be needed to determine the CS values precisely at low temperatures (Note 3).

3.2.4. Monitoring Protein NMR Signal Change with Temperature

Pressure valve tubes are used to keep the protein sample at a He pressure of approximately 8 bar to minimize evaporation at high temperature. Temperature intervals should be small enough to easily follow chemical shift changes of the protein NMR signals. These intervals are typically from 3 to 5 K (Fig. 2). However,

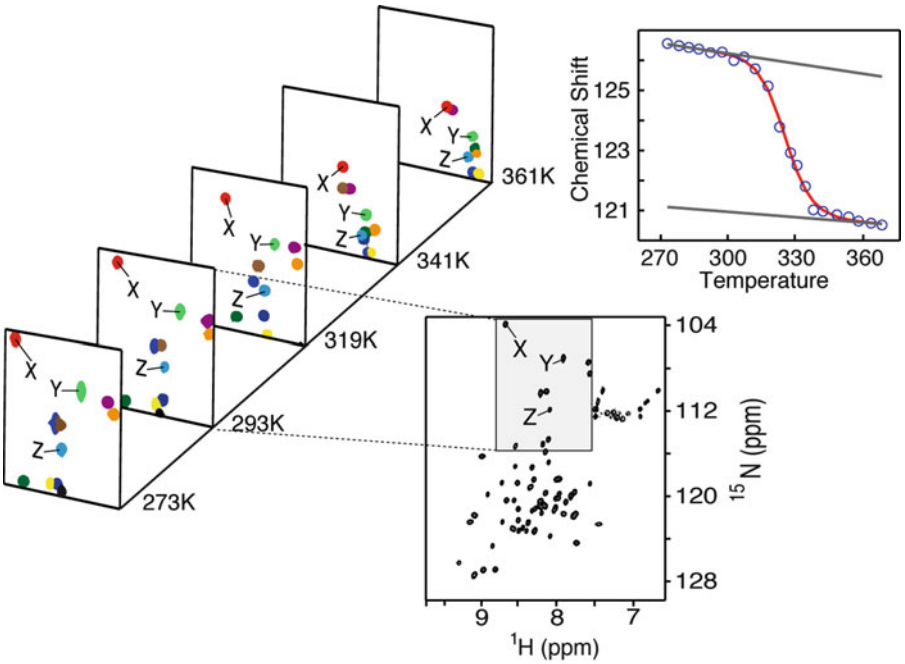


Fig. 2. Graphic representation of the NMR experiments to monitor the equilibrium unfolding process of a folding archetype at atomic resolution. The high-resolution NMR spectrum of the protein in native conditions needs to be assigned (*lower right*). Then the signals are tracked down as the protein becomes progressively unfolded in the series of spectra at different temperatures (*left top diagonal*) to generate the atomic unfolding curves that are used in the analysis (*upper right*).

they can be reduced if chemical shift changes are significantly large or if there is substantial signal overlap (Note 4). It is also important to check the status of the protein sample during such long series of experiments (Note 5)

3.3. Analysis of Individual Atomic Equilibrium Unfolding Curves

3.3.1. Classification of Atomic Unfolding Curves

The atomic unfolding curves generated by the NMR analysis at different temperatures (Note 6) need to be classified into three groups according to the unfolding behavior of the particular atom: (1) two-state (single transition); (2) three-state (double transition); and (3) other (unfolding curves with multiple transitions or no obvious transition—e.g., a straight line). A first stage classification is performed visually, classifying curves as two-state when there is a clear sigmoidal shape and as three-state when the shape is that of a double sigmoidal. All other curves are included in the third group. The two-state and three-state groups are then rechecked calculating numerically the first derivative of the unfolding curve. In this case, two-state curves should have a single maximum and three-state curves, two well-defined maxima. Any curves failing the derivative criterion are then included in the third group. A final check based on comparison of fitting residuals is performed. In this case, the atomic curves classified in groups 1 and 2 are fitted to two-state and three-state

models (see below) and the best fit using a Fisher-test criterion defines the final adscription of the atomic curve to either group.

3.3.2. Analysis of Two-State-Like Unfolding Curves

The atomic unfolding curves need to be analyzed with a simple thermodynamic procedure (Note 7). The curves of group 1 are fitted to a simple two-state model. In this model the observed chemical shift signal $\langle cs \rangle$ is given by equation

$$\langle cs \rangle = \left(cs_{\mathrm{f}} + cs_{\mathrm{f}}^{\mathrm{s}}(T - T_o)\right) \cdot p_{\mathrm{f}} + (1 - p_{\mathrm{f}}) \cdot \left(cs_{\mathrm{u}} + cs_{\mathrm{u}}^{\mathrm{s}}(T - T_o)\right)$$

where cs_{f} and $cs_{\mathrm{f}}^{\mathrm{s}}$ are the intercept and slope of the pre-transition baseline, and cs_{u} and $cs_{\mathrm{u}}^{\mathrm{s}}$ are the intercept and slope of the post-transition baseline. Both baselines are assumed to have linear temperature dependence. p_{f} and p_{u} represent the probabilities of the folded and unfolded states so that $p_{\mathrm{u}} = (1 - p_{\mathrm{f}})$; T_o is an arbitrary reference temperature.

Using the native state as thermodynamic reference, p_{f} is given by

$$p_{\mathrm{f}} = 1 / \left[1 + \exp\left((-\Delta H + T\Delta H / T_{\mathrm{m}}) / RT\right)\right]$$

where T_{m} is the midpoint of the thermal unfolding curve and ΔH is the enthalpy change upon unfolding (reflects the sharpness of the unfolding curve). The analysis involves fitting 6 free floating parameters: the two baselines, T_{m} and ΔH. For simplicity, this model and the three-state model below ignore changes in heat capacity upon unfolding.

3.3.3. Analysis of Three-State-Like Unfolding Curves

The atomic unfolding curves from group 2 are fitted to a three-state model. In this case the observed chemical shift $\langle cs \rangle$ at any given temperature is given by

$$\langle cs \rangle = \left(cs_{\mathrm{f}} + cs_{\mathrm{f}}^{\mathrm{s}}(T - T_o)\right) \cdot p_{\mathrm{f}} + cs_{\mathrm{i}} p_{\mathrm{i}} + \left(cs_{\mathrm{u}} + cs_{\mathrm{u}}^{\mathrm{s}}(T - T_o)\right) \cdot p_{\mathrm{u}}$$

where cs_{f} and $cs_{\mathrm{f}}^{\mathrm{s}}$ represent the intercept and slope of the pre-transition, cs_{i} is the intercept for the intermediate state (for simplicity it is assumed to be temperature independent since this baseline is typically not well resolved), and cs_{u} and $cs_{\mathrm{u}}^{\mathrm{s}}$ represent the intercept and slope of the post-transition. p_{f}, p_{i}, and p_{u} are the folding, intermediate, and unfolding probabilities so that $p_{\mathrm{f}} + p_{\mathrm{i}} + p_{\mathrm{u}} = 1$. The probabilities are calculated with the relationships:

$$p_f = 1 / \begin{bmatrix} 1 + \exp\left((-\Delta H_1 + T\Delta H 1 / T_{1,m}) / RT\right) \\ + \exp\left((-\Delta H_2 + T\Delta H_2 / T_{2,m}) / RT\right) \end{bmatrix}$$

$$P_i = \exp\left((-\Delta H_1 + T\Delta H_1 / T_{1,m}) / RT\right) / \begin{bmatrix} 1 + \exp\left((-\Delta H_1 + T\Delta H_1 / T_{1,m}) / RT\right) \\ + \exp\left((-\Delta H_2 + T\Delta H_2 / T_{2,m}) / RT\right) \end{bmatrix}$$

$$P_u = 1 - p_f - p_i$$

where the ΔH_1 and $T_{1,m}$ are the parameters for the intermediate relative to the native state, and ΔH_2 and $T_{2,m}$ are the parameters for the unfolded state relative to the native state. There are other possibilities for characterizing atomic unfolding curves in a simple way (Note 8).

3.4. Clustering of Atomic Unfolding Curves and Network Analysis

All the atomic unfolding curves should be arranged into a matrix of chemical shifts versus temperature (Note 9). The average atomic unfolding behavior can be simply obtained by performing the singular value decomposition (SVD) of the matrix of atomic unfolding curves. The first component multiplied by the first singular value provides the temperature-averaged chemical shift for the whole atom dataset. The amplitude of the first component (the first row of the V matrix) provides the averaged normalized equilibrium unfolding curve for the whole dataset. This average unfolding curve is then compared with that obtained by a low-resolution collective probe such as that measured by far-UV circular dichroism.

3.4.1. Comparison of Average Atomic Unfolding Behavior with the Global Unfolding Process

3.4.2. Data Clustering

The atomic unfolding curves are clustered into groups according to similarity using a standard clustering algorithm such as K-means or hierarchical clustering (Note 10). Clustering is performed for each of the three groups of atomic unfolding curves independently: N1 clusters for two-state curves (group 1), N2 for three-state curves (group 2), and N3 for the others (group 3). For two-state curves the easiest way to cluster curves is according to their thermodynamic parameters (T_m and ΔH) or to the probability of the native state as function of temperature obtained from the fit. Three-state curves can be clustered the same way but with double number of parameters, or calculating the compounded p_f+p_i curve assuming that the signal of the intermediate is 50 % of that of the native state. The curves belonging to the third group are clustered according to their direct similarity once they have been z-scored to provide a common frame of reference (Note 11). Z-scored unfolding curves are calculated using the following expression:

$$z(T) = \left(x(T) - \mu\right) / \sigma$$

where μ is the mean of all the data being clustered together and σ is the standard deviation.

3.4.3. Calculation of the Thermodynamic Coupling Index Matrix

The thermodynamic coupling index (TCI) is a comparison of the similarity between all the atomic unfolding curves of one residue with all those from another residue. The more similar the stronger is the thermodynamic coupling (Note 12). The comparison between each pair of atomic unfolding curves is carried out calculating the root mean square deviation of the *z*-scored curves (here the procedure is the same regardless of the group adscription).

The thermodynamic coupling index of two residues is then calculated by summing all the possible pairwise comparisons of atomic folding curves (each atom of one residue against each atom of the other residue). Mathematically, this procedure is equivalent to the following expression:

$$\mathrm{TCI}_{x,y} = -\sum_i \sum_j \ln\left(\sqrt{(p_{x,i} - P_{y,j})^2} \,/\, \langle \mathrm{RMSD}_{\mathrm{all}} \rangle\right)$$

where i runs over all atoms of residue x and j over all atoms of residue y. The denominator $\langle \mathrm{RMSD}_{\mathrm{all}} \rangle$ corresponds to the mean RMSD for all atomic unfolding curves in the protein. The TCI matrix is constructed by repeating the same procedure over all possible residue pairs.

3.4.4. Comparing Thermodynamic Coupling Index Matrix and Domain 3D Structure

To obtain a two-dimensional representation of the domain 3D structure calculate the contact map from the atomic coordinates (the pdb file). The contact map is obtained by calculating all residue pairs that have atoms at distances shorter than a given threshold indicative of a close contact in the structure (e.g., a threshold of 0.5 nm). The resulting contact map is a matrix of the same rank of the TCI. For graphical comparison, overlay the two matrices in two colors (e.g. red for contact map and green for TCI). This can be easily achieved using the command *image* of Matlab by placing the contact map on the first layer and the TCI on the second layer. A plotting alternative is to use two different symbols instead of colors (Fig. 3) (Note 13).

3.4.5. Plotting the TCI Matrix as a Network Connectivity Graph

Create a network graph for the folding archetype using its residues as nodes and connecting edges defined by the TCI between the two residues. Use Visone (21) (or any other graph plotting software) defining the edges according to the strength of the TCI between residue pairs. The central and most connected nodes give you the cluster of residues that characterizes the most cooperative unfolding behavior of the protein, and thus represents the global unfolding process.

4. Notes

1. *Requirement of microsecond-folding times.* Proteins with folding times comparable to the characteristic NMR timescale (~0.1–5 ms) will be very hard to study with this method because at temperatures that result in intermediate degrees of unfolding the NMR crosspeaks will experience severe line broadening that will make NMR assignment and precise chem-

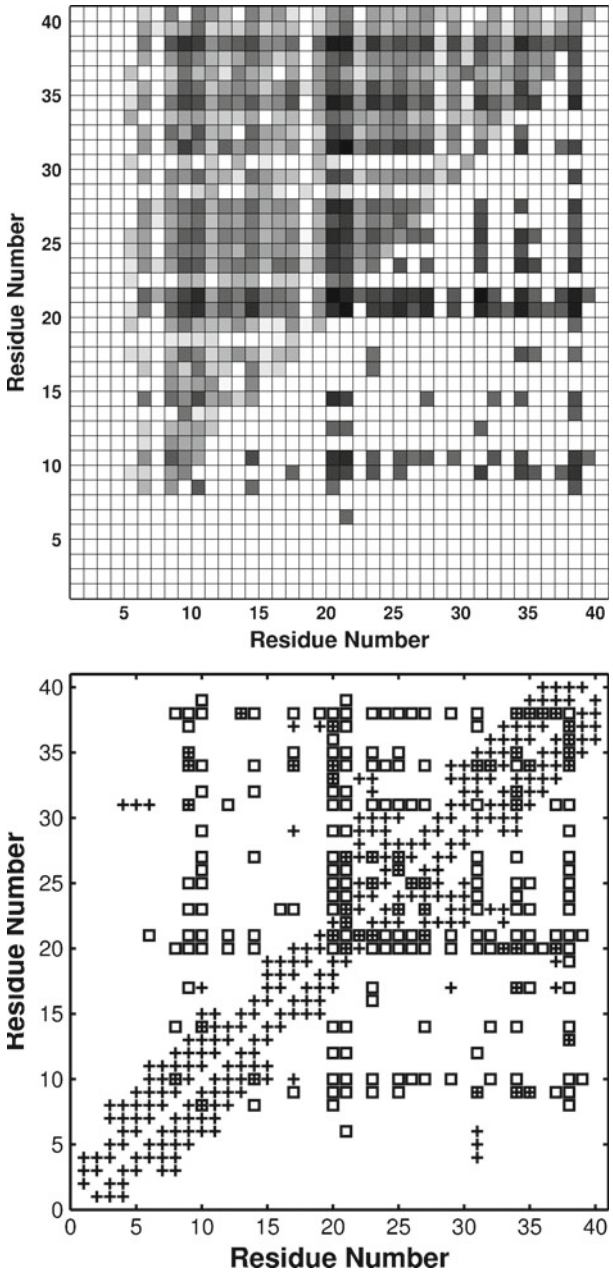


Fig. 3. The matrix of thermodynamic coupling indexes for the protein BBL. The upper panel shows the matrix determined experimentally in shades of gray, where darker signifies stronger coupling. The upper left triangle displays the entire matrix, and the lower right triangle only couplings above a certain threshold (strong couplings). The bottom panel shows the overlay of the strong couplings in the TCI matrix (squares) and the contact map (crosses). In this case the overlapping symbols indicate the residue–residue contacts with strong coupling.

ical shift determination very challenging. In some instances the crosspeaks may disappear altogether.

2. *Handling pressure valve NMR tubes.* Pressure valve NMR tubes are easily connected to a gas manifold. Once the desired pressure is reached the valve can be closed and the gas line can be disconnected from the valve. New Era pressure valve NMR tubes stand a pressure of up to ~20 bar if the glass is 0.77 mm thick. However, the thickness of the wall might slightly reduce NMR signal-to-noise ratio versus regular tubes with thinner glass walls.
3. *Chemical shifts from broad NMR signals.* Protein rotational correlation time strongly affects NMR signal width. The longer the rotational correlation time the broader are the NMR signals. Solvent viscosity increases at low temperature, increasing in turn the protein rotational correlation time and thus resulting in broad signals. The determination of chemical shifts from the shallow maximum of the NMR crosspeak is therefore less accurate at low temperature. For broad signals it is desirable to define a portion of the peak from which a reasonable number of contour levels can be defined. In this case a good estimate of the chemical shift can be obtained by averaging the centers derived from all these contours. The software PIPP for peak picking (20) includes this feature known as contour averaging.
4. *Signal overlap.* NMR signal overlap is common in protein NMR and worsens at high temperature. When two partly overlapping peaks give rise to a broad crosspeak without two distinguishable maxima it is also possible to apply contour averaging (20).
5. *Checking protein sample status.* Some proteins might undergo degradation and/or aggregation at high temperature during the long periods of time required to acquire triple-resonance NMR experiments. Therefore, after each 3D set of experiments it is advisable to acquire simple 2D-[^{1}H-^{15}N]-HSQC spectra under folding conditions to check the status of the protein sample. Spectra acquired before and after the 3D set at a temperature at which the protein is folded should be superimposable. Otherwise it might be necessary to use a newly prepared sample for each set of spectra acquired within the high temperature range.
6. *Chemical shift temperature dependence of protein ^{1}H nuclei.* Chemical shifts of protein ^{1}H nuclei, particularly amide $^{1}H^{N}$, strongly depend on temperature even in the absence of secondary or tertiary structural modifications. As a result, it is more difficult to derive clear information on the protein unfolding process from ^{1}H chemical shift variation with temperature. However, the temperature dependence of protein ^{15}N and ^{13}C nuclei is directly related to structural changes. Thus, amide

^{15}N, $^{13}C_{\alpha}$, and $^{13}C_{\beta}$ chemical shifts are ideal choices for the atom-by-atom analysis of protein folding.

7. *Basic parameters defining an atomic unfolding curve.* The fitting of the atomic unfolding curves to a two-state (or three-state model) renders two (or four) basic thermodynamic parameters: the apparent sharpness of the atomic unfolding curve (obtained directly from ΔH) and the temperature at which the transition is halfway, or denaturation midpoint (i.e., T_m). These parameters together with the group adscription are the defining characteristics of any given atomic unfolding curve.
8. *Additional procedures to obtain denaturation midpoints from atomic unfolding curves.* Denaturation midpoints can also be obtained by an independent model-free method. This method determines the denaturation midpoint by calculating the derivative of the atomic equilibrium unfolding curve using numerical methods. The curve derivative is then analyzed with a simple algorithm that estimates the position of the maximum in the typically noisy derivative data by finding the point that divides the area under the curve into two equal halves (24). The maximum in the derivative is then taken as the denaturation midpoint (24). For curves with two transitions (group 2), the procedure is the same but two maxima are identified. This method does not require to assume any particular thermodynamic model, and provides a cross-validation of the precision of the denaturation midpoints obtained with the two- and three-state fits, which could be affected by poorly defined baselines (25). The calculation of denaturation midpoints by the derivative method is available at http://www.tmg.cib.csic.es/servers.
9. *Preferred atom types for the atom-by-atom analysis.* In principle, preferred atoms are those with chemical shifts that are highly sensitive to structural-conformational changes and which do not have strong intrinsic temperature dependence. The second requirement restricts the use of the amide protons (^{1}HN), which have strong intrinsic temperature dependence due to proton exchange with the surrounding water molecules. Thus, we exclude amide protons from the subsequent analyses that depend heavily on the meaningful comparison between atomic unfolding curves.
10. *Identifying the optimal number of clusters.* The clustering routine should be performed separately for each of the three groups of atomic unfolding curves. The number of clusters to use depends on the total number of curves within each group and their heterogeneity. Clustering methods tend to separate out the most dissimilar unfolding curves into very small clusters containing only one or two atoms and then group all others into a few highly populated clusters. Practically, one can minimize this problem by starting clustering with approximately five times fewer clusters than curves in each group, and then iteratively

decrease the number of clusters until the results produce only a few clusters containing only one or two atomic curves.

11. *Comparison between different unfolding curves.* Because the absolute chemical shift value and its change upon unfolding vary widely depending on the particular atom of interest, it is very important to define a common frame of reference before comparing heterogeneous atomic unfolding curves. One possibility is to compare unfolding behaviors using the basic thermodynamic parameters obtained from the fits to the two-state and three-state models: denaturation midpoint (T_m) and change in enthalpy upon unfolding (ΔH). Another option is to use directly the native signal calculated from the native probability generated by the two-state fit. The same approach can be extended to curves studied with a three-state model by assuming that the intermediate provides a fraction (e.g., 50 %) of the native signal. For curves that do not belong to either group 1 or group 2, or for comparing curves from different groups, the best procedure is to compare directly the chemical shift versus temperature curves once they have been z-scored to provide a common frame of reference.
12. *Interpretation of TCI values.* The TCI is positive when the sum of atomic couplings for two residues is stronger than the mean coupling for the entire dataset (all atoms of the protein under study) and negative when it is weaker. Therefore, positive TCI reflects residues with highly coupled atomic unfolding behaviors.
13. *The TCI matrix versus the contact map.* The mechanistic interpretation of the overlay of contact map and TCI matrix is straightforward. The positions with overlapping symbols (square plus cross) in the figure define the critical network of residue–residue contacts that are responsible for the global unfolding process of the autonomously folding supersecondary structure domain. The green squares represent residues that are not in spatial contact in the 3D structure but are strongly coupled folding-wise (i.e., very similar unfolding process). These residues are typically structurally connected by secondary or tertiary contacts, that is, by mutual coupling to another residue that is in contact with both for secondary contacts. Finally, crosses signify contacts in the 3D structure that do not convey thermodynamic coupling during unfolding.

Acknowledgments

This work was supported by the Marie Curie Excellence Award MEXT-CT-2006-042334, and the grants BFU2008-03237, BFU2008-03278 and CONSOLIDER CSD2009-00088 from the Spanish Ministry of Science and Innovation (MICINN).

References

1. Bryngelson JD, Onuchic JN, Socci ND et al (1995) Funnels, pathways, and the energy landscape of protein-folding—a synthesis. Proteins: Struct Funct Genet 21:167–195
2. Pande VJ (2008) Computer simulations of protein folding. In: Muñoz V (ed) Protein folding, misfolding and aggregation: classical themes and novel approaches, RSC, Cambridge, pp 161–187
3. Jackson SE (1998) How do small single-domain proteins fold? Fold Des 3:R81–R91
4. Kubelka J, Hofrichter J, Eaton WA (2004) The protein folding "speed limit". Curr Opin Struct Biol 14:76–88
5. Muñoz V (2007) Conformational dynamics and ensembles in protein folding. Annu Rev Biophys Biomol Struct 36:395–412
6. Naganathan AN, Doshi U, Fung A et al (2006) Dynamics, energetics, and structure in protein folding. Biochemistry 45:8466–8475
7. Yang WY, Gruebele M (2003) Folding at the speed limit. Nature 423:193–197
8. Muñoz V, Sanchez-Ruiz JM (2004) Exploring protein-folding ensembles: a variable-barrier model for the analysis of equilibrium unfolding experiments. Proc Natl Acad Sci USA 101:17646–17651
9. Naganathan AN, Sanchez-Ruiz JM, Muñoz V (2005) Direct measurement of barrier heights in protein folding. J Am Chem Soc 127:17970–17971
10. Naganathan AN, Perez-Jimenez R, Sanchez-Ruiz JM et al (2005) Robustness of downhill folding: guidelines for the analysis of equilibrium folding experiments on small proteins. Biochemistry 44:7435–7449
11. Muñoz V (2002) Thermodynamics and kinetics of downhill protein folding investigated with a simple statistical mechanical model. Int J Quant Chem 90:1522–1528
12. Garcia-Mira MM, Sadqi M, Fischer N et al (2002) Experimental identification of downhill protein folding. Science 298:2191–2195
13. Sadqi M, Fushman D, Muñoz V (2006) Atom-by-atom analysis of global downhill protein folding. Nature 442:317–21
14. Fung A, Li P, Godoy-Ruiz R et al (2008) Expanding the realm of ultrafast protein folding: gpW, a midsize natural single-domain with alpha + beta topology that folds downhill. J Am Chem Soc 130:7489–7495
15. Marti-Renom MA, Stuart A, Fiser A et al (2000) Comparative protein structure modeling of genes and genomes. Annu Rev Biophys Biomol Struct 29:291–325
16. De Sancho D, Munoz V (2011) Integrated prediction of protein folding and unfolding rates from only size and structural class. Phys Chem Chem Phys 13: 17030–17043
17. Bax A, Grzesiek S (1993) Methodological advances in protein NMR. Acc Chem Res 26:131–138
18. Cavanagh J, Fairbrother WJ III, AGP, Rance M et al (1996) Chemical exchange effects in NMR spectroscopy, in protein NMR spectroscopy: principles and practice, Academic Press, San Diego, p 391–404
19. Delaglio F, Grzesiek S, Vuister GW et al (1995) NMRPipe: a multidimensional spectral processing system based on UNIX pipes. J Biomol NMR 6:277–293
20. Garret DS, Powers R, Gronenborn AM et al (1991) A common sense approach to peak picking in two-, three-, and four-dimensional spectra using computer analysis of contour diagrams. J Magn Reson 95:214–220
21. Brandes U, Wagner D (2004) In: Juenger M, Mutzel P (ed) Visone—analysis and visualization of social networks, in Graph drawing software, Springer, New York, p 321–340
22. Schanda P, Brutscher B (2005) Very fast two-dimensional NMR Spectroscopy for real-time investigation of dynamic events in proteins on the time scale of seconds. J Am Chem Soc 127:8014–8015
23. Amman C, Meier P, Merbach AE (1982) A simple multinuclear NMR thermometer. J Magn Reson 46:319–321
24. Naganathan AN, Muñoz V (2008) Determining denaturation midpoints in multiprobe equilibrium protein folding experiments. Biochemistry 47:6752–6761
25. Sadqi M, Fushman D, Muñoz V (2007) Structural biology—analysis of protein-folding cooperativity—reply. Nature 445:E17–E18

Chapter 14

Artificial Supersecondary Structures Based on Aromatic Oligoamides

Hai-Yu Hu and Chuan-Feng Chen

Abstract

With an intelligent design of the monomers, considerable effort has so far focused on the creation of aromatic oligoamide foldamers which are able to mimic the secondary structures of biopolymers. Supersecondary structure is a growing set of known and classifiable protein folding patterns that provides an important organizational context to this complex endeavor. In this article, we highlight the design, chemical synthesis, and structural studies of artificial supersecondary structures based on aromatic oligoamide foldamers in recent years.

Key words: Supersecondary structure, Aromatic oligoamides, Structure elucidation, Hydrogen bonds, Pi interactions, X-ray diffraction

1. Introduction

The mystery of how a protein sequence specifies a unique structure and function has intrigued chemists' significant interest in the design and development of foldamers (1–5). Foldamers are oligomers synthesized from this vast pool of compounds, the defining characteristic being their folding into well-defined conformations due to one or combination of noncovalent forces. With an intelligent design of the monomers, considerable effort has so far focused on the creation of unnatural oligomers which are able to mimic many secondary structural elements of native peptides and proteins, such as helices (6–16), sheets (17–23), and turns (24–29).

Biopolymers, however, rely not on secondary but mostly on more complicated motifs to mediate their functions. In comparison, little is achieved by isolated secondary folded elements.

Alexander E. Kister (ed.), *Protein Supersecondary Structures*, Methods in Molecular Biology, vol. 932,
DOI 10.1007/978-1-62703-065-6_14,

A supersecondary structure is the term used to describe certain common combinations of secondary structure elements that are observed frequently in protein structures (30). In the hierarchy of protein structure classification, supersecondary structure falls between that of secondary structure and tertiary structure, which can provide a useful way of categorizing distinctive and recurring components of protein structure (31). A great challenge for foldamer research and for chemistry in general is thus to establish design principles and synthetic methods to prepare very large and complex, yet well-organized, molecular architectures comprising of several secondary folded blocks. In recent years, examples of foldamers with supersecondary-like structures have begun to appear and validate the viability of this approach. In the article, we focus only on aromatic amide foldamers which feature a remarkable combination of structural predictability, stability, tunability, and ease of synthesis.

The lessons learned from the successful construction of secondary structural elements can be applied to the design of supersecondary structures. Construction of foldamers at a higher level of complexity involves the design of oligomeric secondary structural elements whose assembly leading to a compact fold can be achieved by exploiting a variety of strategies (Fig. 1). The designed modules for subsequent assembly can be from isolated structural units or tethered to form a single polypeptide chain as in the case of naturally occurring proteins.

2. Helix–Helix Motifs

In recent years, Chen and coworkers have developed a class of aromatic oligoamides based on phenanthroline dicarboxamides, which exhibited well-defined helical secondary structures in solution and in the solid state (Fig. 2) (32). Furthermore, they reported the first artificial aromatic oligoamide based helix-turn-helix (HTH) supersecondary structure, which was composed of two regular helical secondary structures based on oligo(phenanthroline dicarboxamide) strands connected with a binaphthyldiamine as the turn (Fig. 3) (33). The binaphthyldiamine spacer adopts a staggered conformation due to steric hindrance between the tow *ortho* groups, thereby making an angle of 72° between two helical segments and acting like a turn motif as evidenced by an X-ray crystal structure of (±)-**2** (Fig. 4a). When a chiral turn unit was used, a Cotton effect was seen in the CD that increased with increasing chain length (Fig. 4b).

Subsequently, they deduced that by the insertion of suitable connecting units, new foldamers with specific supersecondary structures could be obtained easily. In addition, the rigid aromatic

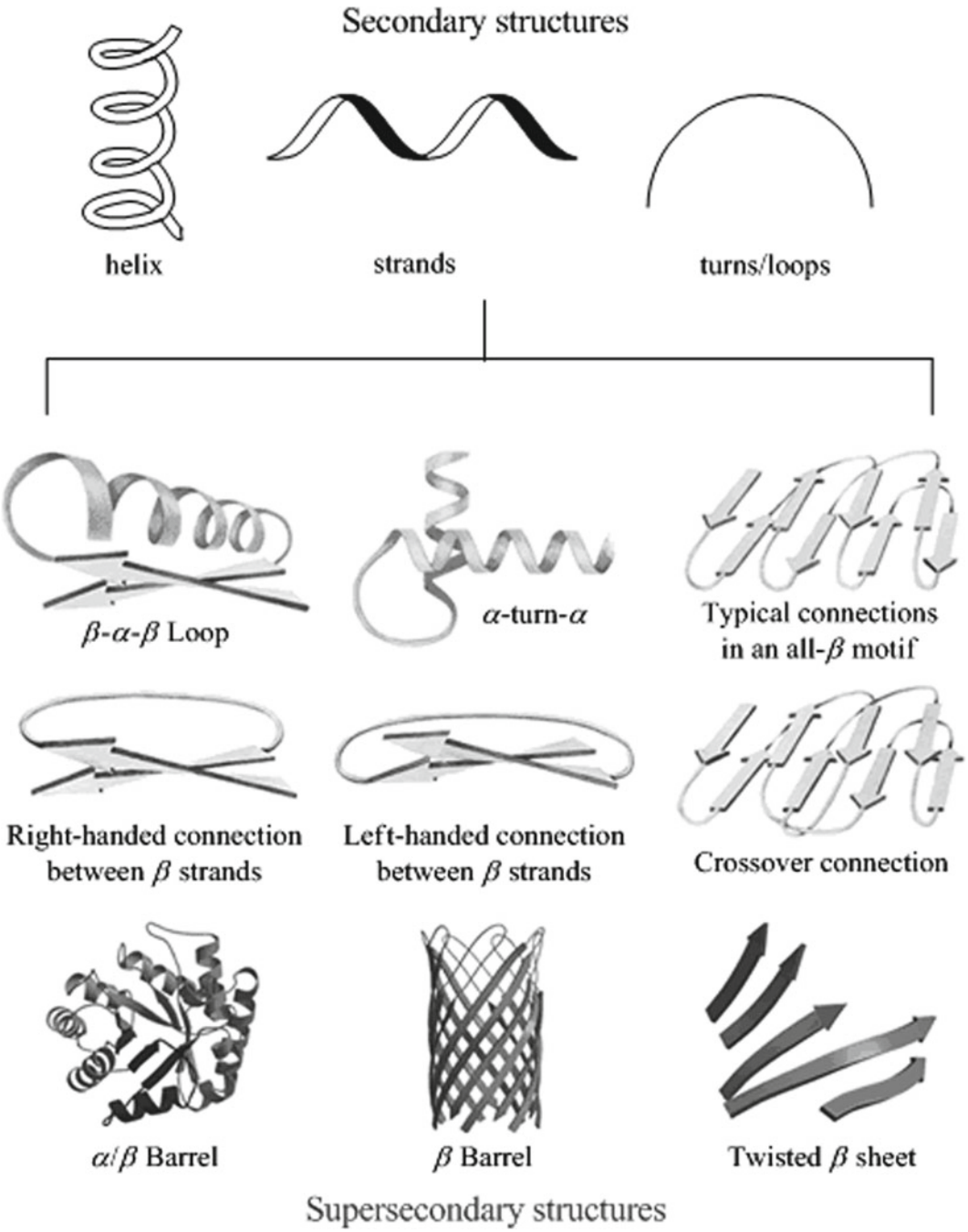


Fig. 1. Schematic representation of secondary structural elements assemble into supersecondary structures.

linkers will improve predictability and stability of the foldamers. Based on their preceding research, connected with 1,8-diaminoanthraquinone, the oligo(phenanthroline dicarboxamide)s exhibited well-defined helical secondary structures under the intramolecular

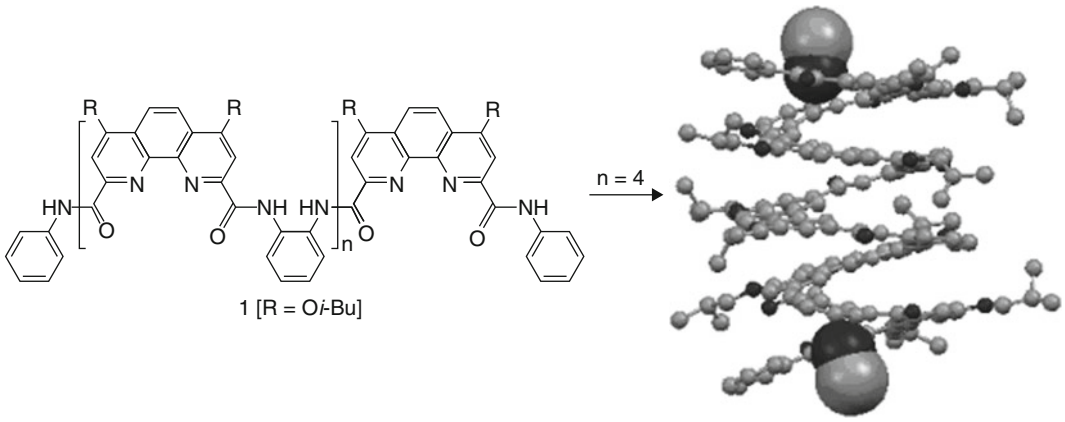


Fig. 2. Structure of aromatic oligomers based on phenanthroline dicarboxamides and crystal structure of **1** (n = 4)@2CH_3OH.

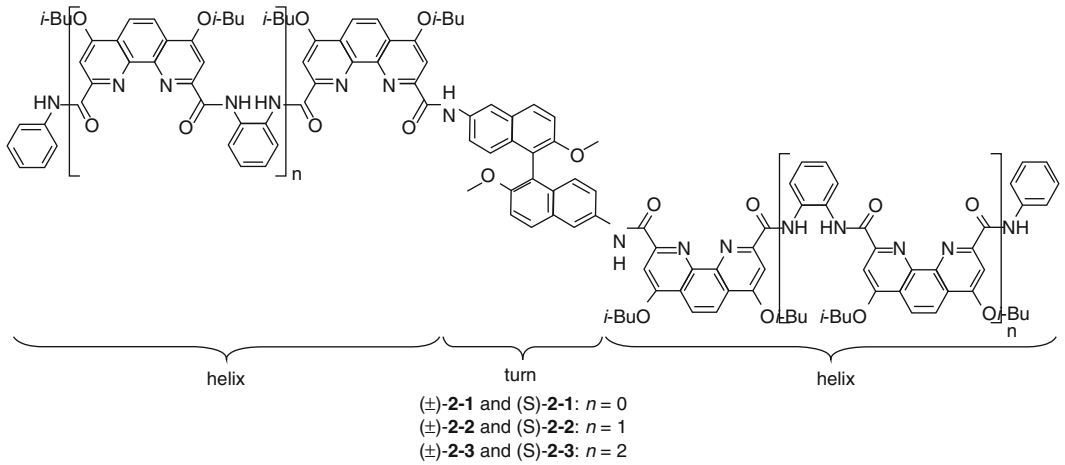


Fig. 3. Structure of the helix-turn-helix motif reported by Chen and coworkers.

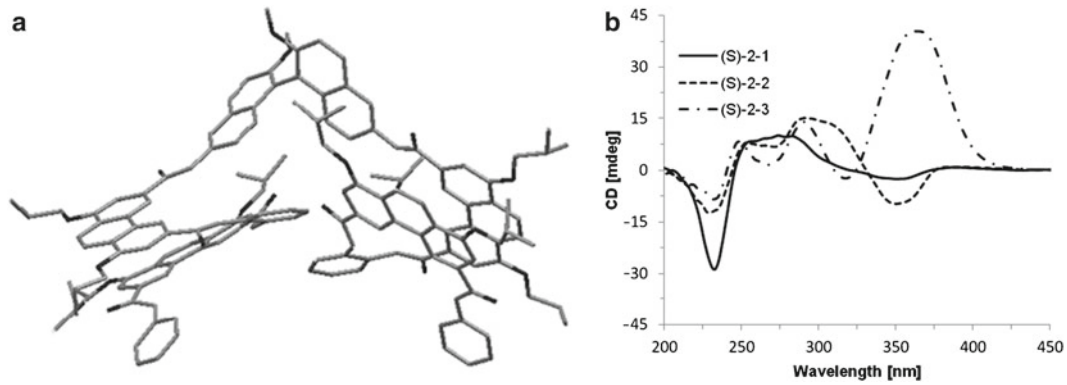


Fig. 4. (**a**) Crystal structure of (±)-**2-2**; (**b**) CD spectra of the molecular strands (*S*)-**2-1** (*solid line*), (*S*)-**2-2** (dashed ine), and (*S*)-**2-3** (dash-dot line) in CH_3CN ($c = 10^{-5}$ M).

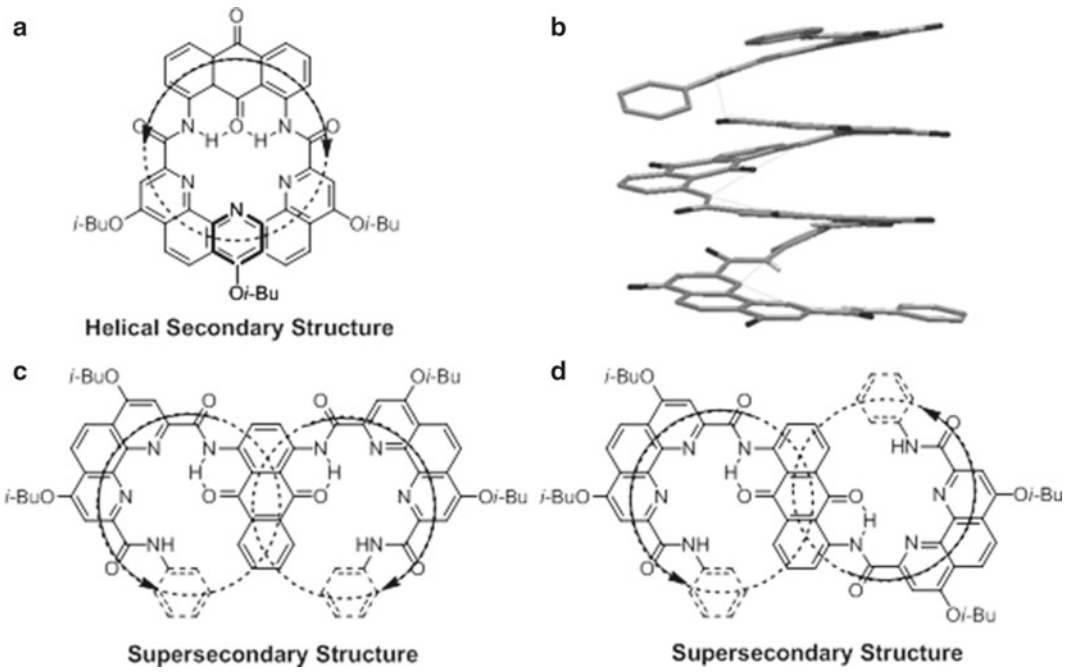
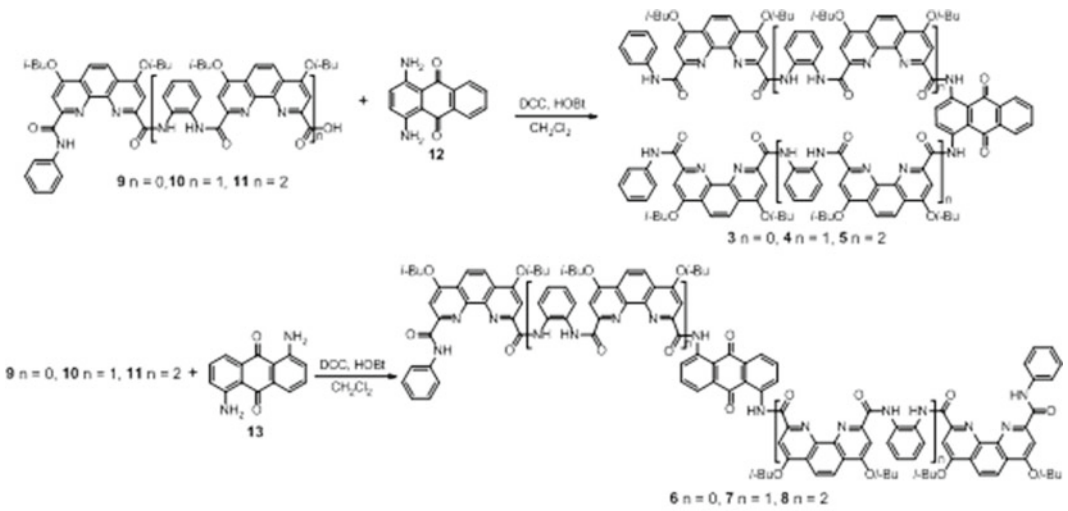


Fig. 5. Schematic representation of the projection of the two helical oligo(phenanthroline dicarboxamide) strands segments in the plane of the 1,8-diaminoanthraquinone spacer (**a**), the crystal structure (**b**) 1,4-diaminoanthraquinone spacer (**c**) and 1,5-diaminoanthraquinone spacer (**d**). Overlap between the circles indicates possible steric hindrance between helices if they extend on the same side of the plane of the linkers. The *arrows* indicate the direction along which each oligomeric segment extends from the linker.

hydrogen bonds (Fig. 5a, b) (34). Consequently, they envisioned that changing the amino position of the diaminoanthraquinone will cause the conformation of these oligo(phenanthroline dicarboxamide)s change from helical structures to supersecondary structures, which would offer a useful and easy avenue for the de novo design of aromatic oligoamide foldamers with distinctive structural architectures. To confirm this thought, the new oligo(phenanthroline dicarboxamide)s 3–8 were designed and synthesized (Scheme 1) in which the diaminoanthraquinone subunits not only are used as the linkers, but also locally set the relative orientation of the secondary elements. Because rotations about the amide nitrogen-aryl linkages at the 1 and 4 positions of the anthraquinone would be restricted by NH…O=C hydrogen bonds, steric hindrance was expected to prevent the two helical segments from extending on the opposite side of the linker, and the helices should be both right-handed (*P-P*) or both left-handed (*M-M*) (Fig. 5c). A similar case could also be applied in the helical supersecondary structures with 1,5-diaminoanthraquinone as the linker, in which steric hindrance would result the two helical segments in opposite handedness (*P-M*) (Fig. 5d) (35).



Scheme 1. Synthetic schemes of aromatic oligoamides **3–8**.

Design concepts were directly validated in the solid state by the single-crystal structures of oligomers **4** and **7**. As shown in Fig. 6a, the oligomer **4** consists of two regular helices linked by the rigid 1,4-diaminoanthraquinone, which formed a supersecondary structure. Rotations about the aryl-NH bonds of the linker are expectedly restricted by strong intramolecular hydrogen bond between the anthraquinone oxygens and the adjacent amide protons with the C=O…N distance of 2.60 Å, and the entire structure is held by the network of conformational restrictions. The two helical segments are found on opposite sides of the anthraquinone. If the two helices are located on the same side, they would bump into each other, which are shown in the *top view* of the structure clearly. Fig. 6b shows the two helices having the same handedness with a complete turn (*M-M*); however, the crystal is racemic with both *P-P* and *M-M* handed forms presented in the unit cell (Fig. 6c), and there is an intermolecular face-to-face π-π stacking between the two phenanthroline rings of the adjacent foldamers with the distance of 3.41 Å. In the case of oligomer **7**, its crystal structure also shows (Fig. 6d, e) a clear supersecondary structure stabilized by a network of intramolecular hydrogen bonds. As expected, the rigid linker 1,5-diaminoanthraquinone set the relative orientation of the two helical structures, and the two helical segments of oligomer **7** were also found on opposite sides of the linker. Compared with oligomer **4**, the structure of oligomer **7** possessed a center of symmetry in the middle of the anthraquinone ring. The two helices thus have opposite handedness, giving rise to *meso*-helicity (36, 37). One conformation (*M-P*) was only presented in the unit cell (Fig. 6f), and there is also an intermolecular face-to-face π-π stacking between the two phenanthroline rings of the adjacent foldamers with the distance of 3.34 Å.

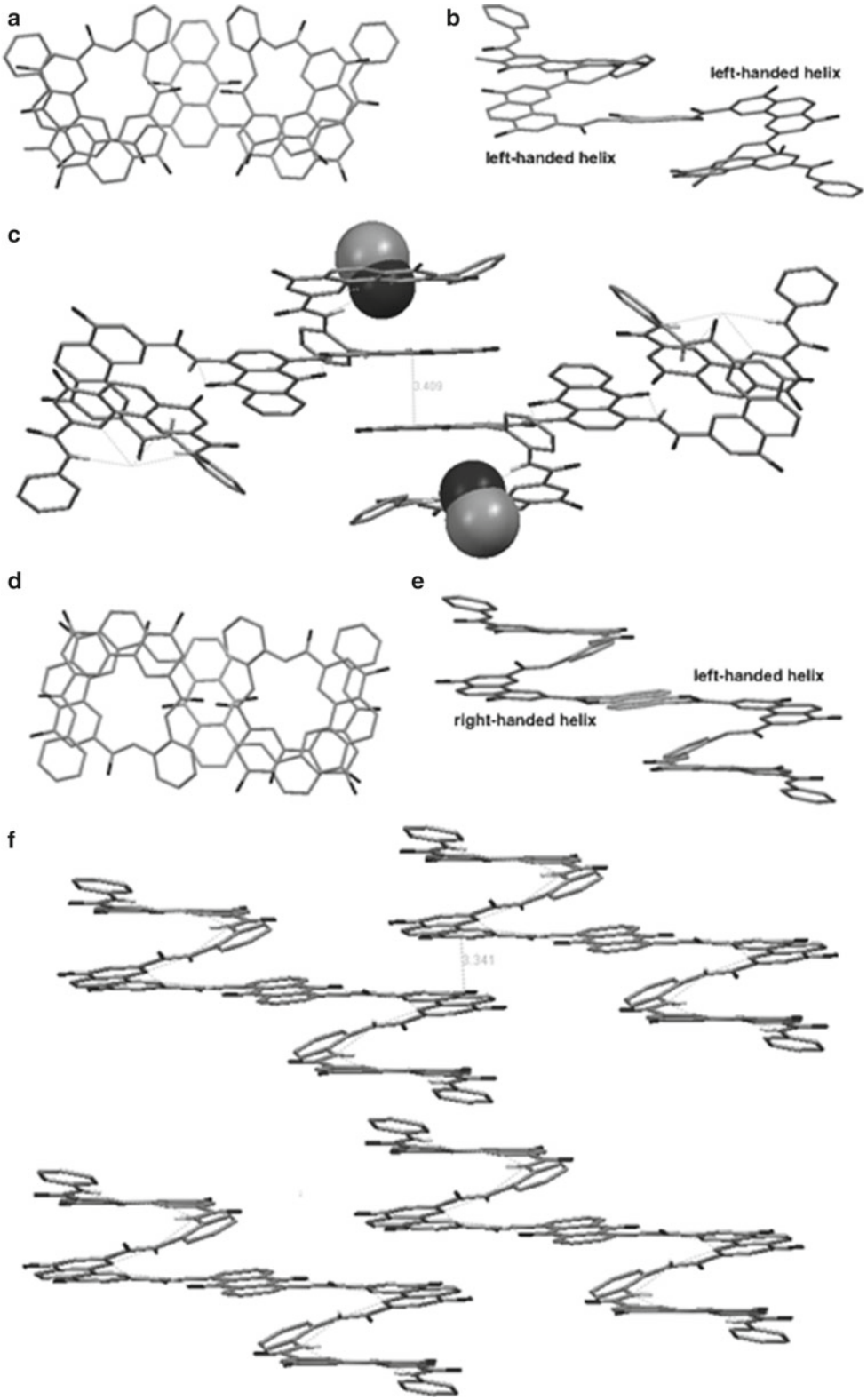


Fig. 6. Crystal structures of **4** (**a**) *top view*, (**b**) *side view* and (**c**) crystal packing (along an axis) with methanol molecules presented at the ends of the helix, and **7** (**d**) *top view*, (**e**) *side view* and (**f**) crystal packing (along an axis). Isobutyl chains and hydrogen atoms are omitted for clarity.

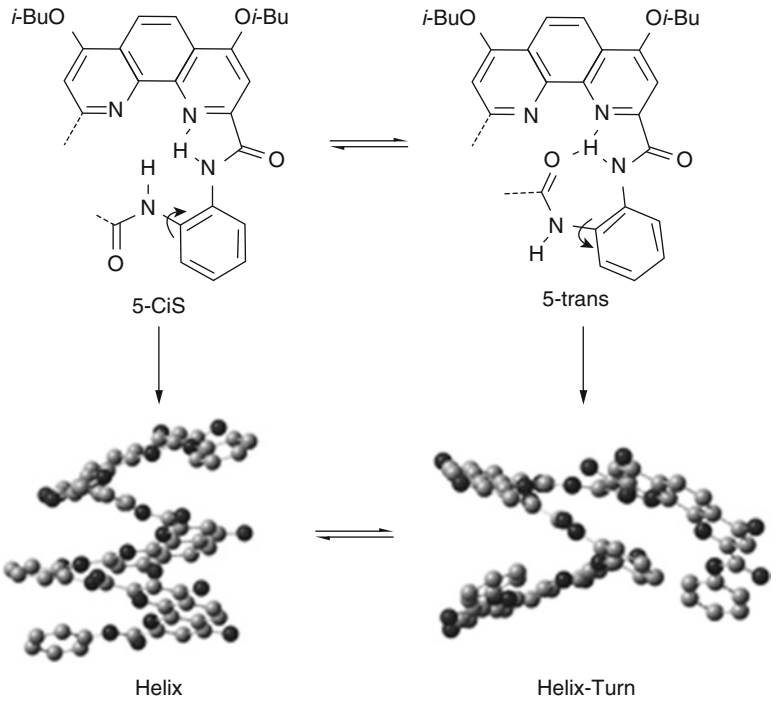


Fig.7. Local variation of steric interaction related to the amide *s*-cis/*s*-trans isomerization process in oligo(phenanthroline dicarboxamide)s.

Similar to biomolecules, the differences in local conformational equilibria of the oligo(phenanthroline dicarboxamide)s could also be dominated by hydrophobic effects at the global structural level. Recently, they have characterized the structural features of the oligo(phenanthroline dicarboxamide)s, and presented their dynamic environment-associated conformational conversion from secondary helical structures to supersecondary helix-turn structures by X-ray crystallographic, variable-temperature ^{1}H NMR, variable-temperature circular dichroism techniques, and computational studies (38). The *o*-phenylenediamide is an important structural unit in the backbone of oligo(phenanthroline dicarboxamide)s. Since there is no specific attractive or repulsive interaction between the two amides of the *o*-phenylenediamide moieties, it can form *s*-cis and *s*-trans conformations under the rotation about the CONH-aryl bond, which can be a rate-limiting step in the folding mechanism. The factors that favor specific isomer geometry about *o*-phenylenediamide can thus contribute significantly toward controlling the structures of the oligo(phenanthroline dicarboxamide)s. In order to investigate the conformational conversion phenomena, the oligomers with three phenanthroline units were chose as the model. When the conformational transition occurs in one of the *o*-phenylenediamide subunits in the oligomer triggering by an *s*-cis to *s*-trans 180° rotation, the simplest environment-associated supersecondary helix-turn structure was formed (Fig. 7).

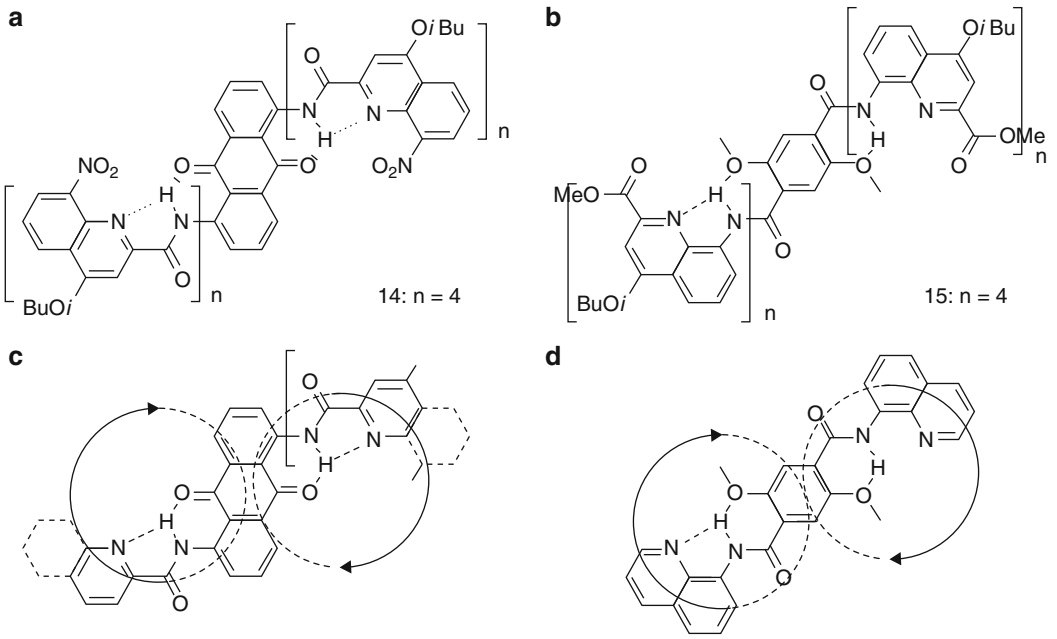


Fig. 8. Structures and synthetic schemes of compounds 14 (a) and 15 (**b**). Schematic representation of the projection of the two helical oligoquinolinecarboxamide segments of 14 in the plane of the diaminoanthraquinone spacer (**c**), and of 15 in the plane of the dimethoxyterephthaloyl spacer (**d**). In both 14 and 15, the surfaces covered by the two helical segments (*circles*) partly overlap, indicating steric hindrance if the helices extend on the same side of the spacer. The *arrows* indicate the direction in which each oligomeric segment extends from the spacer. For a given compound, that both arrows turn in the same direction (clockwise) indicates that the two helical segments have the same handedness if they extend on the same side of the spacer, and opposite handedness if they extend on opposite sides.

Aromatic oligoamides of 8-amino-2-quinoline carboxylic acid adopt particularly stable helical conformations in the solid state and in a wide variety of solvents (39). It provided a firm foundation upon which to build in modular fashion towards large multihelical, folded architectures. In 2004, Huc and his coworkers have used 8-amino-2-quinoline carboxylic acid unit to construct protein-like architectures from totally synthetic building blocks. They describe a strategy based on mutual steric exclusion to orient two helical segments in opposite directions and simultaneously impose an inversion of helix handedness between them (40). First, a rigid linker 1,5-diaminoanthraquinone has been proposed to insert between two helices at the C terminus without disrupting the continuity of the hydrogen bond network and of π–π interactions (Fig. 8a). The single crystal structures of oligomer 14 with two tetrameric quinonlinecarboamide segments showed that the first quinoline group at the C-terminus of each tetrameric quinolinecarboxamide segment is almost coplanar with the anthraquinone ring, and the network of intramolecular hydrogen bonds sets the conformation of each rotatable bond over the entire strand. Tight hydrogen bonds are established between the anthraquinone oxygens and the adjacent amide protons. The structure possesses a center of symmetry

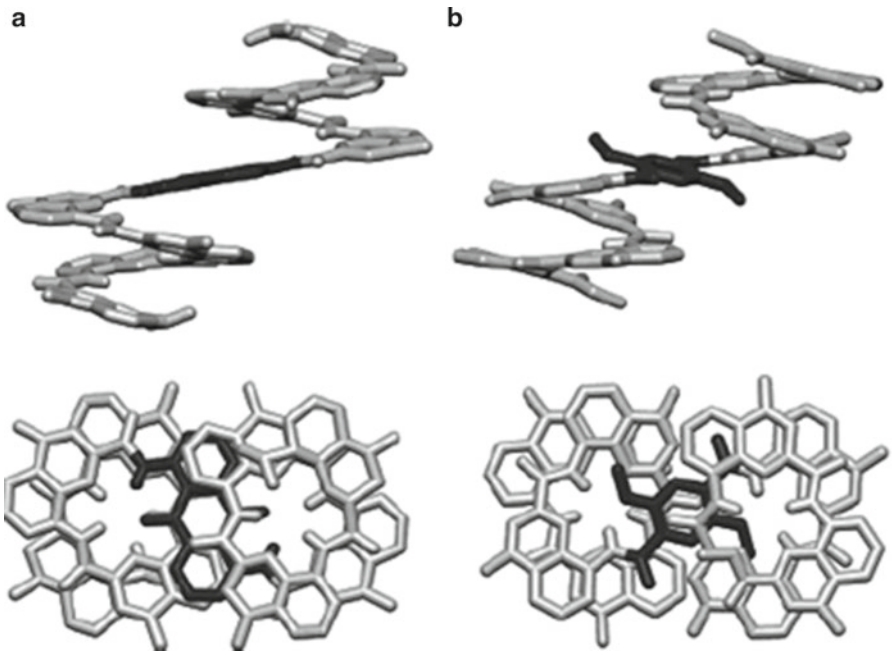


Fig. 9. *Side views* (*top*) and *top views* (*bottom*) of stick representations of the crystal structure of **14** (**a**) and of the crystal structure of **15** (**b**). The diaminoanthraquinone moiety in **14** and the dimethoxyterephthalic unit in **15** are shown in *red*. Included solvent molecules, isobutyl groups, and carbon hydrogens have been omitted for clarity.

in the middle of the anthraquinone ring, and the asymmetric unit contains only half a molecule. The two helices thus have opposite handedness, giving rise to *meso*-helicity (Fig. 9a).

A similar reasoning has be applied for linking two helices at the *N*-terminus using a 2,5-dimethoxyterephthaloyl linker (Fig. 8b). The structure of oligomer **15** was characterized in the solid state (Fig. 9b) by single-crystal X-ray diffraction analysis. As expected, rotations about the aryl-carbonyl bonds of the linker are restricted by NH–O hydrogen bonds and add to the network of conformational restrictions that holds the entire structure. Thus, the linker belongs to both helical segments. There is slightly tilted (25°) out of the plane of the two adjacent quinoline rings. It lies parallel to the next two quinoline rings in the sequence between which it is sandwiched. The two helical segments of **15** are found on opposite sides of the linker, leading to a centrosymmetric, *meso*-helical structure. Supramolecular *meso*-helices have occasionally been encountered in the solid state.

The conformation of **17** features several intrinsically chiral elements that are all expected to undergo dynamic exchange: the right (*P*) or left (*M*) handedness of the two helical segments, and the Λ or Δ configuration of the metal complex. A total of six species, three enantiomeric pairs of diastereomers, is thus expected: two pairs in which both helices have the same handedness *P*Λ*P*/*M*Δ*M* (**17a**) and *P*Δ*P*/*M*Λ*M* (**17b**) and one pair in which the two helices have opposite handedness *P*Λ*M*/*P*Δ*M* (**17c**).

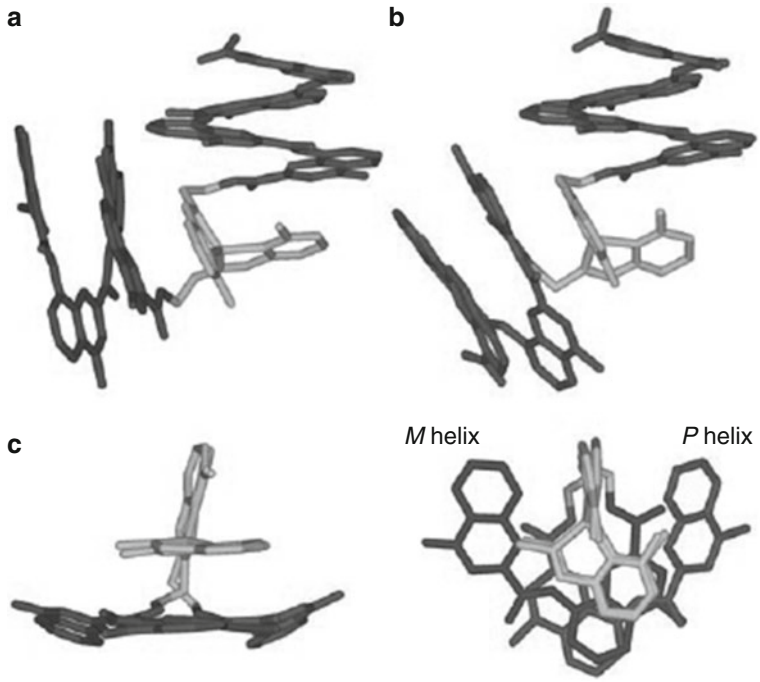


Fig. 10. Crystal structures showing (**a**) the *P*Λ*M*/*P*Δ*M* (**17c**) and (**b**) *P*Λ*P*/*M*Δ*M* (**17a**) conformations of **17**. (**c**) *Top* and *side views* of the overlay of fragments of the above complexes showing two Δ Cu^{I} complexes (in *grey*) and the first two quinoline residues of an *M* helix and of a *P* helix. Side chains, BF_4^- ions, and included solvent molecules are omitted for clarity.

Crystallographic investigations proved particularly successful as they allowed the characterization of four out of the six possible forms of **17** (Fig. 10). In both structures, each 2-iminopyridine moiety is perpendicular to the terminal quinoline ring of the helix to which it belongs due to the gauche conformation of the ethylene spacer (Fig. 11) (Scheme 2).

3. Strands

The β-strand consists of a highly extended or "sawtooth" amino acid arrangement and is the simplest peptide secondary structure motif. β-Strands lack intramolecular hydrogen bonds between backbone residues and typically interact with complementary peptide chains to form β-sheets. These supersecondary structures are key recognition elements in protein–protein and protein–DNA interactions relevant to cell proliferation, infectious diseases, and neurological disorders (42–45).

Gong and coworkers reported H-bonded duplexes based on the zippering of oligoaimde strands bearing complementary H-bonding sequences, which were featured by programmable

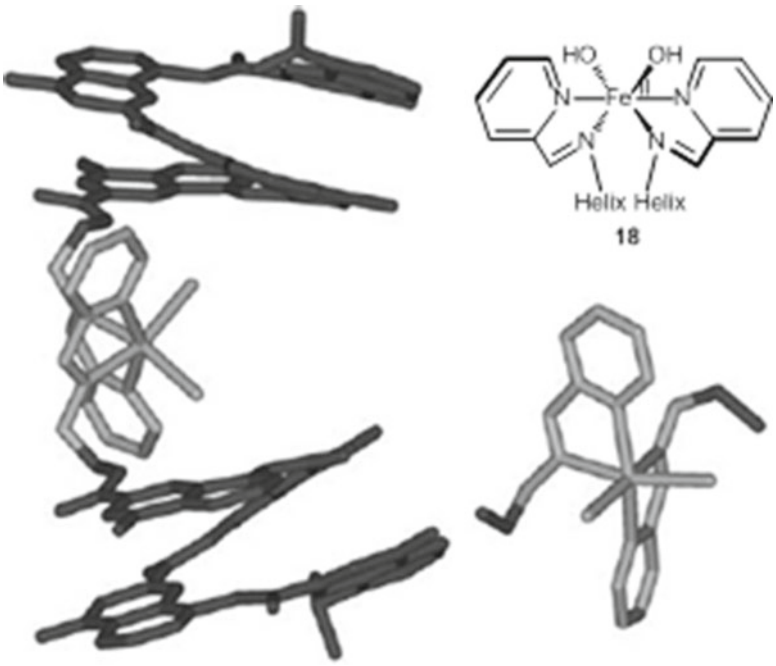
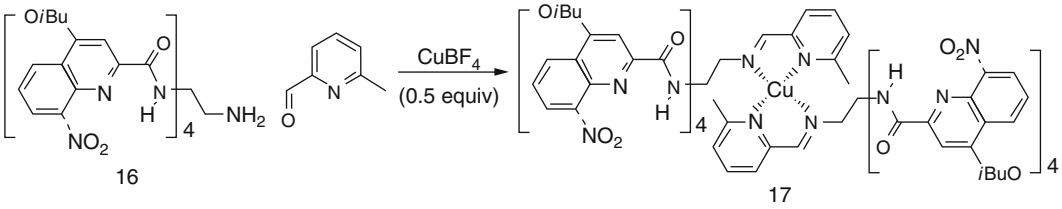


Fig. 11. Formula and crystal structure of **18**. Left-handed helices are shown in *dark grey* and the metal complex in *light grey*. A separate view of the metal complex is shown in the bottom right. Side chains, hydrogen atoms, and included solvent molecules are omitted for clarity.



Scheme 2. Equilibrium between **16** and tetrahedral Cu^{I} complex **17**. Helical chirality (*P*/*M*) and chirality at the metal center (Λ/Δ) result in a mixture of three racemic pairs of diastereomers for **17**: *P*Λ*P*/*M*Δ*M* (**17a**), *P*Δ*P*/ *M*Λ*M* (**17b**), and *P*Λ*M*/*P*Δ*M* (**17c**).

sequence specificity and tunable stability. Since the single strands of the H-bonded duplexes adopt an extended conformation similar to that of β-strands, these single strands can be regarded as *β*-strand mimics and the corresponding duplexes as two stranded *β*-sheet mimics (46–49). They have recently reported intramolecularly hydrogen-bonded foldamer strands that recognize and cross-link even in aqueous solutions. As shown in Fig. 12, the linear oligoamides were modified with *S*-trityl groups, allowing the reversible formation of disulfide bonds. The designed self-assembling strand

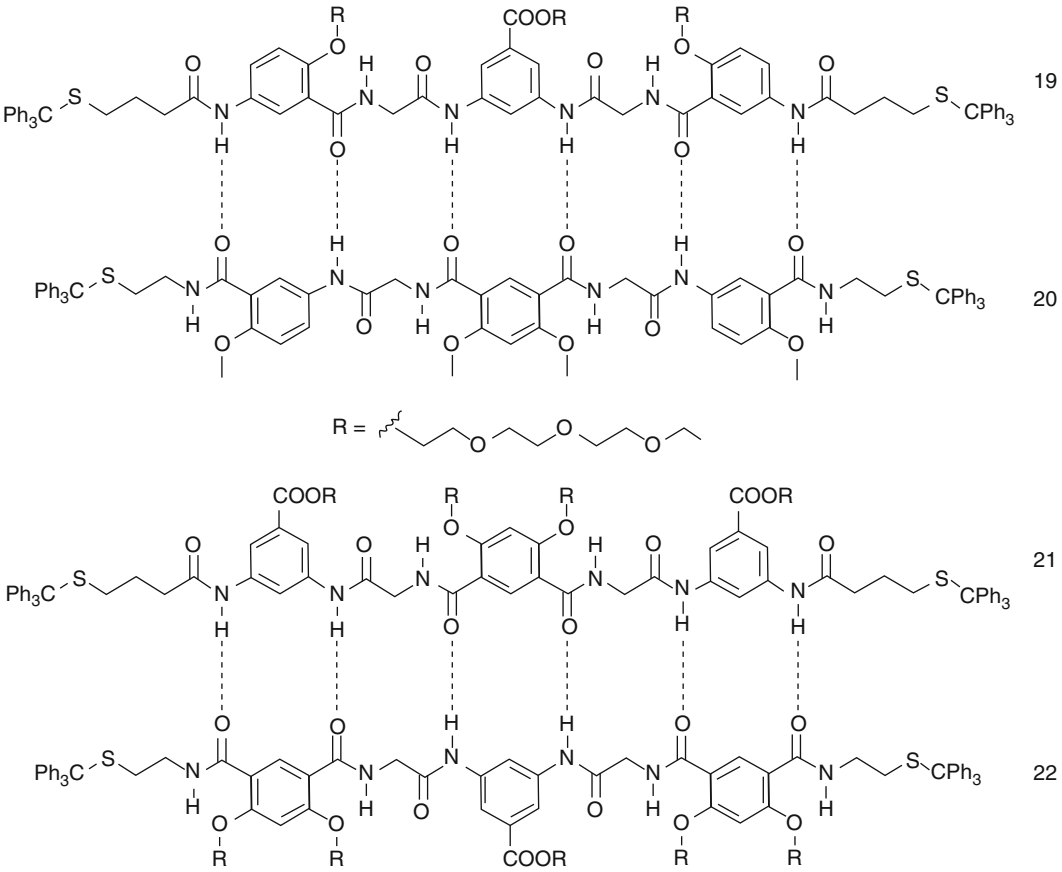


Fig. 12. The complementary strands designed by Gong and coworkers that are capable of recognizing each other even in aqueous media.

pairs **19**–**20** and **21**–**22** can make up to six hydrogen bond pairs with each other and incorporate terminal *S*-trityl groups that can be directly oxidized to disulfides with iodine via a sulfenyl iodide intermediate and the release of trityl cation. The disulfide cross-linking reactions of oligoamides capable of pairing via two, four, and six intermolecular H-bonds, along with several control strands, were examined by ESI, MALDI-TOF, reverse phase HPLC, and two-dimensional NMR.

4. Turns

Sanjayan group developed a repeat β-turn structure motif derived from Aib-Pro-Adb (3-amino-4,6-dimethoxy benzoic acid) building blocks (Fig. 13). The aryl–NH of the Adb unit

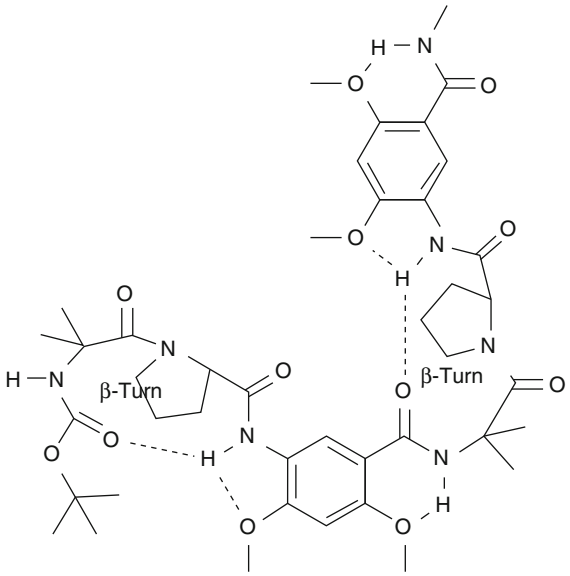


Fig. 13. Repeat β-turn structure motif reported by Sanjayan and coworkers.

makes additional hydrogen bond mediated contacts with the dimethoxy group. The *β*-turn conformation was seen in the X-ray crystal structure of the monomer. 2D NOESY NMR spectra of the dimer in $CHCl_3$ supported the formation of repeating β-turn motifs. The findings suggest that constrained aliphatic-aromatic amino acid conjugates would offer new avenues for the de novo design of foldamers with distinctive structural architectures (50).

5. Conclusion and Outlook

In summary, aromatic oligoamides will continue to provide useful systems to test and extend our understanding of how proteins fold into their native three-dimensional structures. Despite some success in the formation of supersecondary structures from aromatic oligoamides, a major challenge will be the assembly of foldamer secondary/supersecondary structures into more complex protein-like tertiary structures. Overall, the recent work reviewed here provides an excellent foundation for the creation of supersecondary structures. The chemical and biological applications of foldamers with supersecondary will become even more attractive.

Acknowledgment

We thank the National Natural Science Foundation of China (20625206) and the National Basic Research Program (2011CB932501) for financial support.

References

1. Hecht SM, Huc I (2007) Foldamers: structure, properties and applications. Wiley-VCH, Weinheim, Germany
2. Gellman SH (1998) Foldamers: a manifesto. Acc Chem Res 31:173–180
3. Hill DJ, Mio MJ, Prince RB et al (2001) A field guide to foldamers. Chem Rev 101:3893–4011
4. Cheng RP, Gellman SH, DeGrado WF (2001) β-Peptides: from structure to function. Chem Rev 101:3219–3232
5. Seebach D, Hook DF, Glattli A (2006) Helices and other secondary structures of β- and γ-peptides. Biopolymers 84:23–37
6. Seebach D, Overhand M, Kuehnle FNM et al (1996) 3-Peptides: synthesis by Arndt-Eistert homologation with concomitant peptide coupling. Structure determination by NMR and CD spectroscopy and by X-ray crystallography. Helical secondary structure of a P-hexapeptide in solution and its stability towards pepsin. Helv Chim Acta 79:913–941
7. Nelson JC, Saven JG, Moore JS et al (1997) Solvophobically driven folding of nonbiological oligomers. Science 277:1793–1796
8. Appella DH, Christianson LA, Klein DA et al (1999) Synthesis and structural characterization of helix-forming β-peptides: trans-2-aminocyclopentanecarboxylic acid oligomers. J Am Chem Soc 121:7574–7581
9. van Gorp JJ, Vekemans JAJM, Meijer EW (2004) Facile synthesis of a chiral polymeric helix; folding by intramolecular hydrogen bonding. Chem Commun 2004:60–61
10. Claridge TDW, Long DD, Baker et al (2005) Helix-forming carbohydrate amino acids. J Org Chem. 70:2082–2090
11. Violette A, Averlant-Petit MC, Semetey V et al (2005) N, N′-Linked oligoureas as foldamers: chain length requirements for helix formation in protic solvent investigated by circular dichroism, NMR spectroscopy, and molecular dynamics. J Am Chem Soc 127:2156–2164
12. Menegazzo I, Fries A, Mammi S et al (2006) Synthesis and structural characterization as 12-helix of the hexamer of a β-amino acid tethered to a pyrrolidin-2-one ring. Chem Commun 2006:4915–4917
13. Goto H, Katagiri H, Furusho Y et al (2006) Oligoresorcinols fold into double helices in water. J Am Chem Soc 128:7176–7178
14. Vasudev PG, Ananda K, Chatterjee S et al (2007) Hybrid peptide design. Hydrogen bonded conformations in peptides containing the stereochemically constrained γ-Amino acid residue, gabapentin. J Am Chem Soc 129:4039–4048
15. Ousaka N, Sato T, Kuruda R (2008) Sequence-specific unusual $(1\rightarrow2)$-type helical turns in α/β-hybrid peptides. J Am Chem Soc 130:463–365
16. Baruah PK, Gonnade R, Rajamohanan et al (2007) BINOL-based foldamers access to oligomers with diverse structural architectures. J Org Chem. 72:5077–5084.
17. Krauthäuser S, Christianson LA, Powell DR et al (1997) Insertion of methylene units into the turn segment of designed β-hairpin peptides. J Am Chem Soc 119:11719–11720
18. Seebach D, Abele S, Gademann K et al (1999) Pleated sheets and turns of β-peptides with proteinogenic side chains. Angew Chem Int Ed 38:1595–1597
19. Gong B, Yan Y, Zeng H et al (1999) A new approach for the design of supramolecular recognition units: hydrogen-bonded molecular duplexes. J Am Chem Soc 121:5607–5608
20. Nowick JS (1999) Chemical models of protein β-sheets. Acc Chem Res 32:287–296
21. Woll MG, Lai JR, Guzei IA et al (2001) Parallel sheet secondary structure in gamma-peptides. J Am Chem Soc 123:11077–11078
22. Kendhale A, Gonnade R, Rajamohanan PR et al (2006) Isotactic N-alkyl acrylamide oligomers assume self-assembled sheet structure: first unequivocal evidence from crystal structures. Chem Commun 2006:2756–2758
23. Baruah PK, Sreedevi NK, Majumdar B et al (2008) Sheet-forming abiotic hetero foldamers. Chem Commun 2008:712–714
24. Chung YJ, Christianson LA, Stanger HE et al (1998) A β-peptide reverse turn that promotes

hairpin formation. J Am Chem Soc 120:10555–10556
25. Yang D, Li B, Ng F-F et al (2001) Synthesis and characterization of chiral N-O turns induced by α-aminoxy acids. J Org Chem 66:7303–7312
26. Chen F, Zhu N-Y, Yang D (2004) γ4-aminoxy peptides as new peptidomimetic foldamers. J Am Chem Soc 126:15980–15981
27. Salaun A, Potel M, Roisnel T et al (2005) J Org Chem 70:6499–6502
28. Baruah PK, Sreedevi NK, Gonnade R et al (2007) Enforcing periodic secondary structures in hybrid peptides: a novel hybrid foldamer containing periodic γ-turn motifs. J Org Chem 72:636–639
29. Chen F, Song K-S, Wu Y-D et al (2008) Synthesis and conformational studies of γ-aminoxy peptides. J Am Chem Soc 130:743–755
30. Rao ST, Rossmann MG (1973) Comparison of super-secondary structures in proteins. J Mol Biol 76:241–256
31. Creighton TE (1999) Encyclopedia of molecular biology, vol 1–4. Wiley, New York.
32. Hu Z-Q, Hu H-Y, Chen C-F (2006) Phenanthroline dicarboxamide-based helical foldamers: stable helical structures in methanol. J Org Chem 71:1131–1138
33. Hu H-Y, Xiang J-F, Yang Y et al (2008) Folding-induced selective helix-turn-helix supersecondary structure based on oligo(phenanthroline dicarboxamide)s. Org Lett 10:69–72
34. Hu H-Y, Xiang J-F, Cao J et al (2008) Hydrogenation of helical 9,10-anthraquinone analogues. Org Lett 10:5035–5038
35. Hu H-Y, Xiang J-F, Chen C-F (2009) Conformationally constrained aromatic oligoamide foldamers with supersecondary structure motifs. Org Biomol Chem 7:2534–2539
36. Blay G, Fernädez I, Pedro JR et al (2003) A hydrogen-bonded supramolecular meso-helix. Eur J Org Chem 2003:1627–1630
37. Plasseraud L, Maid H, Hampel F et al (2001) A meso-helical coordination polymer from achiral dinuclear [Cu2(H3CCN)2(μ-pydz)3][PF6]2 and 1,3-bis(diphenylphosphanyl) propane-synthesis and crystal structure of ∞1{[Cu(μ-pydz)2][PF6]} (pydz = pyridazine). Chem Eur J 7:4007–4011
38. Hu H-Y, Wei X, Hu Z-Q et al (2009) Probing the dynamic environment- associated conformational conversion from secondary to supersecondary structures in oligo(phenanthroline dicarboxamide)s. J Org Chem 74:4949–4957
39. Huc I (2004) Aromatic oligoamide foldamers. Eur J Org Chem 2004:17–29
40. Maurizot V, Dolain C, Leydet Y et al (2004) Design of an inversion center between two helical segments. J Am Chem Soc 126:10049–10052
41. Delsuc N, Hutin M, Campbell et al (2008) Metal-directed dynamic formation of tertiary structure in foldamer assemblies: orienting helices at an angle. Chem Eur J. 14:7140–7143.
42. Somers WS, Phillips SEV (1992) Crystal structure of the met represser-operator complex at 2.8 Å resolution reveals DNA recognition by β-strands. Nature 359:387–393
43. Puglisi JD, Chen L, Blanchard S et al (1995) Solution structure of a bovine immunodeficiency virus tat-TAR peptide-RNA Complex. Science 270:1200–1203
44. Derrick JP, Wigley DB (1992) Crystal structure of a streptococcal protein G domain bound to an Fab fragment. Nature 359:752–754
45. Colon W, Kelly JW (1992) Partial denaturation of transthyretin is sufficient for amyloid fibril formation in vitro. Biochemistry 31:8654–8660
46. Gong B, Yan YF, Zeng HQ et al (1999) A new approach for the design of supramolecular recognition units: hydrogen-bonded molecular duplexes. J Am Chem Soc 121: 5607–5608
47. Li MF, Yamato K, Ferguson JS et al (2006) Sequence-specific association in aqueous media by integrating hydrogen bonding and dynamic covalent interactions. J Am Chem Soc 128:12628–12629
48. Li MF, Yamato K, Ferguson JS et al (2008) Sequence-specific, dynamic covalent crosslinking in aqueous media. J Am Chem Soc 130:491–500
49. Bialecki JB, Yuan LH, Gong B (2007) A branched, hydrogen-bonded heterodimer: a novel system for achieving high stability and specificity. Tetrahedron 63:5460–5469
50. Srinivas D, Gonnade R, Ravindranathan S et al (2007) Conformationally constrained aliphatic—aromatic amino-acid-conjugated hybrid foldamers with periodic β-turn motifs. J Org Chem 72:7022–7025

Part IV

Other Applications of Supersecondary Structure to Protein Biology

Chapter 15

Cross-β-Sheet Supersecondary Structure in Amyloid Folds: Techniques for Detection and Characterization

Raimon Sabaté and Salvador Ventura

Abstract

The formation of protein aggregates is linked to the onset of several human disorders of increasing prevalence, ranging from dementia to diabetes. In most of these diseases, the toxic effect is exerted by the self-assembly of initially soluble proteins into insoluble amyloid-like fibrils. Independently of the protein origin, all these macromolecular assemblies share a common supersecondary structure: the cross-β-sheet conformation, in which a core of β-strands is aligned perpendicularly to the fibril axis forming extended regular β-sheets. Due to this ubiquity, the presence of cross-β-sheet conformational signatures is usually exploited to detect, characterize, and screen for amyloid fibrils in protein samples. Here we describe in detail some of the most commonly used methods to analyze such supersecondary structure.

Key words: Amyloid, Beta-fold, Fibril, Cross-beta-sheet, Protein aggregation

1. Introduction

In the cell, the final protein conformation at equilibrium is the result of a delicate and multi-step balance regulated by diverse intrinsic and extrinsic factors. In this way, when polypeptide chains emerge from the ribosome, their spontaneous conformational folding into native and functional structures can be competed by self-aggregation side-reactions leading to the formation of insoluble β-sheet-enriched structures. Moreover, even once proteins have attained their native structure, conformational fluctuations under stress conditions might promote their self-assembly and subsequent deposition. This competition between folded and aggregated states cannot be avoided, because many of the biophysical traits that promote folding also tend to favor interactions leading to the formation of the intermolecular β-sheets that sustain the common core of

Alexander E. Kister (ed.), *Protein Supersecondary Structures*, Methods in Molecular Biology, vol. 932,
DOI 10.1007/978-1-62703-065-6_15,

aggregated structures (1). The aggregated state represents in fact a ground state for protein folding, alternative to that populated by the native state, and accordingly protein aggregation reactions are now recognized as major contributors shaping the folding energy landscapes of protein (2). Protein misfolding and aggregation has become a widely active area of research in recent years, mainly because of the connection between the formation of insoluble protein deposits in human tissues and the development of dozens of conformational diseases. These protein deposits are constituted mainly by fibrillar structures known as amyloids that are characterized by a polypeptide backbone organization in a cross-β structure consisting of β-strands stacked perpendicular to the fibril axis. It is important to note that all proteins shown to form amyloid fibrils to date share this common fold in their aggregated state, despite the fact that they do not share any sequential or structural similarities in their respective native states. For many years, the characterization of the cross-β-sheet motif with high resolution techniques remained elusive and its presence has been inferred from the results obtained using a battery of assays including transmission electron microscopy (TEM), atomic force microscopy (AFM), staining with amyloidtropic dyes such as Thioflavins (Th) and Congo Red (CR), limited proteolysis, or checking out the seeding capacity characteristic of amyloid assemblies. Concomitantly, secondary structure analysis by circular dichroism, Fourier transformed infrared spectroscopy, or X-ray diffraction of fibrils has been used to identify the characteristic cross-β-sheet signature in these protein aggregates. Only recently, structural studies using X-ray crystallography and solid-state NMR have made possible to visualize with atomic detail the series of interactions that allow the formation of this highly ordered and densely packed fold. In this review we try to provide readers with a detailed list of the most commonly used low/medium-resolution methods and techniques and their application to identify and verify the presence of amyloid cross-β-sheet secondary structures in protein assemblies. High-resolution techniques are not included here because they require large equipment and/or high expertise, being thus accessible only to a reduced set of laboratories; excellent papers and reviews on such approaches have been published recently (3–9).

2. Materials

2.1. Dye Staining

2.1.1. Congo Red Binding

1. Reagents.
   1.1 Congo Red (Sigma Chemical Company, St. Louis, MO, USA).
   1.2 milliQ water (Millipore, Billerica, MA, USA).

1.3 Other chemical reagents and buffers can be obtained from (Sigma Chemical Company, St. Louis, MO, USA).

1.4 Nucleopore polycarbonate membranes with a 0.4-μm nominal pore size (Whatman, Bandury, OX, UK).

2. Dye preparation.

2.1 CR stock solution is prepared by dissolving the required CR amount in milliQ water to obtain a dye concentration of 200 mM (2×).

2.2 When required, 1 vol. of 2× Tris–HCl buffer at pH 7.5 can be added to obtain a 100 mM solution of CR (1×) (see Note 1).

2.3 CR stock solution is filtered through polycarbonate membranes to remove dye aggregates.

2.4 Additional point. To minimize dye adsorption, the glassware and cuvettes are silanated with 2% (v/v) dichloromethylsilane/toluene solution and then rinsed with methanol.

CR UV–Vis Absorbance

1. Reagents. See item 1 in Subheading 2.1.1.
2. Equipment.

2.1 Cary 100 or 400 UV/Vis spectrophotometer (Varian, Palo Alto, CA, USA).

2.2 GraphPad Prism 5 (GraphPad Software Inc., La Jolla, CA, USA).

Spectrophotometric Determination of Bound CR

1. Reagents. See item 1 in Subheading 2.1.1.
2. Equipment. See item 1 in Subheading 2.1.1 and item 2 in Subheading "CR UV–Vis Absorbance".

2.1 Centrifuge Eppendorf 5424 (Eppendorf International Corporation, Hamburg, Germany).

CR Birefringence Assay

1. Reagents. See item 1 in Subheading 2.1.1.
2. Equipment. See item 2.1 in the Subheading "Spectrophotometric Determination of Bound CR".

2.1 Optic microscope with cross-polarized light (Leica DMRB, Heidelberg, Germany).

CR Fluorescence Microscopy Assay

1. Reagents. See item 1 in Subheading 2.1.1.
2. Equipment. See item 2.1 in the Subheading "Spectrophotometric Determination of Bound CR".

2.1 Leica fluorescence DMBR microscope with a narrow green band filter (Leica Microsystems AG, Heidelberg, Germany).

2.1.2. Thioflavin (Th) Binding

1. Reagents.
   1.1 Thioflavin-S and T (Sigma Chemical Company, St. Louis, MO, USA).
   1.2 Other reagents: See items 1.2 and 1.3 in Subheading 2.1.1.
2. Equipment. See item 2.1 in Subheading 2.1.1 and item 2.1 in the Subheading "CR UV–Vis Absorbance".
3. Dye preparation.
   3.1 Th-S and Th-T stock solution is prepared by dissolving the required Th-S and Th-T (Sigma Chemical Company, St. Louis, MO, USA) amount in milliQ water to obtain a dye concentration of 250 μM (see Notes 2 and 3).
   3.2 Th-T stock solution is filtered through polycarbonate membranes with a 0.4-μm nominal pore size to remove the Th-S and Th-T aggregates. If the final concentration of stock solution wants to be confirmed a molar absorptivity of 36 000 M^{-1} cm^{-1} at 412 nm can be used (10).

Thioflavin-T Relative Fluorescence Assay

1. Reagents. See item 1 in Subheading 2.1.2.
2. Equipment.
   2.1 Cary Eclipse spectrofluorimeter (Varian, Palo Alto, CA, USA).

Thioflavin-T Steady Fluorescence Anisotropy Assay

1. Reagents. See item 1 in Subheading 2.1.2.
2. Equipment. See item 2 in Subheading "Thioflavin-T Relative Fluorescence Assay".

Determination of Thioflavin-T Binding by Induced Circular Dichroism

1. Reagents: See item 1 in Subheading 2.1.2.
2. Equipment: Jasco 810 spectropolarimeter (JASCO International Co. Ltd., Tokyo, Japan).

Thioflavin-S and Thioflavin-T Binding by Fluorescence Microscopy

1. Reagents. See item 1 in Subheading 2.1.2.
2. Equipment. See item 2 in Subheading "CR Fluorescence Microscopy Assay".

2.2. Secondary Structure Analysis

2.2.1. Detection of β-Sheet rich Structures by Circular Dichroism

1. Reagents. See items 1.2 and 1.3 in Subheading 2.1.1.
2. Equipment. See item 2 in Subheading "Determination of Thioflavin-T Binding by Induced Circular Dichroism".

2.2.2. Detection of β-Sheet Rich Structures by Fourier Transformed Infrared

1. Reagents. See items 1.2 and 1.3 in Subheading 2.1.1.
   1.1 Deuterated water (D_2O) (Sigma Chemical Company, St. Louis, MO, USA).
2. Equipment.

2.1 The PeakFit package (Systat Software, San Jose, CA, USA) can be used for nonlinear peak-fitting.

2.2 For Absorption FT-IR spectroscopy analysis.

2.2.1 FTS-6000 FT-IR spectrophotometer (BioRad, Hemel Hempstead, UK) equipped with a liquid N_2-cooled mercury cadmium telluride detector (Bio-Rad Laboratories, Inc., Hercules, CA, USA).

2.3 For ATR FT-IR spectroscopy analysis.

2.3.1 Bruker Tensor 27 FT-IR Spectrometer (Bruker Optics Inc., Karlsruhe, Germany) with a Golden Gate MKII ATR accessory.

2.3.2 OPUS MIR Tensor 27 software (OPUS OPtics User Software, Bruker Optics Inc., Karlsruhe, Germany).

3. Sample preparation.

3.1 Exchangeable hydrogen molecules in the protein are replaced by deuterons by dissolving the protein in D_2O ~2.0 mM, leaving the sample at room temperature for 12 h followed by lyophilization.

3.2 The previous step would be repeated until the labile hydrogen molecules in the sample are completely replaced by deuterons ≈98% exchange.

3.3 The extent of deuteration can be analyzed by electrospray mass spectrometry.

3.4 The pH of the solution can be adjusted with a D_2O based buffer solution.

3.5 Typically a 1 or 2 mM final protein concentration is necessary.

2.2.3. X-Ray Diffraction

1. Reagents. See items 1.2 and 1.3 in Subheading 2.1.1.
2. Rigaku X-ray diffractometer (Rigaku Americas, The Woodlands, Texas USA) with rotating anode Cu *K*α microfocus (11) and RAxis 4++ detector.
3. CLEARER, specific software for the analysis of X-ray fiber diffraction patterns (12).

2.3. Limited Proteolysis. Visualization by SDS-PAGE and Mass Spectroscopy

1. Reagents. See items 1.2 and 1.3 in Subheading 2.1.1.

1.1 Proteinase-K and trypsin can be obtained from Sigma Chemical Company (St. Louis, MO, USA).

1.2 3,5-Dimethoxy-4-hydroxycinnamic (sinapinic acid) and a-Cyano-4-hydroxy-cinnamic acid can be obtained from Sigma Chemical Company (St. Louis, MO, USA).

2. Equipment.
   2.1 Ultraflex MALDI-TOF mass spectrometer (Bruker Daltonics, Karlsruhe, Germany).
   2.2 The Quantity One software from Bio-Rad Laboratories, Inc. (Hercules, CA, USA) can be used to scan the SDS-PAGE gels at high-resolution.
   2.3 For Edman N-terminal sequencing: ABI Procise Model 492 Edman Micro Sequencer connected to an ABI Model 140 °C PTH Amino Acid Analyzer from Perkin Elmer Applied Biosystems (Foster City, CA, USA).

2.4. Aggregation Kinetics: Seeding and Cross-Seeding Analysis

1. Reagents. See items 1.2 and 1.3 in Subheading 2.1.1.
2. Equipment. For turbidity, relative fluorescence, CD ellipticity, and FT-IR see item 2.1 in Subheading "CR UV–Vis Absorbance", item 2.1 in Subheading "Thioflavin-T Relative Fluorescence Assay", item 2 in Subheading "Determination of Thioflavin-T Binding by Induced Circular Dichroism", and item 2.2.2 in Subheading 2.2.2, respectively. For nonlinear regression fitting see item 2.2 in Subheading "CR UV–Vis Absorbance".

2.5. Microscopic Techniques

2.5.1. Transmission Electronic Microscopy

1. Reagents. See items 1.2 and 1.3 in Subheading 2.1.1.
   1.1 Uranyl acetate can be obtained from Sigma Chemical Company (St. Louis, MO, USA)
2. Equipment.
   2.1 Hitachi H-7000 transmission electron microscope (Hitachi, Tokyo, Japan) operating at an accelerating voltage of 75 kV.

2.5.2. Atomic Force Microscopy

1. Reagents. See items 1.2 and 1.3 in Subheading 2.1.1.
2. Equipment.
   2.1 Highly oriented pyrolytic graphite (HOPG) from NT-MDT Co. (Moscow, Russia).
   2.2 Multimode atomic force microscope equipped from Veeco Instruments, Inc. (Santa Barbara, CA, USA).
   2.3 In tapping mode, Veeco NP-S probes form Bruker Optics Inc. (Karlsruhe, Germany).

3. Methods

Because all the below described methods and techniques report on the presence of cross-β-sheet conformations in the aggregated state, but most of them provide only indirect measurements of

such property, researchers should select the right number and combination of tests to perform in order to obtain conclusive data pointing to the amyloid nature of the species under analysis. We want to note that the present review does not address the detection of amyloid assemblies in clinical studies and therefore that the lab protocols represent a generic list of useful methods for the analysis of amyloid-like aggregates formed in vitro or obtained after purification from in vivo or ex vivo experiments.

3.1. Dye Staining

Nowadays the scientific community can chose among a wide range of amyloid-dyes useful for different purposes. Herein, we focus our attention on the most widely used dyes: Congo Red and Thioflavins.

3.1.1. Congo Red Binding

CR, first synthesized in 1883, is the sodium salt of benzidinediazo-bis-1-naphthylamine-4-sulfonic acid (13) and has been widely used as histopathological staining test of amyloid plaques. In 1989 Klunk and coworkers showed that this histological hydrophilic diazo dye binds to insulin and poly-L-lysine fibrils and proposed its application for the quantification of amyloid-like aggregates (14, 15) (see Fig. 1). Since then it has been widely used in the study of conformational diseases and in the detection of amyloid structures (16). Changes in absorbance, birefringence, and fluorescence are detectable when the dye binds to cross-β-sheet supersecondary structure.

CR UV–Vis Absorbance

In the presence of intermolecular β-sheet structure, as it is the case of amyloid fibrils, the CR absorbance spectra shifts from orange-red to pink. Its binding mechanism has been elucidated by Scatchard analysis assuming an independent binding mode and several algorithms have been developed for quantification of CR complexes on the basis of spectral changes (e.g., for Aβ40 aggregates). However, we advise to calculate the absorptivity of free and bound CR for each amyloid protein and experimental condition, in order to obtain accurate binding data (14, 15, 17–20).

1. CR spectra can be determined using a UV/Vis spectrophotometer in the 375–675 nm range using a matched pair of quartz cuvettes of 1 cm optical length placed in cell holder thermostated at the required temperature.

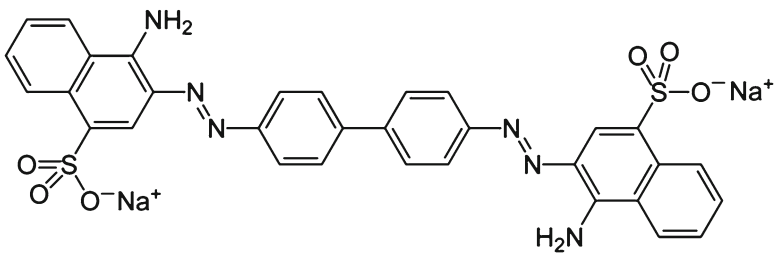


Fig. 1. Chemical structure of Congo red.

2. Usually, the presence of amyloid fibrils in the solution promotes significant light scattering that interferes with the binding assay. Therefore, this signal has to be subtracted from the resulting CR spectra (see Note 4).
3. Although CR binding to amyloids is commonly a fast process, the samples are better incubated for 10–15 min before measurements.
4. To detect the typical amyloid band at ≈541 nm, the differential CR spectrum is plotted by subtracting the spectrum of free CR from that of bound CR.
5. Determination of the binding constants and the amount of bound peptide.
   5.1 If the absorptivities of free and bound CR are known (see Note 5), the binding can be calculated using the typical one-site binding equation in a curve fitting program (i.e., Graphpad Prism):

$$[CR_{bound}] = \frac{B_{max} * [CR]}{K_d + [CR]} \quad (1)$$

   where $[CR_{bound}]$ is the concentration of CR bound to amyloid peptides, B_{max} is the maximum number of binding sites expressed in concentration, $[CR]$ is the CR concentration, and K_d is the process constant.

   5.2 The amyloid peptide concentration can be also determined using the previously determined molar absorptivities of the free and bound species for our amyloid protein and experimental conditions. This spectrophotometric method is based in the use of two wavelengths, in which the molar absorptivities of both the free and bound forms of the dye and peptide must be known, to determine the amount of bound dye. Wavelengths of 541 and 403 nm, which correspond to those of the maximal spectral difference between free and bound CR, are used in the following equation:

$$[\mathrm{CR}_{\mathrm{bound}}] = \frac{\left(A^{541}_{\mathrm{total}} / \varepsilon^{541}_{\mathrm{free}}\right) - \left(A^{403}_{\mathrm{total}} / \varepsilon^{403}_{\mathrm{free}}\right)}{\left(\varepsilon^{541}_{\mathrm{bound}} / \varepsilon^{541}_{\mathrm{free}}\right) - 1} \quad (2)$$

   where A^{541}_{total} and A^{403}_{total} are the absorbances of CR–protein complex at 541 and 403 nm, respectively, and $\varepsilon^{541}_{\mathrm{free}}$, $\varepsilon^{541}_{\mathrm{bound}}$, $\varepsilon^{403}_{\mathrm{free}}$ and $\varepsilon^{403}_{\mathrm{bound}}$ are the molar absorptivities for the free and bound CR at each wavelength. When the stoichiometry of binding is known, the amount of fibrillar peptide can be approximated from $\mathrm{CR}_{\mathrm{bound}}$. For example, for Aβ40 peptide, because 1 μg/ml of fibrillar species corresponds approximately to 0.1 μM $\mathrm{CR}_{\mathrm{bound}}$, a factor of 10 can be used to convert Eq. 2 into an equation that allows approximating the concentration of fibrillar Aβ40 ($[\mathrm{A\beta40}_{\mathrm{fib}}]$):

$$[\mathrm{A}\beta 40_{\mathrm{fib}}] = \frac{A_t^{541}}{4,780} - \frac{A_t^{403}}{6,830} - \frac{[\mathrm{CR}]}{0.477} \quad (3)$$

where [CR] is the concentration of the dye in the original solution (17, 18).

Spectrophotometric Determination of Bound CR

This spectrophotometric method represents an interesting alternative to the method described above. It includes a procedure to precipitate amyloid bound CR by centrifugation. Only the concentration of the nonprecipitated dye, which remains in solution, should be quantified (20). Thus, the method only requires knowing the molar absorptivity of the free dye.

1. CR spectra can be determined using a UV/Vis spectrophotometer in the 375–675 nm range using a matched pair of quartz cuvettes of 1 cm optical length placed in cell holder thermostated at the required temperature.
2. The samples are usually incubated for 10–15 min before analysis (see Note 6).
3. After incubation the samples are centrifuged at 14,000 × *g* for 30 min.
4. The spectra of soluble fractions are analyzed in the 375–675 nm range.
5. Knowing the molar absorptivity of free CR allows determining the concentration of free CR in the soluble fraction and therefore, subtracting this value from the initial CR concentration in the assay permits to calculate the amount of CR bound to amyloid material.

CR Birefringence Assay

Birefringence, or double refraction, is the decomposition of a ray of light into two rays when it passes through certain anisotropic materials, such as crystals. The fixation of CR molecules along amyloid fibrils axis usually causes apple-green birefringence when viewed through cross-polarized light providing a more specific assessment of the amyloid nature of protein aggregates than CR absorbance measurements (21, 22).

1. Protein samples are incubated for 1 h in the presence of 50 μM CR.
2. The samples are centrifuged at 14,000 × *g* for 5 min. Note that this step is optional but recommended when the concentration of amyloid material is low. Although not strictly necessary, the excess of CR can be washed up by centrifugation, removal of the soluble fraction and resuspension of the aggregated fraction containing the amyloid material in milliQ water for three-times.

3. The precipitated fraction is placed on a microscope slide and sealed.
4. The CR birefringence can be detected under cross-polarized light using an optic microscope.

CR Fluorescence Microscopy Assay

Congo red fluorescence is an alternative method to detect amyloids. It is based on the visualization of amyloid-bound CR under UV light. Since, in 1965 Putchler and coworkers showed CR for first time CR as a stain for fluorescence microscopy of amyloids (23), this technique has been used for the histological detection of an increasing number of amyloid deposits. This method is highly sensitive and stringent (24).

1. Protein samples have to be incubated for 1 h in the presence of 50 μM CR.
2. The samples are centrifuged at 14,000 × *g* for 5 min. Note that this step is optional but recommended when the concentration of amyloid material is low. The precipitated fraction is placed on a microscope slide and sealed.
3. The CRF can be detected under UV light using an optic microscope with a narrow green band filter.

3.1.2. Thioflavin (Th) Binding

Thioflavins have been widely used for both histology and biophysical studies of amyloid formation and detection. Two Th variants are commonly used: Thioflavin-S and T.

Thioflavin-S (Th-S) is a mixture of compounds that results from the methylation of dehydrothiotoluidine with sulfonic acid, has been vastly used in the histological stain of amyloid aggregates. However, as a mixture of compounds, its molar concentration cannot be accurately calculated and its high fluorescence background impedes that this dye can be used to quantify amyloids (25).

Thioflavin-T (Th-T) consists of a pair of benzothiazole and benzaminic rings freely rotating around a shared C–C bond (see Fig. 2). The rotation around this C–C bond promotes the stabilization of the relaxed and nonfluorescent state of the dye limiting the fluorescence yield. In contrast, fixation into amyloid fibrils blocks the rotation capacity of the dye leading to the predominance of the nonrelaxed and fluorescent state. Accordingly, Th-T fixation upon binding to amyloid-like aggregates promotes a strong

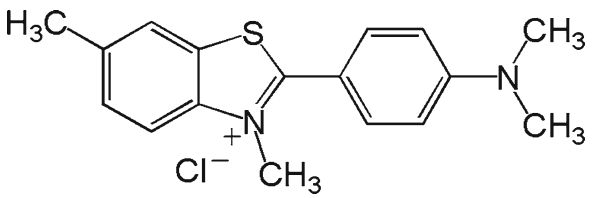


Fig. 2. Chemical structure of Thioflavin-T.

increase in fluorescence emission intensity around 480 nm when excited at 450 nm (26, 27). Note that the excitation and emission wavelengths might change slightly depending of each particular amyloid structure.

Despite the fact that the mechanism underlying fibril induced ThT-fluorescence remains largely unknown at the molecular level, Th-T is probably the most vastly used "specific" amyloid-like dye. Diverse techniques can be used to detect the presence of the characteristic intermolecular cross-β-sheet structure by Th-T, including fluorescence enhancement, anisotropy and fluorescence microscopy, or detection of the cotton-effect by circular dichroism (see Note 7) (27, 28).

Thioflavin-T Relative Fluorescence Assay

Because Th-T does not interfere with amyloid aggregation and its binding is usually very fast, this can be considered as the default protocol for monitoring amyloid aggregation kinetics.

1. 25 μM of Th-T is added to each protein sample (see Note 8).
2. The Th-T relative fluorescence (RF) can be measured immediately using a spectrofluorometer with an excitation wavelength of 450 nm and an emission range from 470 to 570 nm at room temperature (see Note 9).
3. The time-dependent changes in maximal Th-T emission fluorescence are commonly recorded at ~480 nm.

Thioflavin-T Steady Fluorescence Anisotropy Assay

Th-T fluorescence enhancement is strongly dependent on fibril morphology. In some cases, amyloid fibrils can fix Th-T without detection of significant fluorescence enhancement because the rotation movement of the Th-T molecule is not sufficiently impeded. In these occasions, Th-T fluorescence anisotropy provides an alternative technique for the study of amyloid aggregation as well as for kinetic studies (29).

1. 25 μM of Th-T is added to each protein sample.
2. Th-T fluorescence anisotropy can be measured using a spectrofluorometer with polarization filters by measuring the fluorescent emission of Th-T at 480 nm after excitation at 450 nm at room temperature.
3. The anisotropy values can be calculated using the following equation:

$$A = \frac{(I_{VV} - GI_{VH})}{(I_{VV} + 2GI_{VH})} \quad (4)$$

where A is the anisotropy, I is the relative fluorescence and G represents the ratio of to I_{HH}, and V and H in the subscript represent the vertical or the horizontal position of the excitation and the emission polarizers (see Note 10).

Determination of Thioflavin-T Binding by Induced Circular Dichroism

Despite technical limitations (e.g., the high protein concentration necessary for assay) the determination of the Th-T "cotton-effect" represents a stringent method to confirm the formation of amyloid structures. The Th-T molecule is twisted when it is fixed to amyloid fibril. This induced chiral activity of an otherwise achiral molecule can be detected using CD. The negative value of the cotton effect indicates that Th-T and therefore the fibril that imprints this chiral bias upon the Th-T molecule, display a left-handed twist.

1. 500 μM of Th-T are added to 500 μM of each protein sample (see Note 11).
2. CD spectra are recorded at a spectral resolution of 1 cm^{-1} and a scan rate of 15 nm/min in a wavelength range from 375 to 525 nm at room temperature using a spectropolarimeter with a quartz cell of 0.1 cm path length.

Thioflavin-S and Thioflavin-T Staining by Fluorescence Microscopy

The staining protocol used for the visualization of amyloid plaques in tissues can be easily applied in the detection of purified or partially purified amyloid fibrils and amyloid-like aggregates (3).

1. The protein samples are incubated for 1 h in the presence of 125 μM of Th.
2. The samples are centrifuged at 14,000 × *g* for 5 min. As an optional process, the excess of Th can be washed up by centrifugation (removing the soluble fraction) and resuspension in milliQ water for three times.
3. The precipitated fractions are placed on a microscope slide and the coverslip sealed with clear nail polish (see Note 12).
4. The fluorescence images of fibrillar aggregates can be obtained at 40-fold magnification under UV light with a fluorescence microscope.

3.2. Secondary Structure Content Analysis

Amyloid fibrils or amyloid-like aggregates are considered as thread-like protein aggregates with a core region formed by repetitive arrays of β-sheets oriented perpendicular to fibril axis forming the known cross-β structure (30). Since these β-sheet rich structures exhibit specific circular dichroism (CD) and Fourier Transformed Infra Red (FT-IR) spectra and X-ray diffraction patterns, these techniques have become standard assays for amyloid detection.

3.2.1. Detection of β-Sheet-Rich Structures by Circular Dichroism

The β-sheet secondary structure in amyloids displays a characteristic minimum at 217 nm in the far-UV region of the CD spectrum that occasionally can be displaced slightly to higher wavelengths due to the stacking of aromatic residues in these tightly packed structures (3, 30).

1. The protein is placed at the required concentration (usually ranging from 5 to 20 μM) in a quartz cell of 0.1 cm (or 1 cm) path length.

2. CD spectra are usually determined at a spectral resolution of 1 cm^{-1} and a scan rate of 15 nm/min in a wavelength range from 190 to 250 nm at room temperature using a spectropolarimeter.
3. In order to determine the secondary structure components of each sample, the CD spectra can be deconvoluted with the aid of the K2D2 Suite (http://www.ogic.ca/projects/k2d2/) or similar programs (see Notes 13–15).

3.2.2. Detection of β-Sheet Rich Structures by Fourier Transformed Infrared

Amyloid fibrils display a characteristic band at 1,620–1,630 cm^{-1} in the amide I region of the infrared spectra that is attributed to the tightly bound intermolecular β-strands in the amyloid core. In addition, a secondary band at ~1,692 cm^{-1}, which historically has been assigned to antiparallel β-sheet conformation, can be also detected. Note that this secondary band corresponds in fact to the splitting of the main band and cannot considered as a conclusive signature for a β-sheet antiparallel structure. Solution absorption FT-IR is the most extended technique infrared technique to test the secondary structure of proteins. Nevertheless, because amyloid fibrils tend to precipitate Attenuated Total Reflectance (ATR) FT-IR in which the amyloid aggregates can be deposited and analyzed in the solid state is becoming increasingly popular. A large number of methods and variations for infrared analysis have been described. Herein we describe a basic and simplified method, useful for conventional analysis of the secondary structure content of protein assemblies.

1. Absorption FT-IR spectroscopy analysis (31, 32).
   1.1 The samples of amyloid-like proteins are air-dried.
   1.2 Exchangeable hydrogen atoms are replaced by deuterium by dissolving the dried proteins in D_2O or deuterated buffer.
   1.3 The protein samples are inserted between CaF_2 windows using a 50 mm Mylar spacer. The sample holder is connected to a thermostatic bath set to required temperature.
   1.4 Infrared spectra can be recorded with an FT-IR spectrophotometer equipped with a liquid nitrogen-cooled mercury/cadmium telluride detector and purged with a continuous flow of nitrogen gas.
   1.5 Usually about 250 interferograms are recorded at room temperature with a spatial resolution of 1–2 cm^{-1}.
   1.6 For each single spectrum, water vapor is subtracted and the baseline corrected.
   1.7 The area of the spectrum between 1,700 and 1,600 cm^{-1} is normalized by fitting through overlapping Gaussian curves. The amplitude, center, and bandwidth at half of the maximum amplitude and area of each Gaussian

function are calculated using of a nonlinear peak fitting program.

1.8 Second derivatives of the spectra can be used to determine the frequencies at which the different spectral components are located.

2. ATR FT-IR spectroscopy analysis (32, 33).

2.1 Usually the sample does not need previous manipulation; nonetheless, when the buffer spectrum interferes strongly with the measurements, sample centrifugation and buffer exchange for milliQ water is recommended (2 or 3 washing repetitions would suffice to solve the problem).

2.2 5–10 mL of protein sample are placed in a FT-IR Spectrometer with a Golden Gate MKII ATR accessory.

2.3 The samples are dried under N_2.

2.4 The final spectrum consists of 20 independent scans, measured at a spectral resolution of 1 cm^{-1} over the 1,700–1,600 cm range.

2.5 All spectral data are acquired and normalized with the aid of OPUS MIR Tensor 27 software or similar. Note that the buffer spectra have to be subtracted from each single spectrum.

2.6 Second derivatives of the spectra can be used to determine the frequencies at which the different spectral components are located.

2.7 Additionally, infrared spectra can be fitted through overlapping Gaussian curves and the amplitude, center, and bandwidth at half of the maximum amplitude and area of each Gaussian function calculated by use of a nonlinear peak fitting program.

3.2.3. X-Ray Diffraction

X-ray diffraction of aligned amyloid fibrils displays a characteristic pattern with meridional reflection at 4.7–4.8 Å and equatorial reflection at 10–11 Å compatible with the presence of a cross-β structure (34, 35) (see Note 16).

1. A droplet of solution of amyloid fibrils of is placed between two wax filled capillary tubes on a stretch frame.
2. Amyloid fibrils are allowed to dry to form a partially aligned fiber sample.
3. X-ray diffraction data can be collected using a X-ray diffractometer with a rotating anode Cu K_α microfocus (11) and RAxis 4++ detector with a specimen-to-detector distance of 160 mm and exposure times of 10–20 min.
4. X-ray diffraction patterns are examined using CLEARER (12), specific software for the analysis of X-ray fiber diffraction

patterns and diffraction simulation from atomic structural models or similar.

5. Positions of diffraction signals are measured and potential unit cell dimensions are explored using the unit cell determination algorithm within CLEARER.

3.3. Limited Proteolysis. Analysis by SDS-PAGE and Mass Spectroscopy

Proteinase K (pK) is a serine protease that has been commonly used in molecular biology to digest protein and remove contamination from preparations of nucleic acid. In addition, limited proteolysis can be used to probe the domain structure of proteins (36). Thus, while α-helix, random-coil, and β-turn region can be easily digested for pK, the β-sheet regions and particularly the cross-β-sheet characteristic of amyloid fibrils are highly resistant to pK digestion. The limited proteolysis assay allows identifying the core of amyloid-like aggregates. Usually the soluble and aggregated forms of the amyloid protein are analyzed and their digestion patterns are compared. Two approaches are used: First, determining the extent of digestion at different pK concentrations in a fixed time period and second, analyzing the time course of the digestion for fixed pK concentrations. The progress of time-course digestion experiments is usually resolved on SDS-PAGE or Tricine-SDS gels.

1. The aggregated protein is prepared at 50–100 μM final concentration at neutral pH and 37 °C (conditions wherein the pK is more active).
2. The required pK concentration (an enzyme to substrate ratio from ~1:50 to ~1:5,000) in PBS buffer (pH 7.0) is added to start the reaction.
3. The reaction is stopped by heating the sample for 10 min at 95 °C after addition of 1 vol of denaturing electrophoresis loading buffer (i.e., Laemmli buffer).
4. The samples are charged in a SDS-PAGE or Tricine-SDS gels depending on the expected molecular weight of the resulting protein fragments.
5. Gels can be stained with Coomassie brilliant blue (or alternatively by silver-staining) and scanned at high resolution using adequate software.
6. PK-resistant protein extraction.
   6.1 The protein band is cut out (using an scalpel) obtaining a gel slice that is placed in a microcentrifuge tube previously rinsed with 60% acetonitrile.
   6.2 The gel slice is destained in 100 ml of destaining solution (25 mM ammonium bicarbonate in 50% acetonitrile) for 20–30 min. This step is repeated until the gel slice becomes completely destained (usually three to four times). Alternatively, gel slices can be destained overnight at 4 °C.

6.3 The gel slice is dehydrated in 100 ml of 100% acetonitrile for 5–10 min and dried at room temperature.

6.4 30 ml of 2% acetonitrile in 0.1% formic acid are added to the samples and incubated at room temperature for 15 min. The samples are then vortexed briefly and sonicated for 1 min.

6.5 The sample is vacuum dried in a vacuum centrifuge for 45–60 min until it is dry. The eluted peptides are now ready for analysis by mass spectrometry.

7. Samples are prepared by mixing equal volumes of the protein solution and a matrix solution as sinapinic acid (10 mg/mL) dissolved in aqueous acetonitrile (30%) with trifluoroacetic (0.1%)) by the dried droplet method.
8. The molecular masses of the pK-resistant fragments are analyzed by MALDI-Mass spectroscopy (MALDI-MS).

An alternative and fast method to detect PK-resistant cores is the kinetic determination of the reaction by spectroscopy and subsequent sample analysis by mass spectroscopy (to determine the residues forming the core) or by electronic microscopy (in order to visualize the presence of PK-resistant cores displaying an amyloid morphology).

1. The aggregated protein is prepared at a final concentration of 50–100 μM at neutral pH and 37 °C (conditions wherein the pK is more active).
2. The required pK concentration (an enzyme to substrate ratio from ~1:50 to ~1:5,000) in PBS buffer (pH 7.0) is added to start the reaction.
3. The digestion is monitored for 100–200 min by UV/Vis spectroscopy at 350 nm by measuring the reduction in the light scattering signal.
4. At different reaction times, aliquots of the samples are centrifuged and the insoluble part resuspended in water and frozen in liquid nitrogen to block the digestion process.
5. The samples are then analyzed by electronic microscopy, mass spectroscopy or SDS-PAGE.

3.4. Aggregation Kinetics: Seeding and Cross-Seeding Analysis

Protein aggregation kinetics are monitored by measuring the protein transition from the nonaggregated to the aggregated state. The amyloid polymerization reaction can be followed using different reporters like turbidity, relative Th-T fluorescence, Th-T anisotropy, bis-ANS binding, dicrhoism circular, or FT-IR signals (3, 29, 37). In the seeding and cross-seeding assays, a solution of preaggregated peptide (usually ranging from 1 to 10% of the soluble monomer concentration) is added at the beginning of the reaction. The experiments can be carried out with or without

controlled stirring and an initial soluble monomer concentration in the 10–100 μM range, for most amyloid proteins.

Most amyloid aggregation processes can be described as concentration dependent nucleation-polymerization reactions in which three phases can be distinguished: (1) nucleation or lag phase, (2) elongation reaction and (3) plateau phase. Determination of the nucleation and elongation parameters are essential for a detailed description of the kinetic reaction (38, 39). Kinetic constants can be derived from time course experiments exploiting the fact that the aggregation process of soluble proteins into amyloids can be described as an autocatalytic reaction described by the equation

$$f = \frac{\rho\left\{\exp\left[(1+\rho)kt\right]-1\right\}}{\left\{1+\rho^{*}\exp\left[(1+\rho)kt\right]\right\}} \tag{5}$$

under the boundary condition of $t = 0$ and $f = 0$, where $k = k_e a$ (when a is the protein concentration) and ρ represents a dimensionless constant to describe the ratio of k_n to k.

By nonlinear regression of f against t, the values of ρ and k can be easily obtained, as well as the rate constants k_c (elongation constant) and k_n (nucleation constant). The extrapolation of the growth portion of the sigmoidal curve to the abscissa $f = 0$ and to the highest ordinate value of the fitted plot permit to approximate the two time constant (t_0 and t_1), which correspond to the lag time and to the time at which the aggregation is almost completed (39).

1. An adequate protein concentration, usually 5–100 μM, is used.
2. The Th-T fluorescence, absorbance, CD ellipticity or the chosen parameter to follow the soluble to aggregate transition is recorded in the required time interval.
3. The obtained curve is normalized as a function of the protein fraction (see Note 17).
4. Finally, the resultant curve can be analyzed by fitting the above-mentioned equation Eq. 4 using a nonlinear regression program (e.g., GraphPad Prism). The apparent rate constants are derived from these regressions, as described.

3.5. Microscopic Techniques

Different microscopic techniques have been used to analyze amyloid fibrils morphology. Among them, transmission electronic microscopy (TEM) and AFM are probably the most used.

3.5.1. Transmission Electronic Microscopy

TEM is the default microscopic technique for the detection of amyloid fibrils and for their morphological analysis, usually using negative staining to get image contrast. We propose here a simplified and fast protocol for amyloid detection.

1. Place 5–20 μl of the protein sample at a concentration of 0.1–10 μM (depending on the capacity of the particular protein to fix into the support) on a carbon-coated copper grid. The sample should be centrifuged and the buffer exchanged by milliQ water when it can form crystals upon drying, interfering thus with TEM imaging.
2. The protein samples are left to stand for 5 min.
3. The grids are washed with distilled water.
4. Then the samples are stained with uranyl acetate (2%, w/v) for another 2 min before analysis.
5. Finally the grids are washed with distilled water and left to dry for analysis.
6. Amyloids are usually unbranched protein fibrils with diameters in the 10 nm range. The fibrils are easily visualized with 10,000 to 100,000-fold magnification.

3.5.2. Atomic Forces Microscopy

Tapping-mode AFM or scanning force microscopy (SFM) is a high-resolution type of scanning probe microscopy. The technique is increasingly used when a detailed study of amyloid fibril formation in solution is required (33).

1. The protein samples are previously centrifuged and resuspended in double-deionized water after removal of the supernatant (see Note 18).
2. The images are taken in liquid media using a liquid cell without the O-ring seal.
3. About 50 μL of protein solution (at required concentration) is deposited on cleaved highly oriented pyrolytic graphite (HOPG) and allowed to adsorb for about 20 min before the measurements are started.
4. Amyloid fibrils images are obtained with a multimode atomic force microscope equipped with a 12 μm scanner.
5. Veeco NP-S probes can be used to scan the samples in tapping mode at a scan rate of 0.5 or 1 Hz.

4. Notes

1. At acidic pH (<pH 3) CR cannot be used for amyloid fibril detection, since no spectral change is observed in these conditions.
2. When a high purity grade is required for high sensitive measurements, Th-T can be recrystallized in demineralized water.

3. The concentration of stock solutions can be increased depending on the requirements (i.e., for IDC assays).
4. For an optimal spectroscopic determination, a CR concentration from 5 to 20 μM has to be, when this is possible.
5. Both molar absorptivities can be easily determined by calculating the linear dependence of the absorbance on the concentration of CR in the absence and in presence of an excess of the amyloid to be analyzed (slope of the respective graphical representations).
6. It is advisable to use a low CR concentration (e.g., <5 μM) in a such a way that the precipitated amount of the dye would represent at least 5% of the total CR concentration.
7. Th-T can be used in a large range of solution pHs (>pH 2).
8. Th-T and protein concentrations can be changed for each amyloid type and experimental conditions, without affecting the experiment outcome.
9. For fast determinations, like screenings or kinetic measurements the scan from 470 to 570 nm can be obviated, without relevant accuracy loss in the final fitting.
10. For a rigid system the maximum anisotropy value is 0.4, whereas for a freely rotating small molecule the anisotropy values are considerably smaller.
11. Th-T and protein concentrations can be adapted to each amyloid type and experimental conditions.
12. The slides should be imaged within the next few days if not immediately.
13. The obtained ratios provide only a crude approximation to the composition that, in certain occasions, cannot reflect the real aggregate structure.
14. For aggregation kinetics a fixed wavelength, i.e., 217 nm, that monitors increase in β-sheet content is sufficient.
15. Buffers with high ionic strength or chiral molecules should to be avoided because they dramatically increase the noise or background in the spectra.
16. Note that neutron scattering is used, often in combination with X-ray diffraction techniques, to study the structure of amyloid fibers. X-rays are scattered by the electrons surrounding atom nuclei, whereas neutrons are scattered by the nuclei themselves. Small-angle neutron scattering (SANS) and Small-angle X-ray scattering (SAXS) provide information about size, shape, and extent of aggregation of the species under consideration and measurement of mass per unit length.

17. Different equations that take into account the pre- and postexponential slopes can be used.
18. This process might repeated for three times to eliminate specific organic constituents such as dimethyl sulfoxide, which adsorb on HOPG, if these are present in the incubation buffer.

Acknowledgment

This work was supported by grants BFU2010-14901 from Ministerio de Ciencia e Innovación (Spain), 2009-SGR-760 and 2009-CTP-00004 from AGAUR (Generalitat de Catalunya). SV has been granted an ICREA Academia award (ICREA).

References

1. de Groot NS, Sabate R, Ventura S (2009) Amyloids in bacterial inclusion bodies. Trends Biochem Sci 34(8):408–416
2. Jahn TR, Radford SE (2008) Folding versus aggregation: polypeptide conformations on competing pathways. Arch Biochem Biophys 469(1):100–117
3. Dasari M, Espargaro A, Sabate R et al (2011) Bacterial inclusion bodies of Alzheimer's disease beta-amyloid peptides can be employed to study native-like aggregation intermediate states. Chembiochem 12(3):407–423
4. Hubbell WL, Cafiso DS, Altenbach C (2000) Identifying conformational changes with site-directed spin labeling. Nat Struct Biol 7(9):735–739. doi:10.1038/78956
5. Pelczer I, Carter BG (1997) Data processing in multidimensional NMR. Methods Mol Biol 60:71–155
6. Sawaya MR, Sambashivan S, Nelson R et al (2007) Atomic structures of amyloid cross-beta spines reveal varied steric zippers. Nature 447(7143):453–457
7. Tycko R (2006) Molecular structure of amyloid fibrils: insights from solid-state NMR. Q Rev Biophys 39(1):1–55
8. Tycko R (2011) Solid-state NMR studies of amyloid fibril structure. Annu Rev Phys Chem 62:279–299
9. Wasmer C, Lange A, Van Melckebeke H et al (2008) Amyloid fibrils of the HET-s(218–289) prion form a beta solenoid with a triangular hydrophobic core. Science 319(5869):1523–1526
10. Groenning M, Olsen L, van de Weert M et al (2007) Study on the binding of Thioflavin T to beta-sheet-rich and non-beta-sheet cavities. J Struct Biol 158(3):358–369
11. Zhavoronkov N, Gritsai Y, Bargheer M et al (2005) Microfocus Cu K(alpha) source for femtosecond X-ray science. Opt Lett 30(13):1737–1739
12. Makin O, Sikorski P, Serpell L (2007) CLEARER: a new tool for the analysis of X-ray fibre diffraction patterns and diffraction simulation from atomic structural models. J Appl Crystallogr 40:966–972
13. Steensma DP (2001) "Congo" red: out of Africa? Arch Pathol Lab Med 125(2):250–252
14. Klunk WE, Pettegrew JW, Abraham DJ (1989) Two simple methods for quantifying low-affinity dye-substrate binding. J Histochem Cytochem 37(8):1293–1297
15. Klunk WE, Pettegrew JW, Abraham DJ (1989) Quantitative evaluation of Congo red binding to amyloid-like proteins with a beta-pleated sheet conformation. J Histochem Cytochem 37(8):1273–1281
16. Kodali R, Wetzel R (2007) Polymorphism in the intermediates and products of amyloid assembly. Curr Opin Struct Biol 17(1):48–57
17. Klunk WE, Jacob RF, Mason RP (1999) Quantifying amyloid by Congo red spectral shift assay. Methods Enzymol 309: 285–305
18. Klunk WE, Jacob RF, Mason RP (1999) Quantifying amyloid beta-peptide (Abeta) aggregation using the Congo red-Abeta (CR-abeta) spectrophotometric assay. Anal Biochem 266(1):66–76

19. Inouye H, Nguyen JT, Fraser PE et al (2000) Histidine residues underlie Congo red binding to A beta analogs. Amyloid 7(3):179–188
20. Sabate R, Estelrich J (2003) Pinacyanol as effective probe of fibrillar beta-amyloid peptide: comparative study with Congo red. Biopolymers 72(6):455–463
21. Schutz AK, Soragni A, Hornemann S et al (2011) The amyloid-Congo red interface at atomic resolution. Angew Chem Int Ed Engl. doi:10.1002/anie.201008276
22. Sabate R, Espargaro A, Saupe SJ et al (2009) Characterization of the amyloid bacterial inclusion bodies of the HET-s fungal prion. Microb Cell Fact 8:56
23. Puchtler H, Sweat F (1965) Congo red as a stain for fluorescence microscopy of amyloid. J Histochem Cytochem 13(8):693–694
24. Giorgadze TA, Shiina N, Baloch ZW et al (2004) Improved detection of amyloid in fat pad aspiration: an evaluation of Congo red stain by fluorescent microscopy. Diagn Cytopathol 31(5):300–306
25. LeVine H 3rd (1999) Quantification of beta-sheet amyloid fibril structures with thioflavin T. Methods Enzymol 309:274–284
26. Naiki H, Gejyo F (1999) Kinetic analysis of amyloid fibril formation. Methods Enzymol 309:305–318
27. Sabate R, Lascu I, Saupe SJ (2008) On the binding of Thioflavin-T to HET-s amyloid fibrils assembled at pH 2. J Struct Biol 162(3):387–396
28. Dzwolak W, Pecul M (2005) Chiral bias of amyloid fibrils revealed by the twisted conformation of Thioflavin T: an induced circular dichroism/DFT study. FEBS Lett 579(29):6601–6603
29. Sabate R, Saupe SJ (2007) Thioflavin T fluorescence anisotropy: an alternative technique for the study of amyloid aggregation. Biochem Biophys Res Commun 360(1):135–138
30. Chiti F, Dobson CM (2006) Protein misfolding, functional amyloid, and human disease. Annu Rev Biochem 75:333–366
31. Bouchard M, Zurdo J, Nettleton EJ et al (2000) Formation of insulin amyloid fibrils followed by FTIR simultaneously with CD and electron microscopy. Protein Sci 9(10):1960–1967
32. de Groot NS, Parella T, Aviles FX et al (2007) Ile-phe dipeptide self-assembly: clues to amyloid formation. Biophys J 92(5):1732–1741
33. Sabate R, Espargaro A, de Groot NS et al (2010) The role of protein sequence and amino acid composition in amyloid formation: scrambling and backward reading of IAPP amyloid fibrils. J Mol Biol 404(2):337–352
34. Madine J, Jack E, Stockley PG et al (2008) Structural insights into the polymorphism of amyloid-like fibrils formed by region 20–29 of amylin revealed by solid-state NMR and X-ray fiber diffraction. J Am Chem Soc 130(45): 14990–15001
35. Morris K, Serpell L (2010) From natural to designer self-assembling biopolymers, the structural characterisation of fibrous proteins & peptides using fibre diffraction. Chem Soc Rev 39(9):3445–3453
36. Hubbard SJ (1998) The structural aspects of limited proteolysis of native proteins. Biochim Biophys Acta 1382(2):191–206
37. Collins SR, Douglass A, Vale RD et al (2004) Mechanism of prion propagation: amyloid growth occurs by monomer addition. PLoS Biol 2(10):e321
38. Jarrett JT, Lansbury PT Jr (1993) Seeding "one-dimensional crystallization" of amyloid: a pathogenic mechanism in Alzheimer's disease and scrapie? Cell 73(6):1055–1058
39. Sabate R, Gallardo M, Estelrich J (2003) An autocatalytic reaction as a model for the kinetics of the aggregation of beta-amyloid. Biopolymers 71(2):190–195

Chapter 16

Analyzing Oligomerization of Individual Transmembrane Helices and of Entire Membrane Proteins in *E. coli*: A Hitchhiker's Guide to GALLEX

Florian Cymer, Charles R. Sanders, and Dirk Schneider

Abstract

Genetic systems, which allow monitoring interactions of individual transmembrane α-helices within the cytoplasmic membrane of the bacterium *Escherichia coli*, are now widely used to probe the structural biology and energetics of helix-helix interactions and the consequences of mutations. In contrast to other systems, the GALLEX system allows studying homo- as well as heterooligomerization of individual transmembrane α-helices, and even enables estimation of the energetics of helix–helix interactions within a biological membrane. Given that many polytopic membrane proteins form oligomers within membranes, the GALLEX system represents a unique and powerful approach to monitor formation and stability of oligomeric complexes of polytopic membrane proteins in vivo.

Key words: Membrane protein, Oligomerization, In vivo, GALLEX, Helix–helix interaction, TOXCAT

1. Introduction

In a typical genome, one-fourth to one-third of all open reading frames are predicted to encode α-helical transmembrane (TM) proteins (1). While it was originally assumed that α-helical membrane proteins are simple bundles of ideal TM helices that span a membrane perpendicularly, recent high-resolution structures have indicated that the structure of α-helical membrane proteins is far more complex (2). Nevertheless, for some of these proteins the seminal "two-stage model of membrane protein folding" provides an adequate description of folding, especially in case of single-span TM proteins that form oligomers. According to this model individual

Alexander E. Kister (ed.), *Protein Supersecondary Structures*, Methods in Molecular Biology, vol. 932,
DOI 10.1007/978-1-62703-065-6_16,

TM helices integrate independently into the membrane followed by formation of interhelical contacts to form intramolecular or intermolecular (for oligomers) helical TM helix bundles (3). The "two-stage" model has recently been extended by a subsequent third stage, which includes integration of cofactors, rearrangements of individual protein parts, or oligomerization of multispan membrane proteins (4, 5).

Several monomeric membrane proteins appear to form functional complexes that are the result of homo- and/or heterooligomerization of individual polytopic TM proteins, and such oligomerization events might often be highly dynamic since formation and dissociation of such complexes might be regulated by the cell and/or signaling molecules (6). Oligomerization of individual polytopic membrane proteins can control the protein function; however, these events remain far from being completely understood to this day (7–9).

While proteins residing within biological membranes are critically involved in many cellular processes, it is surprising how little is known regarding the in vivo (oligomeric) structure and function of such proteins. This is mainly due to a lack of proper techniques to study membrane proteins, which are difficult to handle and to study for various reasons (10). To study membrane protein folding and/or unfolding in vitro, membrane proteins are typically extracted from membranes by detergents and the detergent-solubilized membrane proteins are further analyzed. However, detergent molecules often mimic a biological membrane only poorly, and thus, it is desirable to study interactions of individual TM α-helices as well as oligomerization of entire membrane proteins within biological membranes. Therefore, techniques have been developed allowing to measure interactions of individual TM helices within the inner membrane of the bacterium *Escherichia coli* (11), and these systems measure stage two of the mentioned two-stage model. The earlier developed systems, the Tox^R and TOXCAT system, are based on the Tox^R transcription activator of the bacterium *Vibrio cholera* (12, 13). The Tox^R DNA-binding domain is genetically fused to a TM helix of interest, and homodimerization of a TM helix results in dimerization of the DNA-binding domains, leading to reporter gene activation. Consequently, a measured reporter gene activity can be correlated to the homodimerization propensity of the TM helix of interest. The GALLEX system was developed later to directly measure not only homotypic, but also heterotypic interactions of individual TM helices (14). In the GALLEX-system, a TM helix of interest is genetically fused to a wild-type or a mutated DNA-binding domain of the *E. coli* LexA protein. Only a dimeric LexA DNA-binding domain can bind to a promoter/operator region, resulting in repression of a reporter gene (*lacZ*) activity. Besides measuring the strength of a given homo- or

heterotypic TM helix–helix interaction (15–18), the GALLEX-system also allows estimating energetics of TM helix-helix interactions within a biological membrane (19, 20). Furthermore, the system has recently been used to study the oligomerization propensity of a full length *E. coli* multispan membrane protein in vivo (7).

The concept of the GALLEX system, the necessary compounds and a detailed protocol for measuring homotypic and heterotypic interactions of TM domains in the *E. coli* membrane, are described herein.

2. Materials

2.1. Plasmid Construction and Transformation of E. coli

The *E. coli* strains and the plasmids have to be chosen whether homo- or heterooligomerization is measured, as outlined in Fig. 1a. For measuring homotypic interactions, the *E. coli* reporter strain SU101 is used together with the pBLM 100 plasmid, whereas SU202 cells are used together with both pALM148 and pBLM100-derived plasmids to measure heterotypic interactions. For measuring heterotypic interactions, the wild-type DNA binding domain of LexA (encoded on pBLM100) has to be expressed together with a mutated LexA DNA-binding domain, which is encoded on the pALM148 plasmid. The mentioned plasmids contain the restriction sites for cloning the TM-segment as shown in Fig. 1b.

Material	Relevant features	Reference
E. coli strains		
SU101	*lexA71::Tn5 (Def)sulA211* Δ*(lacIPOZYA)169/ F©lacIqlacZDM15::Tn9/sulA op+/op+::lacZ*	(21)
SU202	*lexA71::Tn5 (Def)sulA211* Δ*(lacIPOZYA)169/ F©lacIqlacZDM15::Tn9/sulA op408/op+::lacZ*	(21)
NT326	F⁻ *araD139* Δ*lacU1169 rpsL thi* Δ*malE444 recA1*	(22)
Plasmids		
pBLM100 (Km^R)	Derivative of pBR322. Contains P_{tac} promoter and $lacI^q$ gene from pMal-p2x (New England Biolabs), *bla* (ampicillin resistance gene), pMB1 origin of replication, *rob*, (*neo* (kanamycin resistance gene))	(14)
pALM148 (Km^R)	Derivative of pACYC148. Contains P_{tac} promoter and $lacI^q$ gene from pMal-p2x (New England Biolabs), *tet* (tetracycline resistance gene), p15A origin of replication, (*neo* (kanamycin resistance gene))	(14)

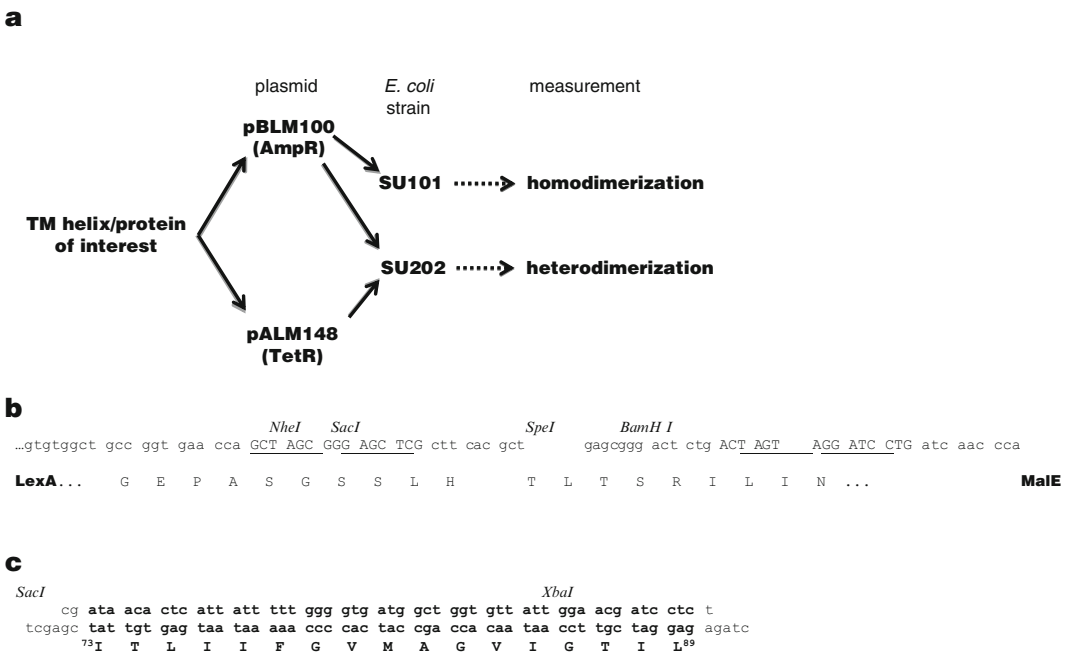


Fig. 1. (**a**) Overview of the plasmids and the used *E. coli* reporter strains. A gene coding for the TM helix of interest is cloned into the pBLM100 plasmid. The generated plasmid is then transformed into *E. coli SU101* for monitoring homodimerization. For monitoring heterodimerization of two different TM helices, one helix is cloned into the pALM and the other into the pBLM plasmid. After cotransformation of both generated plasmids into *E. coli* SU202, heterodimerization can be monitored. The pBLM plasmid contains an ampicillin and the pALM a tetracycline resistance cassette. (**b**) Cloning site of both the pALM and pBLM plasmids. In the plasmids used for cloning, a kanamycin resistance cassette is ligated in between the *SacI* and *SpeI* restriction sites (see Note 4). (**c**) The sequence of the recombinant DNA cassette coding for the GpA TM helix. The nucleotides in bold-faced encode the amino acids of the TM helix, while the nucleotides in light grey are added for cloning to the GpA sequence. The encoded amino acids are given in capital letters.

1. *E. coli* strains and plasmids used in the GALLEX assay.
2. Annealing buffer (10×): 200 mM Tris pH 7.5, 100 mM $MgCl_2$, 500 mM NaCl.
3. LB medium (23): weigh 10 g NaCl, 10 g tryptone, 5 g yeast extract and add distilled water to volume of 1 L. Autoclave the solution. For preparation of LB agar plates, additionally add 15 g agar per 1 L LB medium prior to autoclaving. After autoclaving, cool down the medium in a water bath to about 50 °C. When LB agar is at approximately 50 °C, add antibiotics at required concentrations and pour agar plates.
4. Antibiotics: ampicillin (100 mg/mL in water), chloramphenicol (30 mg/mL in ethanol), kanamycin (30 mg/mL in water), tetracycline (10 mg/mL in 1:1 vol/vol distilled water–ethanol). Sterilize the prepared antibiotic stock solutions by filtration and store aliquots at −20 °C. Wrap aliquots of the tetracycline solution in aluminum foil.
5. $CaCl_2$ solution: 0.1 M $CaCl_2$ in distilled water. Autoclave and store at 4 °C.

6. Isopropyl-β-D-thiogalactopyranoside (IPTG): prepare 1 M stock solution in distilled water. Filter-sterilize solution prior to usage. Always prepare freshly!

2.2. β-Galactosidase Activity Assay

1. 5× Z-buffer: 300 mM $Na_2HPO_4 \cdot 7\ H_2O$, 200 mM $NaH_2PO_4 \cdot H_2O$, 50 mM KCl, 5 mM $MgSO_4 \cdot 7\ H_2O$. Sterilize by filtering.
2. 1× Z-buffer: dilute 5× Z-buffer five times. Add 27 μl β-mercaptoethanol (2-ME) per 10 mL 1× Z buffer (final concentration = 50 mM). Always prepare 1× Z-buffer freshly!
3. 0.1 % sodium dodecylsulfate (SDS) solution in water.
4. *o*-nitrophenyl-β-D-galactopyranoside (ONPG) solution: 4 mg/L in 1× Z buffer (see Note 1).
5. 1 M Na_2CO_3 in water.

2.3. Maltose Complementation Assay

1. Dissolve 10 g glucose or 10 g maltose, respectively, in 100 mL distilled water. Filter-sterilize the respective stock solutions.
2. M9-minimal medium plates: 10 g $Na_2HPO_4 \cdot 7\ H_2O$, 5 g KH_2PO_4, 5 g NH_4Cl, 2.5 g NaCl. Add distilled water to 1 L and add 15 g agar. After autoclaving, cool down the medium in a water bath to about 50 °C. Add 10 μM IPTG from the 1 M stock solution (see above). Split medium solution. To one part add 1/20 of the glucose stock solution and to the other part 1/20 of the maltose stock solution. Add necessary antibiotics to select for pALM containing cells (10 μg/mL tetracycline) or pBLM containing cells (100 μg/mL ampicillin). Pour agar plates.

2.4. Test for Protein Insertion into the E. coli Inner Membrane

3. Lysis buffer: mix 2.5 μL of a 1 M $MgCl_2$ solution (in distilled water), 2.5 μL DNase of a 10 mg/mL stock solution prepared in distilled water, 5 μL of a 10 mg/mL lysozyme stock solution (in distilled water), and 90 μL of distilled water.
4. Sodium hydroxide solution: prepare a 0.1 M NaOH solution in distilled water.
5. Trichloroacetic acid (TCA): prepare a 50 % TCA solution in distilled water. Store solution on ice.

3. Methods

3.1. Outline

Individual steps of the GALLEX-assay are briefly outlined in Fig. 1a. A gene or gene fragment encoding a TM protein or TM helix of interest is ligated to the plasmid pBLM100 and/or pALM148. The pBLM100-derived GALLEX plasmid encodes a chimeric protein consisting of the *E. coli* LexA DNA binding domain, followed by the cloned gene (or gene fragment). In addition, single TM helices are typically also fused to the *E. coli*

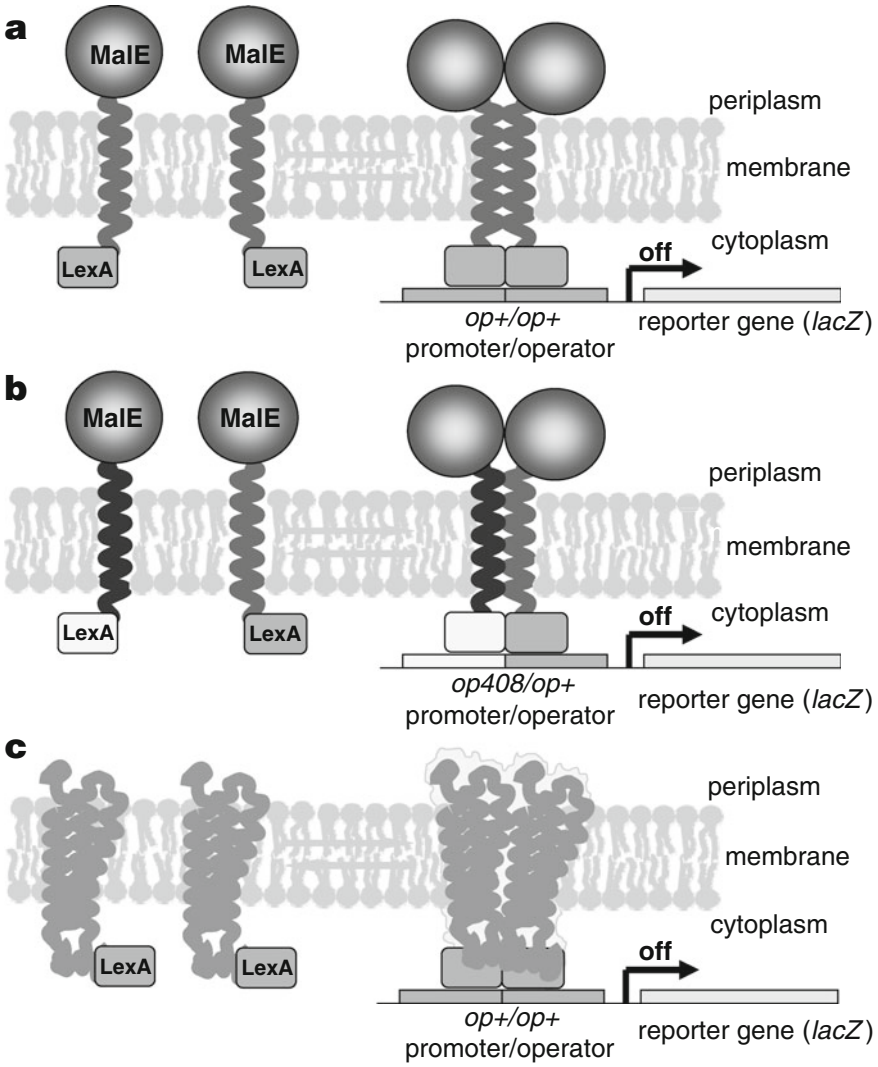


Fig. 2. Outline of the GALLEX assay for measuring TM helix–helix interaction in a biological membrane. The TM domain anchors the chimera in the cytoplasmic membrane of *E. coli* with the C-terminal MalE domain located in the periplasm and the LexA DNA-binding domain in the cytoplasm. Interaction of the TM domains leads to formation of LexA homo- (**a**) or heterodimers (**b**), which can bind to an operator region. The binding of the LexA dimer results in repression of the reporter gene (*lacZ*) activity. Fusion of the LexA DNA-binding domain to a polytopic *E. coli* inner membrane protein allows in vivo oligomerization of larger proteins to be monitored (**c**).

MalE protein at their respective C-terminus (see Note 2). As the chimeric protein is expressed from the pBLM-derived plasmid under the control of the P_{tac} promoter, expression of the chimeric protein is induced by adding IPTG. The hydrophobicity of the TM helix functions as a membrane insertion signal, placing the LexA domain in the *E. coli* cytoplasm and the MalE domain in the periplasm of *E. coli* (Fig. 2a, b).

When a TM domain homooligomerizes, the fused LexA DNA-binding domains come into close contact to form a dimeric LexA DNA-binding domain that is able to bind to a genomically located

op+/op + promoter/operator region, which controls expression of the *lacZ* reporter gene, encoding the β-galactosides. β–galactosidase activity can be measured and correlated to the oligomerization propensity (21). As LexA is a repressor, a strong interaction propensity corresponds to a low, and a weak interaction to a high β-galactosidase activity.

For measuring heterooligomerization, the respective domains are cloned into the plasmids pBLM100 and pALM148. In contrast to pBLM, the pALM plasmid codes for a mutated LexA DNA-binding domain that binds to a different promoter/operator region. Interaction of a TM protein fused to the wild-type LexA DNA-binding domain with another protein fused to the mutated LexA DNA-binding domain, results in formation of a LexA heterodimer. In *E. coli* SU202, the *lacZ* reporter gene is placed under the control of an op408/op + hybrid operator (21). This asymmetric promoter is composed of half of the wild-type promoter plus an altered half (Fig. 2b) and therefore only allows binding of a LexA heterodimer composed of one wild-type and a mutated LexA DNA-binding domain (LexA408) (21).

3.2. Homooligomerization of Single TM α-Helices: Plasmid Construction and Transformation

To measure TM helix homooligomerization, the pBLM-derived plasmids are used together with the *E. coli* strain SU101 (Figs. 1a and 2a).

1. Digest the pBLM100 plasmid with two suitable restriction enzymes, as indicated in Fig. 1b. Purify the digested plasmid by gel extraction (see Notes 3 and 4).
2. Based on the known protein sequence of a TM protein, determine the DNA sequence, which encodes the TM helix, and add the respective restriction site to the selected DNA-sequence. An example of the TM helix of human glycophorin A (GpA) is given in Fig. 1c. The designed sense and antisense oligonucleotides are subsequently custom-synthesized by choice. Add 10 pmol of each oligonucleotide to a 50 μl reaction containing the 1× annealing buffer. Incubate at 95 °C for 15 min and allow the reaction to slowly cool down to 40 °C to generate the double-stranded DNA cassette. Place on ice afterwards (see Note 5).
3. Ligate the annealed oligonucleotides to the respective restriction-digested GALLEX plasmid. Add 2 μL of the in vitro generated DNA cassette to 100 ng of restriction-digested plasmid DNA.
4. Confirm correct ligation by DNA sequencing (see Note 6).
5. Prepare competent SU101 cells. Pick a single, fresh colony of SU101 from an LB plate and grow cells in 3 mL LB + 30 mg/mL chloramphenicol overnight (see Note 7). The next morning inoculate 100 mL LB medium + 30 mg/mL kanamycin with 0.5 mL of the overnight culture. When an OD_{600} of about 0.6

is reached, place cells on ice for 10 min. Pellet cells by centrifugation (3,000×*g*, 10 min, 4 °C), discard supernatant and resuspend the cell pellet in 10 mL ice-cold 0.1 M $CaCl_2$. Incubate on ice for 10 min. Centrifuge as above, discard supernatant and resuspend cells in 2 mL ice-cold $CaCl_2$ solution.

6. Transform competent SU101 cells with the respective GALLEX plasmid by heat shock. First, add plasmid to 200 µL of competent cells and incubate on ice for 30 min. Next, heat shock the cells for 1 min at 42 °C and then incubate on ice for 10 min subsequently. Add 800 mL LB medium and incubate at 37 °C in a rotary incubator for 1 h. Finally, streak 200 µL of the cell culture out onto LB agar plates containing 100 µg/mL ampicillin (see Note 6), and incubate plates overnight at 37 °C.
7. The next day, pick 3–5 single colonies from each transformation and inoculate 2 mL LB media containing 100 µg/mL ampicillin, 5 µg/mL chloramphenicol, 5 µg/mL kanamycin and 0.01 mM IPTG (see Note 8). Grow cells overnight at 37 °C on a shaker at 200 rpm.
8. The next day, dilute cells 1:40 into 10 mL of fresh LB medium containing antibiotics and IPTG. Grow cells in a 50 mL Erlenmeyer flask at 37 °C on a shaker at 200 rpm (see Note 9).
9. Harvest the cells at an $OD_{600} = 0.6$ (after approximately 1.5–2 h).
10. Perform β-galactosidase activity assay.

3.3. Homooligomerization of Single TM α-Helices: β-Galactosidase Activity Assay (24)

β-galactosidase activity is measured to determine repression of the *lacZ* gene activity and thus to correlate the interaction of a TM helix to an optical readout.

1. Take OD_{600} of 1 mL of cells. Record reading.
2. Aliquot 100 µl of cells into a 2 mL plastic tube (use 100 µL media for the blank) and add 900 µl of 1× Z buffer with 2-ME.
3. Lyse cells with 10 µL of 0.1 % SDS and two drops of chloroform. Vortex for 10 s.
4. Equilibrate to room temperature and start reaction by adding 200 µL ONPG (4 mg/mL in Z buffer) to each tube. Invert the tubes twice. Add the ONPG solution from sample to sample in 10 s increments to the next sample (see Note 10). Incubate the reaction at room temperature. Record the time it took for the yellow color to appear (see Note 11).
5. Stop the reaction by adding 0.5 mL of the 1 M Na_2CO_3 solution. Invert the tube twice. Add the Na_2CO_3 solution to the next samples in 10 s increments to ensure that all reactions have been run for exactly the same time (see Note 12).
6. Spin cells down in a table-top centrifuge for 1 min at maximum speed.

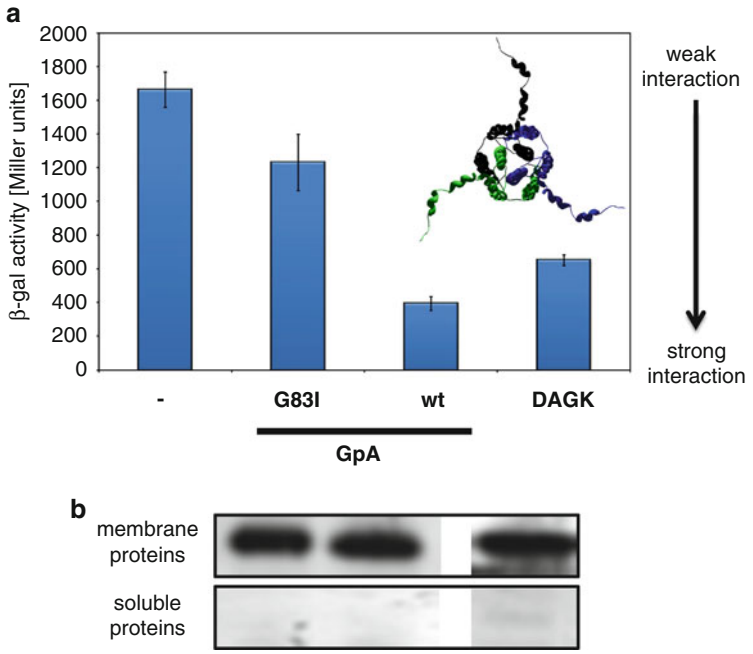


Fig. 3. (**a**) Homodimerization of wild-type and the G831-mutated GpA TM domain. As a control, the plasmid pBR322, which is the origin of the pBLM100 plasmid, was also transformed into *E. coli* SU101 (–). Introduction of the G83I mutation leads to a dramatically increased β-galactosidase activity and thus to a loss of the interaction capacity. Also the full length *E. coli* DAGK exhibits strong oligomerization within the *E. coli* inner membrane. Bars represent the β-galactosidase activities of three independent measurements. The inlet shows a top view on the structure of the homotrimeric DAGK (pdb code: 2KDC) looking down on the membrane from the cytosolic perspective. DAGK has three TM helices per subunit. (**b**) Western blot analyses of *E. coli* membranes expressing the respective LexA fusion proteins (left to right: GpA G83I, GpA wt, DAGK). The chimeric proteins are expressed at near identical levels into the *E. coli* inner membrane and are localized exclusively in the membrane fraction.

7. Transfer 1 mL of the supernatant to a cuvette and record absorbance at 550 nm and at 420 nm.
8. Calculate β-galactosidase activity (in Miller units): $(1{,}000 \times OD_{420} - (1.75 \times OD_{550}))/(\text{incubation time (min)} \times \text{vol. (0.1 mL)} \times OD_{600})$ (see Note 13).
9. To properly assess a monitored β-galactosidase activity, it has been found to be useful to also routinely measure interactions of the strongly dimerizing GpA TM helix (low β-galactosidase activity, positive control) as well as of the weakly dimerizing GpA G83I mutant (high β-galactosidase activity, negative control). Self-association of these single span TM proteins is very well characterized (see Notes 13 and 14). An example is shown in Fig. 3.

3.4. Test for Protein Insertion into the E. coli Cell Membrane

Different chimeric proteins can have different expression levels, even when identical concentrations of the inducer IPTG are used. As a determined interaction propensity depends on the actual concen-

trations of the chimeric protein in the *E. coli* membrane (7, 19), interaction propensities of different TM proteins can only be compared properly whenever the relative protein concentration in the membrane fraction is also almost identical. Therefore, the relative content of the expressed protein incorporated into the membrane has to be analyzed.

1. Take an equivalent of 1 mL of an $OD_{600} = 0.6$ of the culture prepared to measure the β-galactosidase activity (compare above) and spin cells down in a table-top centrifuge (see Note 15). If OD_{600} is not exactly 0.6, adjust the volume used so as to normalize the number of cells from sample to sample.
2. Resuspend cells in 100 μL lysis buffer, incubate at room temperature for 1 h and place on ice afterwards.
3. Add 150 μL of ice-cold water. Safe 125 μL of the suspension for further analyses (→ whole cell extract).
4. Add 1 volume of ice-cold 0.1 M NaOH to the remaining 125 μL, and vortex at the highest setting for 1 min.
5. Centrifuge for 15 min at 4 °C at maximum speed in a table-top centrifuge
6. Remove the supernatant, which represents the soluble and membrane-associated protein fraction. The pellet contains the integral membrane proteins, which were not extracted by the NaOH treatment.
7. Add 0.5 volumes of ice-cold 50 % TCA to the whole cell extract and to the soluble protein fraction and incubate on ice for 30 min.
8. Centrifuge for 15 min at 4 °C and at maximum speed in a table-top centrifuge and discard the supernatant.
9. Resuspend the precipitated proteins in 1 mL of ice-cold acetone and incubate on ice for 5 min. Centrifuge as before and discard the supernatant. Dry the precipitate in a fume hood.
10. Resuspend the entire cell extract, the soluble protein fraction and the membrane fraction in 50 μL 1× SDS sample buffer and run an SDS-PAGE.
11. Perform a Western blot analysis using a commercially available anti-LexA antibody to compare expression levels of the analyzed proteins (for an example see Fig. 3).

3.5. Maltose Complementation Assay

The maltose complementation assay is used to confirm proper orientation of expressed chimeric proteins in the inner *E. coli* membrane. It is of particular importance to exclude an incorrect TM topology (i.e., LexA domain in the *E. coli* periplasm) if a TM helix does not interact, and thus, *lacZ* activity is not repressed. Since the *E. coli* strain NT326 lacks endogenous MalE protein, the cells cannot grow on media where maltose is the only carbon source. Only if the

expressed chimeric protein is oriented in the proper direction and the MalE domain of the chimeric protein is located in the periplasm, cells can grow on this minimal medium. As a positive control, growth is confirmed on glucose plates in order to exclude a lack of growth due to toxic effects of the chimeric protein (see Note 16).

1. Transform NT326 cells with the GALLEX plasmids used for the β-galactosidase assay. Use the protocols to prepare competent cells and for transformation as described above. Streak transformed cells out on LB agar plates containing 100 μg/mL ampicillin. Incubate the plates at 37 °C overnight (see Note 17).
2. Pick a single colony from the LB plates and streak out on M9 agar plates containing maltose or glucose, respectively, as well as 100 μg/mL K ampicillin. Incubate plates at 37 °C until cells become clearly visible (after about 3 days).

3.6. Heterooligomerization of Single TM α-Helices

To monitor heterooligomerization, two different chimeric proteins have to be coexpressed within the *E. coli* SU202 reporter strain (Figs. 1b and 2b). A TM helix fused to the wild-type LexA DNA-binding domain (encoded on pBLM100) has to interact with a TM helix fused to the mutated LexA DNA-binding domain (encoded on pALM148) (see Note 18).

1. Plasmid construction is identical as described in Subheading 3.1. In addition to pBLM, the pALM plasmid also needs to be restriction-digested and used in the ligation reaction. Restriction sites are identical to pBLM (Fig. 1b). pALM-carrying cells are selected from plates containing 10 μg/mL tetracycline. As the protocol for transformation of SU202 differs slightly, the respective steps are described in the following (see Note 19).
2. Prepare competent *E. coli* SU202 cells as described in Subheading 3.2.
3. Transform *E. coli* SU202 with the pALM-derived plasmid and streak transformed cells out on LB agar plates containing 10 μg/mL tetracycline, 5 μg/mL kanamycin, and 5 μg/mL chloramphenicol. Incubate the plates at 37 °C overnight (see Note 20).
4. Pick a single colony the next morning and inoculate 3 mL LB-media containing 10 μg/mL tetracycline. Grow cells at 37 °C for about 4 h until an OD_{600} of about 0.6 is reached. Use the entire culture and prepare competent cells following the protocol above (scale down to the smaller volume!).
5. Transform competent cells with the pBLM-derived plasmid as outlined above. Streak transformed cells out on an LB agar plate containing 100 μg/mL ampicillin and 10 μg/mL tetracycline. Incubate plate overnight at 37 °C.

6. Pick 3–5 single colonies of each transformant and inoculate 2 mL LB-medium with 100 μg/mL ampicillin, 10 μg/mL tetracycline, 5 μg/mL chloramphenicol, 5 μg/mL kanamycin, and 0.01 mM IPTG (see Note 8). Incubate overnight at 37 °C at 200 rpm of shaking.
7. The next day, dilute the overnight culture 1:40 into 10 mL of fresh LB medium containing the antibiotics used previously as well as 0.01 mM of IPTG (see Note 8).
8. Harvest cells at an $OD_{600} = 0.6$.
9. Perform β-galactosidase assay as described in Subheading 3.3

3.7. Homooligomerization of Full Length Multispan Membrane Proteins

Usually, genetic systems, such as the GALLEX-system, are used to monitor interactions of individual TM helices. However, the GALLEX-system can also be used to follow oligomerization of larger, polytopic membrane proteins within the *E. coli* inner membrane. As an example we have analyzed homotrimerization of the *E. coli* diacylglycerol kinase (DAGK) within the *E. coli* inner membrane. The strategy and protocol is mostly identical to the one described for monitoring homooligomerization of single TM helices (Subheading 3.1–3.3) and differs only in some areas on some points. A schematic overview is shown in Fig. 2c.

1. Restriction-digest the pBLM plasmid with two suitable restriction enzymes (Fig. 1b) and purify the restriction-digested plasmid from an agarose gel.
2. Design oligonucleotides that allow amplification of a gene of interest and contain the respective restriction site of the GALLEX plasmids pBLM100 at the 5′ and 3′ end and include the stop-codon at the 3′ end to avoid genetic fusion to the MalE domain (see Note 21).
3. The *dagK* gene might be amplified by a PCR from *E. coli* genomic DNA or from an already cloned plasmid (25). Here, the *dagK* gene was amplified from a plasmid.
4. Restriction-digest the PCR product with the respective restriction enzymes and isolate the restriction-digested PCR fragment from an agarose gel.
5. Ligate the restriction-digested PCR product to the respective restriction-digested plasmid pBLM100.
6. Confirm correct ligation by sequencing.
7. Transform the plasmid into *E. coli* SU101 and perform the β-galactosidase assay as described in Subheadings 3.2 and 3.3.
8. Membrane integration of the chimeric protein might be tested as described in Subheading 3.4.

4. Notes

1. The ONPG solution should always be prepared freshly. ONPG is difficult to dissolve in buffer. Try to crush the ONPG chunks and filter the solution afterwards using a sterile filter and a syringe to remove nondissolved ONPG.
2. Fusion of the DNA-binding domain is absolutely necessary for the assay. Fusion of the MalE domain to the C-terminus of a TM helix has been shown to facilitate membrane integration of the chimeric protein (26). Furthermore, it is convenient to use the fused MalE domain for detection and for topology analyses. However, other proteins might also be fused C-terminally to a TM helix (27).
3. The combination *Sac*I/*Spe*I has successfully been used many times (14, 16–18, 28). *Nhe*I/*Bam*HI has also already been used successfully (5, 7, 29, 30).
4. A kanamycin resistance cassette has originally been introduced in between the restriction sites for cloning reasons (14). However, having this cassette introduced into the actual cloning site is advantageous, as this allows clearly separation of the correctly restriction-digested plasmid from other plasmids, which have, for example, been restriction-digested by only a single enzyme. Cutting the kanamycin resistance cassette out of the plasmid reduced the size of the plasmid by about 1,200 bp.
5. While a DNA fragment encoding a TM helix of interest can, in principal, also be amplified from genomic DNA by a PCR reaction, generating a DNA cassette by in vitro annealing of two synthesized DNA sequences, turned out to be far more practical. As the 5′ ends of the restriction-digested plasmids still contain phosphate groups, which are needed for the subsequent ligation reaction, the primers do not need to be phosphorylated.
6. The primer: *ggattcgtctgttgcaggaag* might be used for sequencing starting upstream of the *lexA* gene, and thus, using this primer results in sequencing of the DNA region encoding the LexA DNA-binding domain (about 260 bp), the cloned DNA fragment as well as the beginning of the MalE domain.
7. Finally, the cells are resistant to three or four antibiotics. Growing the cells on multiple antibiotics appears to stress the cells dramatically such that they grow very slowly. Therefore, either do not always use the antibiotics originally used to generate the *E. coli* reporter strain (kanamycin and chloramphenicol) or use only lower concentrations. At certain steps it might even be beneficial to use only one of these two antibiotics, with the other one being used in a later incubation step. By doing so, one can still ensure that the correct strain,

having the two chromosomally-located resistance genes, has been maintained.

8. The actually monitored β-galactosidase activity depends on the interaction propensity of the analyzed TM helices, which is a basis of the GALLEX-assay. However, the monitored interaction propensity also depends on the concentration of an expressed fusion protein in the *E. coli* inner membrane. The protein concentration can be varied by addition of different amounts of IPTG. Thus, the IPTG concentration might be adjusted from case to case to obtain optimal results in the β-galactosidase activity measurements.
9. When different measurements are compared, be sure to use identical conditions for growth, induction etc. Changing the growth temperature or the speed of the rotary incubator might already influence the measured interaction propensities. Growing the final cultures, which are used in the actual β-galactosidase assay, at a volume of 10 mL in a 50 mL Erlenmeyer flask has, in our hands, been found to minimize trial to trial variability. At a minimum, only compare measurements performed on the same day under exactly the same conditions.
10. It has turned out that it is a convenient to pipette the ONPG solution into the next sample with an increment of 10 s. This leaves time to pipette, to mix, and to move to the next sample.
11. The β-galactosidase assay should be run for at least 3 min. In the case that the reaction is observed to be completed sooner, use only 50 μL cellular extract for the measurement.
12. As all samples should be given the maximum available time for the ONPG reaction, place the negative control, the plasmid carrying the GpA G83I TM helix (or an empty expression plasmid), at the very beginning of the sample line. Monitor and keep an eye on the negative control. When the sample becomes noticeably yellow, stop all reactions by adding the $NaHCO_3$ solution.
13. The actually determined β-galactosidase activities, as measured in *Miller units* may vary in between measurements, whereas the relative interaction propensities, i.e., the differences between the measured β-galactosidase activities, are approximately constant. Thus, it might be beneficial to normalize a set of data, e.g., to an internal control, such as to the interaction propensity of the wild-type GpA TM helix.
14. The GpA wild-type and G832-mutated TM helices are used as positive and negative controls, respectively. In case a clear difference in the measured β-galactosidase activities is not observed between the respective *E. coli* cultures, discard the entire sample set.

15. Use freshly harvested cells for this assay. In cases where cells were frozen previously, some of the expressed protein is sometimes found in the soluble protein fraction. While the exact reason for this is not clear, it is possible that small membrane fragments get detached from the *E. coli* cytoplasmic membrane and are thus found in the soluble protein fraction afterwards.
16. The MalE assay is typically used to prove the correct TM topology of the chimeric protein. However, the cells should probably already grow on the minimal medium plates with maltose if only a fraction of the protein is located within the membrane having the MalE domain facing the periplasm. A better approach is to prepare spheroplasts and to digest all proteins and protein domains facing the periplasm by a proteinase K digestion (12). If one analyzes the relative content of the LexA domain before and after the protease treatment by Western blot analysis, it is safe to assume that all proteins have the correct topology. However, as such spheroplast assays are tricky and time-consuming, and as it is rather unlikely that a protein will have a dual topology, the MalE assay is usually performed.
17. As a control, also transform NT326 cells with the plasmid pMal-p2 and pMal-c2 (both from New England Biolabs), which express the MalE domain to the periplasm or cytoplasm of *E. coli*, respectively. Thus, pMal-p2 will confer growth on the maltose minimal medium (positive control) and pMal-c2 should not allow the cells to grow on maltose minimal medium. As a further negative control, also streak out cells, which have been transformed with the empty pBLM plasmid.
18. In case of homooligomerization, the observed results can clearly be interpreted since only one single equilibrium is monitored (Fig. 4). When heterooligomerization is analyzed, it is possible that multiple equilibria are involved, e.g., when the individual TM helices also form homo- as well as heterooligomers.

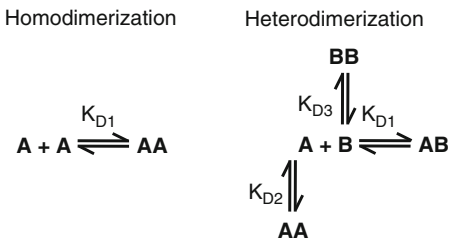


Fig. 4. Homodimerization of a TM domain might be described by a simple equilibrium involving a single K_D value, in which case a homodimerization measurement can be directly analyzed. If interaction of two different helices is analyzed, multiple equilibria might be involved. The individual helices might not only form heterooligomers but also homooligomers. Thus, multiple equilibria might be involved and the measured apparent heterodimerization propensity has to be interpreted with caution.

Therefore, one should always also measure homooligomerization propensities of an analyzed TM helix within the SU101 cells. Furthermore, while the plasmid number per cell of the pBLM and pALM plasmids are very similar (about 20 vs. 15 copies per cell), the small difference in the copy number might also affect the observed interaction propensities. Thus, always measure an interaction in both directions, i.e., clone helix 1 as well as helix 2 in both pALM and pBLM and measure both possible plasmid combinations in *E. coli* SU202.

19. The generated strains appear to be rather unstable. Therefore, always use freshly prepared cells and conduct all steps immediately after one another. For example, do not store transformed cells for a day at 4 °C, as this can influence the measurement.
20. In our experience, cells that are transformed with the pALM-derived plasmid, are more stable. Thus, transform the pALM-derived plasmids first and use the generated *E. coli* strain for transformation of the second, pBLM-derived, plasmid. Cotransformation of the two plasmids, e.g., by using electroporation, might also work but has not been tested yet.
21. Contrary to the GALLEX assay aiming to analyze interactions of single TM helices, a MalE domain might not be genetically fused to the C-terminus of polytopic TM proteins, as they can be assumed to efficiently target to the membrane and efficiently encode their own topology.
22. While the GALLEX assay allows measurement of homo- as well as heterooligomerization, the assay is still limited to monitoring parallel interactions between TM helices and antiparallel interactions cannot be analyzed. Furthermore, all attempts to use the GALLEX assay for screening of libraries have failed thus far.
23. While the assay has proven to be very useful, measurements can be hampered by several potential problems. If the expressed TM domains interact with other *E. coli* proteins, the monitored interaction propensity might be affected significantly. Furthermore, interactions of individual TM helices appear to be sensitive to the length of a TM helix and to the location of an interacting helical surface relative to the fusion domains. A TM helix might be ligated to the pBLM plasmid in different length etc., to test for optimal conditions to measure homooligomerization. However, it might be virtually impossible to screen all possible combinations of two different TM helices expressed from the pALM and pBLM-derived plasmids, respectively, to identify optimal assay conditions.

Acknowledgments

This work was supported by grants from the Deutsche Forschungsgemeinschaft (SCHN 690/2–4 and GRK 1478), the Stiftung Rheinland-Pfalz für Innovation, the centre of complex matter (COMATT), and the University of Mainz to DS. CRS is supported by US NIH grants RO1 GM47485 and G54 GM94608.

References

1. von Heijne G (2011) Introduction to theme "membrane protein folding and insertion". Annu Rev Biochem 80:157–160
2. von Heijne G (2011) Membrane proteins: from bench to bits. Biochem Soc Trans 39:747–750
3. Popot JL, Engelman DM (1990) Membrane protein folding and oligomerization: the two-stage model. Biochemistry 29:4031–4037
4. Engelman DM, Chen Y, Chin CN et al (2003) Membrane protein folding: beyond the two stage model. FEBS Lett 555:122–125
5. Prodöhl A, Volkmer T, Finger C et al (2005) Defining the structural basis for assembly of a transmembrane cytochrome. J Mol Biol 350:744–756
6. Cymer F, Anbazhagan V, Schneider D (2011) Transmembrane helix–helix interactions are modulated by the sequence context and by lipid bilayer properties. Biochim Biophys Acta 1818:963–73
7. Cymer F, Schneider D (2009) A single glutamate residue controls the oligomerization, function, and stability of the aquaglyceroporin GlpF. Biochemistry 49:279–286
8. Berkefeld H, Fakler B, Schulte U (2010) Ca^{2+}-activated K^+ channels: from protein complexes to function. Physiol Rev 90:1437–1459
9. Bornhovd C, Vogel F, Neupert W et al (2006) Mitochondrial membrane potential is dependent on the oligomeric state of F1F0-ATP synthase supracomplexes. J Biol Chem 281:13990–13998
10. Bowie JU (2005) Solving the membrane protein folding problem. Nature 438:581–589
11. Schneider D, Finger C, Prodöhl A et al (2007) From interactions of single transmembrane helices to folding of a-helical membrane proteins: analyzing transmembrane helix–helix interactions in bacteria. Curr Protein Pept Sci 8:45–61
12. Russ WP, Engelman DM (1999) TOXCAT: a measure of transmembrane helix association in a biological membrane. Proc Natl Acad Sci USA 96:863–868
13. Langosch D, Brosig B, Kolmar H et al (1996) Dimerisation of the glycophorin A transmembrane segment in membranes probed with the ToxR transcription activator. J Mol Biol 263:525–530
14. Schneider D, Engelman DM (2003) GALLEX, a measurement of heterologous association of transmembrane helices in a biological membrane. J Biol Chem 278:3105–3111
15. Finger C, Escher C, Schneider D (2009) The single transmembrane domains of human receptor tyrosine kinases encode self-interactions. Sci Signal 2:ra56
16. Schneider D, Engelman DM (2004) Involvement of transmembrane domain interactions in signal transduction by a/b integrins. J Biol Chem 279:9840–9846
17. Schneider D, Engelman DM (2004) Motifs of two small residues can assist but are not sufficient to mediate transmembrane helix interactions. J Mol Biol 343:799–804
18. Escher C, Cymer F, Schneider D (2009) Two GxxxG-like motifs facilitate promiscuous interactions of the human ErbB transmembrane domains. J Mol Biol 389:10–16
19. Finger C, Volkmer T, Prodöhl A et al (2006) The stability of transmembrane helix interactions measured in a biological membrane. J Mol Biol 358:1221–1228
20. Prodöhl A, Weber M, Dreher C et al (2007) A mutational study of transmembrane helix–helix interactions. Biochimie 89:1433–1437
21. Dmitrova M, Younes-Cauet G, Oertel-Buchheit P et al (1998) A new LexA-based genetic system for monitoring and analyzing protein heterodimerization in Escherichia coli. Mol Gen Genet 257:205–212
22. Treptow NA, Shuman HA (1985) Genetic evidence for substrate and periplasmic-binding-protein recognition by the MalF and MalG proteins, cytoplasmic membrane components of the Escherichia coli maltose transport system. J Bacteriol 163:654–660

23. Sambrook J, Russel DW (2001) Cold Spring Habour Press USA. Cold Sping Harbor, New York, NY
24. Miller JH (1992) Cold Spring Harbor Laboratory Press. Cold Spring Harbor, New York, NY
25. Lau FW, Bowie JU (1997) A method for assessing the stability of a membrane protein. Biochemistry 36:5884–5892
26. Kolmar H, Hennecke F, Gotze K et al (1995) Membrane insertion of the bacterial signal transduction protein ToxR and requirements of transcription activation studied by modular replacement of different protein substructures. EMBO J 14:3895–3904
27. Lis M, Blumenthal K (2006) A modified, dual reporter TOXCAT system for monitoring homodimerization of transmembrane segments of proteins. Biochem Biophys Res Commun 339:321–324
28. King G, Dixon AM (2010) Evidence for role of transmembrane helix-helix interactions in the assembly of the Class II major histocompatibility complex. Mol Biosyst 6:1650–1661
29. Volkmer T, Becker C, Prodöhl A et al (2006) Assembly of a transmembrane b-type cytochrome is mainly driven by transmembrane helix interactions. Biochim Biophys Acta 1758:1815–1822
30. Fuhrmann E, Bultema JB, Kahmann U et al (2009) The vesicle-inducing protein 1 from Synechocystis sp. PCC 6803 organizes into diverse higher-ordered ring structures. Mol Biol Cell 20:4620–4628

Chapter 17

Supersecondary Structure Prediction of Transmembrane Beta-Barrel Proteins

Van Du T. Tran, Philippe Chassignet, and Jean-Marc Steyaert

Abstract

We introduce a graph-theoretic model for predicting the supersecondary structure of transmembrane β-barrel proteins—a particular class of proteins that performs diverse important functions but it is difficult to determine their structure with experimental methods. This ab initio model resolves the protein folding problem based on pseudo-energy minimization with the aid of a simple probabilistic filter. It also allows for determining structures whose barrel follows a given permutation on the arrangement of β-strands, and allows for rapidly discriminating the transmembrane β-barrels from other kinds of proteins. The model is fairly accurate, robust and can be run very efficiently on PC-like computers, thus proving useful for genome screening.

Key words: Transmembrane proteins, Beta-barrels, Protein structure prediction, Supersecondary structure, Permuted structure, Greek key, Ab initio modeling, Pseudo-energy model

1. Introduction

Proteins are classified into three major classes according to their overall three-dimensional structures and their functional roles: fibrous, globular, and membrane proteins. Fibrous proteins, which tend to be elongated fibers, are generally strong and insoluble, and thus play structural roles in organisms. Globular proteins, which comprise a large variety of structures, are soluble in aqueous environment. Hence, these proteins generally have compact structures with polar residues on the surface and hydrophobic residues in the core. Membrane proteins exist in the cell membrane—a phospholipid bilayer with hydrophobic core. They typically have hydrophobic exposed regions in order to be stable in such an environment. Some proteins slightly adhere to the membrane, while others are

Alexander E. Kister (ed.), *Protein Supersecondary Structures*, Methods in Molecular Biology, vol. 932,
DOI 10.1007/978-1-62703-065-6_17,

embedded in the lipid bilayer. Among the latter, some proteins, namely, transmembrane proteins, entirely span the biological membrane one or several times (*polytopic* proteins).

Transmembrane proteins play many important roles in the functioning of cells such as enzymes, receptors, transporters, and channels. They are also involved in many human diseases including heart disease, cancer, Alzheimer's, depression, migraine, retinitis pigmentosa, hereditary deafness, diabetes, etc. (1, 2). As a result, they are the targets of a majority of current medicine. These proteins make up 20–30% of identified proteins in most whole genomes. However, determining the structure of transmembrane proteins with experimental methods, such as X-ray crystallography or NMR, is difficult as they are totally destabilized by the change of environment after the removal from the membrane. Solved transmembrane protein structures constitute only about 1–2% of the RCSB Protein Data Bank (PDB) (3–7). Therefore, structure prediction by computational methods for this class of proteins is of particular importance for both biological and medical sciences.

Transmembrane proteins are divided into two main types regarding their conformation: α-helical bundles and β-barrels (TMB). The TMB proteins, which are much less abundant than α-helical transmembrane proteins in the PDB, are found in the outer membrane of Gram-negative bacteria, mitochondria, and chloroplasts. Contrarily to a great progress in structure prediction on α-helices (7), due to a tiny number of determined TMBs, the learning-based predictions for these proteins are still far from being reliable, although various techniques have been recently developed for discriminating TMB proteins from globular and transmembrane α-helical proteins (8–10), and for predicting TMB secondary structures (9–14).

Freeman et al. (8) introduced a statistical approach for recognition of TMB proteins based on known physicochemical properties. Gromiha et al. (9, 10) used the amino acid compositions of both globular and outer membrane proteins (OMPs) to discriminate OMPs and developed a feed forward neural network-based method to predict the transmembrane segments. Ou et al. (11) proposed a method based on radial basis function networks to predict the number of β-strands and membrane spanning regions in β-barrel OMPs. Randall et al. (12) tried to predict the TMB secondary structure with one-dimensional recursive neural network using alignment profiles. Bagos et al. (13) produced a consensus prediction from different methods based on hidden Markov models, neural networks, and support vector machines (9, 15–21). Waldispühl et al. (14) used a structural model and pair-wise interstrand residue statistical potentials derived from globular proteins to predict the supersecondary structure of TMB proteins. Most of these rely on the learning assumptions in the underlying models as well as the sampling of proteins in their training dataset. As only a few TMB

structures are found, it is arguable whether these approaches can work well for recognizing and folding TMB proteins that are not homologous to those currently known.

Moreover, the Greek key motifs are the topological signature of many β-barrel and β-sandwich structures (22). It raises an open question whether the TMB structures are not merely a series of β-strands where each is bonded to the preceding and succeeding ones in the primary sequence, but they may contain Greek key or Jelly roll motifs as well, for instance, the C-terminal domain of the outer membrane usher protein PapC (PDB:3L48). This level of structure may be described as a permutation on the order of the bonded strands.

In this chapter, we present an ab initio model for structure prediction of TMB proteins based on minimizing free energy in a graph-theoretic framework, which is able to deal with permuted TMB structures. This approach performs well in structure prediction with comparable results to those of the existing algorithms. It is also efficient at discriminating the TMB proteins from transmembrane α-helical and globular proteins (23, 24).

2. Materials

We took TMB proteins from the PDBTM database (25) to construct a probabilistic model. The CD-HIT tool (26) is used to restrain the redundancy in these proteins. A threshold of 40% similarity was applied to reduce the dataset, resulting in 49 sequences (PDBTM40). We retain only the monomeric barrels, i.e., the sequences that form a unique complete barrel. Thus, PDBTM40 finally contains the 41 sequences 1AF6_A, 1BH3_A, 1BXW_A, 1BY3_A, 1FEP_A, 1ILZ_A, 1OH2_Q, 1P4T_A, 1PNZ_A, 1QJ8_A, 1TLW_A, 1UXF_A, 1UYN_X, 1XKW_A, 2ERV_A, 2F1T_A, 2FGQ_X, 2GSK_A, 2HDF_A, 2IAH_A, 2IWW_A, 2J1N_A, 2O4V_A, 2ODJ_A, 2POR__, 2R4P_A, 2VDF_A, 2WJQ_A, 2X4M_A, 3A2R_X, 3AEH_A, 3BRZ_A, 3CSL_A, 3DWO_X, 3EFM_A, 3EMN_X, 3FHH_A, 3FID_A, 3GP6_A, 3JTY_A, 3NJT_A.

3. Methods

Physicochemical properties and a simple probabilistic model based on a sliding window are applied to discard the segments of amino acids that are obviously not involved in any β-barrel structures as a membrane spanning β-strand. The given protein will be folded into a TMB structure with available putative β-strands using the

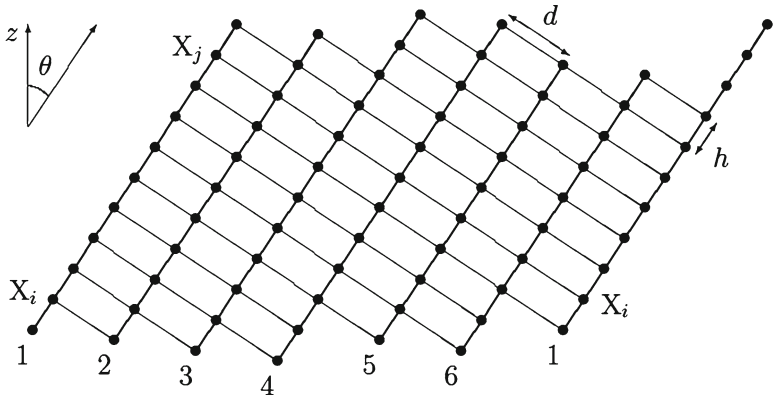


Fig. 1. The simplified geometry of a β-barrel, a schematic planar view for six strands (strand 1 is duplicated for clarity). *Thick lines* denote the peptide bonds that link consecutive amino acids along their strand. *Thin lines* denote the hydrogen bonds that link the amino acids of two adjacent strands. In this example, the shear number is $S=8$, which is the ordinal distance between amino acids X_i and X_j.

pseudo-energy minimization model. If the protein cannot be folded into a β-barrel according to the energy minimization framework, it is classified as a non-TMB protein (see Note 1).

We discuss some geometric (Subheading 3.1) and physicochemical (Subheading 3.2) constraints that a protein must satisfy to be a TMB before presenting the simple model used for filtering the membrane spanning β-strands. We give our concrete folding problem definition in the next section, followed by the description of a dynamic programming approach to solve the problem.

3.1. Geometric Framework for β-Barrels

The backbone geometry of a regular β-barrel (27–29) is entirely determined by n, the number of strands composing the barrel, and by S, the shear number (see Note 2). In such a perfect case, S is unambiguously defined as the ordinal distance between an amino acid X_i and an amino acid X_j that is located on the same strand with X_i and linked to X_i through a path of hydrogen bonds (see Fig. 1). Structural constants are h (≈3.3 Å), the jump per amino acid along a strand, and d (≈4.4 Å), the mean distance between adjacent strands, given by the peptide bond and hydrogen bond geometries, respectively. The other geometric characteristics, such as θ, the slant angle of the strands relative to the barrel z-axis, are given from n, S, h, and d (30):

$$\tan\theta = \frac{hS}{dn}.$$

Angle θ, in association with a given membrane thickness, is involved in the energetic rules and restricts the membrane spanning β-strand length. Then, n and S have to be fixed as parameters.

When the *shear number* S is given, the *relative shears* between adjacent strands remain as $n-1$ degrees of freedom. As a convention,

we consider the relative shears on the periplasmic side of the barrel. So, $\forall i > 1$, s_i, the relative shear of strand i with respect to strand $i-1$ (strand $n+1$ being identified with 1), is measured on strand $i-1$ as the ordinal distance between the undermost amino acid of strand $i-1$ and the one that is directly hydrogen-bonded to the undermost amino acid of strand i. For the example of Fig. 1, the sequence of relative shears s_i is (1 1 1 2 1 2). The sum of elementary consecutive relative shears naturally defines the shear between two farthest strands, thus we have the constraint: $\sum_{1<i\leq n+1} s_i = S$. We define, by generalization, the *shear number* of a β-sheet (i.e., an open β-barrel) by $S = \sum_{1<i\leq n} s_i$. Each β-strand is directed with respect to the sequence order for its amino acids and the *upward/downward* orientation (from extracellular side to periplasmic side and reversely) of this strand, relatively to the barrel z-axis, defines another degree of freedom.

Finally, considering a β-strand as a ribbon where the amino acids direct their side-chains alternatively on both sides (*inward* and *outward* the barrel), we distinguish only two ways of facing, neglecting small swivel adjustments.

3.2. Physicochemical Constraints

On the amphipathic β-strand of TMB proteins, the side-chains of amino acids are directed toward the membrane and the channel alternatively. Hydrophilic and polar side-chains orient toward the aqueous interior, while hydrophobic ones are in contact with the hydrophobic bilayer (31). We have used the Kyte–Doolittle scale (32) to evaluate the hydrophobicity $H(r)$ of each amino acid r. In this scale, a higher value represents higher hydrophobicity, and vice versa. A segment of $j-i+1$ consecutive amino acids $r_i,\ldots,r_j$ is a potential membrane spanning β-strand if one side is hydrophobic and the other side is hydrophilic. Formally, we define

$$H^{\mathrm{e}}_{i,j} = \left\langle H(r_{2k}) \right\rangle, \quad i \leq 2k \leq j$$

$$H^{\mathrm{o}}_{i,j} = \left\langle H(r_{2k+1}) \right\rangle, i \leq 2k+1 \leq j$$

as the average hydrophobicity on the respective even and odd numbered sides. A segment $r_i,\ldots,r_j$ is a potential membrane spanning β-strand if

$$\max\{H^{\mathrm{e}}_{i,j}, H^{\mathrm{o}}_{i,j}\} > \eta^- \wedge \min\{H^{\mathrm{e}}_{i,j}, H^{\mathrm{o}}_{i,j}\} < \eta^+,$$

where η^- is a lower bound for the hydrophobic side and η^+ is an upper bound for the hydrophilic side. We use the values $\eta^- = -1$ and $\eta^+ = 1$, which were obtained through training. Then, with respect to the TMB structure, the segment $r_i,\ldots,r_j$ is defined as *odd inward* oriented if $H^{\mathrm{o}}_{i,j} < H^{\mathrm{e}}_{i,j}$, and *odd outward* oriented if $H^{\mathrm{e}}_{i,j} < H^{\mathrm{o}}_{i,j}$.

3.3. Filtering

In order to identify substrings as potential membrane spanning β-strands (the vertices) or turns/loops (the edges), we introduce a simple probabilistic model that acts as a primary filter. We use a sliding window (segment) as a sequence of consecutive *l*-residue subsegments (or blocks) (see Note 3). Let r denote the occurrence of a given block ($r = r_1, r_2, \ldots, r_l$) and let c be the event that a block is found in a given conformation (β-strand or turn/loop). The information that c gets from r is defined as:

$$\mathrm{I}(c\,;r) = \log\frac{P(c\mid r)}{P(c)} = \log\frac{f_{c,r}\,/\,f_{*,r}}{f_{c,*}\,/\,f_{*,*}},$$

where $f_{c,r}$ represents the frequency observed in the training dataset for a block r to be found in conformation c and we denote for short (33):

$$f_{*,r} = \sum_{c} f_{c,r},$$

$$f_{c,*} = \sum_{r} f_{c,r},$$

and

$$f_{*,*} = \sum_{c}\sum_{r} f_{c,r}.$$

Thus, $I(c;r)$ measures the influence of r on the occurrence of s. If $I(c;r) = 0$, there is no influence; whereas $I(c;r) > 0$ indicates that r is favorable to the occurrence of c and vice versa. Formally, the preference of r in favor of c as opposed to $\overline{c}$, where $\overline{c}$ is any conformation different from c (34), is:

$$I(c:\overline{c}\,;r) = I(c\,;r) - I(\overline{c}\,;r) = \log\frac{f_{c,r}\,/\,f_{c,*}}{f_{\overline{c},r}\,/\,f_{\overline{c},*}}.$$

A simple measure is associated to each segment $r = r_1, r_2, \ldots, r_p$ that helps determine if it is likely a β-strand or a coil. It is defined as the sum of information on all *l*-residue blocks:

$$\tilde{I}(c:\overline{c};r_1,r_2,\ldots,r_p) = \sum_{i=1}^{p-l+1}\frac{I(c:\overline{c};r_{i+1},r_{i+2},\ldots,r_{i+l-1}) - \log\rho}{p-l+1}.$$

The segment is then considered as a candidate for conformation c if $\tilde{I}(c:\overline{c};r_1,r_2,\ldots,r_p) > 0$.

The nonredundant training set of TMB proteins described in Subheading 2 is used to learn this probabilistic model. Due to the small size of the training set, we apply the filter with a relatively low threshold at $\rho = 2/3$ to avoid overfitting (see Note 4). This ensures that, on the average, each block r is accepted in conformation c if the propensity for r to be in c (i.e., $f_{c,r}\,/\,f_{c,*}$) is at most 1.5 times

less than the propensity for r to be in $\overline{c}$ (i.e., $f_{\overline{c},r} / f_{\overline{c},\star}$). Only substrings that pass these very stringent criteria are considered to be putative strands.

3.4. Folding Problem Definition

Let Γ be the sequence of the N amino acids constituting the primary structure of a given protein. We consider $G(V, E, \Phi_{\text{intr}}, \Phi_{\text{adj}}, \Phi_{\text{loop}})$, the weighted directed acyclic graph (DAG) (35) built from Γ as follows:

1. *Vertices.* Let $V = V_+ \cup \{\triangleleft, \triangleright\}$ be the set of vertices. Each vertex of V_+ represents a candidate secondary structure item defined as a β-strand associated with a given set of parameters. It corresponds to a contiguous part (a substring, defined by its starting and ending indices $1 \leq \tau < \upsilon \leq N$) of Γ that satisfies given conformational constraints (such as length, propensity to be a β-strand…). The associated parameters describe the discretized spatial laying of this part relatively to the whole structure. So, combining the *upward/downward* and *inward/outward* degrees of freedom introduced in Subheading 3.1, we consider four different orientations for each given candidate β-strand. We could also consider the different instances of relative shear to multiply the number of vertices, but we do not for reasons to be clarified later. A canonical order is defined on V_+ as the lexicographic order on tuples formed by the respective starting/ending indices in Γ and the associated parameters. The length constraint implies that the number of candidate substrings and thus $|V|$, the number of vertices, are bounded above by kN for a small value k. To simplify further definitions, a dummy vertex $\triangleleft$ will be used to represent an empty substring at the start of Γ and, similarly, $\triangleright$ will represent an empty substring at the end of the sequence. To extend the order on all of the vertices, we set $\triangleleft < v < \triangleright$, $\forall v \in V_+$.
2. *Edges.* Let $E \subset V \times V$ be the set of directed edges. Intuitively, an edge in graph G corresponds to a turn or a loop that connects two consecutive β-strands. To be more precise, $\forall v$, $w \in V_+$, with τ_v, υ_v, τ_w, υ_w denoting their respective starting and ending indices, (v, w) is an edge, if $\upsilon_v < \tau_w - 2$ and the substring of amino acids from $\upsilon_v + 1$ to $\tau_w - 1$ satisfies the constraints that allow to form a turn or a loop (conditions on length, flexibility, propensity, …) also depending on the relative laying of the two substructures. We have the elementary property:

$$\forall v, w \in V_+, (v, w) \in E \Rightarrow v < w$$

for the lexicographic order, and this ensures the DAG structure. The set E also contains edges of the form $(\triangleleft, v)$ that compose the subset of starting vertices the leading substrings satisfying specific constraints. Similarly, E contains edges of the form $(v, \triangleright)$ that compose the subset of ending vertices, with a

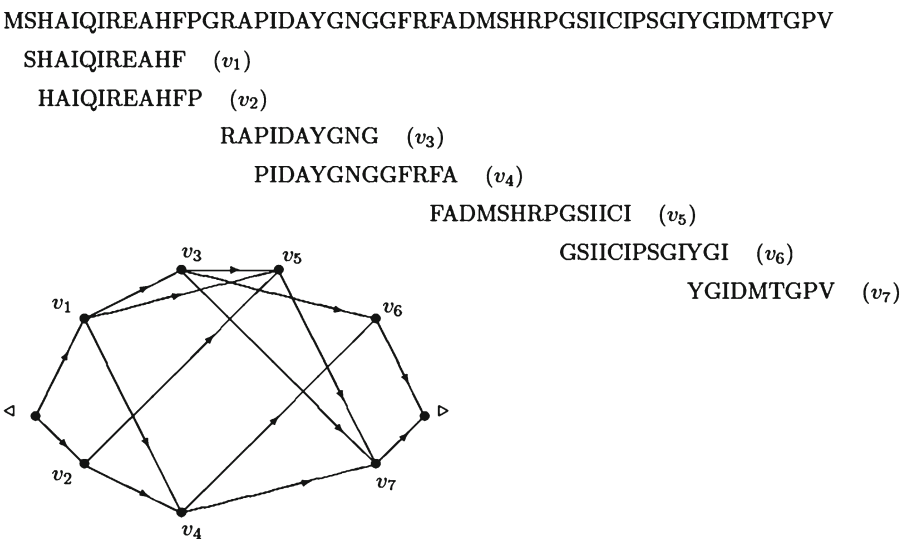


Fig. 2. An example of the graph structure.

satisfactory trailing substring. Again, the length constraints applied to the substrings associated to edges imply that $|E|$, the number of edges, is ($O|V|$) or $O(N)$.

Figure 2 gives a small example of such a graph (to simplify, only one orientation has been considered). An edge like (v_1, v_2) is forbidden, since the two corresponding substrings overlap. Edges such as (v_2, v_3) or (v_2, v_6) are also forbidden, since the inserted substrings are respectively too short for a turn or too long for a loop.

3. *Energy attributes.* The attributes that complete the definition of the graph G are pseudo-energy functions (see Note 5) defined as follows:
   - $\forall v \in V_+$, $\Phi_{\text{intr}}(v)$ represents the intrinsic energy of the given strand in the given orientation. This term is the sum of both the internal energy of the substructure, i.e., the interactions between its own amino acids, and the interaction energy with the environment (e.g., membrane and channel) apart from the rest of the considered protein. Note that $\Phi_{\text{intr}}(\triangleleft) = \Phi_{\text{intr}}(\triangleright) = 0$.
   - $\forall (v, w) \in V_+ \times V_+$, $\Phi_{\text{adj}}(v, w, s)$ represents the interaction energy of the pair (v, w) when the two corresponding strands are placed side by side along the barrel, with respect to the respective orientation parameters associated to the vertices and accordingly to the relative shear s. The energy will take into account the number of contacts and different side-chain interactions as packing of hydrophobic cores and bonding abilities.

     Then, $\forall (v, w) \in V_+ \times V_+, \Phi_{\text{adj}}(v, w) = \min_s \Phi_{\text{adj}}(v, w, s)$ is the interaction energy of the pair (v, w) for an optimal relative

shear. It is further assumed that Φ_{adj} is defined over a superset of E, since we consider the case where two adjacent strands are not consecutive along the sequence. We also introduce the particular values: $\Phi_{adj}(\triangleleft, v) = \Phi_{adj}(v, \triangleright) = 0, \forall v \in V$.

- An associated function s_{adj} is defined such that: $\forall (v,w) \in V_+ \times V_+, \Phi_{adj}\left(v, w, s_{adj}(v,w)\right) = \Phi_{adj}(v,w)$, which is a relative shear that leads to the optimal interaction energy. An arising question is why the orientation degrees of freedom are described as a multiplicity of nodes but the *relative shear* degrees of freedom are considered when calculating the Φ_{adj} terms. A first answer comes from the fact that wrong orientations are rather absolute and will result in pruning the sets E and V, while the *shear* parameters are not so discriminative. The main reason is that we consider "floating" parts in which adjacencies are already set, while a *relative shear* between any two parts is not yet known. In such a situation, attaching the *relative shears* to node pairs allows a significant factorization.
- $\forall (v,w) \in E$, $\forall t \in \{1, 2, \ldots, n-1\}$ and $\forall s$—a relative shear, $\Phi_{loop}(v, w, t, s)$ is related to the intrinsic energy of the turn/loop between the strands v and w (consecutive along the sequence) when they are placed at a distance t along the barrel with a relative shear s. The distance $t = 1$ corresponds to the case where the strands are placed consecutively on the barrel, while an integer value $t > 1$ will correspond to the case where $t-1$ other strands are interleaf.

  To simplify, we also use $\Phi_{loop}(\triangleleft, v)$ or $\Phi_{loop}(v, \triangleright)$ for denoting the intrinsic energy of the outer fragment attached respectively to a starting or an ending vertex v. As such a fragment has a free side, the position parameters may be dropped.

  Then, in the usual case of two β-strands that fold as a hairpin, the related energy is considered to be $\Phi_{adj}(v,w) + \Phi_{loop}(v, w, 1, s_{adj}(v,w))$. It is supposed a relative flexibility for turns and loops, so, when a fold is feasible, Φ_{loop} is weak compared to Φ_{adj} and the relative placement of the two β-strands is enforced to be s_{adj}. Nevertheless, Φ_{loop} will result in a strong penalty in the case of an unfeasible turn or loop, for example a loop with a majority of hydrophobic residues.

4. *Protein folding problem*. Given a graph $G(V, E, \Phi_{intr}, \Phi_{adj}, \Phi_{loop})$ defined as above, two integers n, S and a permutation σ, we look for the path Λ in G that maximizes the following objective function:

$$\Phi = \sum_{v \in \Lambda} \Phi_{intr}(v) + \sum_{(v,w) \in \Lambda} \Phi_{loop}(v,w) + \sum_{(v,w) \in \sigma(\Lambda)} \Phi_{adj}(v,w)$$

such that $\sum_{(v,w) \in \sigma(\Lambda)} s_{adj}(v,w) = S$.

3.5. Dynamic Programming Approach

1. *Solving as the longest path problem*

- We first consider the case of an open structure, as a β-sheet, where the adjacency of strands follows their order in the amino acids sequence. We involve here the constraint $\sum_{1<i\leq n} s_i = S$. Hence, solving such a structure will resume in finding a path Λ in G and the overall "energy" for this structure results in a sum:

$$\Phi = \sum_{v\in\Lambda} \Phi_{\text{intr}}(v) + \sum_{(v,w)\in\Lambda} \left(\Phi_{\text{adj}}(v,w) + \Phi_{\text{loop}}(v,w,1,s_{\text{adj}}(v,w))\right)$$

- Considering a minimization of Φ, the protein folding problem will turn into finding the path from $\triangleleft$ to $\triangleright$ that maximizes the criterion $C = -\Phi$. Let C_v^h be the maximum value for C over all the paths from $\triangleleft$ to v, with a shear number of h of the corresponding β-sheet, then $C_{\triangleleft}^0 = 0$, and, $\forall v \in V \setminus \{\triangleright\}$, $\forall h$, C_v^h is defined as:

$$C_v^h = \max_{u\in V,\,(u,v)\in E} \left(C_u^{h-s_{\text{adj}}(u,v)} - \Phi_{\text{intr}}(v) - \Phi_{\text{adj}}(u,v) - \Phi_{\text{loop}}(u,v,1,s_{\text{adj}}(u,v))\right).$$

- Since the graph is a DAG, the longest path problem is solved with a well known dynamic programming scheme (35) of complexity $O(|V|)$ in space and $O(|V|+|E|)$ in time, that is also $O(N)$ for both, from the structural constraints that relate $|V|$, $|E|$ and N. The objective is the computation of $C_{\triangleright}^S$ and the optimal structure is then reconstructed by a usual traceback postprocessing. Note that, for each path, we only have to consider its last vertex, thus we have to track single index states.
- For a barrel secondary structure, we have to consider a closing spatial adjacency between the last and the first strands. The dynamic programming scheme is almost the same as the previous, except that we also have to keep track of the first vertex of any path. So, $\forall v \in V_+$, such that $(\triangleleft, v) \in E$, let $C_{(v,v)}^0 = -\Phi_{\text{intr}}(v) - \Phi_{\text{loop}}(\triangleleft, v)$, then the general recurrence is: $\forall v, w \in V_+$, $\forall h$, such that $(\triangleleft, v) \in E$,

$$C_{(v,w)}^h = \max_{u\in V,\,(u,w)\in E} \left(C_{(v,u)}^{h-s_{\text{adj}}(u,w)} - \Phi_{\text{intr}}(w) - \Phi_{\text{adj}}(u,w) - \Phi_{\text{loop}}(u,w,1,s_{\text{adj}}(u,w))\right)$$

and a special closing step is needed: $\forall v \in V_+$, $\forall h$, such that $(\triangleleft, v) \in E$,

$$C_{(v,\triangleright)}^h = \max_{u\in V,(u,\triangleright)\in E} \left(C_{(v,u)}^{h-s_{\text{adj}}(u,v)} - \Phi_{\text{adj}}(u,v) - \Phi_{\text{loop}}(u,\triangleright)\right)$$

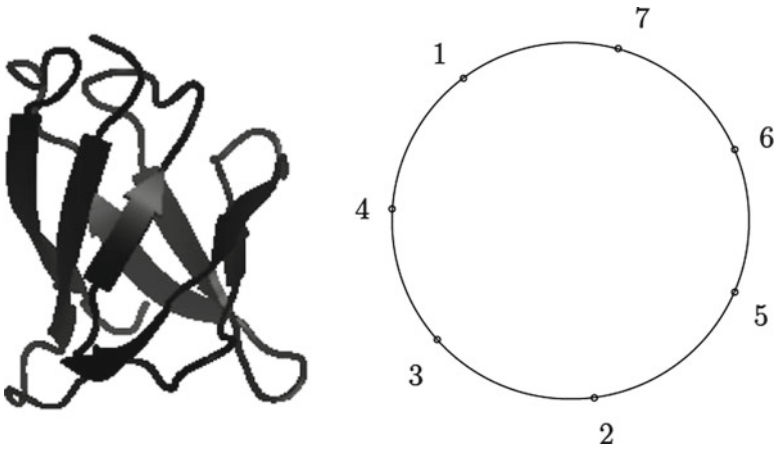


Fig. 3. An example of a β-barrel with a Greek key motif.

The goal is to calculate $\max_{v\in V,(\lhd,v)\in E} C^{S}_{(v,\rhd)}$. Thus the scheme is of complexity $O(|V|^2)$ in space and $O(|V||E|)$ in time, which is also $O(N^2)$ for both, from the structural constraints. This may produce paths of any length and the constraint of n strands is applied as a cut in recurrence.

2. *Generalization*
In a more general case, we consider permutations to deal with the fact that the arrangements of the strands along the barrel do not follow their order in the sequence. This usually occurs with Greek key motifs or more rarely with Jelly roll motifs. Hence, the protein folding problem becomes finding the longest path Λ in a graph with respect to a given permutation σ, i.e., the vertices of Λ, seen on a circle as in Fig. 3, are permuted according to σ.

Let σ be a circular permutation of $\{1, 2, \ldots, n\}$. When $1, 2, \ldots, n$ number the positions along the barrel, values $\sigma(1), \sigma(2), \ldots, \sigma(n)$ will give the respective ranks of the strands in the sequence order. A reference position along the barrel is fixed by setting $\sigma(1) = 1$. The Greek key example of Fig. 3 is described by the permutation $\sigma = (1, 4, 3, 2, 5, 6, 7)$.

The dynamic programming scheme now consists in building a barrel by consecutively adding a new strand following the graph edges. Such a strand will be inserted at the position defined by the given permutation. Useful values are the ranks (in the sequence order) of the two strands between which a given one will be inserted. For instance, with the given example of Fig. 3, the fourth strand will be inserted between the first and the third strands.

Let now k denote the level of construction ($1 \le k \le n$), that is the number of strands already placed.

Proposition 1 The k^{th} strand (in the sequence order) is inserted between the two strands whose ranks (in the sequence order) are $\mathbf{left}_k$ and $\mathbf{right}_k$, defined as:

$$\mathbf{left}_k = \begin{cases} \sigma(\sigma^{-1}(k)-1) & \text{if } \sigma^{-1}(\mathrm{k}) > 1 \\ \sigma(n) & \text{otherwise} \end{cases}$$

$$\mathbf{right}_k = \begin{cases} \sigma(\sigma^{-1}(k)+1) & \text{if } \sigma^{-1}(k) < n \\ 1 & \text{otherwise} \end{cases}$$

With the current example, we get:
$\mathbf{left}_1 = 7, \mathbf{left}_2 = 3, \mathbf{left}_3 = 4, \mathbf{left}_4 = 1, \mathbf{left}_5 = 2, \mathbf{left}_6 = 5, \mathbf{left}_7 = 6$
$\mathbf{right}_1 = 4, \mathbf{right}_2 = 5, \mathbf{right}_3 = 2, \mathbf{right}_4 = 3, \mathbf{right}_5 = 6, \mathbf{right}_6 = 7,$
$\mathbf{right}_7 = 1$

An important piece of information to store for the dynamic programming scheme is the set of "active" indices. They are ranks of the strands (in the sequence order) that are either not definitively bonded on both sides along the barrel or not linked along the sequence, and thus have to be kept as degrees of freedom. So, in the given example, we have to keep in mind every valid instance as the first and third strands until an optimal choice is recorded for each instance as a fourth strand. At that time, any instance as a fourth strand is kept as a candidate for a link with a fifth strand, by a turn or loop, while the instances as the second strand are also kept for proceeding to the insertion of a fifth strand.

Definition 1 Two ranks i and j, which refer to the sequence order, are said "adjacent" if:

$$\left|\sigma^{-1}(i) - \sigma^{-1}(j)\right| \in \{1, n-1\},$$

where the case $n-1$ is intended for the adjacency that will close the barrel.

Proposition 2 The set of "active" indices (in the sequence order) at level k is defined by:

$$\mathbf{conf}_k = \{k\} \cup \{i \,|\, (1 \le i \le k) \wedge (\exists j : k < j \le n \,|\, i, j \text{ are "adjacent"})\}$$

With the current example, we get:

$\mathbf{conf}_1 = \{1\}$ $\mathbf{conf}_2 = \{1, 2\}$ $\mathbf{conf}_3 = \{1, 2, 3\}$
$\mathbf{conf}_4 = \{1, 2, 4\}$ $\mathbf{conf}_5 = \{1, 5\}$ $\mathbf{conf}_6 = \{1, 6\}$

Thus, in this example, the maximal complexity in space, $O(N^3)$, is reached for the set of subsolutions with four strands. Then looping over this set, for computing the set of subsolutions with five strands, will also cost $O(N^3)$ in time, since for each four-strand subsolution the choice for a fifth strand is bounded to a small value by the structural constraints embedded as edges in the graph.

Now we have to decide at which minimal level k each term $\mathbf{\Phi}_{\text{adj}}$ or $\mathbf{\Phi}_{\text{loop}}$ is determined and can be integrated in the dynamic programming scheme. For the $\mathbf{\Phi}_{\text{adj}}$ terms, it is simply asserted that the previous or the next strand along the barrel is already placed when $\mathbf{left}_k < k$ or $\mathbf{right}_k < k$, respectively.

Proposition 3 For all k, we have:

$$\mathbf{left}_k < k \Leftrightarrow \mathbf{left}_k \in \mathbf{conf}_{k-1},$$

$$\mathbf{right}_k < k \Leftrightarrow \mathbf{right}_k \in \mathbf{conf}_{k-1}$$

This results from the definition of the "active" indices of $\mathbf{conf}_{k-1}$. To simplify the further energy expression, we use the following notation for an "ifelse" function:

$$\mathbf{if}_k(i,x) = \begin{cases} x & \text{if } i < k \\ 0 & \text{otherwise} \end{cases}$$

For the $\mathbf{\Phi}_{\text{loop}}$ terms, the problem is to wait until the relative shear between the two ends of a turn or loop is solved by the interleaf adjacencies. So, in the given example, the energy of the loop between the first and second strands can only be evaluated when the fourth strand has been laid and the optimal relative shear $s^*_{\text{adj}}(v_1, v_2) = s_{\text{adj}}(v_1, v_4) + s_{\text{adj}}(v_4, v_3) + s_{\text{adj}}(v_3, v_2)$ is known.

Definition 2 Let Δ_k be the relation on positive integers, defined as: $\forall i, j$,

$$i \Delta_k j \Leftrightarrow \begin{bmatrix} i = j \\ (i \le k) \wedge (j \le k) \wedge (i, j \text{ are "adjacent"}) \end{bmatrix}$$

then let Δ^*_k denote the equivalence relation defined by the transitive closure of Δ_k and let $A_k = \{i < k \mid i \Delta^*_k (i+1)\}$.

Thus, $i \in A_k$ means that the i^{th} and $(i+1)^{\text{th}}$ strands are geometrically linked by adjacencies when the k^{th} substructure is laid and we can compute by composition an optimal relative shear s^*_{adj}.

We now focus on the set $\delta A_k = A_k - A_{k-1}$, $\forall k > 1$.

Proposition 4 For all k, we have:

$$(k-1) \in \delta A_k \Leftrightarrow \mathbf{left}_k \Delta^*_{k-1} (k-1) \vee \mathbf{right}_k \Delta^*_{k-1} (k-1)$$

Proposition 5 For all $i < k-1$,

$$i \in \delta A_k \Leftrightarrow \begin{cases} i \notin A_{k-1} \\ \begin{bmatrix} \mathbf{left}_k \Delta^*_{k-1} i \wedge \mathbf{right}_k \Delta^*_{k-1} (i+1) \\ \mathbf{right}_k \Delta^*_{k-1} i \wedge \mathbf{left}_k \Delta^*_{k-1} (i+1) \end{bmatrix} \end{cases}$$

Definition 3 Let $T_k \subset V_+^{|\mathbf{conf}_k|}$ denote the set of all tuples of $|\mathbf{conf}_k|$ vertices such that there is at least one path (of k edges) starting from $\triangleleft$ and leading through these vertices in order.

For any instance $z \in T_k$ of such a tuple and, $\forall i \in \mathbf{conf}_k$, let $z[i]$ denote the i^{th} vertex of a corresponding path.

This notation (not to be confused with z_i, the i^{th} component of tuple z) is not ambiguous since, from definition, the vertex $z[i]$ is in common to any path associated to z. Particularly, $z[k]$ is the last vertex of any path associated to z.

Proposition 6 For all $z \in T_k$, the set of tuples corresponding to paths of length $k-1$ that can be extended to a path corresponding to z is defined as:

$$\mathbf{pre}(z) = \{y \in T_{k-1} \mid ((y[k-1], z[k]) \in E)$$

$$\wedge(\forall i \in \mathbf{conf}_k \cap \mathbf{conf}_{k-1}, y[i] = z[i])\}$$

Let $C_{k,z}^h$ be the maximum value for C over all paths starting from $\triangleleft$ and leading in order through the vertices of a given tuple $z \in T_k$ with a shear number of h of the corresponding β-barrel. The general recurrence is $\forall z \in T_k$,

$$C_{k,z}^h = \max_{y \in \mathbf{pre}(z)} (C_{k-1,y}^{h-s_{\text{adj}}(y[\mathbf{left}_k], z[k]) - s_{\text{adj}}(z[k], y[\mathbf{right}_k])} - \mathbf{\Phi}_{\text{intr}}(z[k])$$
$$- \mathbf{if}_k(\mathbf{left}_k, \mathbf{\Phi}_{\text{adj}}(y[\mathbf{left}_k], z[k])) - \mathbf{if}_k(\mathbf{right}_k, \mathbf{\Phi}_{\text{adj}}(z[k], y[\mathbf{right}_k]))$$
$$- \sum_{i \in \delta A_k} \mathbf{\Phi}_{\text{loop}}(y[i], y[i+1], \sigma^{-1}(i+1) - \sigma^{-1}(i), s_{\text{adj}}^*(y[i], y[i+1])))$$

Note that, from Proposition 3, $\forall y \in T_{k-1}$, if $\mathbf{left}_k < k$ then the vertex $y[\mathbf{left}_k]$ is defined (and the same is worth for $\mathbf{right}_k$). We can verify that each $\mathbf{\Phi}_{\text{adj}}$ term is finally counted once in the sum, at the level corresponding to the position of its further vertex in sequence order. The optimum is found at $k = n$ and $h = S$.

Corollary 1 The complexities both in time and space are $O\left(\sum_{k=2}^{n} \left(\frac{|V|}{n}\right)^{|\mathbf{conf}_k|}\right)$, that is $O(nN^{\max_k |\mathbf{conf}_k|})$.

For any permutation, we have $|\mathbf{conf}_{n-k}| \leq \min\{1+2k, n-k\}$, $\forall k = 0, 1, \ldots, n-1$. Hence $\max_k |\mathbf{conf}_k| \leq 1 + (2n-2)/3$. For a permutation that only differs from the identity permutation by disjoint Greek key motifs, i.e., $\{1, 2, \ldots, i_1, K_1, i_1+5, \ldots, i_2, K_2, i_2+5, \ldots, K_j, \ldots, n\}$ where $K_j = \{i_j+3, i_j+2, i_j+1, i_j+4\}$, it is easy to see that $\max_k |\mathbf{conf}_k| \leq 4$.

It is possible to compute the optimum in $O(nN^2)$ running time for structures corresponding to the identity permutation and from $O(nN^2)$ to $O(nN^4)$ for structures containing disjoint Greek key motifs.

4. Notes

1. A threshold on overall energy can also be involved to enhance the discrimination. We studied the per-strand energy value for a variety of TMB proteins including the training dataset and other TMB proteins. Even though this value is always higher than 0.9 for these proteins, we chose 0.85 as a threshold to avoid overfitting. Note that this does not affect the prediction results and is only used for discriminating the TMB proteins from the others.
2. The number of strands n and the shear number S determine the geometry of the barrel, particularly the membrane spanning part of the segments, and are thus involved in the computation of energy terms. When n and S are known, the algorithm can enforce these values and fold the protein accordingly. The values for n, which are usually even, are governed by the consideration on the length of the sequence, the thickness of membrane and the length of turns or loops, and are experimentally found between 8 and 22 (31). We note that all known β-barrels have a positive shear number (36) and are slanted "to the right," as illustrated in Fig. 1. The values for S are even and experimentally observed between n and $2n$ (28, 29). The problem is then solved by the dynamic programming under the constraints of a given (n, S). A small number of couples (n, S) have to be explored and our algorithm is fast enough for that.
3. We use $l=3$ in our implementation, that seems suitable for such a small number of training TMB sequences.
4. The lower the parameter ρ, the more independent to the training dataset the predictor. This can reduce the discrimination ability of the model. However, it may be useful to discover some "new" TMB protein.
5. Side-chain interactions between contiguous residues along a segment on the same side and interactions with the environment of channel or bilayer define the intrinsic energy of the corresponding vertex. The pairing energy of two adjacent segments in the barrel is computed by optimizing the relative positions between constituent amino acids. These energies involve hydrogen bonds in main chains, electrostatic interactions

between side-chains, hydrophobic effect as well as environmental effect. More specifically, the extracellular and intracellular environments with distinct hydrophobicity indices can have significantly different hydrophobic effects. In addition, the membrane thickness gives constraints on segment size and helps identify the interactions inside or outside the membrane region. We use here as a parameter, the default value of 3 nm for the membrane thickness, thus a thickness of eight residues (37, 38). The features on size, polarity (39), and flexibility (40) of turns and loops are also taken into consideration, i.e., turns and loops satisfy threshold constraints on their polarity and flexibility indices and their length. Their energies are approximated as reduced to the hydrophobicity term (32).

The Dunbrack backbone-dependent rotamer library (41) and the partial charges from GROMOS force field (42) are used to compute pair-wise interaction energies. The hydrophobic interaction between two side-chains u, v is assessed by the amount of contacts between nonpolar groups, calculated by taking the average on all rotamer pairs of the two side-chains $e_{u,v} = \left\langle e_{uv|rotamers} \right\rangle$. Each side-chain plays a role of a group of partial charges in the electrostatic interaction. The main-chain hydrogen bond is measured by the electrostatic potential energy between peptide CO and NH groups.

Acknowledgments

The authors would like to thank Saad Sheikh for reading through the chapter and all the INRIA AMIB Team members, especially Mireille Régnier, Yann Ponty, Julie Bernauer, and Balaji Raman for helpful discussions.

References

1. Cobbold C et al (2003) Aberrant trafficking of transmembrane proteins in human disease. Trends Cell Biol 13(12):639–647
2. Marsico A et al (2007) A novel pattern recognition algorithm to classify membrane protein unfolding pathways with high-throughput single-molecule force spectroscopy. Bioinformatics 23(2):e231–e236
3. Arora A, Tamm LK (2001) Biophysical approaches to membrane protein structure determination. Curr Opin Struct Biol 11: 540–547
4. Berman HM et al (2000) The Protein Data Bank. Nucleic Acids Res 28:235–242
5. Casadio R, Fariselli P, Martelli PL (2003) In silico prediction of the structure of membrane proteins: Is it feasible? Brief Bioinform 4(4):341–348
6. Taylor PD et al (2006) Beta-barrel transmembrane proteins: Enhanced prediction using a Bayesian approach. Bioinformation 1(6): 231–233
7. Fleishman SJ, Ben-Tal N (2006) Progress in structure prediction of alpha-helical membrane proteins. Curr Opin Struct Biol 16(4): 496–504
8. Freeman TC Jr, Wimley WC (2010) A highly accurate statistical approach for the prediction

of transmembrane beta-barrels. Bioinformatics 26(16):1965–1974

9. Gromiha MM, Ahmad S, Suwa M (2005) TMBETA-NET: discrimination and prediction of membrane spanning β-strands in outer membrane proteins. Nucleic Acids Res 33:W164–W167
10. Gromiha MM, Ahmad S, Suwa M (2004) Neural network-based prediction of transmembrane β-strand segments in outer membrane proteins. J Comput Chem 25:762–767
11. Ou Y-Y, Chen S-A, Gromiha MM (2010) Prediction of membrane spanning segments and topology in β-barrel membrane proteins at better accuracy. J Comput Chem 13: 217–223
12. Randall A et al (2008) TMBpro: secondary structure, β-contact and tertiary structure prediction of transmembrane β-barrel proteins. Bioinformatics 24:513–520
13. Bagos P, Liakopoulos T, Hamodrakas S (2005) Evaluation of methods for predicting the topology of beta-barrel outer membrane proteins and a consensus prediction method. BMC Bioinformatics 6:7
14. Waldispühl J et al (2006) Predicting transmembrane β-barrels and interstrand residue interactions from sequence. Protein Struct Funct Bioinform 65:61–74
15. Ahn CS, Yoo SJ, Park HS (2003) Prediction for beta-barrel transmembrane protein region using HMM. KISS 30(2):802–804
16. Bagos PG et al (2004) PRED-TMBB: a web server for predicting the topology of β-barrel outer membrane proteins. Nucleic Acids Res 32:W400–W404
17. Bigelow HR et al (2004) Predicting transmembrane beta-barrels in proteomes. Nucleic Acids Res 32:2566–2577
18. Jacoboni I et al (2001) Prediction of the transmembrane regions of β-barrel membrane proteins with a neural network-based predictor. Protein Sci 10:779–787
19. Martelli PL et al (2002) A sequence-profile-based HMM for predicting and discriminating β-barrel membrane proteins. Bioinformatics 18(Suppl 1):S46–S53
20. McGuffin LJ, Bryson K, Jones DT (2000) The PSIPRED protein structure prediction server. Bioinformatics 16:404–405
21. Natt NK, Kaur H, Raghava GPS (2004) Prediction of transmembrane regions of β-barrel proteins using ANN- and SVM-based methods. Protein Struct Funct Bioinform 56:11–18
22. Zhang C, Kim SH (2000) A comprehensive analysis of the Greek key motifs in protein β-barrels and β-sandwiches. Protein Struct Funct Genet 40:409–419
23. Tran TVD et al (2011) Energy-based classification and structure prediction of transmembrane beta-barrel proteins. Proc IEEE ICCABS 2011:159–164
24. Tran TVD, Chassignet P, Steyaert J-M (2011) Prediction of permuted super-secondary structures in beta-barrel proteins. Proc ACM SAC 2011:110–111
25. Tusnády GE, Dosztányi Z, Simon I (2005) PDB_TM: selection and membrane localization of transmembrane proteins in the Protein Data Bank. Nucleic Acids Res 33: D275–D278
26. Li W, Godzik A (2006) Cd-hit: a fast program for clustering and comparing large sets of protein or nucleotide sequences. Bioinformatics 22(13):1658–1659
27. Marsh D (2000) Infrared dichroism of twisted beta-sheet barrels. The structure of E. coli outer membrane proteins. J Mol Biol 297: 803–808
28. Murzin AG, Lesk AM, Chothia C (1994) Principles determining the structure of β-sheet barrels in proteins I. A theoretical analysis. J Mol Biol 236:1369–1381
29. Murzin AG, Lesk AM, Chothia C (1994) Principles determining the structure of β-sheet barrels in proteins II. The observed structures. J Mol Biol 236:1382–1400
30. Chou KC, Carlacci L, Maggiora GM (1990) Conformational and geometrical properties of idealized beta-barrels in proteins. J Mol Biol 213:315–326
31. Tamm LK, Hong H, Liang B (2004) Folding and assembly of β-barrel membrane proteins. Biochim Biophys Acta 1666:250–263
32. Kyte J, Doolittle RF (1982) A simple method for displaying the hydropathic character of a protein. J Mol Biol 157:105–132
33. Fano R (1961) Transmission of information. Wiley, New York
34. Gibrat J-F, Garnier J, Robson B (1987) Further developments of protein secondary structure prediction using information theory. New parameters and consideration of residue pairs. J Mol Biol 198:425–443
35. Cormen TH et al (2009) Introduction to algorithms, 3rd edn. MIT Press, Cambridge, MA
36. Liu WM (1998) Shear numbers of protein β-barrels: definition, refinements and statistics. J Mol Biol 275:541–545
37. Lewis BA, Engelman DM (1983) Lipid bilayer thickness varies linearly with acyl chain length in fluid phosphatidylcholine vesicles. J Mol Biol 166(2):211–217

38. Rawicz W et al (2000) Effect of chain length and unsaturation on elasticity of lipid bilayers. Biophys J 79(1):328–339
39. Grantham R (1974) Amino acid difference formula to help explain protein evolution. Science 185:862–864
40. Bhaskaran R, Ponnuswamy PK (1988) Amino acid scale: average flexibility index. Int J Pept Protein Res 32:242–255
41. Dunbrack RL, Cohen FE (1997) Bayesian statistical analysis of protein side-chain rotamer preferences. Protein Sci 6(8):1661–1681
42. van Gunsteren WF et al (1996) Biomolecular simulation: the GROMOS96 Manual and User Guide. vdf Hochschulverlag AG an der ETH Zurich and BIOMOS b.v.: Zurich, Groningen

Chapter 18

Functional Structural Motifs for Protein–Ligand, Protein–Protein, and Protein–Nucleic Acid Interactions and their Connection to Supersecondary Structures

Akira R. Kinjo and Haruki Nakamura

Abstract

Protein functions are mediated by interactions between proteins and other molecules. One useful approach to analyze protein functions is to compare and classify the structures of interaction interfaces of proteins. Here, we describe the procedures for compiling a database of interface structures and efficiently comparing the interface structures. To do so requires a good understanding of the data structures of the Protein Data Bank (PDB). Therefore, we also provide a detailed account of the PDB exchange dictionary necessary for extracting data that are relevant for analyzing interaction interfaces and secondary structures. We identify recurring structural motifs by classifying similar interface structures, and we define a coarse-grained representation of supersecondary structures (SSS) which represents a sequence of two or three secondary structure elements including their relative orientations as a string of four to seven letters. By examining the correspondence between structural motifs and SSS strings, we show that no SSS string has particularly high propensity to be found interaction interfaces in general, indicating any SSS can be used as a binding interface. When individual structural motifs are examined, there are some SSS strings that have high propensity for particular groups of structural motifs. In addition, it is shown that while the SSS strings found in particular structural motifs for nonpolymer and protein interfaces are as abundant as in other structural motifs that belong to the same subunit, structural motifs for nucleic acid interfaces exhibit somewhat stronger preference for SSS strings. In regard to protein folds, many motif-specific SSS strings were found across many folds, suggesting that SSS may be a useful description to investigate the universality of ligand binding modes.

Key words: Protein structure, Protein–ligand interaction, Protein–protein interaction, Protein–nucleic acid interaction, Protein structure comparison, Protein structure classification

1. Introduction

All protein molecules need to interact with other molecules in order to perform their functions in a living system. Accordingly, a number of studies have been carried out to elucidate the characteristic features of interactions of proteins with small molecules, other

Alexander E. Kister (ed.), *Protein Supersecondary Structures*, Methods in Molecular Biology, vol. 932,
DOI 10.1007/978-1-62703-065-6_18,

proteins as well as nucleic acids (DNA and RNA). Since all interactions are accommodated by specific arrangements of atoms in protein structures, one approach to understand the diversity and universality of the mechanisms of protein functions is to compare and classify the interface structures. Comparative studies of interface structures have been extensively performed (1–7). While all the structural information regarding interactions of proteins are provided at atomic resolution, sometimes it is useful to grasp typical structural patterns in a more abstract description. Here we employ one-dimensional representation of supersecondary structures (SSS) and analyze the correspondence between SSS strings and structural motifs of interaction interfaces.

There are a number of prerequisites in order to carry out this type of studies. First of all, the original data source is the Protein Data Bank (PDB) (8). To extract useful information requires a deep understanding of the data structure of the PDB, which is reviewed first to the extent necessary for the present purpose. Then, we describe a method for comparing and aligning structures of interaction interfaces at atomic resolution. Since there are large amounts of data in the current PDB, a very efficient method is required to perform all-against-all comparison of all the interfaces that are currently known. We also provide the precise definition of SSS strings used in the present study. Finally, we study the correspondence between structural motifs of interaction interfaces at atomic resolution and SSS strings. In particular, we examine if there exist any SSS strings that are preferred for interaction interfaces in general or for particular motifs.

2. Materials

2.1. Protein Data Bank

To study protein structures thoroughly and comprehensively, a good understanding of the PDB data structure is required. We review various file formats provided by the PDB. There are currently three data formats: PDB, mmCIF (9), and PDBML (10).

1. The PDB format has been used, with some revisions, since the beginning of the PDB. However, it carries some historical limitations such as the limited number of columns, inconsistent numbering of residues, and lack of cross-references between data elements in addition to too many “exceptional” cases, which makes it difficult to handle the PDB format files in a consistent manner.
2. The mmCIF (macromolecular crystallographic information file) (9) mostly solves the problems of the PDB format by rigorously defining the syntax and vocabularies for specifying data elements (see Note 1). The basic units of data are categories

and category items which are defined in the PDB exchange dictionary (see Note 2). The categories classify the type of data such as citation, entity (molecular entities in an entry), exptl (experimental techniques), etc. There are some 300 categories defined. The category items specify the properties or attributes associated with each category. For example, the title of the primary citation article is specified by the _citation.title item. As in relational database schema, each category is specified by its primary key which may be one or a specific combination of the category items, and the relationships between two categories are specified by the foreign keys. Certain related categories are grouped into category groups. For example, the citation and citation_author categories are both under the citation_group. Thus, mmCIF allows one to identify the relevant data elements and other elements related to them in a consistent and comprehensive way. One drawback of the mmCIF format is its special-purpose syntax called the STAR syntax (11), which may be a minor obstacle for the casual user.

3. The PDBML format (see Note 3) (10) is a direct translation of the mmCIF format into an XML (extensible markup language) format (see Note 4). The XML elements and attributes in PDBML are defined by the PDBML XML schema which also includes the definitions of the primary keys and foreign keys (cross-references) of category elements. Being an XML format, PDBML allows to extract specific data items using standard XML tools. Some of the drawbacks of PDBML are that PDBML files are often large in size and that it is sometimes difficult for humans to read the contents of PDBML files because actual contents are buried in XML tags.

2.2. Biological Units

Since many proteins function in complex with other molecules, it is important to identify their biological units. For crystallographic structures, the structural information provided in the PDB is not necessarily that of biological unit, but often that of asymmetric unit. Information for generating biological units is provided in the PDB data which we summarize below based on the mmCIF or PDBML formats.

1. The pdbx_struct_assembly category provides general information about biological units. There may be multiple elements of this category for a PDB entry because the author-defined and software-predicted (12) assemblies may disagree. Each element is given a unique identifier and provides some annotations such as oligomeric count and the method of identification.
2. The pdbx_struct_assembly_gen category lists the molecular entities whose atomic coordinates are transformed as well as the transformations (operations) to be applied. The molecular entities are identified by their “chain identifiers.” Note that

these chain identifiers are not the same as those in the PDB format files (which are referred to as auth_asym_id in mmCIF/PDBML), but are the label_asym_id provided in the struct_asym category of mmCIF/PDBML (see Note 5). A transformation is specified by its identifier which corresponds to an element in the pdbx_struct_oper_list category (see below).

3. The pdbx_struct_oper_list category provides the rotation matrix and the translation vector for transforming the atomic coordinates of molecules. By combining the atomic coordinates provided in the original data and those obtained by the transformations, a biological unit is generated.

2.3. Secondary Structures

Secondary structures are defined by regular patterns of backbone conformations and hydrogen bonds. While there are widely used software packages such as DSSP (13) and Stride (14) for extracting secondary structure information from atomic coordinates, basic information on helices and sheets are also available in the PDB data. We review how to extract such information from mmCIF/PDBML.

1. The information of secondary structures that are defined locally along the backbone is specified in the struct_conf category. These structures may include α helices, 3_{10} helices, turns, and isolated β strands. The types of these local structures are provided in the conf_type_id item which refers to the struct_conf_type category. Although a very detailed classification of local structures is defined in the PDB exchange dictionary for both proteins and nucleic acids, the current data actually contain only two types, namely, HELX_P (helix with handedness and type unspecified) and TURN_P (turn with unspecified type) for proteins.
2. The struct_conf element specifies the segment for which a secondary structure is defined by providing the range of residue positions. The residue position at the beginning of a segment is specified by beg_label_asym_id, beg_label_comp_id, beg_label_seq_id which refer to label_asym_id (chain ID), label_comp_id (residue name), label_seq_id (serial number of amino acid sequence) in the atom_site category (atomic coordinates) (see Note 6). The end of a segment is specified in a similar manner with items prefixed with "end_…" instead of "beg_…"
3. Several categories are dedicated for annotating β sheets, which include, but not limited to, the struct_sheet, struct_sheet_range and struct_sheet_order categories. The struct_sheet category element lists the identifier of a β sheet and the number of β strands contained in the β sheet. The struct_sheet_range

category lists individual β strand segments in a similar manner as the struct_conf category, and the struct_sheet_order category describes the relative orientation (i.e., parallel or antiparallel) for a pair of β strands.

2.4. Interaction Interfaces

1. Some information on functionally relevant sites such as active or ligand binding sites are available in categories such as struct_site, struct_site_gen, or struct_site_keywords. However, this information is based on the residue-level description, and hence more detailed atomic-level description must be obtained from the atomic coordinates in the atom_site category.
2. The atom_site category roughly corresponds to the ATOM or HETATM lines in the PDB format files, but it also contains some additional information that eases the systematic manipulation of atomic coordinates.
3. In the following, we define an interaction interface of a subunit as a set of at least 10 atoms that are in contact within 5 Å with some atoms of a ligand (nonpolymers, proteins, or nucleic acids).

3. Methods

3.1. The GIRAF Structure Search and Alignment Method

To analyze the relationship between interaction interfaces and supersecondary structures, we use the GIRAF (Geometric Indexing and Refined Alignment Finder) structure search and alignment method (15). The GIRAF method can extremely efficiently search similar interface structures for a given query structure, and produces atom-wise structural alignments. We briefly describe how such efficient search is achieved as well as how sequence-order-independent alignments are refined.

3.2. Compiling a Database of Interface Structures

To use GIRAF, we first need to compile a relational database (RDB) of interface structures. Interfaces are identified as described in the Subheading 2.4.

1. The Delaunay tessellation is applied to the atomic coordinates of a subunit, which produces a set of tetrahedra whose vertices comprise atoms of the subunit.
2. The tetrahedra that contain at least one atom of an interface are selected, and other tetrahedra are discarded. The tetrahedra in which at least two vertices have the same atom type are also discarded, and vertices are ordered according to atom types. The atom types include main-chain N, C_α, C′, O, C_β, and side-chain N, C, O, S.

3. Each selected tetrahedron is characterized by its edge lengths, tetrahedron's volume, the atom type of the vertices, the chirality of the tetrahedra, as well as the atom composition within the half-sphere in the normal direction of each tetrahedron face. There are in total 43 features that characterize a tetrahedron. A B-tree index (16) is created for these 40 features (see Note 7).
4. A tetrahedron is also used for defining a local coordinate system of an interface structure. A tetrahedron characterized by structural features and a local coordinate frame is referred to as a refset in the following.
5. When all protein binding interfaces (~350,000) of all PDB entries (~70,000) are compiled into the RDB, approximately 40 million refsets are identified.
6. Atomic coordinates as well as other relevant information of the interfaces are also stored in the RDB.

3.3. Geometric Index Search

In order to search the database for a query interface structure, we first need to characterize the query structure using the same structural features as the interfaces in the RDB. The efficiency of GIRAF search rests on the assumption that if two interfaces are similar, they are characterized by similar refsets and vice versa. The geometric index (GI) search identifies a set of similar refsets by exploiting the index of geometric features of tetrahedra.

1. The Delaunay tessellation is applied to the query structure and the corresponding refsets are identified. The resulting refsets are stored in a temporary table in the RDB.
2. The database table containing the refsets of interfaces is queried using an SQL (17) statement joining the template and query refset tables with conditions on structural features allowing some variations. As a result of this query, a set of pairs of similar refsets (one for the query, the other for a template in the RDB) is obtained.
3. For each pair of returned refsets, we transform the atomic coordinates of template interfaces as well as those of the query structure based on the local coordinate frames defined by the respective refsets.
4. Since the structures of the query and a template are represented in a local coordinate frame, the spatial proximity of their atomic coordinates can be directly compared. Thus, the number of query-template atom pairs of the same atom type overlapping within a certain cutoff radius (2.5 Å is used in the current study) is counted. If the count is not sufficiently high (10 in the current study), the template structure is discarded in the following stages.

3.4. Refinement of Atomic Alignment

The atomic overlap count identified in the GI search does not provide one-to-one correspondence (alignment) between query and template atoms. It yields many-to-many correspondences in general because an atom in the query structure may be close to multiple atoms in the template structure in local coordinate frames and vice versa. Since the topology of protein structure is not linear at atomic level, we cannot use the dynamic programming algorithm to obtain an alignment. Instead we can use the Hungarian method (18, 19) for that purpose. The Hungarian method is an efficient algorithm for finding one-to-one correspondence between two sets of points which are represented as a data structure called bipartite graph (see Note 8). There are a few different approaches to apply the Hungarian method based on refsets. The one described below was inspired by a mathematical framework based on the Gromov–Hausdorff distance (20, 21). An advantage of this particular approach is that it allows alignments of flexible structures such as those found in domain motions.

1. For a given pair of potentially similar interfaces identified in the GI search, find all the corresponding refset pairs. Note that since there are many refsets associated with each of the query and template structures, multiple query-template refset pairs may yield plausible atomic overlaps.
2. Create a bipartite graph with the one group of nodes consisting of the query atoms and the other group of the template atoms, and with edges between all pairs of the query and template atoms. Assign null (0) weights to the edges.
3. Pick a refset pair and transform the atomic coordinates of the query and template based on the corresponding refsets.
4. If the query and template atoms are of the same atom type and closer than a specified cutoff length (2.5 Å in the current study) in the local coordinate frames, add a weight to the corresponding edge. The weight for the edge connecting the query atom q and template atom t is defined by

$$w(q,t) = \max\left(1 - d(q,t) / d\mathrm{c}, 0\right) \qquad (1)$$

   where $d(q,t)$ is the distance between the two atoms in the local coordinate frames and d_c is the cutoff distance (2.5 Å).
5. Iterate the steps 3 and 4 and for all the refset pairs (see Note 9).
6. Remove the edges with weight 0 if any.
7. Apply the Hungarian method to find the optimal alignment that maximizes the sum of edge weights.
8. The GIRAF score of the alignment between a query Q and template T is given as

$$s(Q,T) = \frac{N(Q,T)\sum_{a} w(q_a, t_a)}{\min\left[N(Q), N(T)\right]} \quad (2)$$

where $N(Q,T)$ is the number of aligned atom pairs, $w(q_a,t_a)$ is the weight of the edge between atoms q_a and t_a, which is summed over aligned atom pairs, and $N(Q)$ and $N(T)$ are the number of interface atoms of the query and the template, respectively.

3.5. Classification of Interaction Interface Structures

Based on an all-against-all comparison of interfaces, we define structural motifs by complete-linkage clusters (see Note 10) of similar interface structures (5, 6). For the clustering, we imposed GIRAF score of 15 as the threshold of the similarity.

1. We have used all the 70,231 PDB entries as of December 29, 2010 from which all biological units were generated. There were 197,690 subunits in 79,826 biological units which contained at least one ligand binding interfaces.
2. The ligands include nonpolymers except for water molecules (see Note 11), and proteins (annotated as polypeptide(L) in PDBML) with at least 25 amino acid residues, and nucleic acids (annotated as polydeoxyribonucleotide, polyribonucleotide, or polydeoxyribonucleotide–polyribonucleotide hybrid in PDBML) (see Note 12).
3. There were 410,254 nonpolymer binding interfaces, and an all-against-all comparison and subsequent complete-linkage clustering yielded 5,869 structural motifs with at least 10 members.
4. There were 346,288 protein binding interfaces, and an all-against-all comparison and subsequent complete-linkage clustering yielded 7,678 structural motifs with at least 10 members.
5. There were 20,338 nucleic acid binding interfaces, and an all-against-all comparison and subsequent complete-linkage clustering yielded 398 structural motifs with at least 10 members.

3.6. Defining Supersecondary Structures

Supersecondary structures (SSS) are spatial arrangements of a few consecutive secondary structure elements (SSE) that are frequently observed in protein structures. Sometimes SSS is simply represented as a string of consecutive SSE types (22). Here we employ a string of letters representing secondary structure elements interleaved with symbols representing relative orientations between two adjacent SSE's (Fig. 1).

1. A SSE is either helix or strand as defined in the corresponding categories of PDBML (i.e., struct_conf and struct_sheet_range; see Subheading 2.3).

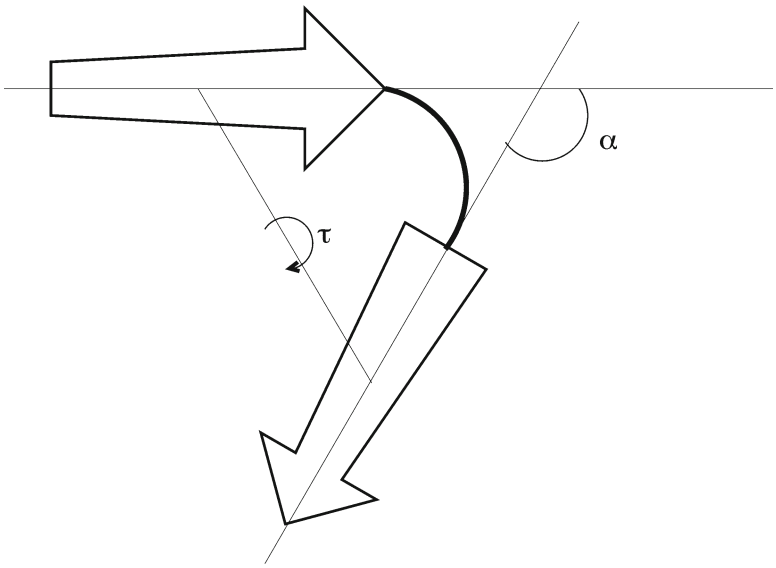


Fig. 1. Relative orientation of two secondary structure elements.

2. Geometrically, each SSE is represented as a vector whose center resides at the center of mass of the backbone atoms (N, C_α, C′) in the SSE and whose direction is the principal axis defined by the coordinates of the backbone atoms in the SSE (23).
3. Symbolically, each SSE is represented by a letter, "H" for a helix and "E" for a strand.
4. Geometrically, the relative orientation between two adjacent SSE's is quantified by the bend angle between the two SSE vectors, $\alpha \in [0,\pi]$, and the torsion angle between the two SSE vectors connected by a vector between their center points, $\tau \in (-\pi,\pi)$.
5. Symbolically, the bend angle α is represented by the symbol "+" if it is less than equal to $\pi/2$ (i.e., acute or right angles), or "−", otherwise (i.e., obtuse angles). Similarly, the torsion angle τ is represented by "+" if it is nonnegative, or "−" if it is negative.
6. In this study, a SSS consists of two or three consecutive SSE's which is symbolically represented as a string of letters and symbols indicating the SSE's and their relative orientations. For example, a string "H+−E" represents a SSS consisting of a helix followed by a strand with $\alpha \le \pi/2$ and $\tau < 0$. In total, 144 $(=2^4+2^7)$ distinct strings are possible for combinations of two or three SSE's, and all of them are observed in the PDB. Note that some 2-SSE SSS's may be a part of some 3-SSE SSS's, but we do not remove such redundancy in the following analysis.

3.7. General Trends of Supersecondary Structures in Interaction Interfaces

To have a general overview, the SSS strings that are most frequently found in interaction interfaces irrespective of structural motifs were computed. To do so requires a careful treatment of the redundancy in the data set. Thus, we first normalize the SSS count in each structural motif and then count the frequency of each SSS using the normalized per-motif counts.

1. When a SSE segment includes some atoms in an interface to a ligand, then the SSS that includes the SSE is defined as an interface SSS.
2. The number $C(m,s)$ of each SSS string s is counted for each interface motif m. Then, the normalized count $N(m,s)$ is defined as

$$N(m,s) = \frac{C(m,s) + \delta}{\sum_{s'} [C(m,s') + \delta]} \tag{3}$$

   where $\delta = 0.01$.
3. The normalized frequency of each SSS string is given as

$$f(s) = \frac{\sum_{m} N(m,s)}{\sum_{s'} \sum_{m} N(m,s')}. \tag{4}$$

4. As a control, we also computed the complementary normalized number $N(m,s)$ of SSS strings that did not overlap with the interface belonging to the motif but nevertheless were found in the same subunit (see Note 13). Then we calculated the corresponding frequency $\bar{f}(s)$ in a similar manner as $f(s)$. This control set is referred to as the type-1 control set in the following. Then, the log-odd score of SSS string s is given by

$$lod(s) = \log_2[f(s) / \bar{f}(s)]. \tag{5}$$

5. The five most frequent SSS strings are listed in Table 1. We can observe that SSS strings with 2 SSE's with turns (i.e., H–… or E–…) are dominant. For nonpolymer interfaces, however, their rank in log-odd score is rather low, which indicates that these SSS strings are not particularly specific to the binding interfaces. For protein and nucleic acid interfaces, we can see that the SSS strings H–H and H–+H, both of which may be regarded as helix-turn-helix motifs, also exhibit marginally high log-odd scores.
6. Five SSS strings with the highest log-odd scores are listed in Table 2. It is notable that all of these SSS strings contain only helices except for H–+E–+H for nonpolymer interfaces. Nevertheless, all the log-odd scores are lower than 1, indicat-

Table 1
The SSS strings that are most frequently found in interaction interfaces

Rank	Nonpolymer[a]	Protein[a]	Nucleic acid[a]
1	H–H (129, 2.4, –0.12)	H–H (5, 3.2, 0.64)	H–H (5, 3.3, 0.75)
2	H–+H (130, 2.3, –0.21)	H–+H (6, 3.2, 0.64)	H–+H (3, 3.2, 0.81)
3	E–H (132, 2.1, –0.31)	E–+E (135, 2.8, –0.47)	E–+E (25, 3.2, 0.30)
4	E–+H (131, 2.1, –0.29)	E–E (141, 2.6, –0.58)	E–E (30, 3.0, 0.18)
5	E–+E (142, 1.9, –1.05)	H–E (136, 2.4, –0.49)	H–E (54, 2.2, –0.03)

[a]The SSS strings with their rank in log-odd score, frequency and log-odd score in the parentheses

Table 2
The SSS strings with largest log-odd score $\log_2[f(s)/\bar{f}(s)]$

Rank	Nonpolymer[a]	Protein[a]	Nucleic acid[a]
1	H–H+–H (23, 1.0, 0.84)	H–+H–+H (20, 1.0, 0.88)	H–+H–H (13, 1.1, 0.93)
2	H–+H–H (26, 0.9, 0.74)	H–+H–H (21, 0.9, 0.87)	H–+H–+H (14, 1.1, 0.91)
3	H–H–+H (28, 0.9, 0.74)	H–H–+H (24, 0.9, 0.84)	H–+H (2, 3.2, 0.81)
4	H+–H–H (30, 0.9, 0.74)	H–H–H (25, 0.9, 0.83)	H–H–H (17, 1.0, 0.80)
5	H–+E–+H (19, 1.0, 0.73)	H–H (1, 3.2, 0.64)	H–H (1, 3.3, 0.75)

[a]The SSS strings with their rank in frequency, frequency and log-odd score in the parentheses

ing that no SSS strings are particularly specific to interaction interfaces in general.

3.8. Supersecondary Structures in Individual Interaction Interface Motifs

In the previous section, we have seen that no SSS strings are particularly preferred to be found in interaction interfaces in general. Nevertheless, if we examine individual interface motifs, we can find strong preferences as we shall show below.

1. The normalized count $N(m,s)$ of the SSS string s in the motif m is defined as above.
2. The complementary normalized count $\bar{N}(m,s)$ of the SSS string s not found in any interfaces of the subunits in the motif m is also computed as above (see Note 13).
3. The log-odd score $lod(m,s)$, which measures the preference of the SSS string s within the motif m, is defined as

$$lod(m,s) = \log_2\left[\frac{N(m,s)}{\bar{N}(m,s)}\right]. \quad (6)$$

Table 3
Recurring motif-specific SSS strings with respect to type-1 control set

Rank	Nonpolymer[a]	Protein[a]	Nucleic acid[a]
1	H−H (545)	H−+H (1156)	E−+E (70)
2	H−+H (473)	H−H (1135)	E−E (66)
3	H+−H (353)	H+−H (685)	H−H (62)
4	H++H (305)	H++H (678)	H−+H (59)
5	E−H (298)	H−E (600)	H−E (52)
6	H−+H−H (295)	E−+E (595)	E−+H (51)
7	H−H−+H (291)	E−+H (588)	H−+E (45)
8	E−+H (285)	E−H (583)	E−H (40)
9	E−+E−E (278)	E−E (581)	H++H (29)
10	H−+H−+H (276)	H−+E (538)	H+−H (28)

[a]Motif-specific SSS strings with the number of corresponding motifs in the parentheses

4. For each interface motif, we identified SSS strings that satisfied the following criteria:
   (a) The rank in normalized count is within top 5.
   (b) The rank in log-odd score is within top 5.
   (c) The normalized count $N(m,s)$ is greater than 5 (%).
   (d) The log-odd score is greater than 3 (i.e., >8 times the background).

   We refer to those SSS strings that satisfy these criteria as "motif-specific" in the following.
5. We found some SSS string satisfied these criteria for 4,733 (out of 5,869) motifs for nonpolymer interfaces, 5,859 (out of 7,678) motifs for protein interfaces, and 312 (out of 398) motifs for nucleic acid interfaces. Thus, there do exist some SSS strings that are specific to these motifs in the majority of cases.
6. For nonpolymer and protein interface motifs, all 144 possible patterns of SSS strings were found to satisfy the above criteria for at least one motif whereas 99 SSS strings are found for nucleic acid interfaces.
7. Some SSS strings were often found to be motif-specific for many motifs. Examples are listed in Table 3. The abundance of helix-turn-helix-type SSS strings is noticed. In particular, the SSS strings H−+H and H−H are highly abundant in protein interface motifs (6). For nucleic acid interfaces, both helix-turn-helix and β-hairpin are abundant.

3.9. Supersecondary Structures in Individual Interaction Interface Motifs That Differ from Other Interfaces in the Same Subunits

In the previous section, we compared SSS strings in individual motifs with those not in any interfaces (type-1 control set). There are in general multiple interfaces in a protein subunit and the above analysis does not discriminate the differences between the interface of the motif from other interfaces. It may be possible that for a given protein subunit, similar SSS strings may be utilized in different interfaces. In order to examine how individual interfaces in a particular motif differ from other interfaces, we introduce a different SSS string distribution as another control (type-2 control set).

1. Let $C'(m,s)$ the number of the SSS string s of the subunit that belong to the motif m, but not found in the interfaces corresponding to the motif m (see Note 14). The normalized count $N'(m,s)$ is also defined analogously to Eq. 3.
2. The log-odd score $lod'(m,s)$ with $N'(m,s)$ is defined as

$$lod'(m,s) = \log_2\left[\frac{N(m,s)}{N'(m,s)}\right] \tag{7}$$

3. Using the same criteria as above (see Subheading 3.8), we found SSS strings for 1,447 (out of 5,869) motifs for nonpolymer interfaces, 1,270 (out of 7,678) motifs for protein interfaces, and 164 (out of 398) motifs for nucleic acid interfaces. In total, 144, 143, and 69 types of SSS strings were identified for nonpolymer, protein, and nucleic acid interface motifs, respectively.
4. Compared to the case with the type-1 control set, there are fewer motifs for which the motif-specific SSS strings exist, especially for nonpolymer and protein interfaces. This indicates that the SSS architectures are often similar for different interfaces in a particular protein subunit. However, the trends of nucleic acid interfaces are similar for type-1 and type-2 control sets, indicating there is a strong preference for SSS strings in nucleic acid binding.
5. Some recurring motif-specific SSS strings with respect to the type-2 control set are listed in Table 4. While some SSS strings are common to the case with the type-1 control set, the number of motifs in which these SSS strings are found is much smaller.

3.10. Correlation Between Supersecondary Structures at Interfaces and Protein Folds

Certain supersecondary structures are found in many protein folds. In this subsection, we examine how the SSS strings found in interaction interfaces (Table 4) are related to global protein folds (Tables 5–7).

1. The protein folds associated with each motif were identified using the SCOP database (24).
2. For each motif-specific SSS string (Table 4), the corresponding SCOP folds were associated via individual motifs.

Table 4
Recurring motif-specific SSS strings with respect to type-2 control set

Rank	Nonpolymer[a]	Protein[a]	Nucleic acid[a]
1	E−H (85)	H−H (286)	E−+E (59)
2	H−H (71)	H−+H (269)	E−E (58)
3	E−+H (70)	E−E (262)	H−H (52)
4	E−+H−E (68)	H−+E (253)	H−+H (47)
5	E−E (65)	E−+E (253)	E−H (44)
6	E−+E (63)	E−H (230)	E−+H (39)
7	H−+H (62)	H−E (228)	H−+E (32)
8	E−+H−+E (58)	E−+H (221)	H−E (31)
9	E−H−+E (56)	H+−H (170)	H++H (26)
10	H−E−H (56)	H++H (167)	H+−H (24)

[a]Motif-specific SSS strings with the number of corresponding motifs in the parentheses

3. *Nonpolymer interfaces* (Table 5). On the one hand, it is immediately evident that each SSS string is associated with a diverse set of folds. For example, the SSS string H−H is found in SCOP folds a.25 (Ferritin-like), d.153 (NTN hydrolase-like), a.1 (Globin-like), c.2 (NAD(P)-binding Rossmann-fold domains), f.23 (single transmembrane helix). On the other hand, some folds or superfolds (25) such as c.2, c.1 (TIM beta/alpha-barrel), d.58 (Ferredoxin-like) are found to be associated with many SSS strings. Only 4 motif-specific SSS strings (H++E++E, H+−E++E, E++E+−E, and E+−E+−E) corresponded to less than 10 protein folds. On the other hand, studied, 164 out of 408 folds studied were associated with less than 10 motif-specific SSS strings.
4. *Protein interfaces* (Table 6). The tendency is similar to that of nonpolymer interfaces in that each SSS string is associated with a wide range of folds and that some folds are found to be associated with many SSS strings. Many folds such as c.2, a.1, a.25, f.23, etc. found here are also found in nonpolymer interfaces. 16 SSS strings (e.g., E++H+−E, E+−H++E,H++H+−E, etc.) corresponded with less than 10 protein folds and 253 folds (out of 517) with less than 10 motif-specific SSS strings.
5. *Nucleic acid interfaces* (Table 7). Again, the general tendency is similar to other interfaces. However, the folds found here are

Table 5
Recurring motif-specific SSS strings for nonpolymer interfaces (type-2 control set) and associated SCOP folds

Rank	SSS[a]	N-fold[b]	Representative folds[c]
1	E–H	123	c.2, c.1, c.37, d.58, c.16
2	H–H	119	a.25, a.1, f.13, c.1, c.2
3	E–+H	122	c.2, c.37, c.1, c.23, b.6
4	E–+H–E	104	c.1, c.2, d.58, c.56, c.23
5	E–E	82	d.92, d.169, h.1, b.3, c.30
6	E–+E	91	d.92, b.3, d.169, d.230, b.82
7	H–+H	117	a.1, a.25, f.13, c.1, f.43
8	E–+H–+E	109	c.1, c.56, d.58, c.2, c.37
9	E–H–+E	105	c.1, c.56, d.58, c.37, c.16
10	H–E–H	111	c.1, c.37, c.2, c.36, d.58

[a]Motif-specific SSS strings (Table 4)
[b]Number of distinct SCOP folds associated with the SSS string
[c]Five most popular folds

Table 6
Recurring motif-specific SSS strings for protein interfaces (type-2 control set) and associated SCOP folds

Rank	SSS[a]	N-fold[b]	Representative folds[c]
1	H–H	207	a.25, d.153, a.1, c.2, f.23
2	H–+H	211	a.25, d.153, a.1, f.23, c.1
3	E–E	164	d.58, b.40, c.2, b.38, d.73
4	H–+E	138	d.58, d.153, b.40, a.22, c.1
5	E–+E	167	d.58, b.38, b.40, c.47, d.17
6	E–H	151	d.153, d.58, d.74, b.40, c.23
7	H–E	142	d.58, d.153, a.22, b.40, c.1
8	E–+H	151	d.153, d.58, c.23, g.8, b.40
9	H+–H	174	f.23, a.25, c.2, c.1, a.1
10	H++H	165	f.23, a.25, c.1, c.2, a.1

[a]Motif-specific SSS strings (Table 4)
[b]Number of distinct SCOP folds associated with the SSS string
[c]Five most popular folds

Table 7
Recurring motif-specific SSS strings for nucleic acid interfaces (type-2 control set) and associated SCOP folds

Rank	SSS[a]	N-fold[b]	Representative folds[c]
1	E−+E	37	d.58, g.41, b.34, b.40, c.55
2	E−E	31	b.34, c.55, g.41, b.40, a.4
3	H−H	35	a.4, a.22, a.60, b.34, c.55
4	H−+H	32	a.4, a.22, a.75, b.34, a.144
5	E−H	29	a.43, d.141, d.50, b.34, c.55
6	E−+H	25	c.55, d.58, g.39, a.43, d.12
7	H−+E	22	a.22, a.4, d.12, d.59, c.22
8	H−E	20	a.22, a.4, b.34, d.58, c.12
9	H++H	20	g.39, a.4, a.144, b.34, c.53
10	H+−H	23	a.22, a.144, a.4, a.7, b.34

[a]Motif-specific SSS strings (Table 4)
[b]Number of distinct SCOP folds associated with the SSS string
[c]Five most popular folds

somewhat different from those found in nonpolymer or protein interfaces, for these folds are associated with nucleic acid binding: c.55 (Ribonuclease H-like motif), a.4 (DNA/RNA-binding 3-helical bundle), a.22 (Histone-fold), a.144 (PABP domain-like), etc. 71 SSS strings corresponded with less than 10 protein folds and 84 folds (out of 103) with less than 10 motif-specific SSS strings. Although these numbers are large compared to the cases with nonpolymer and protein interfaces, this may be simply due to the less number of proteins in the PDB.

6. In all the cases, a large number of protein folds were associated with each SSS string. This suggests that there are no fold-specific supersecondary structures at the present level of description, and illuminates the universality of supersecondary structures across protein folds.

3.11. Ligand Specificity for Nonpolymer Interfaces

For nonpolymer interfaces, it is possible to associate ligand types with supersecondary structures at interaction interfaces.

1. Nonpolymer ligands associated with individual interfaces were identified.
2. The number of occurrence of each ligand type for each interface motif for nonpolymer interfaces was counted and normalized.

Table 8
Recurring motif-specific SSS strings for nonpolymer interfaces (type-2 control set) and associated ligands

Rank	SSS[a]	N-lig[b]	Representative ligands[c]
1	E−H	47	SO_4, CA, ADP, PO_4, FE
2	H−H	41	SO_4, HEM, ZN, CL, ATP
3	E−+H	41	SO_4, CA, CO, MN, GOL
4	E−+H−E	42	ZN, SO_4, PO_4, MN, MG
5	E−E	44	SO_4, FE, ZN, GOL, MN
6	E−+E	36	SO_4, CA, GOL, SF_4, ACT
7	H−+H	38	HEM, SO_4, GOL, CA, ATP
8	E−+H−+E	37	SO_4, MN, MG, ZN, FES
9	E−H−+E	42	SO_4, ZN, MG, PO_4, FES
10	H−E−H	37	SO_4, GOL, PO_4, CA, ZN

[a]Motif-specific SSS strings (Table 4)
[b]Number of ligand types associated with the SSS string
[c]Five most popular ligands (PDB chemical compound identifiers) associated with the SSS string

3. Each motif corresponds to a relatively small number of ligand types. Out of 5,869 nonpolymer interface motifs, 3,198 (54.5 %) had only 1 associated ligand, and 4,778 (81.4 %) of the motifs had one of the ligands associated with more than half of their member interfaces.
4. The number of ligand types associated with each motif-specific SSS along with most frequently found ligands are given in Table 8.
5. The motif-specific SSS strings are not highly specific to particular ligands in general. For example, the SSS string E−H is associated with 47 types of ligands and most popular ligands include SO_4 (sulfate ion), CA (calcium ion), ADP, PO_4 (phosphate ion), and FE (Fe^{3+} ion).
6. Some ligands are associated with many SSS strings. For example, GOL (glycerol), CA, ZN (zinc ion), and SO_4 are associated with 132, 131, 127, and 125 SSS strings (out of 144), respectively, with $lod'(m,s) > 3$ and $N'(m,s) > 5$.
7. In summary, there are no supersecondary structures that are specific to particular ligands. Put another way, supersecondary structures highlight the universal scaffolds for ligand binding sites.

4. Notes

1. A comprehensive resource regarding mmCIF is http://mmcif.pdb.org/.
2. For the definitions of category groups, categories, and category items, refer to http://mmcif.pdb.org/dictionaries/mmcif_pdbx.dic/Index/.
3. The PDBML format, being an XML format (see Note 4), has an advantage that it is readily extensible. The Protein Data Bank Japan (26), a member of the worldwide PDB (8), provides an extension of PDBML called PDBMLplus (27) in which the original PDBML files are augmented with some additional data.
4. The XML and related technologies are standardized by the World Wide Web consortium (W3C). See http://www.w3.org/standards/xml/.
5. The so-called chain ID in the ATOM or HETATM lines of the PDB format file corresponds to auth_asym_id in the atom_site category of mmCIF/PDBML. The mmCIF/PDBML format files have another identifier for chains, namely, label_asym_id, which are necessarily different for different molecular objects. For example, in the PDB entry 1GOF (28), the protein molecule galactose oxidase and its ligands, copper and sodium ions have the identical auth_asym_id (A), but their label_asym_id's are A, B, and C, respectively.
6. The residue names in the PDB format file correspond to auth_comp_id which may be different from label_comp_id for historical reasons. The latter follows a more standardized convention. Similarly, the serial number of residues in the PDB format file corresponds to auth_seq_id which is actually a text rather than a number so that it may apparently start from a "number" other than 1 and may contain gaps and duplications of numbers; the label_seq_id item in the atom_site category (corresponding to the num item in the entity_poly_seq category) is actually defined as a nonnegative integer and it always starts from 1 without gaps or duplications.
7. The B-tree index for the geometric features consists of 43 columns. Some database management systems (DBMS) do not support multicolumn index of this many columns. In such case, it is necessary to modify the source code of the DBMS (if it is an open source software).
8. A naive implementation of the Hungarian method (the Kuhn–Munkres algorithm) requires $O(|V|^2|E|)$ CPU time where $|V|$

and $|E|$ are the number of nodes and edges, respectively. This may not be efficient for large bipartite graphs. By using a data structure called heap (or priority queue) (29, 30), the CPU time is reduced to $O(|V||E|\log|V|)$. See the report by Gupta and Ying (19) for the details.

9. An alternative approach is to use only one pair of refsets for constructing a bipartite graph to obtain an alignment. The alignment obtained this way is rigid rather than flexible. Based on the rigid alignment, the template structure can be superposed on the query structure. After the superposition, a new bipartite graph can be generated which may be slightly different from the initial graph. Then we can iterate bipartite matching and superposition until convergence (although convergence is not guaranteed). We obtain an alignment for each pair of refsets, and after this procedure is applied to all pair of refsets, the best-scoring alignment is selected. This iterative refinement algorithm was the one that had been employed in earlier versions of GIRAF (15).
10. Complete linkage clustering of a large number of interfaces is a time-consuming process. An efficient technique based on heap (29) is described by Manning et al. (31).
11. Whether an entity is nonpolymer or polymer is specified by the _entity.type item of the entity category in mmCIF/PDBML.
12. The polymer types are specified by the _entity_poly.type item of the entity_poly category in mmCIF/PDBML.
13. A protein subunit may contain multiple interfaces for different ligands, including the interface(s) corresponding to the motif m, and $N(m,s)$ excludes all SSS strings that are found in any interfaces.
14. The type-2 control set includes all the SSS strings that are not part of the interface belonging to the structural motif of interest. Thus, it includes SSS strings in both other interfaces and noninterfaces.

Acknowledgments

This work was supported by a grant-in-aid by National Bioscience Database Center (NBDC), the Japan Science and Technology Agency (JST). A. R. K. was supported in part by the KAKENHI Grant-in-Aid for Young Scientists B (No. 22770149) from the Japan Society for the Promotion of Science (JSPS). H. N. was supported in part by the KAKENHI Grant-in-Aid for Scientific Researches B (No. 23370071) from the JSPS.

References

1. Kinoshita K, Sadanami K, Kidera A, Go N (1999) Structural motif of phosphate-binding site common to various protein superfamilies: all-against-all structural comparison of protein-mononucleotide complexes. Protein Eng 12:11–14
2. Gold ND, Jackson RM (2006) Fold independent structural comparisons of protein–ligand binding sites for exploring functional relationships. J Mol Biol 355:1112–1124
3. Xie L, Bourne PE (2008) Detecting evolutionary relationships across existing fold space, using sequence order-independent profile-profile alignments. Proc Natl Acad Sci USA 105:5441–5446
4. Minai R, Matsuo Y, Onuki H, Hirota H (2008) Method for comparing the structures of protein ligand-binding sites and application for predicting protein–drug interactions. Proteins 72:367–381
5. Kinjo AR, Nakamura H (2009) Comprehensive structural classification of ligand binding motifs in proteins. Structure 17:234–246
6. Kinjo AR, Nakamura H (2010) Geometric similarities of protein-protein interfaces at atomic resolution are only observed within homologous families: an exhaustive structural classification study. J Mol Biol 399:526–540
7. Keskin O, Tsai CJ, Wolfson H, Nussinov R (2004) A new, structurally nonredundant, diverse data set of protein–protein interfaces and its implications. Protein Sci 13:1043–1055
8. Berman H, Henrick K, Nakamura H, Markley JL (2007) The worldwide Protein Data Bank (wwPDB): ensuring a single, uniform archive of PDB data. Nucleic Acids Res 35:D301–D303
9. Westbrook JD, Bourne PE (2000) STAR/mmCIF: an ontology for macromolecular structure. Bioinformatics 16:159–168
10. Westbrook J, Ito N, Nakamura H, Henrick K, Berman HM (2005) PDBML: the representation of archival macromolecular structure data in XML. Bioinformatics 21:988–992
11. Hall SR (1991) The STAR file: a new format for electronic data transfer and archiving. J Chem Inf Comput Sci 31:326–333
12. Krissinel E, Henrick K (2007) Inference of macromolecular assemblies from crystalline state. J Mol Biol 372:774–797
13. Kabsch W, Sander C (1983) Dictionary of protein secondary structure: pattern recognition of hydrogen bonded and geometrical features. Biopolymers 22:2577–2637
14. Frishman D, Argos P (1995) Knowledge-based protein secondary structure assignment. Proteins 23:566–579
15. Kinjo AR, Nakamura H (2007) Similarity search for local protein structures at atomic resolution by exploiting a database management system. Biophysics 3:75–84. doi:10.2142/biophysics.3.75
16. Garcia-Molina H, Ullman JD, Widom J (2009) Database systems: the complete book, 2nd edn. Prentice Hall, Upper Saddle River, NJ, USA
17. Melton J, Simon AR (2002) SQL:1999 Understanding relational language components, Morgan Kaufmann, San Francisco, CA, USA
18. Lawler E (2001) Combinatorial optimization: networks and Matroids, Dover, New York, USA, originally published in 1976
19. Gupta A, Ying L (1999) On algorithms for finding maximum matchings in bipartite graphs. Tech Rep RC 21576(97320), IBM
20. Burago D, Burago Y, Ivanov S (2001) A course in metric geometry, vol. 33 of graduate studies in mathematics, American Mathematical Society, Providence, Rhode Island, USA
21. Mémoli F (2007) On the use of Gromov-Hausdorff distances for shape comparison. In: Botsch M, Pajarola R (eds) Eurographics symposium on point-based graphics 2007, The Eurographics Association, pp 81–90
22. Gerstein M (1997) A structural census of genomes: comparing bacterial, eukaryotic, and archaeal genomes in terms of protein structure. J Mol Biol 274:562–576
23. Mizuguchi K, Go N (1995) Comparison of spatial arrangements of secondary structural elements in proteins. Protein Eng 8:353–362
24. Murzin AG, Brenner SE, Hubbard T, Chothia C (1995) SCOP: a structural classification of proteins database for the investigation of sequences and structures. J Mol Biol 247:536–540
25. Orengo CA, Jones DT, Thornton JM (1994) Protein superfamilies and domain superfolds. Nature 372:631–634
26. Standley DM, Kinjo AR, Kinoshita K, Nakamura H (2008) Protein structure databases with new web services for structural sbiology and biomedical research. Brief Bioinform 9:276–285
27. Kinjo AR, Yamashita R, Nakamura H (2010) PDBj mine: design and implementation of relational database interface for Protein Data Bank Japan, Database 2010, baq021

28. Ito N, Phillips SE, Stevens C, Ogel ZB, McPherson MJ, Keen JN, Yadav KD, Knowles PF (1991) Novel thioether bond revealed by a 1.7 Å crystal structure of galactose oxidase. Nature 350:87–90
29. Cormen TH, Leiserson CE, Rivest RL, Stein C (2009) Introduction to algorithms, 3rd edn. MIT Press, Cambridge, MA, USA
30. Okasaki C (1999) Purely functional data structures. Cambridge University Press, Cambridge, UK
31. Manning CD, Raghavan P, Schuetze H (2008) Introduction to information retrieval. Cambridge University Press, Cambridge, United Kingdom

INDEX

Alexander E. Kister (ed.), *Protein Supersecondary Structures*, Methods in Molecular Biology, vol. 932,
DOI 10.1007/978-1-62703-065-6, © Springer Science+Business Media New York 2013

G

H

I

K

L

M

N

O

P

Q

R

S

T

U

V

X

Z